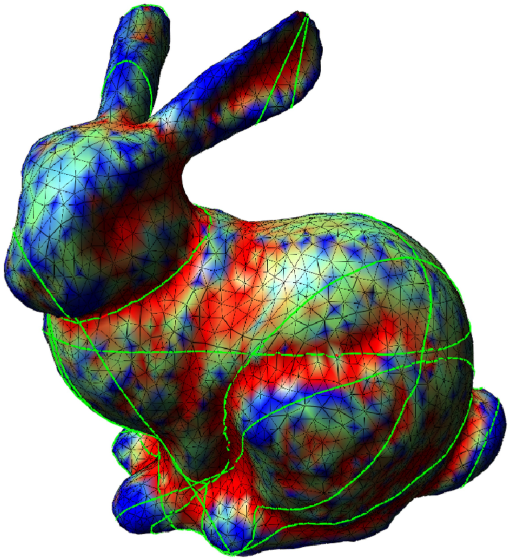
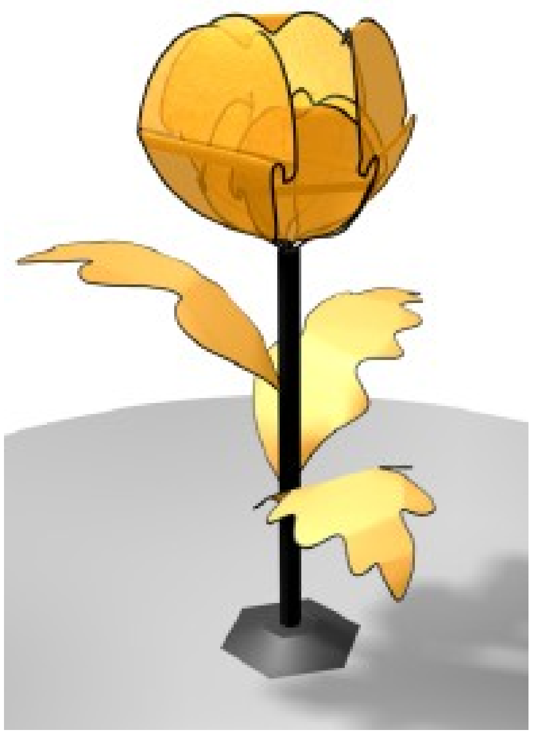
Creating and processing 3D geometry



Marie-Paule Cani
Marie-Paule.Cani@imag.fr

Cédric Gérot
Cedric.Gerot@gipsa-lab.inpg.fr

Franck Hétroy
Franck.Hetroy@imag.fr



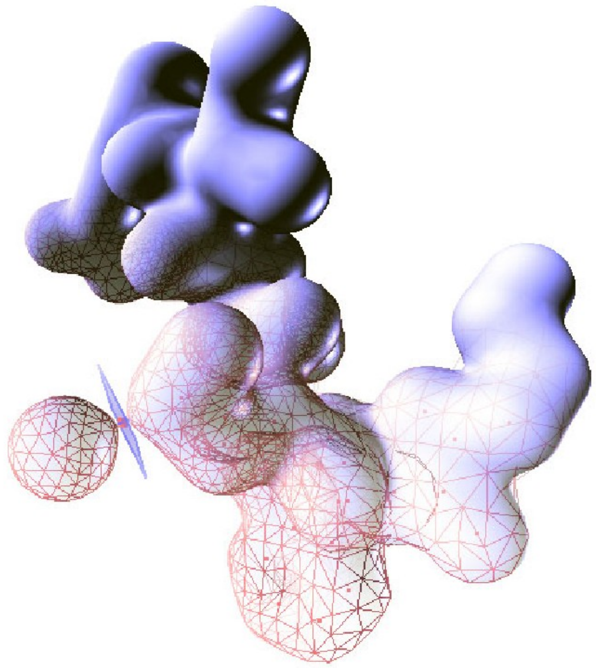
http://evasion.imag.fr/Membres/Franck.Hetroy/Teaching/Geo3D/

Context: computer graphics

- We want to represent objects
  - Real objects
  - Virtual/created objects
- Several ways for virtual object creation
  - Interactive by graphists
  - Automatic from real data
    - 3D scanner, medical angiography, ...
  - Procedural (on the fly)
    - Complex scenes, terrain, ...
- Different uses
  - Display, animation, physical simulation, ...

Course overview

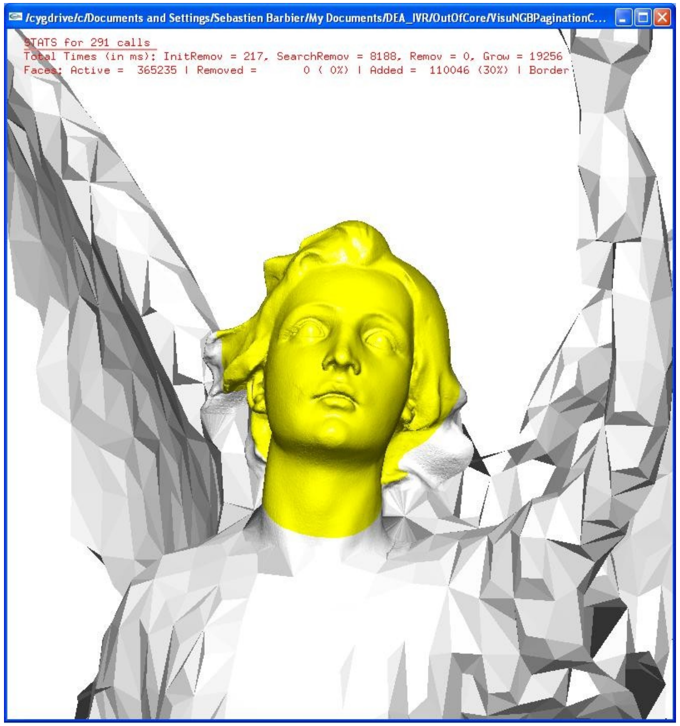
1. Objects representations
   - Volume/surface, implicit/explicit, ...



Real-time triangulation of implicit surfaces

Course overview

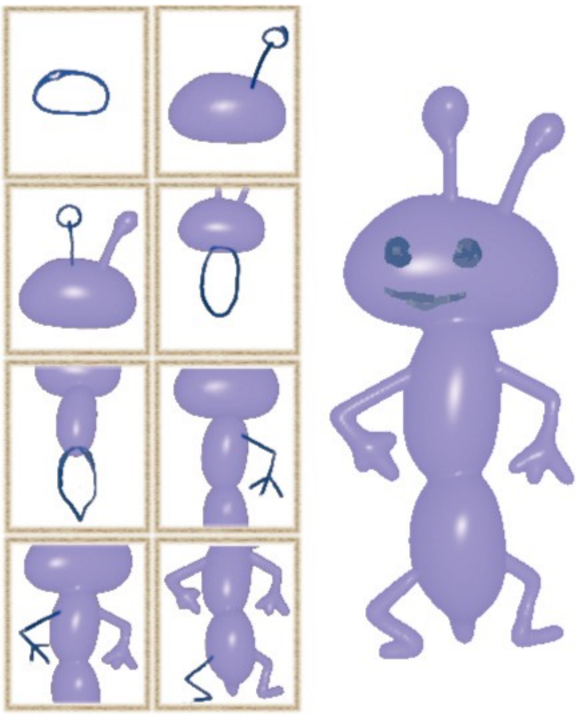
1. Objects representations
   - Volume/surface, implicit/explicit, ...
2. Geometry processing
   - Simplify, smooth, ...



Interactive multiresolution surface exploration

Course overview

1. Objects representations
   - Volume/surface, implicit/explicit, ...
2. Geometry processing
   - Simplify, smooth, ...
3. Virtual object creation
   - Surface reconstruction, interactive modeling



Shape modeling by sketching

Planning (provisional)

Part I – Geometry representations

- Lecture 1 – Oct 9th – FH
  - Introduction to the lectures; point sets, meshes, discrete geometry.
- Lecture 2 – Oct 16th – MPC
  - Parametric curves and surfaces; subdivision surfaces.
- Lecture 3 – Oct 23rd - MPC
  - Implicit surfaces.

Planning (provisional)

Part II – Geometry processing

- Lecture 4 – Nov 6th – FH
  - Discrete differential geometry; mesh smoothing and simplification (paper presentations).
- Lecture 5 – Nov 13th - CG + FH
  - Mesh parameterization; point set filtering and simplification.
- Lecture 6 – Nov 20th - FH (1h30)
  - Surface reconstruction.

Planning (provisional)

Part III – Interactive modeling

- Lecture 6 – Nov 20th – MPC (1h30)
  - Interactive modeling techniques.
- Lecture 7 – Dec 04th - MPC
  - Deformations; virtual sculpting.
- Lecture 8 – Dec 11th - MPC
  - Sketching; paper presentations.

Books

For my part of the course:

- M. Botsch et al., “Geometric Modeling Based on Polygonal Meshes”, SIGGRAPH 2007 Course Notes.

http://graphics.ethz.ch/~mbotsch/publications/sg07-course.pdf

!!! Also test the source code:

http://graphics.ethz.ch/~mbotsch/publications/meshcourse07_code.tgz

Books

For Marie-Paule's part of the course:

- D. Bechmann, B. Péroche eds., “Informatique graphique, modélisation géométrique et animation”, Hermès, 2007.
  - Geometry representations
- M. Alexa et al., “Interactive shape modelling”, Eurographics 2005 Tutorial.
  - Interactive modeling

Factual information

- 9h-10h30+10h45-12h15
- This room (008)
- Mark:
  - 1 final written exam (1/2)
  - Geometry processing paper presentation + demo (1/4)
  - Interactive modeling paper presentation (1/4)

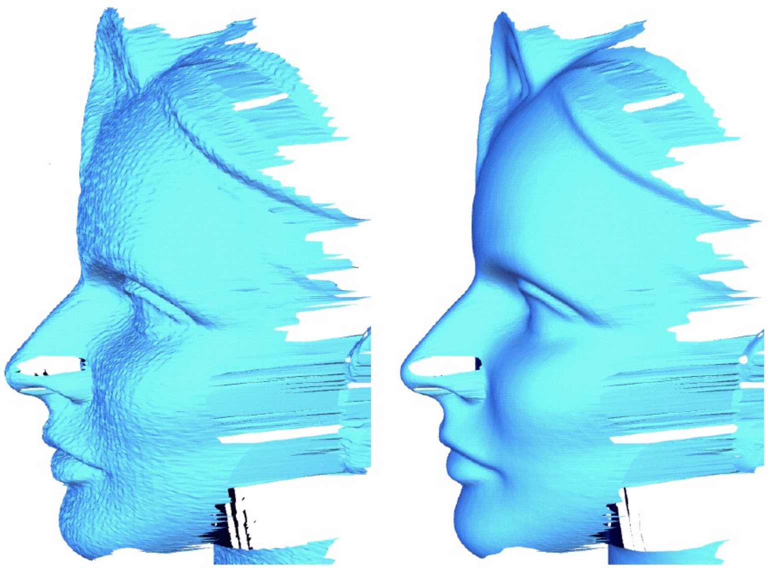
Geometry processing paper

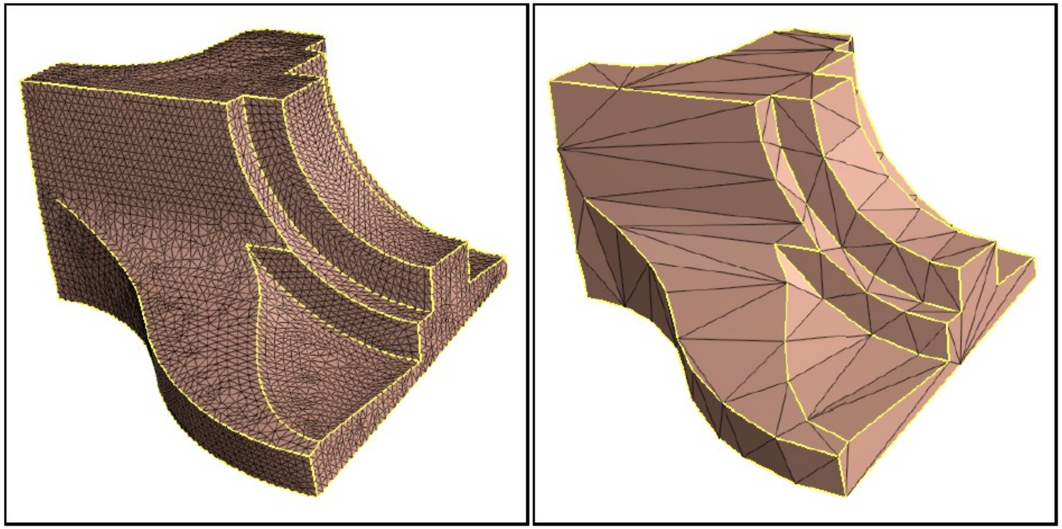
- By groups of 2 students
- You are asked to:
  - Choose a paper among the proposed ones
  - Prepare a short presentation (10 minutes + 5 minutes for questions), which includes a demo
- PDF files and basic interface and data structures on the course's webpage:

http://evasion.imag.fr/Membres/Franck.Hetroy/Teaching/Geo3D

Proposed papers

- Two topics
  - Mesh smoothing (3 papers)
  - Mesh simplification (3 papers)





- Send an e-mail to Franck.Hetroy@imag.fr when chosen
- Presentations: November, 6th

Mesh smoothing papers

1. G. Taubin, “A Signal Processing Approach to Fair Surface Design”, SIGGRAPH 1995.
2. M. Desbrun et al., “Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow”, SIGGRAPH 1999.
3. S. Fleishman et al., “Bilateral Mesh Denoising”, SIGGRAPH 2003

+T.R. Jones et al., “Non-Iterative, Feature-Preserving Mesh Smoothing”, SIGGRAPH 2003.

Mesh simplification papers

1. H. Hoppe, “Progressive Meshes”, SIGGRAPH 1996.
2. R. Klein et al., “Mesh Reduction with Error Control”, Visualization 1996.
3. P. Lindström, “Out-of-Core Simplification of Large Polygonal Models”, SIGGRAPH 2000 => incl. M. Garland & P. Heckbert, “Surface Simplification Using Quadric Error Metrics”, SIGGRAPH 1997.

Today's planning

1. Introduction to the course
2. Geometry representations: introduction
3. Point sets
4. Meshes
5. Discrete geometry

Today's planning

1. ~~Introduction to the course~~
2. Geometry representations: introduction
3. Point sets
4. Meshes
5. Discrete geometry

Geometry representations

- Today:
  - Point sets
  - (Flat) Meshes
  - Voxels
- Next week:
  - Parametric curves and surfaces (splines, ...)
  - Multiresolution meshes
- In two weeks:
  - Implicit surfaces

Geometry representations

- A good introduction to all these representations is in chapter 2 of Botsch et al.'s book
  - Parametric/explicit surfaces: splines, subdivision surfaces, triangle meshes
  - Implicit surfaces
  - Conversion from one rep. to the other
  - Only about surfaces: ~~point sets volumetric rep.~~

Why not one good representation ?

- Multiple applications, different constraints
  - Powerful rep.
    - To handle a large class of objects
    - To create complex objects from simple ones
  - Intuitive rep.
    - To edit the model
    - To animate some parts of it
  - Efficient rep.
    - Memory cost
    - Display/process time cost

Classification: a proposal

- Non structured rep.
  - Point set
  - Polygon soup
- Surface rep.
  - Mesh
  - Parametric
  - Subdivision
  - Implicit
- Volumetric rep.
  - Voxel line/plane/set
  - Octree
  - CSG
- Procedural rep.
  - Fractal
  - Grammar/L-system
  - Particle system
- Image-based rep.

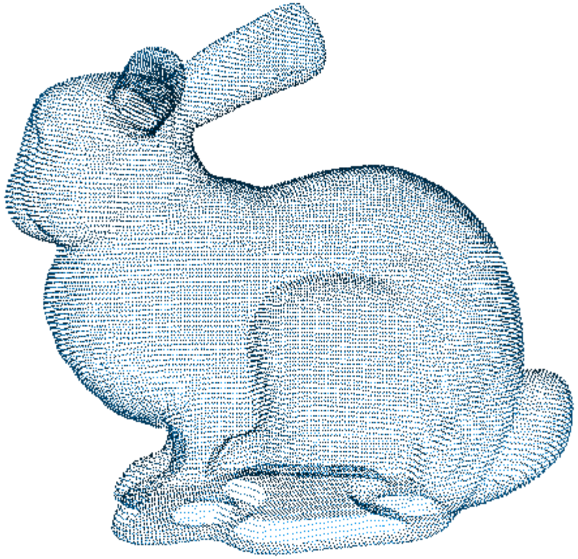
Today's planning

1. ~~Introduction to the course~~
2. ~~Geometry representations: introduction~~
3. Point sets
4. Meshes
5. Discrete geometry

Point sets

- Result of scanner acquisition
- Also image-based modeling
- Main advantages:
  - “Natural” representation
  - Simple and cheap to display
- Main drawbacks:
  - No connectivity info: underlying shape = ?
  - Tedious to edit

NextEngine scanner: soon here!

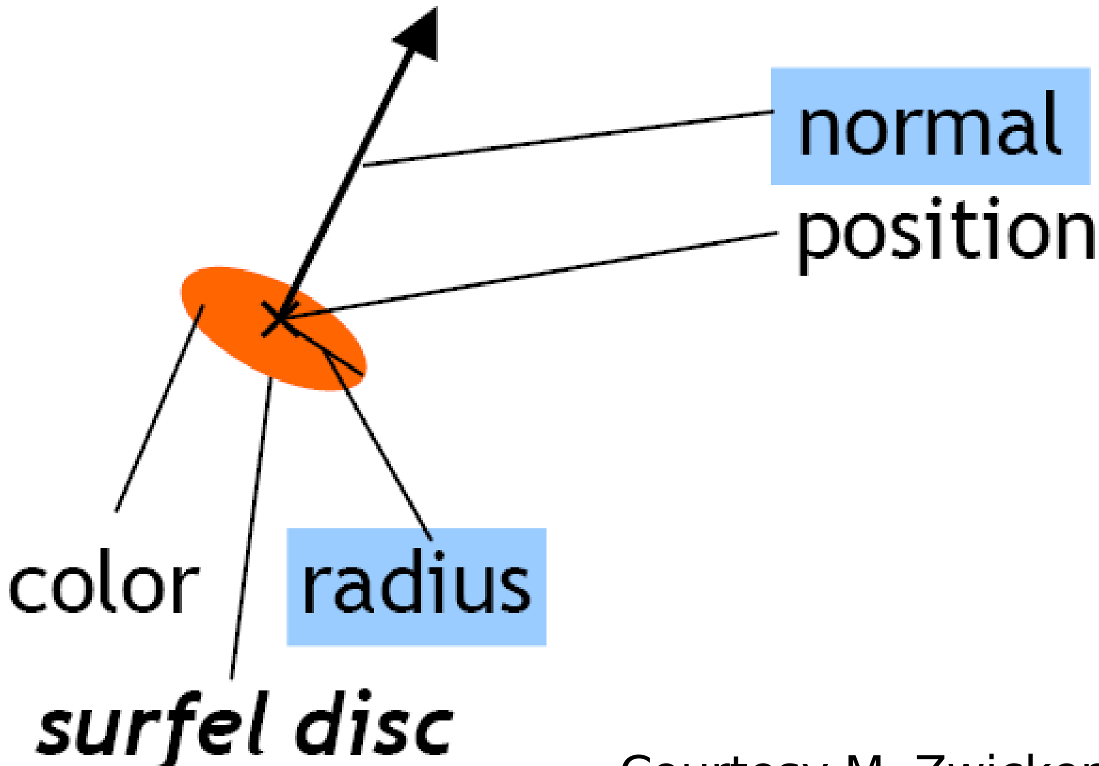


Too simple ?

- If nb of points too low: holes
- However:
  - Currently scanned models have up to several millions points
  - Mesh reconstruction is then time-consuming
  - Memory to store the mesh also a problem (number of faces ~ 2 x number of points)
  - Each face projects onto only one or two pixels !
- That is why surface representation by a point set is more and more used and studied

Point set representation

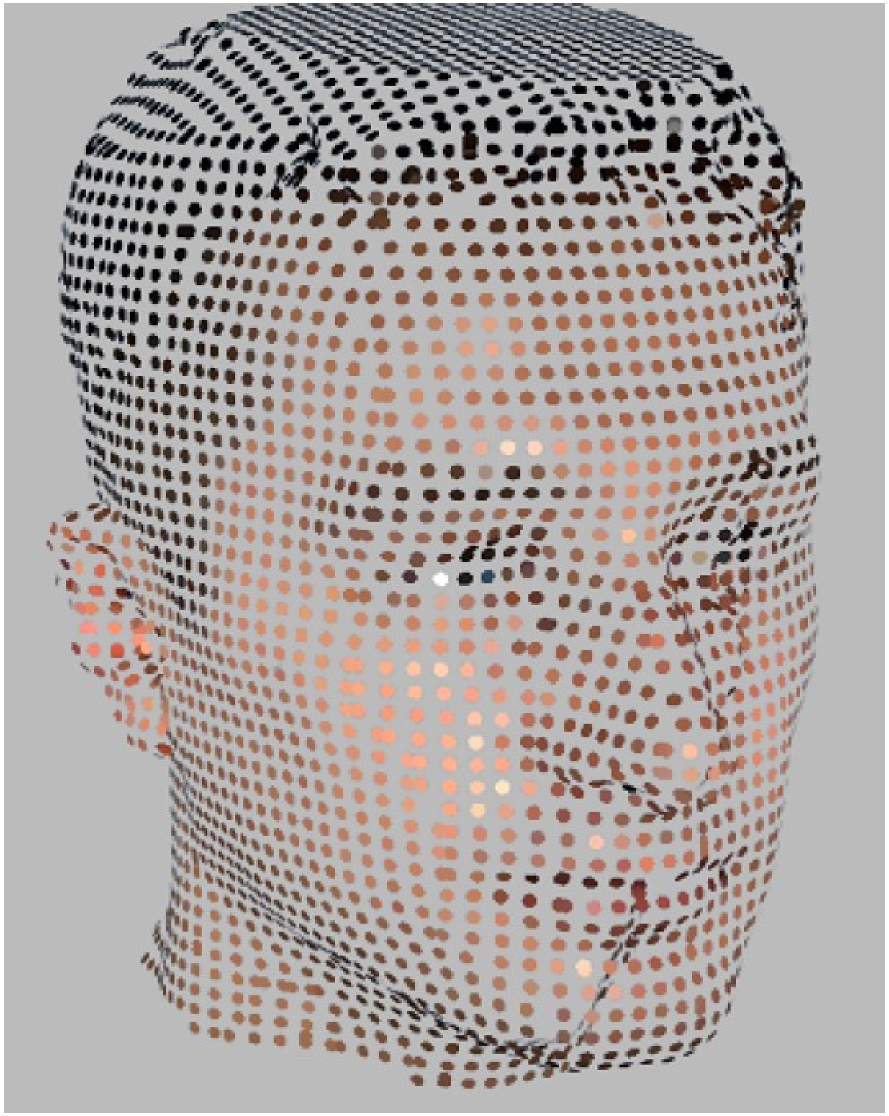
- Points are samples of the underlying surface
- 1 point corresponds to 1 surfel (surface element)
  - Position
  - Color
  - Normal
  - Radius
- Surfel = 2D !



Courtesy M. Zwicker

Surfel

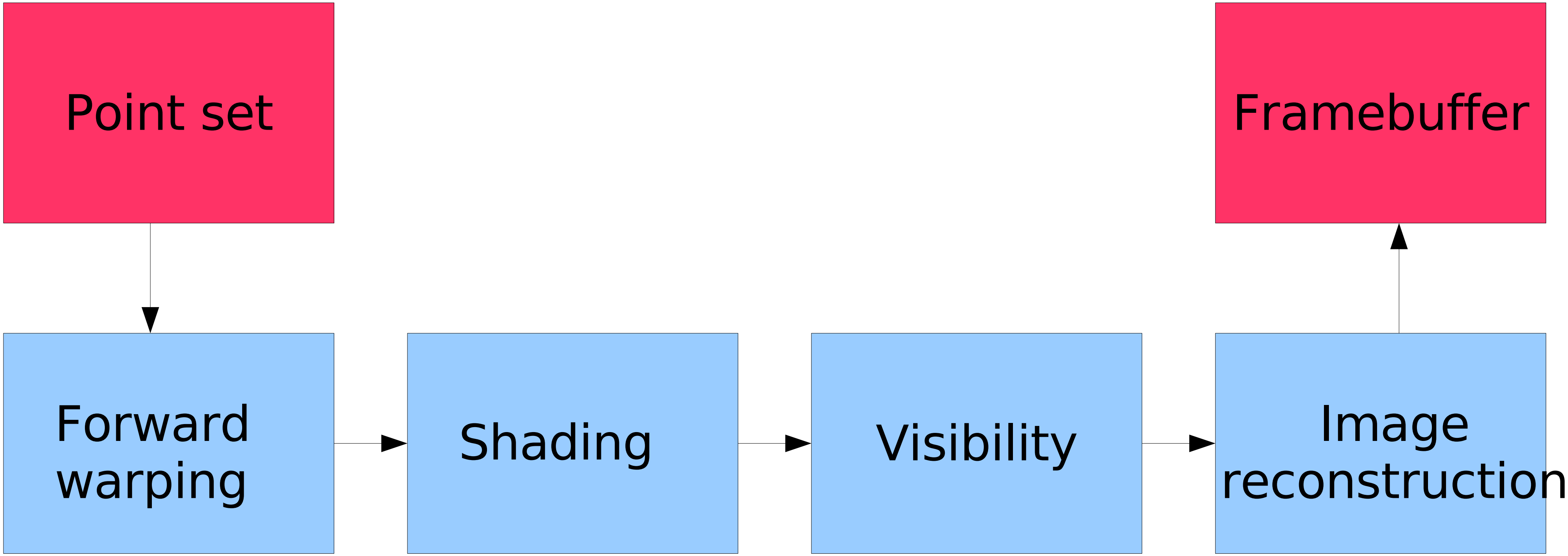
- Surfels are designed mostly for rendering
- Advantage: no mesh reconstruction necessary
  - Time saving
- No surface connectivity information



Courtesy M. Zwicker

Point rendering pipeline

- Credit: M. Zwicker 2002



Forward warping and shading

- Forward warping = perspective projection of each point in the point cloud
- Similar to projection of triangle vertices (mesh case)
- Shading:
  - Per point
  - Conventional models for shading (Phong, Torrance-Sparrow, reflections, etc.)
  - Cf. rendering course

Visibility and image reconstruction

- Performed simultaneously
- Discard points that are occluded from the current viewpoint
- Reconstruct continuous surfaces from projected points

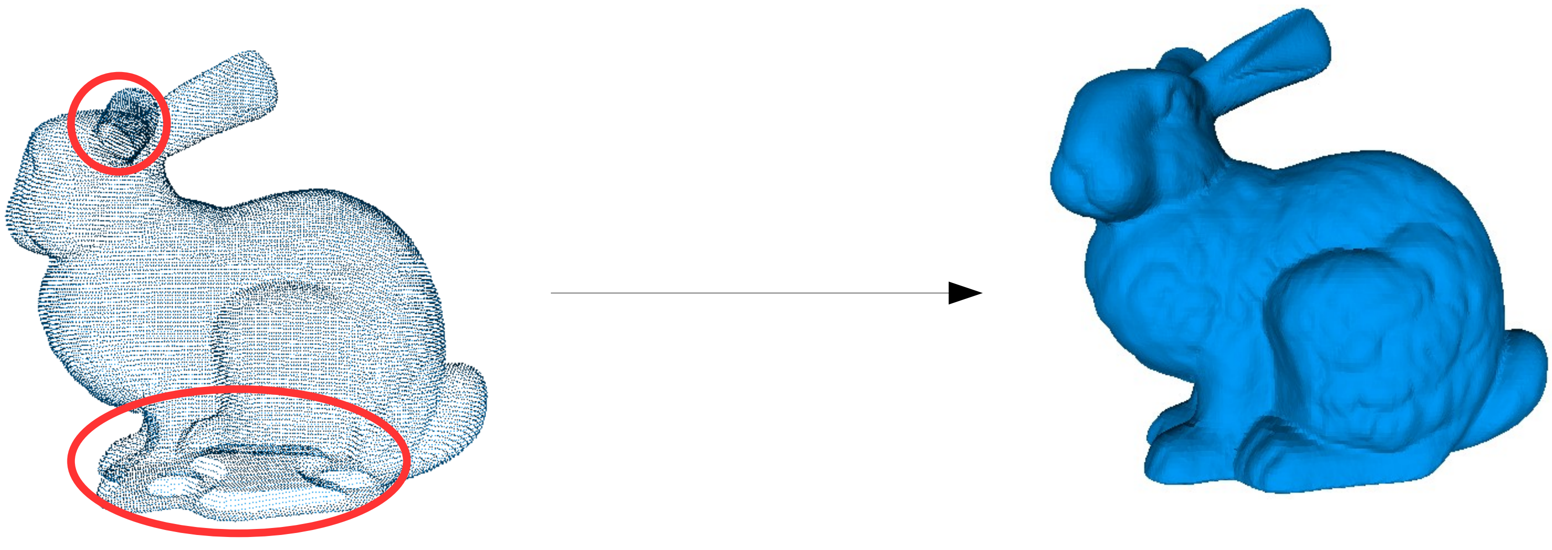
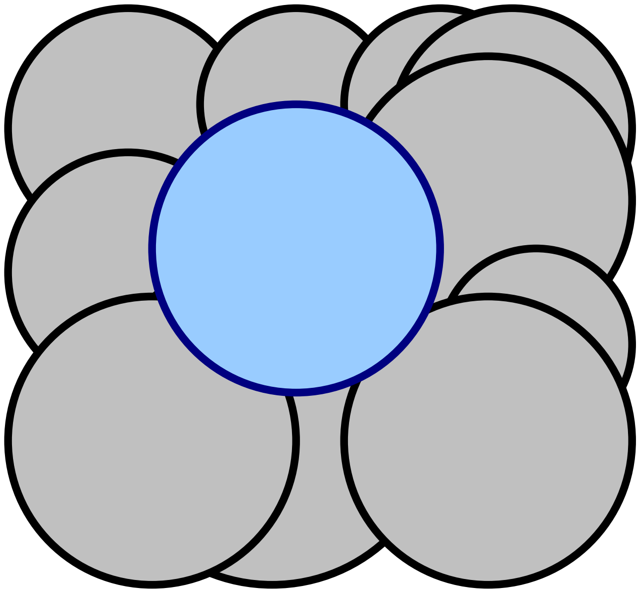


Image reconstruction

- Goal: avoid holes in the image of the surface
- Use surfel radius to cover the surface
- More during the rendering course



Point set processing

- Some work on:
  - Simplification
  - Filtering
  - Decomposition, resampling
- Still lack of robust mathematical theory
  - Cf. the mesh case (session 4)
- (Possible) Topic of the session 5 of this course

Surface approximation

- Almost all other surface representations are based on points
  - Meshes
  - Parametric rep. (splines)
  - Implicit rep.
- A projection-based surface definition is also possible
  - Local polynomial P around each point
  - Project P(0) onto a local reference plane

Books

- M. Alexa et al., “Point-Based Computer Graphics”, SIGGRAPH 2004 Course Notes

  http://graphics.ethz.ch/publications/tutorials/points/

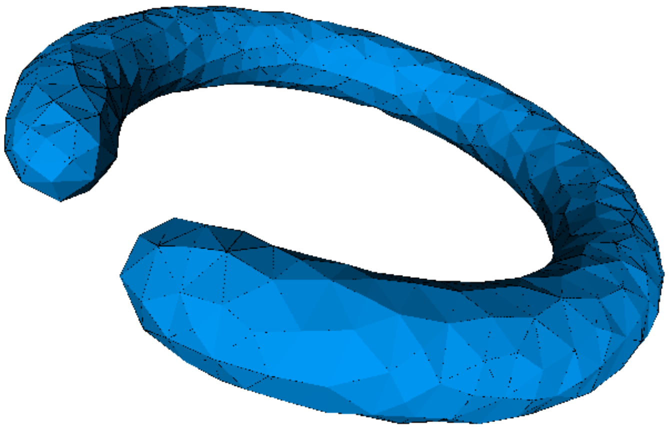
- C. Schlick, P. Reuter, T. Boubekeur, “Rendu par Points”, chapter from “Informatique Graphique et Rendu“, Hermès, 2007
- See also works by Mark Alexa (TU Berlin), Markus Gross et al. (ETH Zürich), Gaël Guennebaud et al. (IRIT Toulouse, now ETH Zürich)

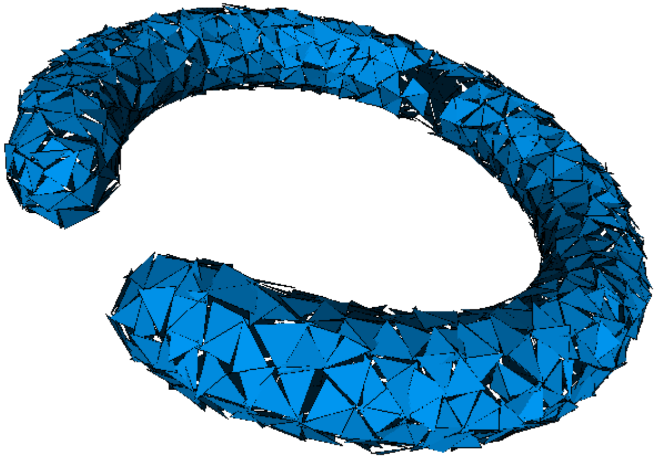
Today's planning

1. ~~Introduction to the course~~
2. ~~Geometry representations: introduction~~
3. ~~Point sets~~
4. Meshes
5. Discrete geometry

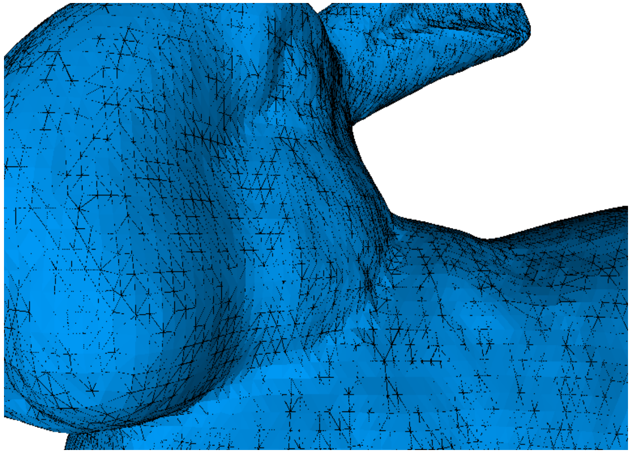
Meshes

- Mesh = (V,E,F)
  - V = set of vertices
  - E = set of edges
  - F = **connected** set of (planar) faces
- Not connected = polygon soup
- Faces can be
  - Triangles
  - Planar quads
  - Any planar, convex polygon

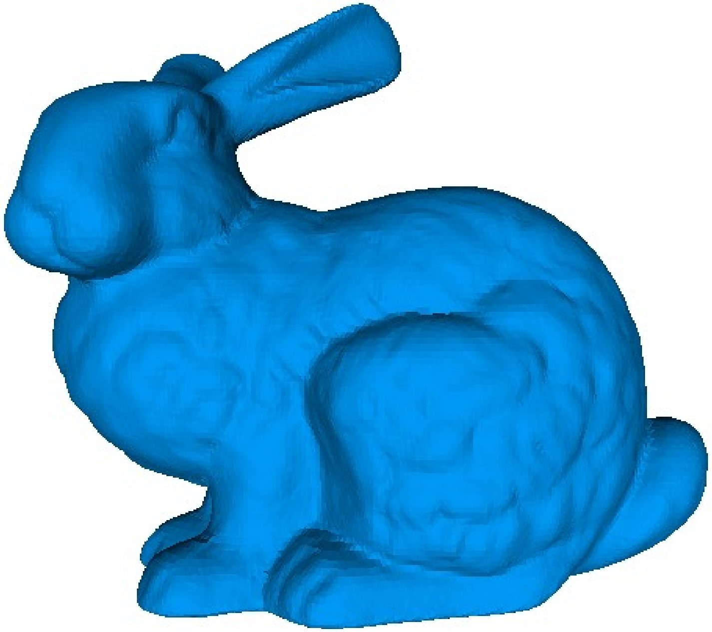




Meshes

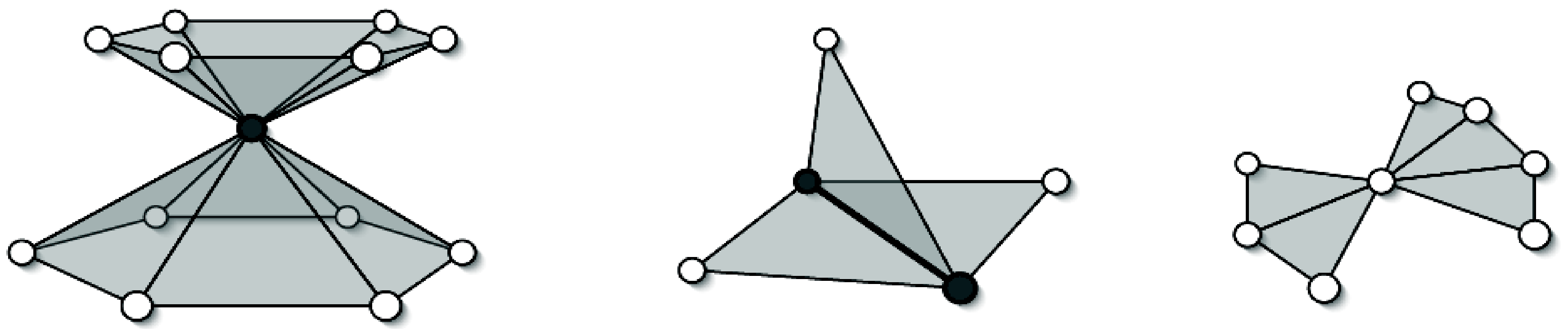


- Main advantage: easy display
- Main drawback: tedious to edit
- Represent continuous piecewise linear surfaces
- Encode
  - (Approximate) geometry
    - OK for planar shapes (CAD)
    - Bad for curved shapes
  - Topology (see 2 slides after)



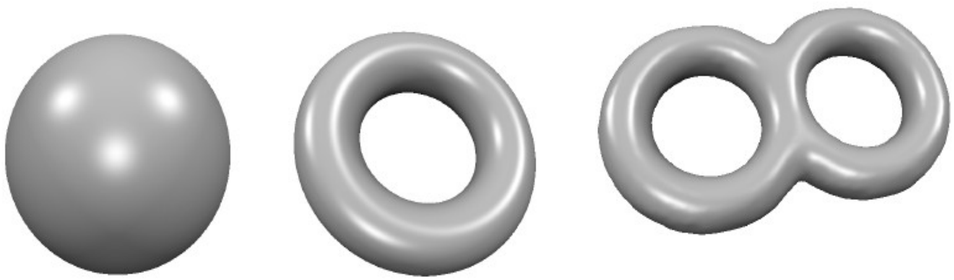
2-Manifold

- Def.: each vertex has a neighborhood on M homeomorphic to a disk
  - Continuous bijection, distance does not matter
- 2-Manifold with boundary: to a [half-]disk
- 3-Manifold, n-manifold, ...
- No singularities:



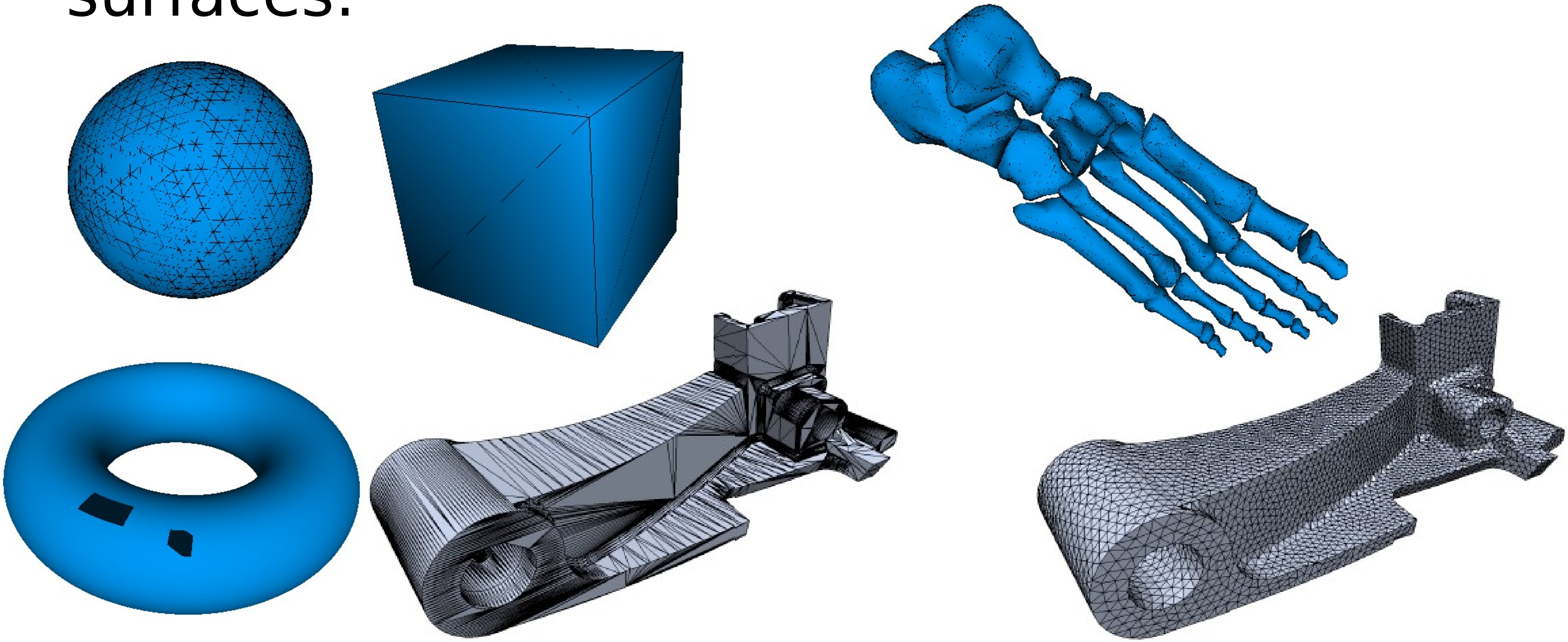
Object topology

- Any manifold's topology is defined by a small set of numbers:
  - Surface: nb c of connected components + nb g of holes + nb b of boundaries
  - Volume: nb of conn. comp. + nb of tunnels + nb of cavities (bubbles) + nb of boundaries
- Euler formula for surface meshes:
  - $V - E + F = \chi = 2(c-g)-b$
  - χ = Euler characteristic
  - g = genus



(Easy) Exercise

- Find the Euler characteristic of the following 6 surfaces:



- And for volumes ?

Mesh data structures

- Ref.: chapter 3 of Botsch et al.'s book
- How to store geometry <u>and</u> connectivity ?
  - STL-like: store triangles, vertices duplicated

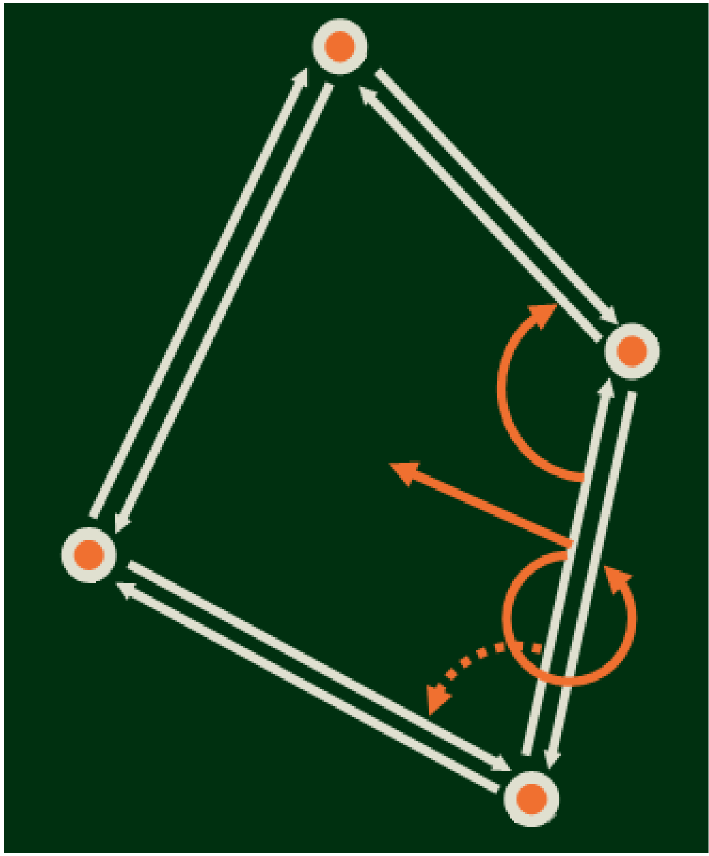
    => no connectivity
  - Shared vertex data structure (OBJ, OFF file formats): vertex list, triangles = triples of indices

    => no neighborhood info
  - Half-edge and variants

    => all is based on **oriented** edges

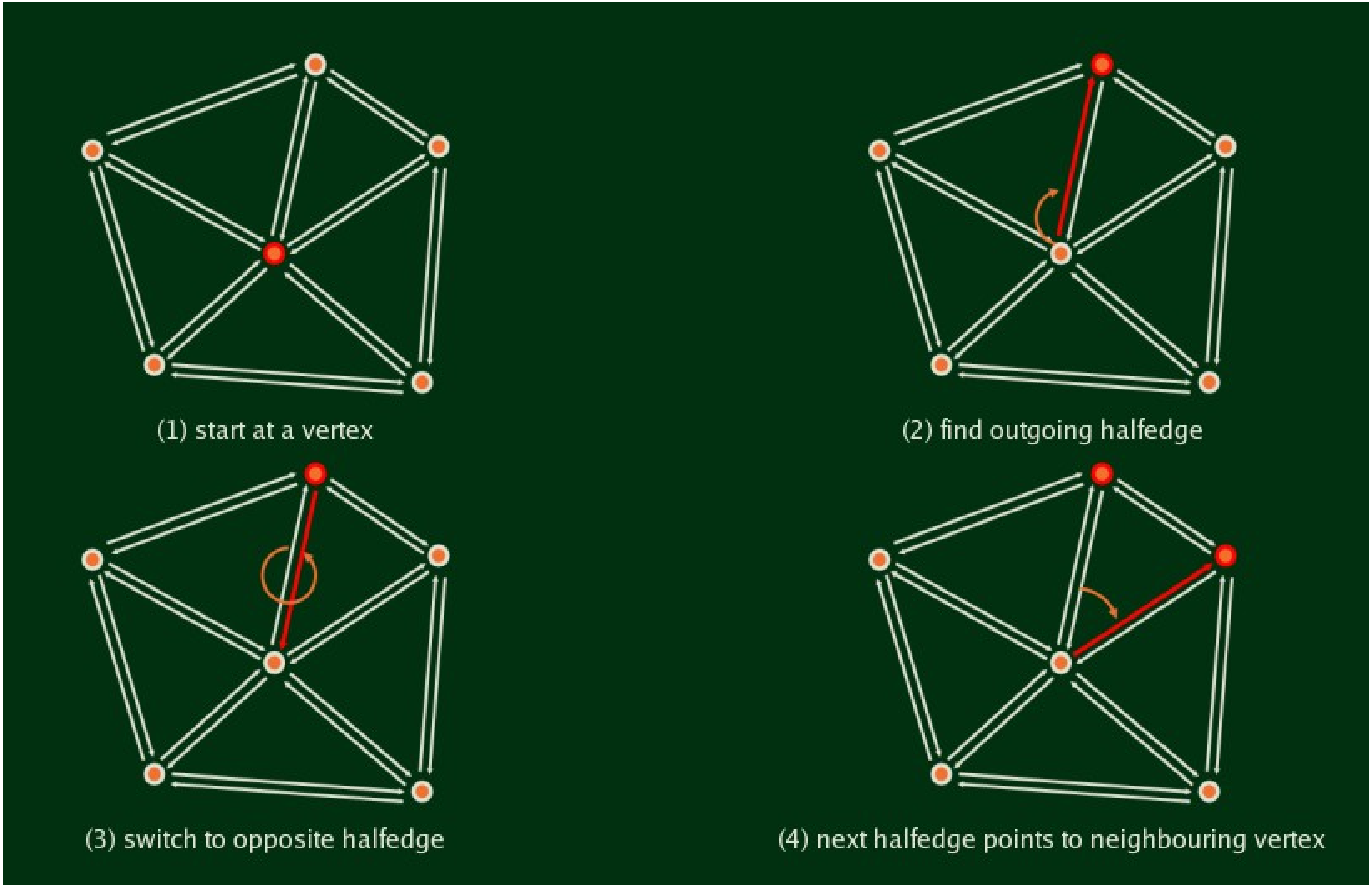
Half-edge data structure

- Three main classes:
  - Vertex
    - Coord, [id,] pointer to **one** outgoing half-edge
  - Half-edge
    - Pointers to the **origin** vertex, to the **next** and to the **opposite** half-edge, to the incident face
  - Face
    - Pointer to **one** incident half-edge



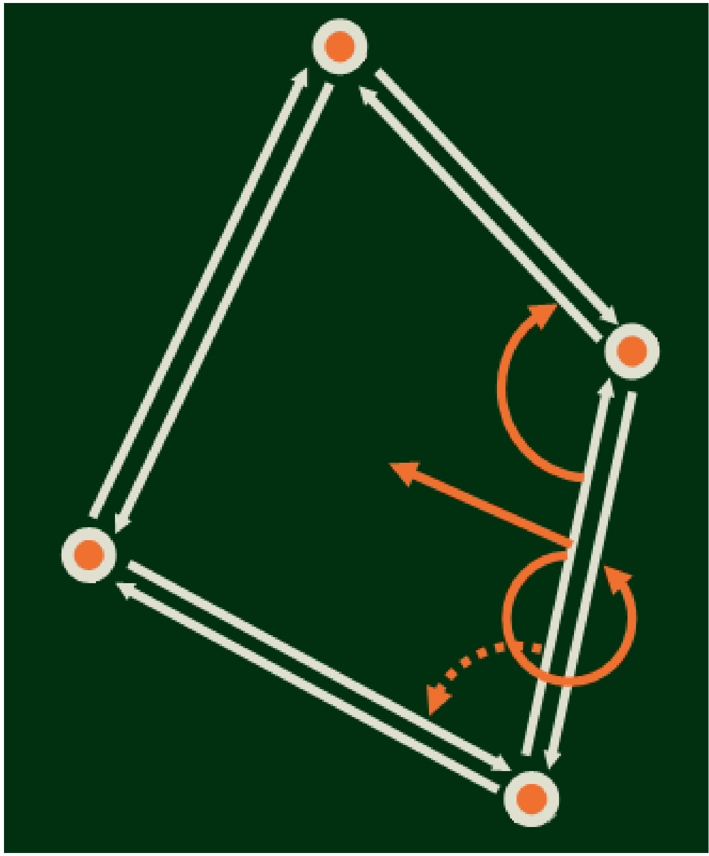
You can add whatever attributes you want (normal, color, ...)

Example: browsing the 1-ring neighborhood of a vertex



Example: browsing the 1-ring neighborhood of a vertex

- Exercise:
  - Write your own half-edge data structure:
    - class Vertex
    - class Edge
    - class Face
  - Write a procedure browseOneRing(Vertex* v) which returns the 1-ring neighborhood of v as a list.



C++ libraries

- CGAL http://www.cgal.org/
  - Developed by a consortium led by INRIA, lots of stuff
  - Widely used by researchers, tutorials
  - Somehow complicated (genericity)
- OpenMesh http://www.openmesh.org/
  - Developed by Mario Botsch at RWTH Aachen
  - Simpler, clearer
  - Lack of documentation
- GTS http://gts.sourceforge.net/
  - Why not ?

Mesh processing

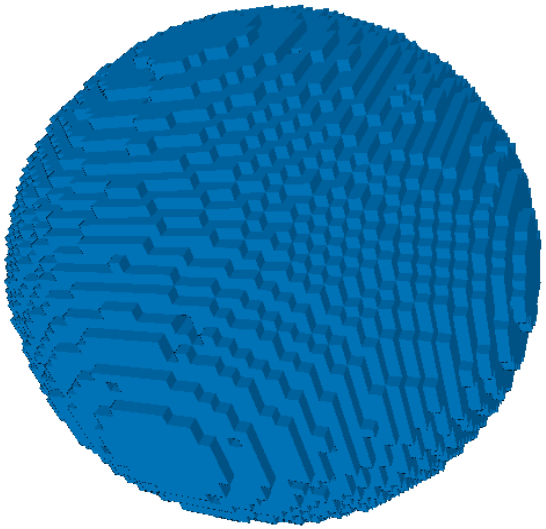
- Lots of work
  - Simplification
  - Smoothing, fairing
  - Parameterization
  - Remeshing
  - Deformation
- See Botsch et al.'s book
- Topic of the sessions 4 and 5 of this course

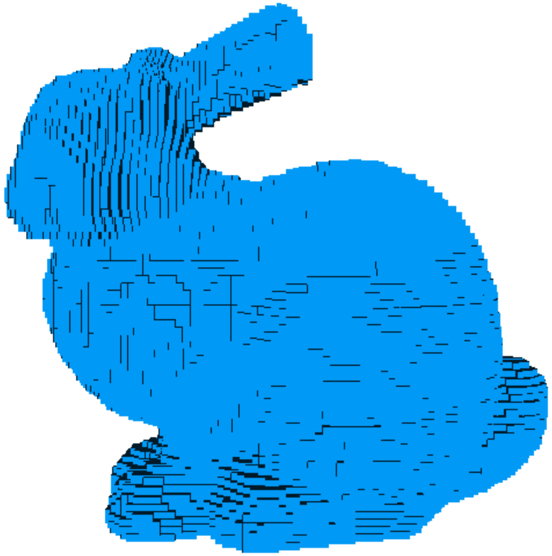
Today's planning

1. ~~Introduction to the course~~
2. ~~Geometry representations: introduction~~
3. ~~Point sets~~
4. ~~Meshes~~
5. Discrete geometry

Voxels

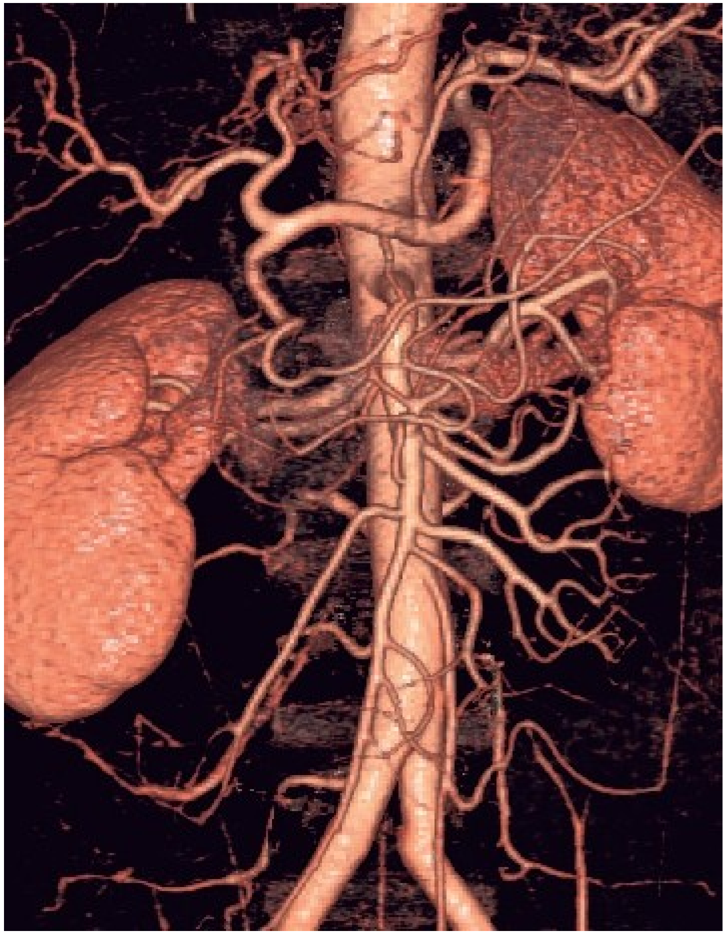
- Volumetric representation
- (Regularly) discretize the 3D space and only keep elements inside the object
- 2D : pixel = PICTure ELement
- 3D : voxel = VOlume ELement
- And also: surfel (surface), texel (texture), ...





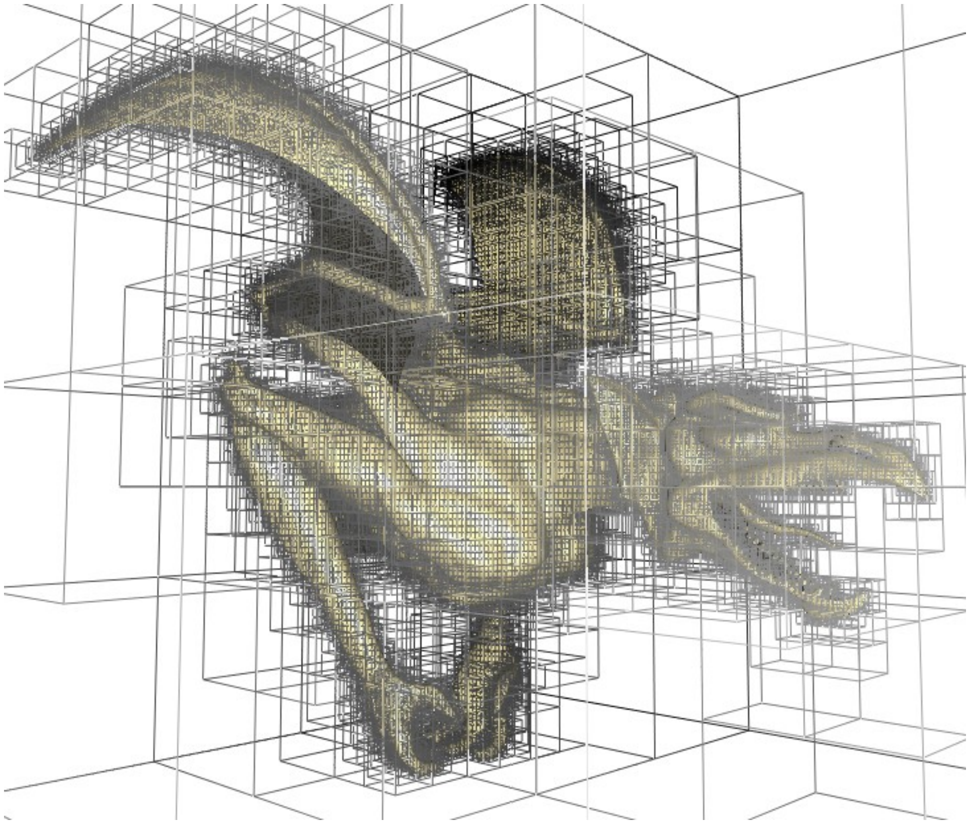
Voxel set acquisition

- Using a function sampled on a grid
  - Numerical simulation
- Tomographic reconstruction (CT scan)
  - Medical area
- Depending on the acquisition/application, voxels contain scalar values (function, density, color, ...)



Octree

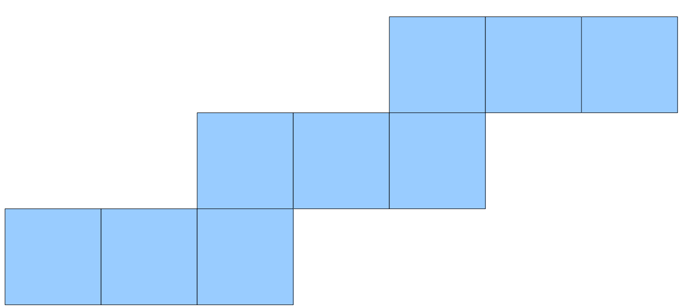
- Voxel hierarchy
- Saves memory
- Interesting for:
  - Spatial queries
  - Collision detection
  - Hidden surface removal (“view frustrum culling”)

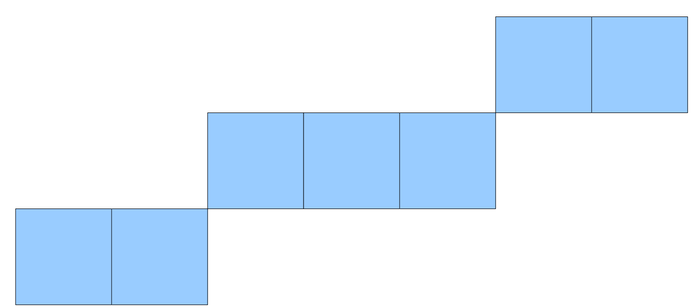


Courtesy S. Lefebvre

An introduction to discrete geometry

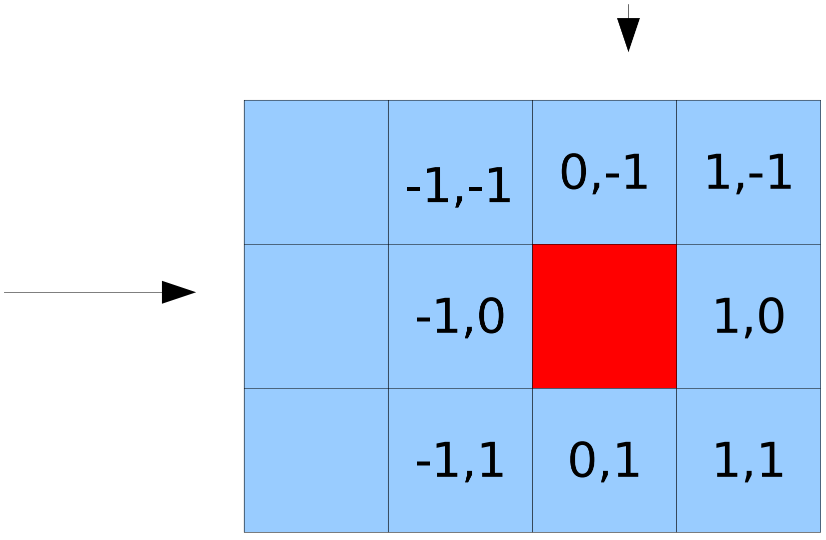
- Theoretical/Mathematical study of **regular** 2D/3D (simple) objects
  - Sampled on a grid
  - Object = point, line, plane
- How to define what is a line of voxels ?
- Adapted algorithms





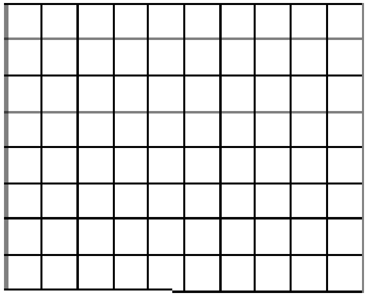
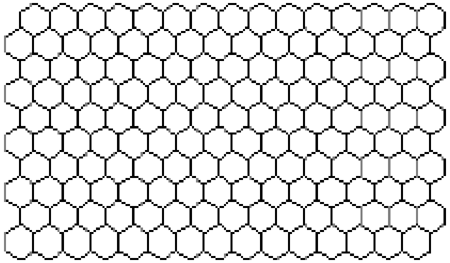
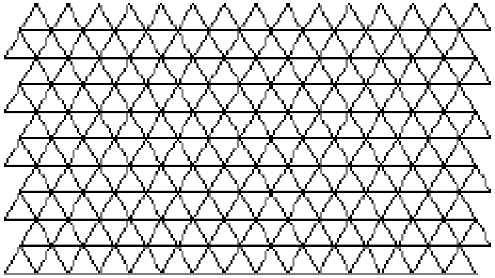
Why a regular grid

- Simple topology
- Easy address to a cell: coordinates
- Easy access from a cell to its neighbors
- Physical reality (sensors)



Cell

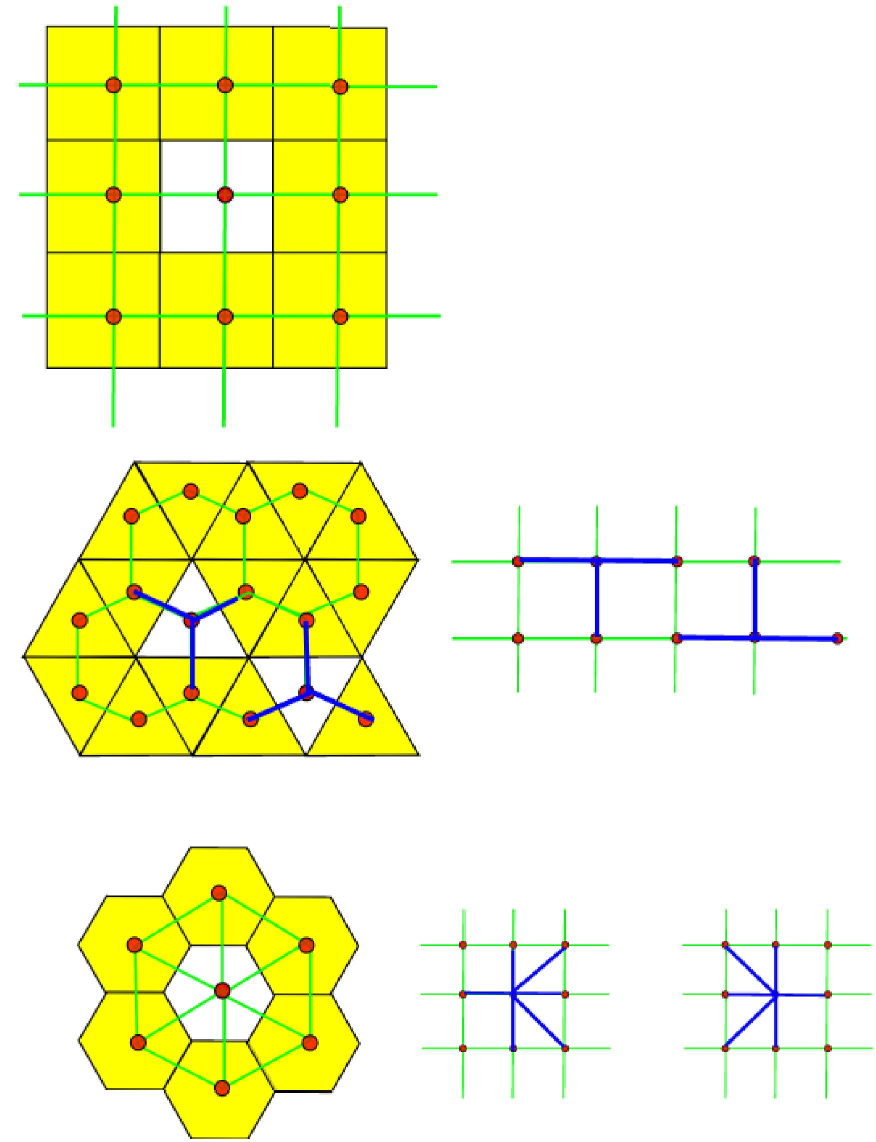
- Usually a convex polygon/polyhedron
- Regular
- The 3 principal cases: square/cube, hexagon/hexahedron, triangle/tetrahedron

Courtesy D. Coeurjolly & I. Sivignon

Advantage of squares/cubes

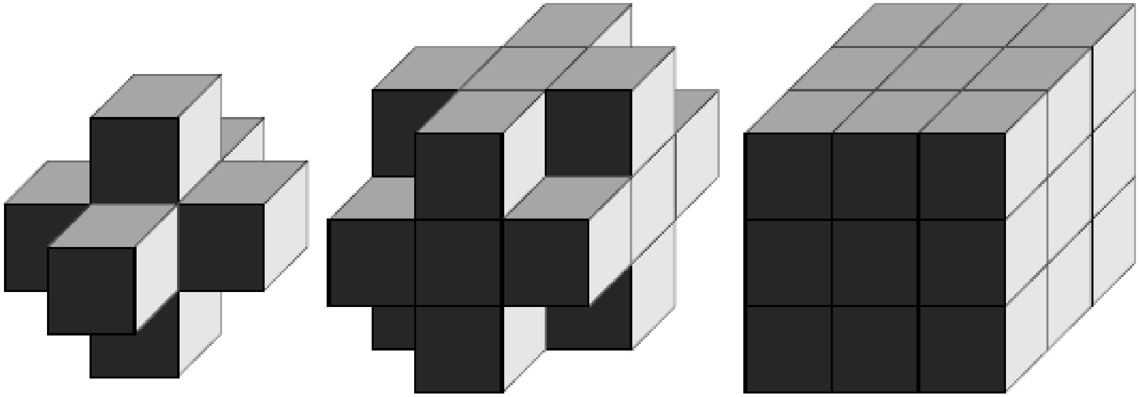
- Square:
  - 4 neighbors
  - 1 configuration
- Triangle:
  - 3 neighbors
  - 2 configurations
- Hexagon:
  - 6 neighbors
  - 2 configurations



Courtesy D. Coeurjolly & I. Sivignon

Adjacency on a voxel grid

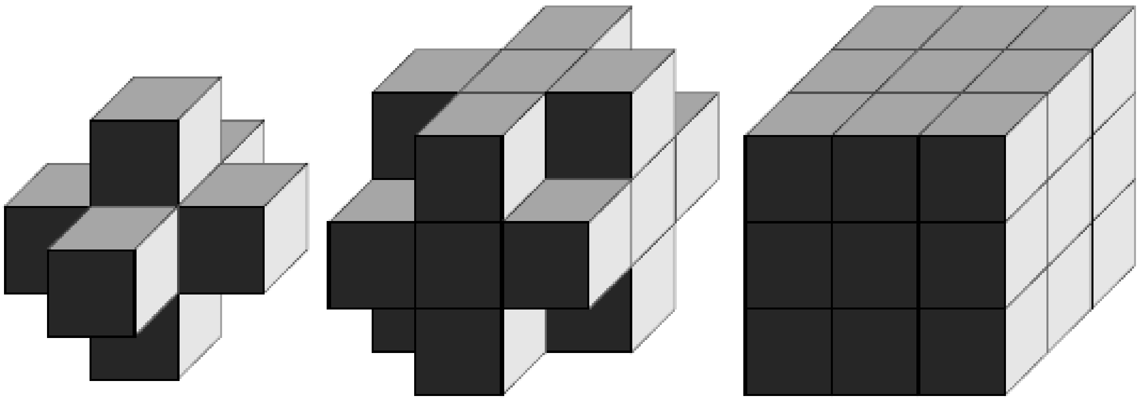
- (Combinatorial) Def.:
  - 6-neighbors = voxels that share a face
  - 18-neighbors = voxels that share a edge
  - 26-neighbors = voxels that share a vertex



Courtesy D. Coeurjolly & I. Sivignon

Adjacency on a voxel grid

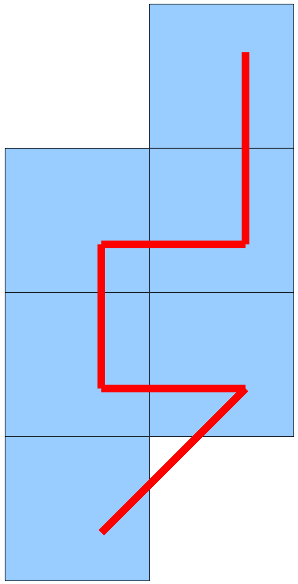
- (Topological) Def.:
  - 2-neighbors = voxels that share a face
  - 1-neighbors = voxels that share a edge
  - 0-neighbors = voxels that share a vertex



Courtesy D. Coeurjolly & I. Sivignon

Basic discrete geometry definitions

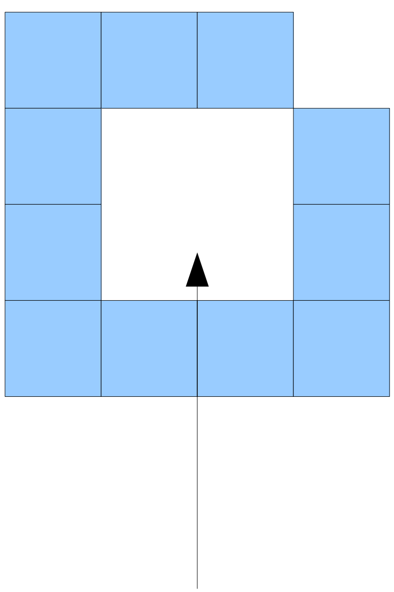
- An ordered set $\{c_1, \ldots, c_n\}$ of discrete cells is a (topological) k-path if $\forall$ i, c_i is a k-neighbor of c_{i-1}
- It is a k-arc if $\forall$ i, c_i has exactly two k-neighbors
- It is a k-curve if it a k-arc + $c_1 = c_n$



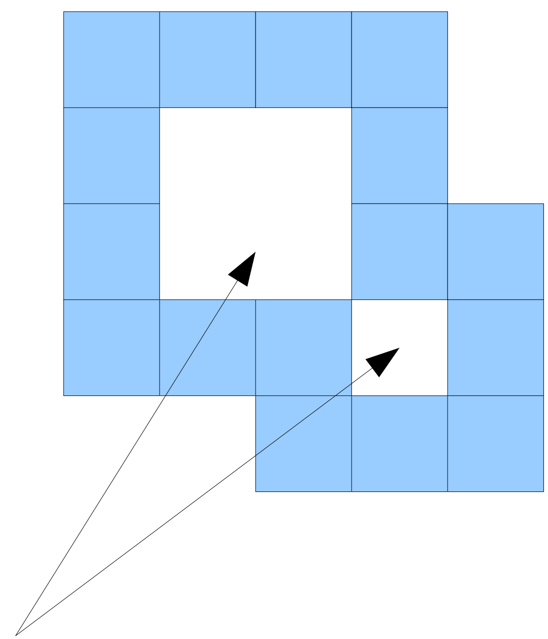
- A set O of discrete cells is a k-object if $\forall$ c,c' in O, one can find a k-path from c to c' in O

Discrete object boundary

- Problem with discrete objects: their boundary is not obvious



Inside or outside ?



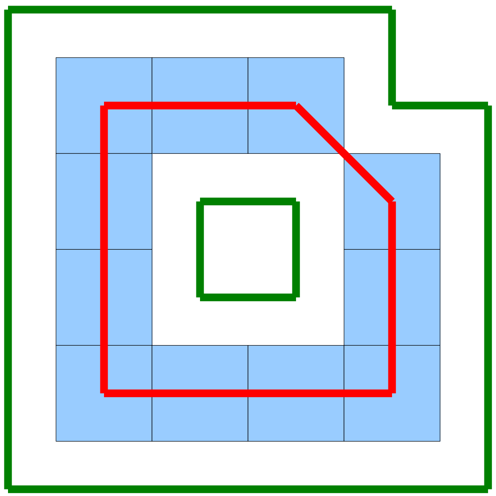
One or two components ?

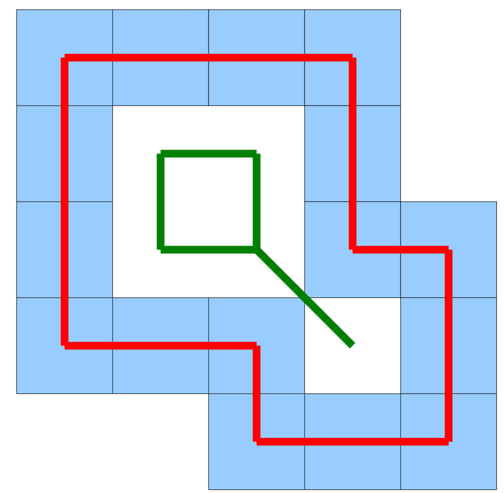
Problem

- Jordan's theorem: every smooth (n-1)-manifold in $\mathbb{R}^n$ disjoints space into two connected domains (the inside and the outside); it is the common boundary of these domains
- Corollary: impossible to find a path from inside to outside
- Need to define the right adjacency !

Adjacency couple

- Need to define one connexity for the (inside) object, and one for the outside

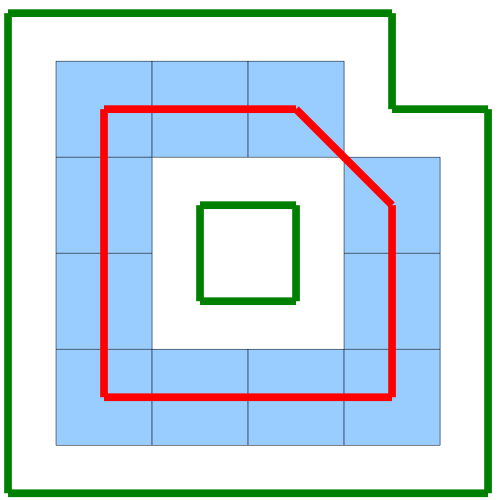


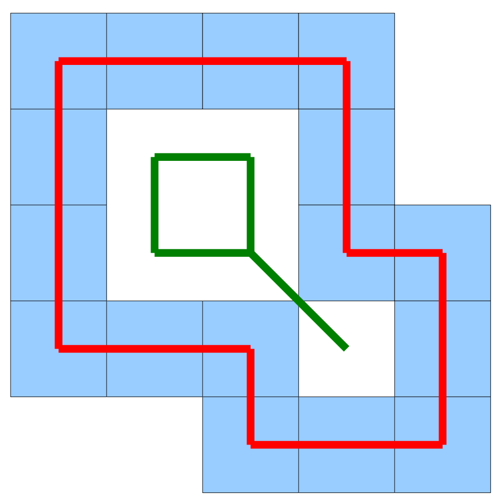


- Exercise: possible couples?

Adjacency couple

- Need to define one connexity for the (inside) object, and one for the outside

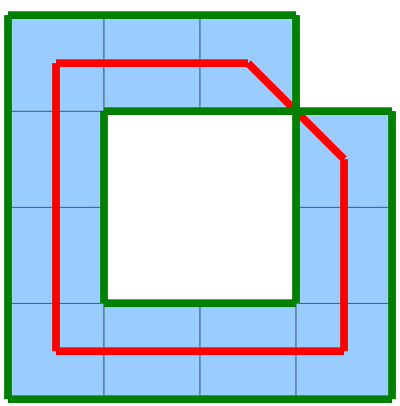


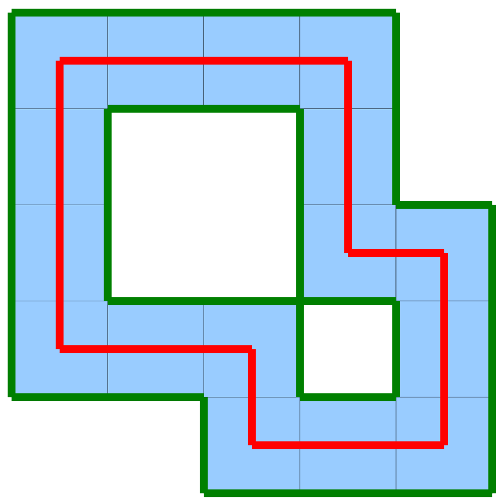


- Possible couples: (6, 18), (6, 26), (18, 6) and (26, 6)

Contour

- Def.: connected set of cell faces between a cell inside the object and a cell outside





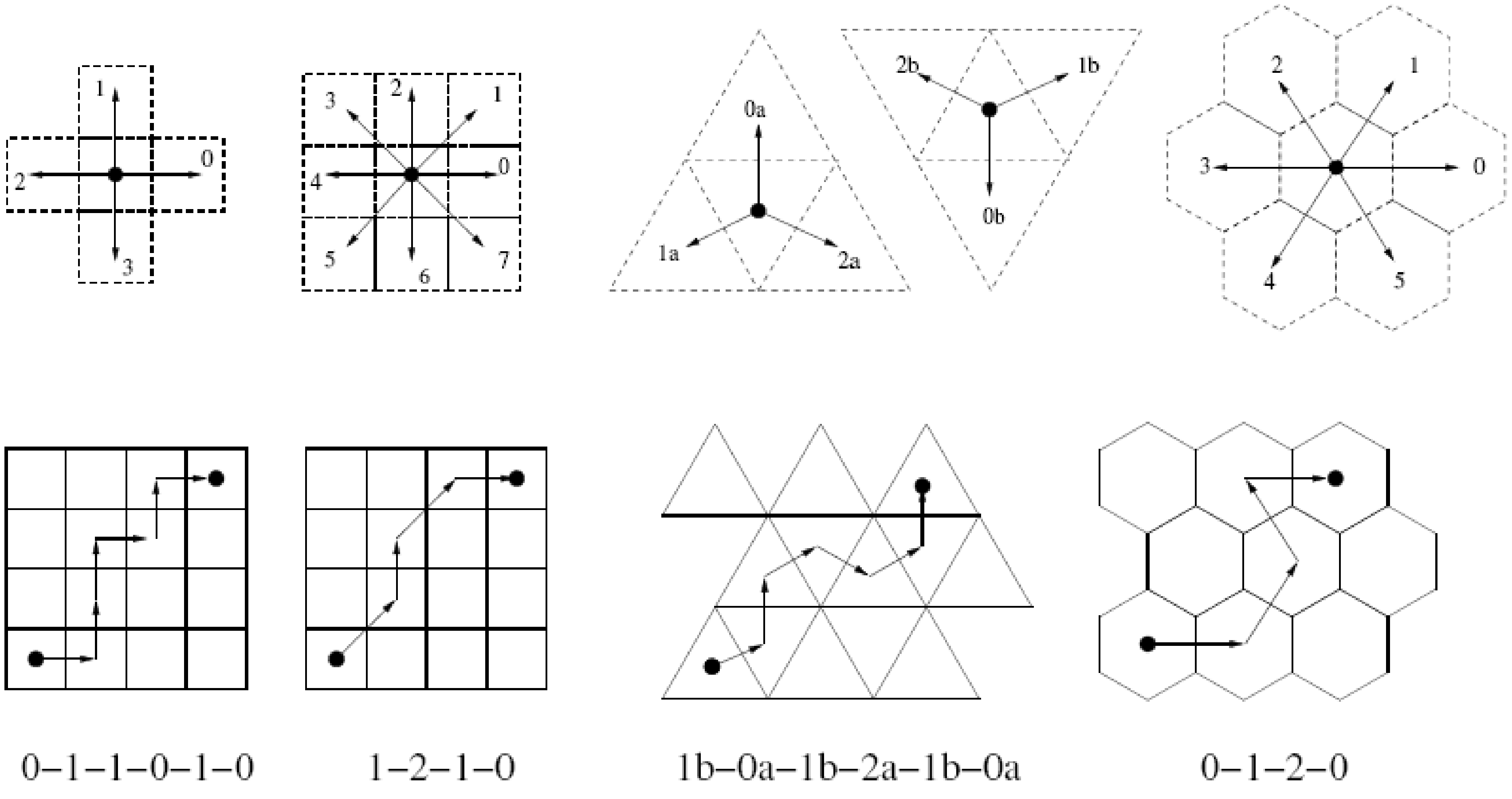
- Coherent with Jordan; depends on the chosen adjacency
- Contour of a volume = surface (to display)

Contour coding

- We want the code to be:
  - Compact: compared to a simple list of the discrete faces coordinates
  - Toggle: the surface can be reconstructed from the code
  - Invariant: w.r.t. some geometrical transforms
  - Informative: about the surface (area, ...)
- In 2D: Freeman code

Freeman code

- Idea: code the path between two consecutive pixels of the discrete curve

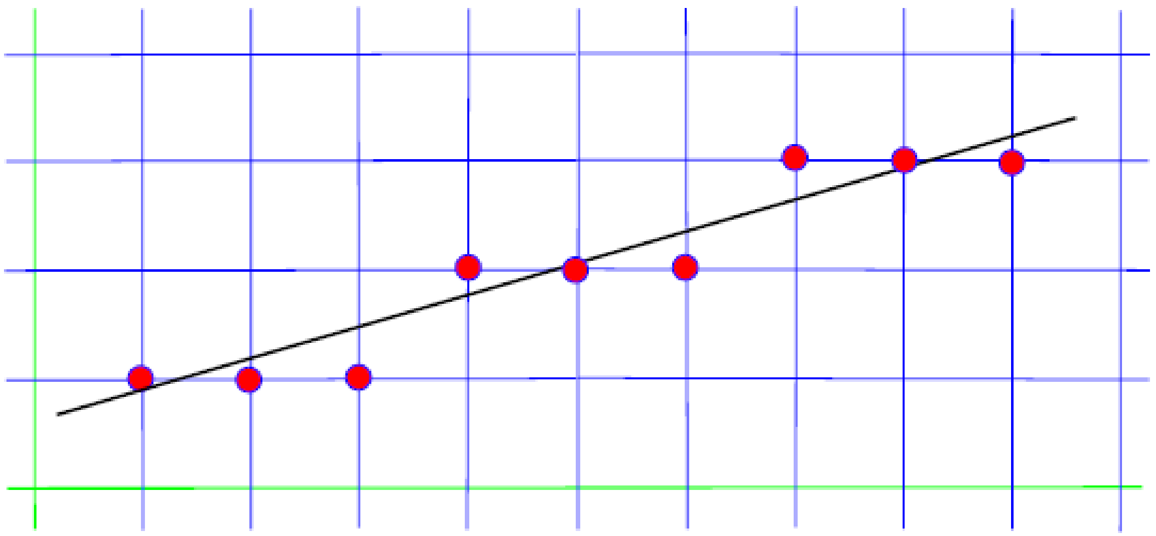


Properties

- Reversible (unicity)
- Geometrical transforms does not affect much the code
  - Translation: just change the origin point
  - Rotation with angle $\pi/2$: $c'=c+2 \bmod 8$ (if 8-adjacency)
- Can give an estimate of the curve length
  - $L := L+1$ if c is even
  - $L := L+\sqrt{2}$ if c is odd

Discrete line (2D)

- How to define a discrete line from a real line ?
- Bresenham algorithm:
  - Choose the closest pixel to the line in the vertical direction (incremental)

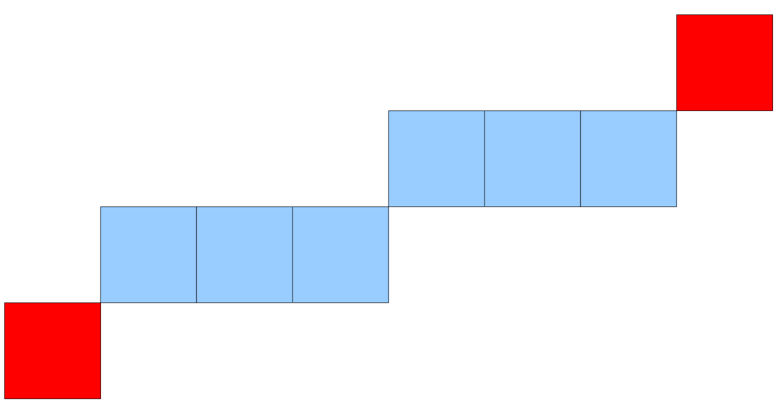


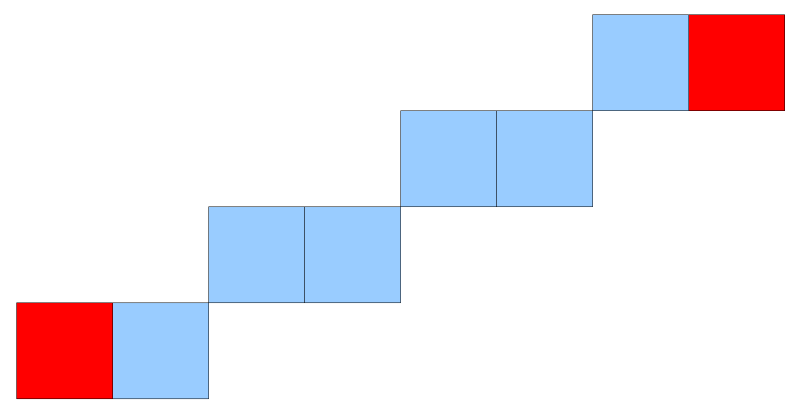
Other definition

- Let D: $y = ax+b$ be the real line. D is the set of points/pixels $p_i = (x_i, y_i)$ with $x_i = i$ and $y_i = \lfloor ax_i + b + 0.5 \rfloor$.
- Properties:
  - D is a 8-arc
  - D can be Freeman-coded with codes 0 and 1 only
  - If a is rational, then the code of D is periodic

Discrete vs. continuous

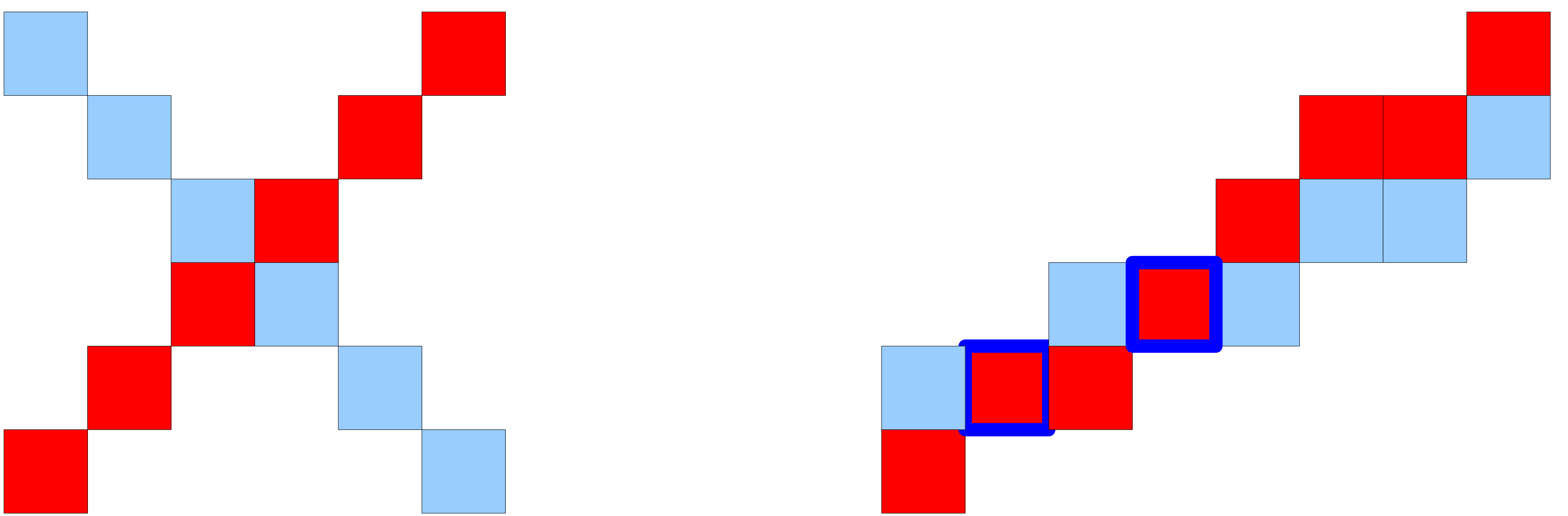
- Euclide 1: given two points A and B, there exists only one line going through A and B.





Discrete vs. continuous

- Euclide 2: Two non parallel lines intersect exactly once.

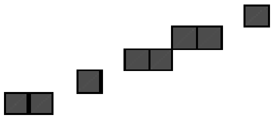


Third definition [Reveillès 1991]

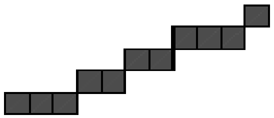
- Arithmetic discrete line:
  - $D(a,b,d,e) = \{ (x, y)$ with x,y,a,b,d,e in $\mathbb{Z}$, $b \neq 0$, $0 \leq ax - by + d < e$ and $\gcd(a,b)=1 \}$.
  - a/b is the line slope, d is the origin offset and e the thickness.
- Exercise:
  - Draw a regular grid.
  - Draw (the beginning of) the following lines: D(3,7,0,5), D(3,7,0,7), D(3,7,0,8), D(3,7,0,10) and D(3,7,0,16).

Third definition [Reveillès 1991]

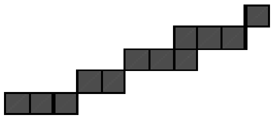
- Arithmetic discrete line:
  - D(a,b,d,e) = { (x, y) with x,y,a,b,d,e in $\mathbb{Z}$, b ≠ 0, 0 ≤ ax - by + d < e and gcd(a,b)=1 }.
  - a/b is the line slope, d is the origin offset and e the thickness.



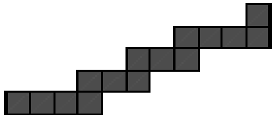
$D(3,7,0,5)$



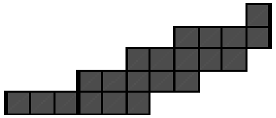
$D(3,7,0,7)$



$D(3,7,0,8)$



$D(3,7,0,10)$



$D(3,7,0,16)$

Courtesy D. Coeurjolly and I. Sivignon

Properties

- Let D(a,b,d,e) be a discrete line. Then:
  - if $e < \max(|a|,|b|)$ then D is disconnected;
  - if $e = \max(|a|,|b|)$ then D is a 8-arc and is called a naive line;
  - if $\max(|a|,|b|) < e < |a|+|b|$ then D has both 4- and 8-connected parts;
  - if $e = |a|+|b|$ then D is a 4-arc and is called a standard line;
  - else D is called a thick line.

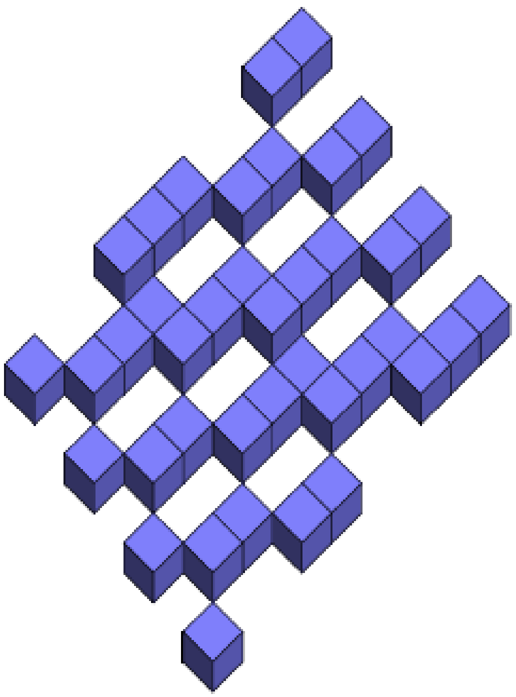
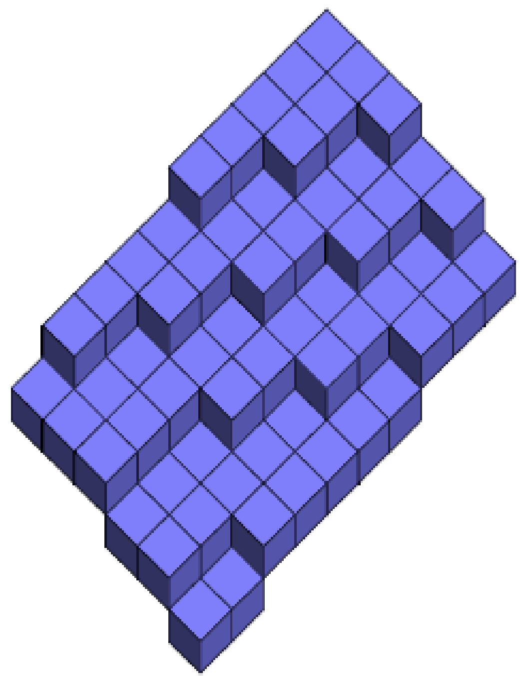
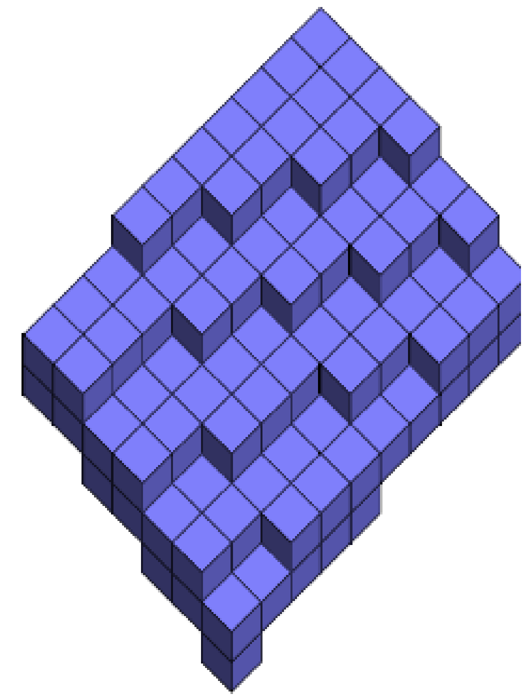
Properties

- Let D be the real line ax-by+d = 0 with a,b,d in $\mathbb{Z}$; suppose $|a| \leq |b|$. Then:
  - the default discretization of D, that is to say the set { (x,y), y = $\lfloor$(-ax-d)/b$\rfloor$} is exactly D(a,b,d,b);
  - the excess discretization of D, that is to say the set { (x,y), y = $\lceil$(-ax-d)/b$\rceil$} is exactly D(a,b,d+b-1,b);
  - ...

Discrete plane (3D)

- Discretization of a real plane:
  - Let d: $z = ax+by+c$ be the real plane. P is the set of points/voxels $p = (x,y,z)$ with x and y in $\mathbb{Z}$ and $z = \lfloor ax + by + c \rfloor$.
- Arithmetic discrete plane:
  - $P(a,b,c,d,e) = \{ (x, y, z)$ with x,y,z,a,b,c,d,e in $\mathbb{Z}$, $d \leq ax + by + cz < d + e$ and $\gcd(a,b,c)=1 \}$.
  - $(a, b, c)^t$ is the plane normal, d is the origin offset and e the thickness.

Some discrete planes

Courtesy D. Coeurjolly and I. Sivignon

P(6,13,27,0,15) P(6,13,17,0,27) P(6,13,17,0,46)

Discrete geometry

This part was inspired by a course given by David Coeurjolly and Isabelle Sivignon (CNRS researchers, LIRIS, Lyon)

Books

- J.-M. Chassery, A. Montanvert, “Géométrie Discrète en Analyse d'Images”, Hermès, 1991
  - Available at INRIA or University library
- F. Feschet, J.-P. Reveillès, “Tracés Géométriques”, chapter from “Informatique Graphique et Rendu“, Hermès, 2007
- R. Klette, A. Rosenfeld, “Digital Geometry, Geometric Methods for Digital Picture Analysis”, Morgan-Kaufmann, 2004

The end

- Next week:
  - Parametric curves and surfaces
  - Subdivision surfaces
  - Lecturer: Marie-Paule Cani
- These slides will be available on the course's webpage:

http://evasion.imag.fr/Membres/Franck.Hetroy/Teaching/Geo3D/