The Computational Geometry Algorithmes Library

Sofware Structure

- The kernel
- The Support Library
- The Basic Library
- Class Factory
- Futur Development

F. Rebufat

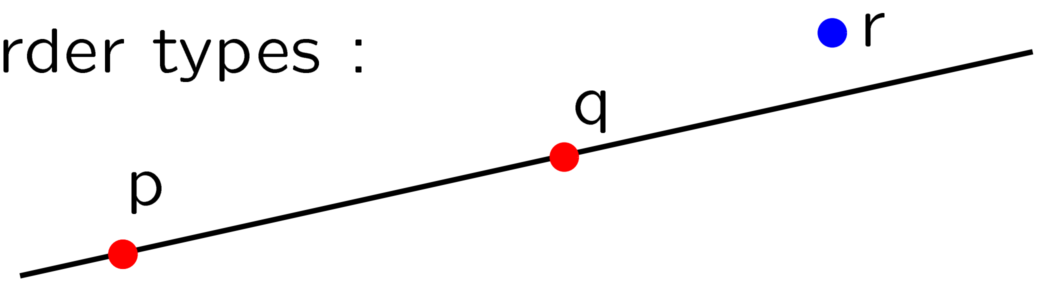
The Kernel overview

- Representation Class :
  - Cartesian
  - Homogeneous
- Utilities : basic.h
  - Order Types — CGAL Orientation
  - Relative Position — CGAL Oriented Side, CGAL Bounded Side
  - Comparison
  - Generic Object — CGAL make object, CGAL assign
- 2D Objects :
  - Point
  - Vector
  - Direction
  - Ray
  - Segment
  - Line
  - Triangle
  - Iso Rectangle
  - Bounding Box
  - Circle

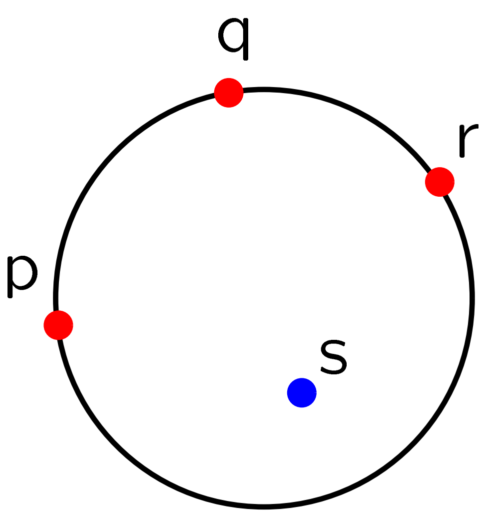
The Kernel overview

- Predicates :

  → Order types :



  → Incircle test



  → Comparison of Coordinates of points

`CGAL compare`

The Kernel overview

- Transformations
  - → CGAL TRANSLATION
  - → CGAL ROTATION
  - → CGAL SCALING

- Intersections
  - → Checking

```
CGAL do intersect(obj1,obj2);
```

  - → Computing

```
CGAL Object CGAL intersection(obj1,obj2);
```

- Squared distance
  - → Computing

```
CGAL squared distance(obj1, obj2);
```

  - → Comparisons

The Kernel overview

- 3D Objects :
  - Point
  - Vector
  - Direction
  - Ray
  - Segment
  - Line
  - Triangle
  - Tetrahedron
  - Bounding Box
  - Plane
- Predicates, Tranformations, Intersections
- Points in d dimension

A mathematic model

- Affine vs Linear geometry

  Point - Origine $\rightarrow$ Vector
  Point + Vector $\rightarrow$ Point
  Point - Point $\rightarrow$ Vector

- Euclidean space representations

  Cartesian : $(x, y) \in R^2$

  Homogeneous : $(h_x, h_y, h_w) \rightarrow (\frac{h_x}{h_w}, \frac{h_y}{h_w}) \in R^2$

  $h_w > 0$

  Catesian : compact, commonly used
  Homogeneous : No division, more robust

- Number types

  Field types : *double, Quotient<RT>, leda real, ...*

  Ring types : *int, Gmpz, double, ...*

Parametrisation

- Point
  - Cartesian
    - *double*
    - *quotient*
    - *leda real*
  - Homogeneous
    - *double*
    - *gmpz*
    - *int*

`CGAL-Point-2<Cartesian<double>>`

F. Rebufat

Evolution

- **Past**

  Two parameters for homogeneous representation

  → Two types can be incompatible

- **Present**

  One parameter for homogeneous

  → Cartesian and homogeneous defined geometric objects but can not be used as traits class

  → No equivalence between `CGAL-Point-2<R>` and `R::Point-2`

  → No possibility to change kernel components
  The user who want another point type has to rewrite everything

- **Futur**

  Allows user defined point type

Support Library

- Number type support
- STL extentions
- Circulators
- Handles
- Timer
- Random generators
- Stream Support
- Geomview stream

INRIA SOPHIA ANTIPOLIS

F. Rebufat

The Basic Library

Data structures and algorithms for CG

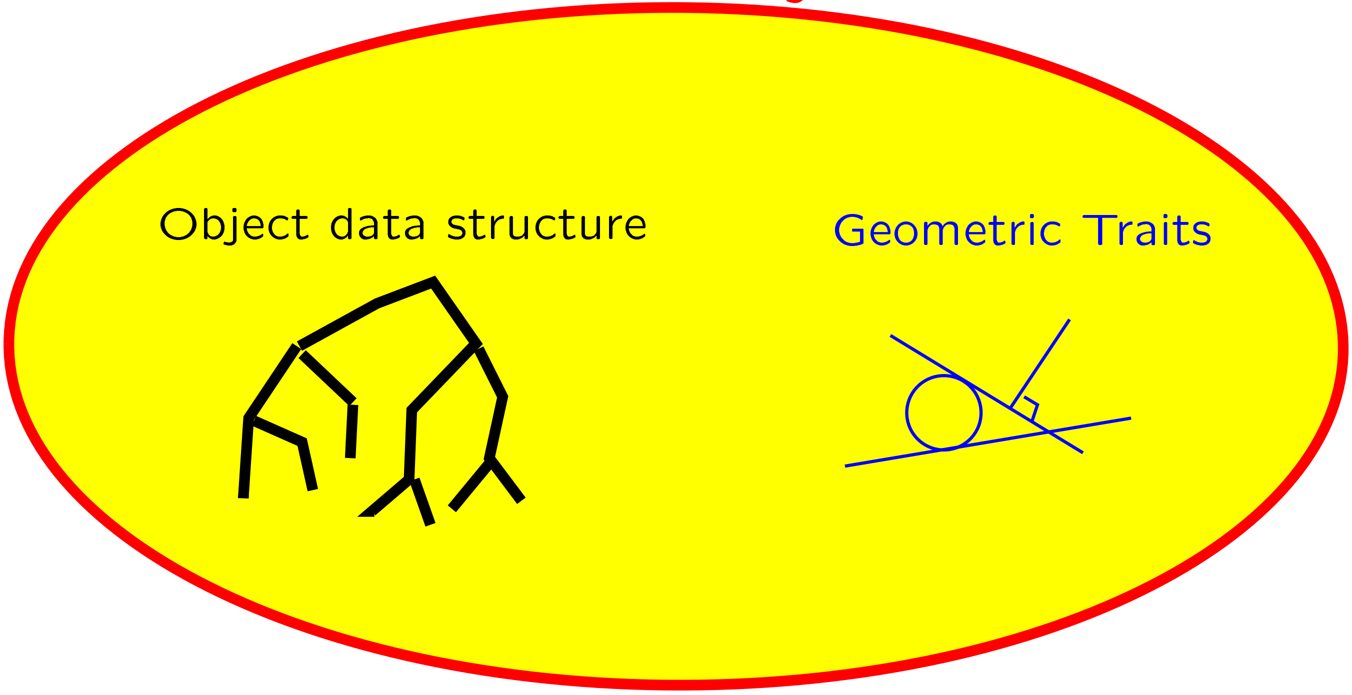
- Polygons
- Halfedge Data Structures
- 3D-Polyhedral Surfaces
- Topological Map
- Planar Map
- Planar Map Overlay
- Triangulations
- Boolean Operations in 2D
- Convex Hulls
- Geometric optimisation
- Search Structures

Applications

- Reconstruction
  - Voronoi diagrams
  - Alpha shapes
  - Nearest neighbors
  - Power diagrams
- Placement and deplacement
  - Minkowki sum
  - Boolean operations
  - Polygons
  - Planar maps
- Meshing, CAD
  - Constrained triangulation 2D/3D
- Research and teatching

Generic Structure

CGAL-Object



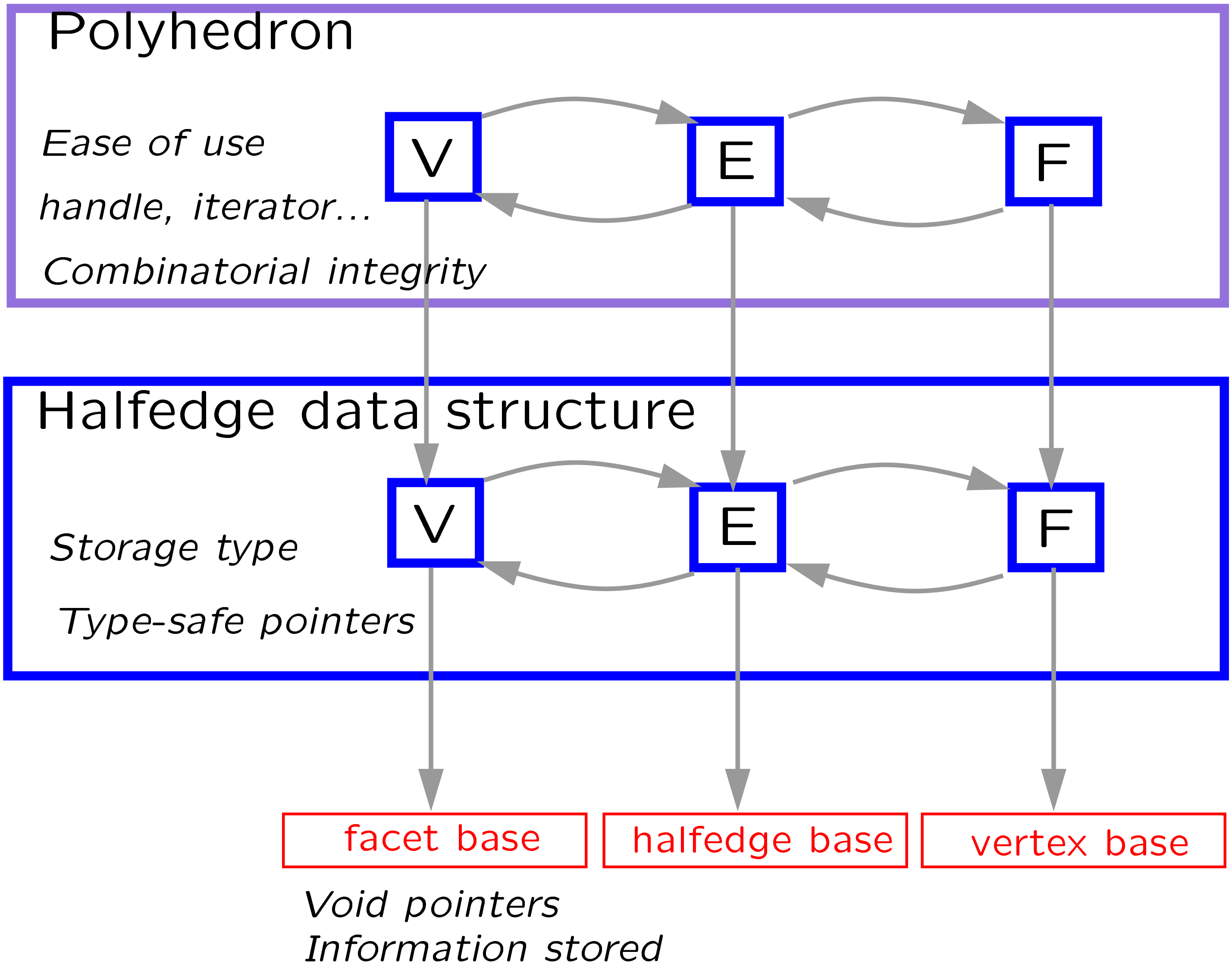
Ex : Polygon<Traits, Container>

Advantage :

users can defined their owns data structures or their own geometry independently

3D Polyhedral Surfaces

Vertex, halfedges and facets
Who knows what?

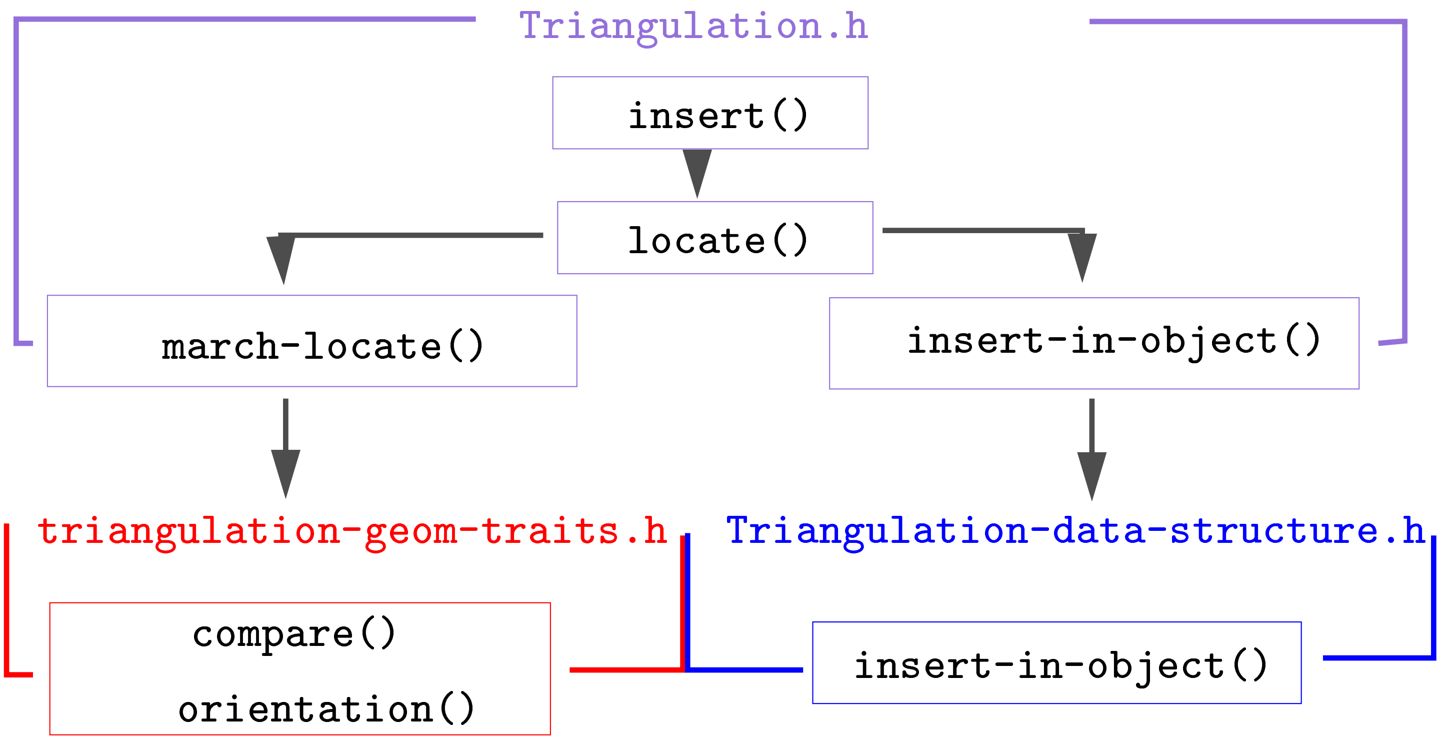


Multi-layers structure

Advantages :

Base objects are independant (void pointers)
Heavy parametrisation → Leggo
No loss of efficiency (??)
Ease of use
Adding informations to objects can be done easily

Drawback : Two levels of cast (efficiency??)



Adding a function in V, E, F classes that involves V, F or E imply redefining all the class hierarchy

A Class Factory

Minimal requierement for each level of the design

- Geometry : Traits class
  - Exported types
  - Geometric Predicates

+++++

- Data structure
  - Creation
  - Access
  - Setting
  - Modifiers
  - Traversing (iterator...)
  - ....

+++

- Base Objects and data structure wrapers
  - Representation
  - Incidences and adjacencies

++

INRIA SOPHIA ANTIPOLIS

F. Rebufat

FUTUR

Actual public release is 1.2

- Supported platforms
  - Linux, Sun (OS, Solaris), Irix (SGI)
  - CC, g++, egcs

Next release will be 2.0

- Add namespace to support Visual C++
  CC will not be supported anymore
- Change Kernel structure
  - More flexibility
    User can used its own point type
- New design for the basic library
  common layers structure
  - More flexibility
  - Simplify instanciation of classes
- New documentation
- New packages :

INRIA SOPHIA ANTIPOLIS

F. Rebufat

New Packages :

- Triangulation 3D
- Incremental constrained triangulation
  - Meshing
- Visualization package
  - OpenGL, OpenInventor
  - Presentation, teaching, debugging
- Operations on curves (conics)
- Applicatives packages
  - Reconstruction (Alpha Shapes)
  - Placements, deplacements

A new triangulation Traits class

Triangulation traits class