

# 3D images analysis workshop

Surfaces and meshes

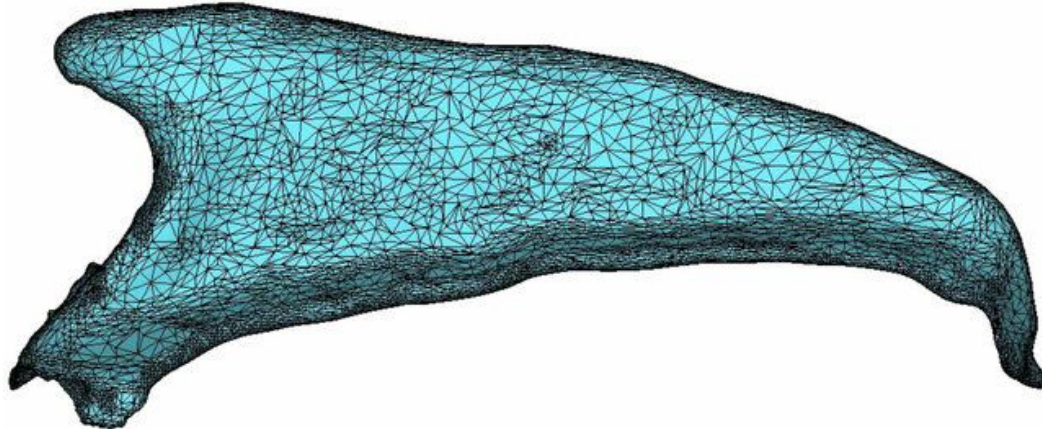
BAECKER Volker  
BENEDETTI Clément  
DEJEAN Matthieu

# Chapters:

1. How to represent a surface?
2. Mesh representations
3. What is a mesh useful for?
4. Acquire a surface/mesh
5. Voxels grid  $\Leftrightarrow$  mesh

# 1. How to represent a surface

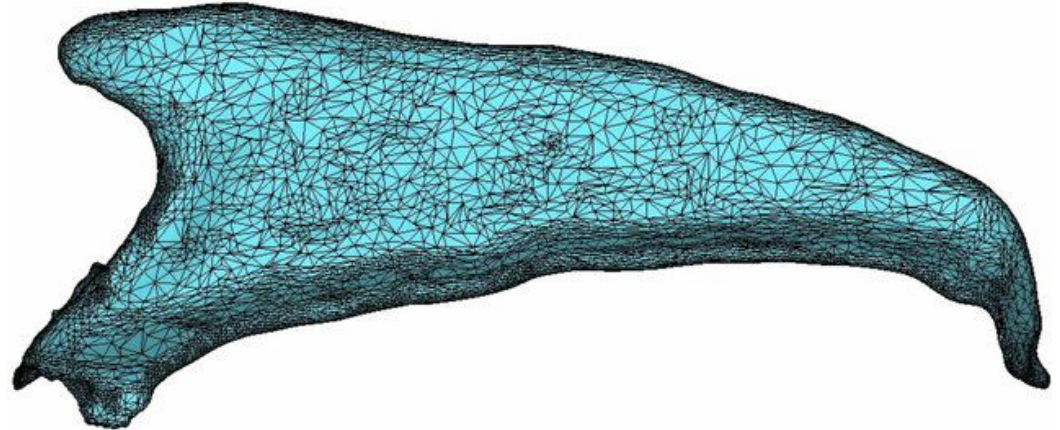
- Surface → superficial hull
- Voxels grid → in depth representation



# 1. How to represent a surface

Graph-like structure:

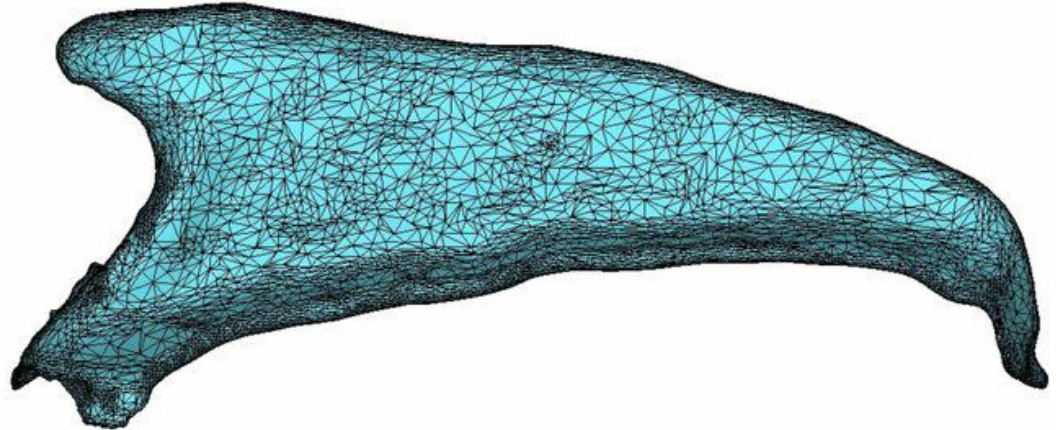
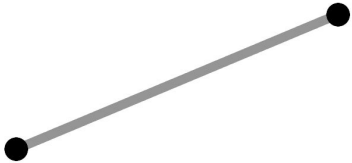
- Vertices (3D point)



# 1. How to represent a surface

## Graph-like structure:

- Vertices (3D point)
- Edges (link between vertices)

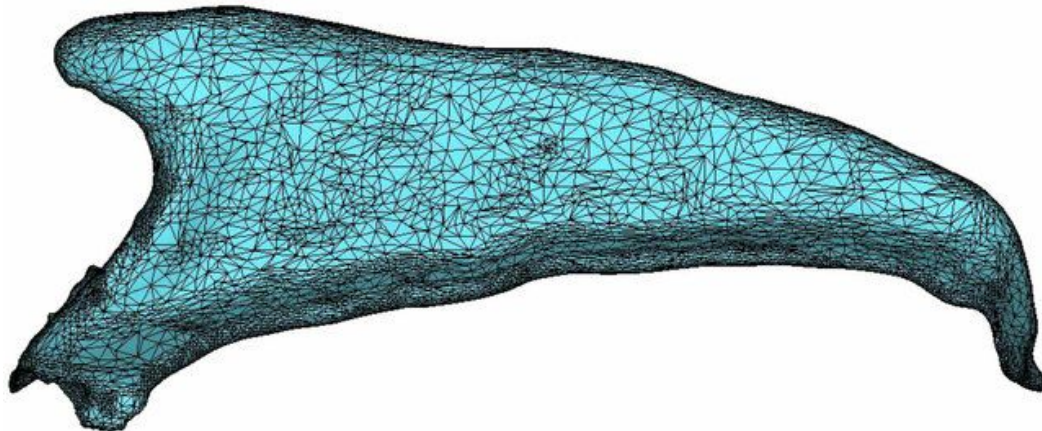
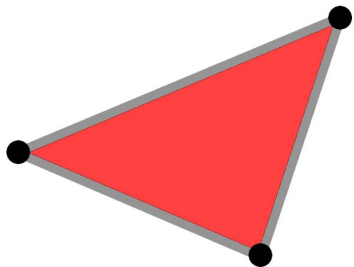


# 1. How to represent a surface

## Graph-like structure:

- Vertices (3D point)
- Edges (link between vertices)
- Faces (polygons)

⇒ **Mesh**



## 2. Mesh representations

Two most commons:

- List of triangles
- Indexed face set

Others (mainly for processing)

- Half-edges
- Winged-edges
- ...

## 2. Mesh representations

### Two most commons:

- List of triangles
- Indexed face set

### Others (mainly for processing)

- Half-edges
- Winged-edges
- ...



## 2. Mesh representations

### a. Triangles list

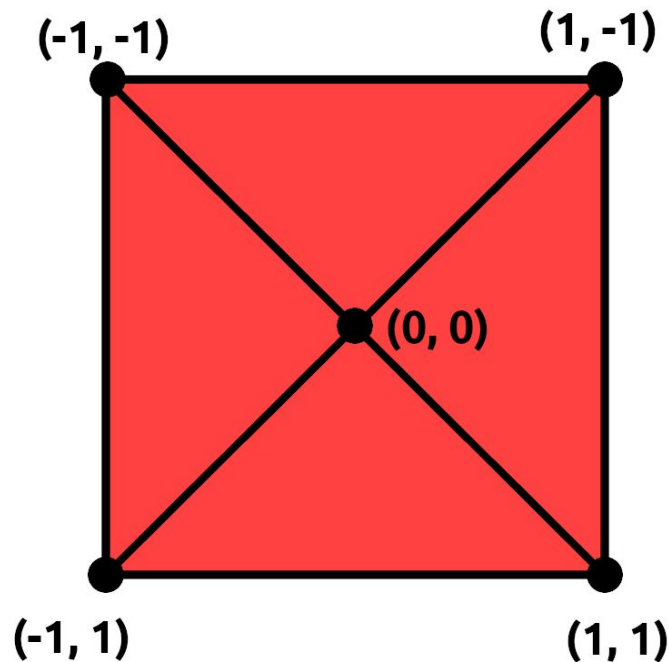
- List of coordinates
- Each vertex is repeated as many times as it participates in a triangle

## 2. Mesh representations

### a. Triangles list

- List of coordinates
- Each vertex is repeated as many times as it participates in a triangle

```
[  
  [ (-1, -1), (1, -1), (0, 0) ],  
  [ (-1, -1), (-1, 1), (0, 0) ],  
  [ (-1, 1), (1, 1), (0, 0) ],  
  [ (1, -1), (1, 1), (0, 0) ]  
]
```



## 2. Mesh representations

### b. Indexed face-set

- List of coordinates
- List of triplets (indices in vertices list)

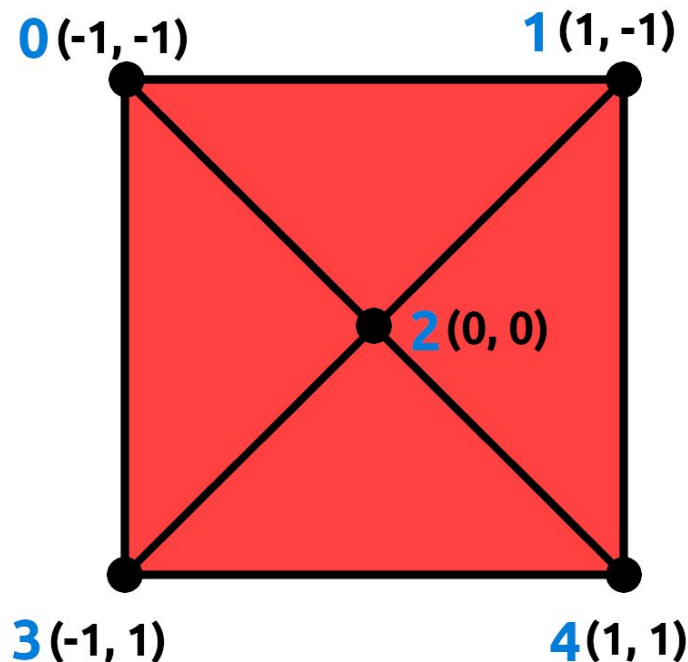
## 2. Mesh representations

### b. Indexed face-set

- List of coordinates
- List of triplets (indices in vertices list)

```
vertices = [  
  (-1,-1), (1,-1), (0,0), (-1,1), (1,1)  
]
```

```
faces = [  
  (0,1,2), (1,2,4), (3,2,4), (0,2,3)  
]
```



## 2. Mesh representations

### c. File formats

	<b>STL</b>	<b>PLY</b>	<b>OBJ</b>
<b>Representation</b>	triangles list	indexed faces	indexed faces
<b>Vertex attributes</b>	No	Yes	Not in the standard
<b>Spread</b>	Extremely	Common	Common

## 2. Mesh representations

### d. Note: Vertex attributes

- Values associated with a vertex:
  - Normal
  - Color
  - Bump
  - ...
- Can include custom values as long as we have a homogeneous structure (== an explicit value for each vertex)

### 3. What are meshes useful for?

#### a. Data visualization

- Meshes are very light → hull around objects
- Meaningful data at low cost

Name Rascasse\_380\_382\_892\_140um.raw.obj

Type TGIF document (application/x-tgif)

Size 66.8 MB (66,777,891 bytes)

Name Rascasse\_140µm.tif

Type TIFF image (image/tiff)

Size 259.1 MB (259,077,909 bytes)

x4

### 3. What are meshes useful for?

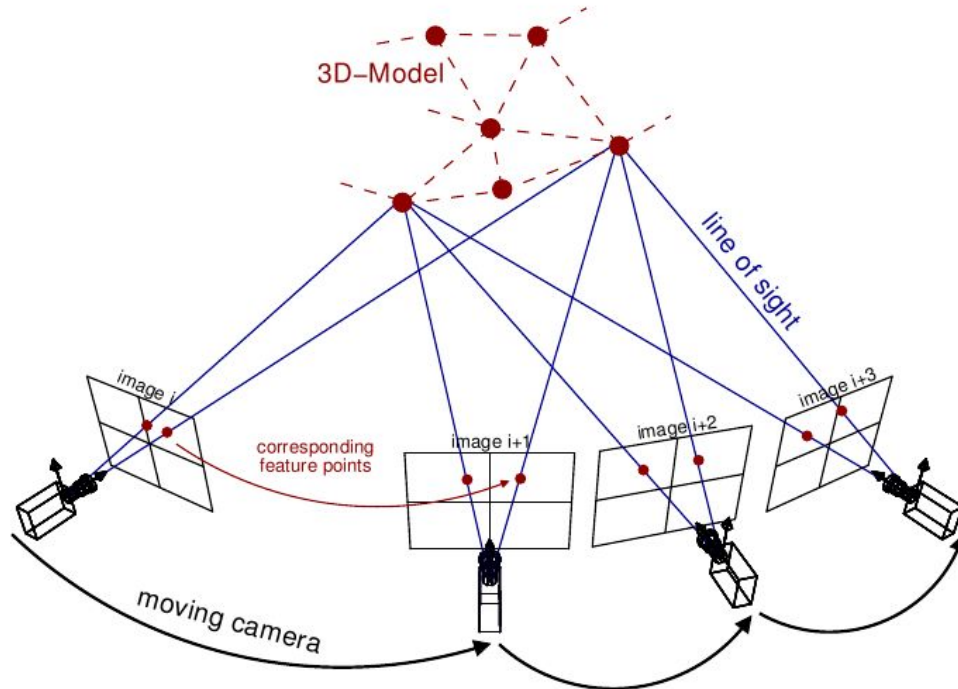
#### b. Other metrics

- Volumes
- 2D surfaces
- Distances between objects
- Topology (roughness, ...)



# 4. Acquisition of surfaces

## a. Photogrammetry



- Start with a set of 2D images
- Identify characteristic points
- Use epipolar geometry to evaluate the position of points according to the camera
- Build a mesh over the points-cloud

# 4. Acquisition of surfaces

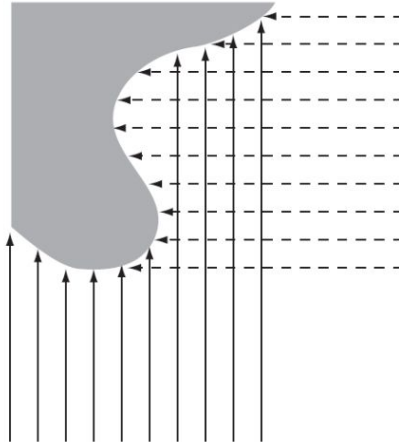
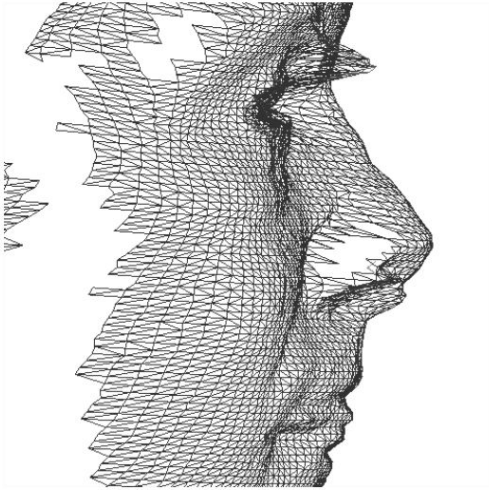
## b. Laser scanning



- Travel time measured to acquire a points cloud.

# 4. Acquisition of surfaces

## b. Laser scanning



- Travel time measured to acquire a points cloud.
- May contain holes in highly angular areas.

## 5. From voxels grid to mesh

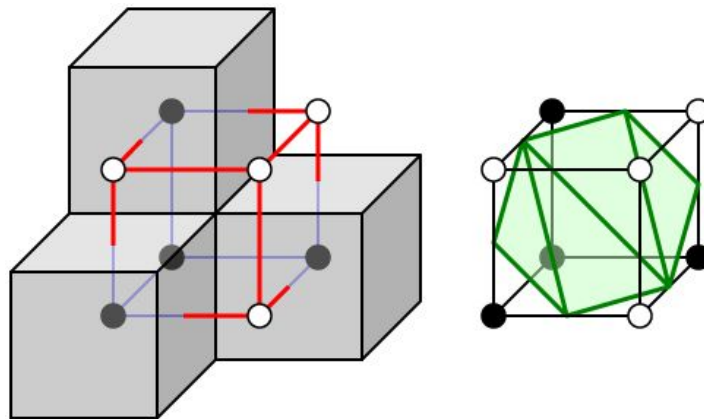
- Extraction of isosurface: common features (ex: intensity)

## 5. From voxels grid to mesh

- Extraction of isosurface: common features (ex: intensity)
- Mask (or labels) to mesh: marching cube algorithm

## 5. From voxels grid to mesh

- Extraction of isosurface: common features (ex: intensity)
- Mask (or labels) to mesh: marching cube algorithm
- Creates an imaginary box:
  - Which corners are in the background
  - Which corners are in the background



# 5. From voxels grid to mesh

- Extraction of isosurface: common features (ex: intensity)
- Mask (or labels) to mesh: marching cube algorithm
- Every possible case is hardcoded

