# Dual-primal skeleton: a thinning scheme for vertex sets lying on a surface mesh

Ricardo Uribe Lobello J.-L. Mari

Aix Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France

05/04/2019

# Overview

# Topological skeleton

## Skeleton

The skeleton is a well-known shape descriptor. It is an entity that is globally centered in a 2D or 3D object, and it characterizes its topology.

Figure: 2D Skeleton.

# Applications

## Applications

- Video tracking.
- Shape recognition.
- Surface sketching.

# State of the art

## State of the art

There are several techniques to extract topological skeletons from :

- Binary 2D images.
- 3D closed volume meshes.
- 3D cubic grids.

# State of the art

## Kudelski and Mari [13]

**From marked vertices on a surface mesh** :This approach uses erosion and special categories of vertices.

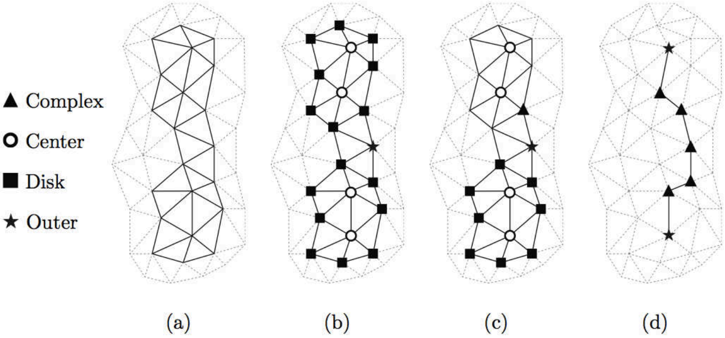



Figure: Description of the method proposed by Kudelski et Mari [13].

# Introduction

## Our objective

The extraction of skeletons from binary information located on the surface of an arbitrary triangulated mesh or area of interest. For example, vertices on the lines of maximal curvature.

# Introduction

## Our objective

The extraction of skeletons from binary information located on the surface of an arbitrary triangulated mesh or area of interest. For example, vertices on the lines of maximal curvature.

## Principles of our approach

- Implementing an homotopic thinning based in the concept of generalized adjacency.
- Replacing the erosion step by a primal-dual-primal iteration.
- Iterative process from primal to dual space.

# Introduction

## Our objective

The extraction of skeletons from binary information located on the surface of an arbitrary triangulated mesh or area of interest. For example, vertices on the lines of maximal curvature.

## Principles of our approach

- Implementing an homotopic thinning based in the concept of generalized adjacency.
- Replacing the erosion step by a primal-dual-primal iteration.
- Iterative process from primal to dual space.

## Contributions

- It can be applied over non developpable surfaces.
- It preserves the topology of the original area of interest.
- It produces skeletons more centered specially in thin areas.

# Our approach

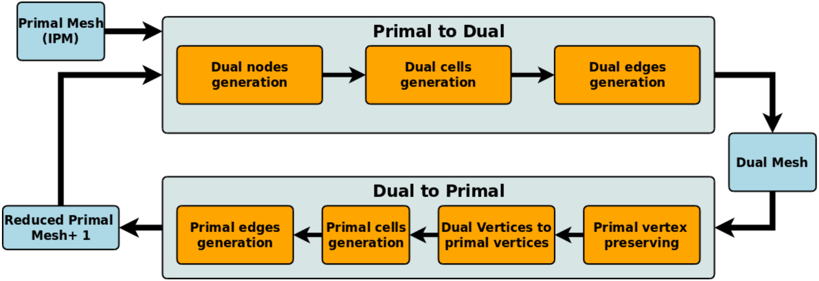



Figure: General structure of the workflow of our approach.

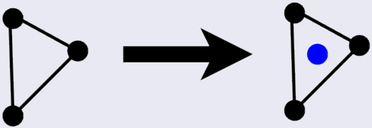
# Primal to dual algorithm

1. Generate all dual vertices.
2. Generate 2-dimensional dual cells (polygons).
3. Generate 1-dimensional dual cells (edges).
4. Transform triangles strips to dual poly-lines.
5. Connect thin and large dual structures sharing a primal edge.
6. Connect thin and large dual structures sharing a primal vertex.

# Primal to dual algorithm

## 1. Generate all dual vertices

(a) Primal edge to dual vertex.

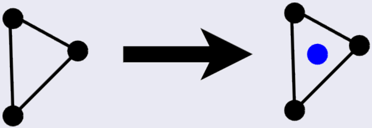


(b) Primal face to dual vertex.

# Primal to dual algorithm

## 1. Generate all dual vertices

(a) Primal edge to dual vertex.



(b) Primal face to dual vertex.

## 2. Generate 2-dimensional dual cells (polygons)

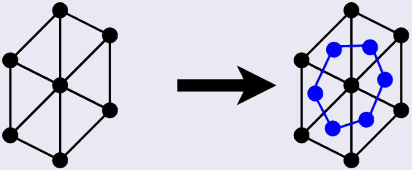


Figure: Primal vertex to dual cell (a polygon).

# Primal to dual algorithm

## 3. Generate 1-dimensional dual cells (edges).

Figure: Primal vertex to dual edge.

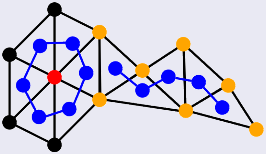
# Primal to dual algorithm

## 3. Generate 1-dimensional dual cells (edges).

Figure: Primal vertex to dual edge.

## 4. Transform triangles strips to dual poly-lines.

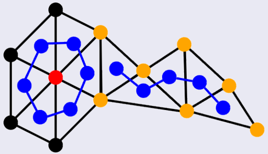
It can be detected by identifying triangles with all 3 vertices not having a 1-ring in the area of interest.



# Primal to dual algorithm

## 5. Connect thin and large dual structures sharing a primal edge.

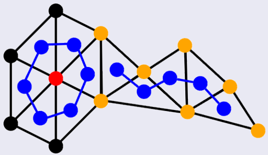
By detecting dual edges endpoints and checking if its primal cell share and edge with a large structure.



# Primal to dual algorithm

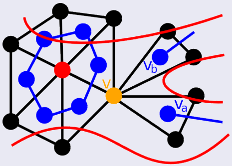
## 5. Connect thin and large dual structures sharing a primal edge.

By detecting dual edges endpoints and checking if its primal cell share and edge with a large structure.



## 6. Connect thin and large dual structures sharing a primal vertex.

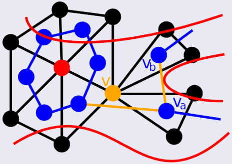
It can be detected by identifying primal vertices with more that one connected component in their incident primal faces.



# Primal to dual algorithm

## 6. Connect thin and large dual structures sharing a primal vertex.

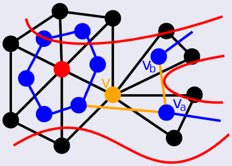
It can be detected by identifying a primal vertex (in yellow) with more that one connected component in their incident primal faces.
First configuration.



# Primal to dual algorithm

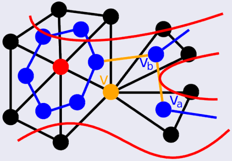
## 6. Connect thin and large dual structures sharing a primal vertex.

It can be detected by identifying a primal vertex (in yellow) with more that one connected component in their incident primal faces.
First configuration.




## 6. Connect thin and large dual structures sharing a primal vertex.

It can be detected by identifying primal vertices with more that one connected component in their incident primal faces.
Second configuration.



# Dual to primal transformation

## Rule 1

All 2-dimensional cells containing a dual vertex not shared by at least 3 dual cells (polygon) are eliminated.

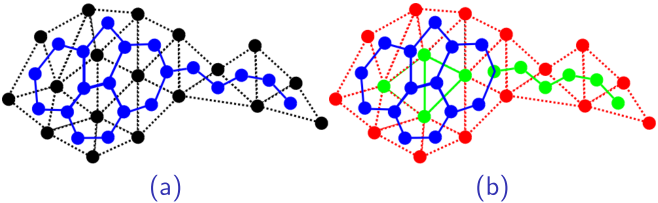
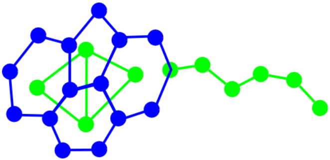



Figure: Primal cells are in black. Dual cells in blue. All red primal cells will be eliminated.

# Dual to primal transformation

## Rule 2

All 2-dimensional cells belonging to thin structures are also eliminated and replaced by dual vertices and dual edges that will be added to the new primal mesh.



(a) (b)

Figure: Green cells and vertices make part of the new primal mesh. All edges connected to a 2-dimensional cell are connected to the new primal vertex inside this dual cell.

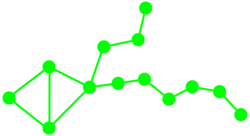
# Special case for mixed meshes

## Mixed meshes (figure a)

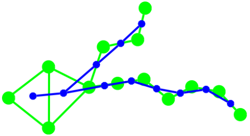
In intermediate iterations, mixed meshes (containing 2D and 1D cells) are generated.

## Generate the dual mesh (figure b)

The dual mesh in blue is generated mostly by using the standard rules explained earlier. However, in mixed areas, dual vertices in edges have to be connected to dual vertices in primal faces generating a junction.



(a)



(b)

# Post-processing

## Smoothing

For the moment, simple smoothing algorithm that moves every vertex to the barycenter of the vertices composing its topological neighborhood.

## Pruning

- Detection of the endpoints of the skeleton.
- If all or the majority of vertices belong to the original mesh, the branch is totally preserved.
- If all vertices do not belong to the original mesh, we eliminate the branch if it is short (a parameter defined by the user).

# Results Kudelski et Mari [13]

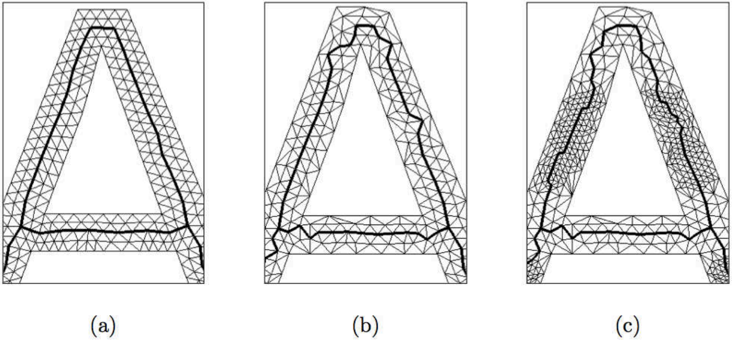
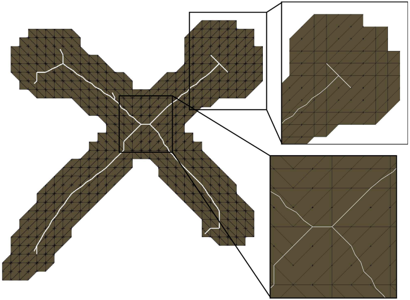
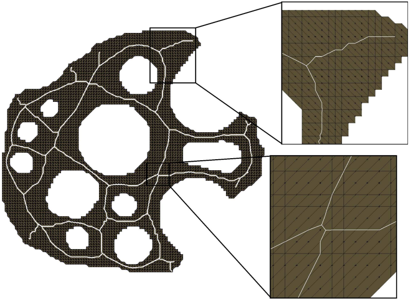


Figure: Results of the application of the approach by Kudelski et Mari [13].

# Results on planar meshes



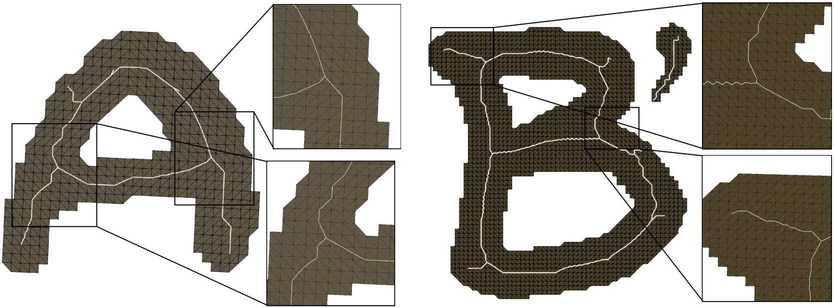
(a) Cross.



(b) Shape.

Figure: Application of our algorithm to 2D meshes. The cleaned skeleton is shown after a smoothing process.

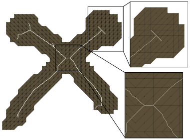
# Results on planar meshes

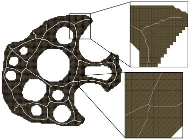


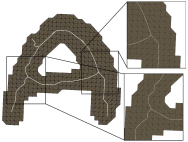
(a) Letter A. (b) Letter B.

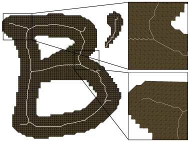
Figure: Application of our algorithm to 2D meshes. The cleaned skeleton is shown after a smoothing process.

# Results on planar meshes


(a) Cross.


(b) Shape.

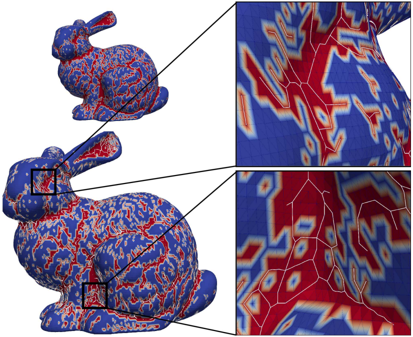

(c) Letter A.


(d) Letter B.

| Data set | # Triangles | # Iterations | Exec. Time (in ms) | Edges final skeleton |
|---|---|---|---|---|
| Cross | 819 | 5 | 246 | 108 |
| Shape | 8556 | 13 | 4529 | 1009 |
| A letter | 812 | 6 | 261 | 103 |
| B Letter | 4307 | 9 | 2049 | 361 |

Table: Information on the execution of the algorithm in the four previous planar meshes. This table shows the number of primal-to-dual and dual-to-primal iterations, the execution time and the total number of edges in the final skeleton.

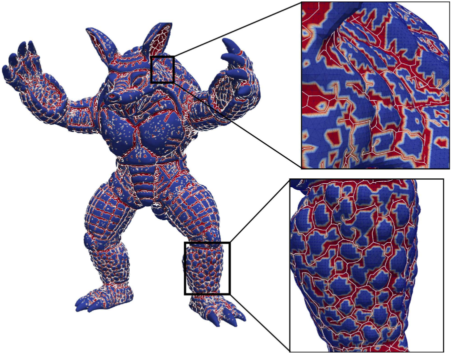
# Results on surfaces



(e) Surface Bunny with 69451 faces and 35947 vertices. The final skeleton has 9053 edges.

Figure: Application to the high curvature areas of a surface (in red). Two closeups show how the skeleton (in white) is located in the middle of the region of interest.

# Results on surfaces



(a) Surface Armadillo with 345944 faces and 172974 vertices. The final skeleton has 46725 edges.

Figure: Application to the high curvature areas of a surface (in red). Two closeups show how the skeleton (in white) is located in the middle of the region of interest.

# Primal to dual basic transformations

# Conclusions and perspectives

## Conclusions

- Our algorithm is robust and capable of extracting a 1-dimensional structure from any region lying on a mesh with arbitrary topology.
- Our approach is dependent on the size and connectivity of the mesh.
- The complexity and execution time can, and should be improved.

# Conclusions and perspectives

## Conclusions

- Our algorithm is robust and capable of extracting a 1-dimensional structure from any region lying on a mesh with arbitrary topology.
- Our approach is dependent on the size and connectivity of the mesh.
- The complexity and execution time can, and should be improved.

## Perspectives

- Our approach can be extended to n-dimensional simplicial meshes.
- Improvement of the localization of the final skeleton.