



<http://liris.cnrs.fr/dgtal>

LORIA 6-8 avril 2011

1 DGtal: a software library for the discrete geometry community 2 Structure

1.1 Objectives

- to make easier discrete geometry for the neophyte (student, researcher from another field, ...)
- to test quickly new ideas, with objective comparison wrt existant works
- to make easier the implementation of demonstrators
- to help spread our research results to other domains
- to pursue a federative project
- ...

1.2 Main features

- to define digital objects in arbitrary dimension
- to propose algorithms for topological and geometric analysis
- to facilitate image analysis with data structures
- to provide I/O mechanisms and visualization tools

1.3 Philosophy

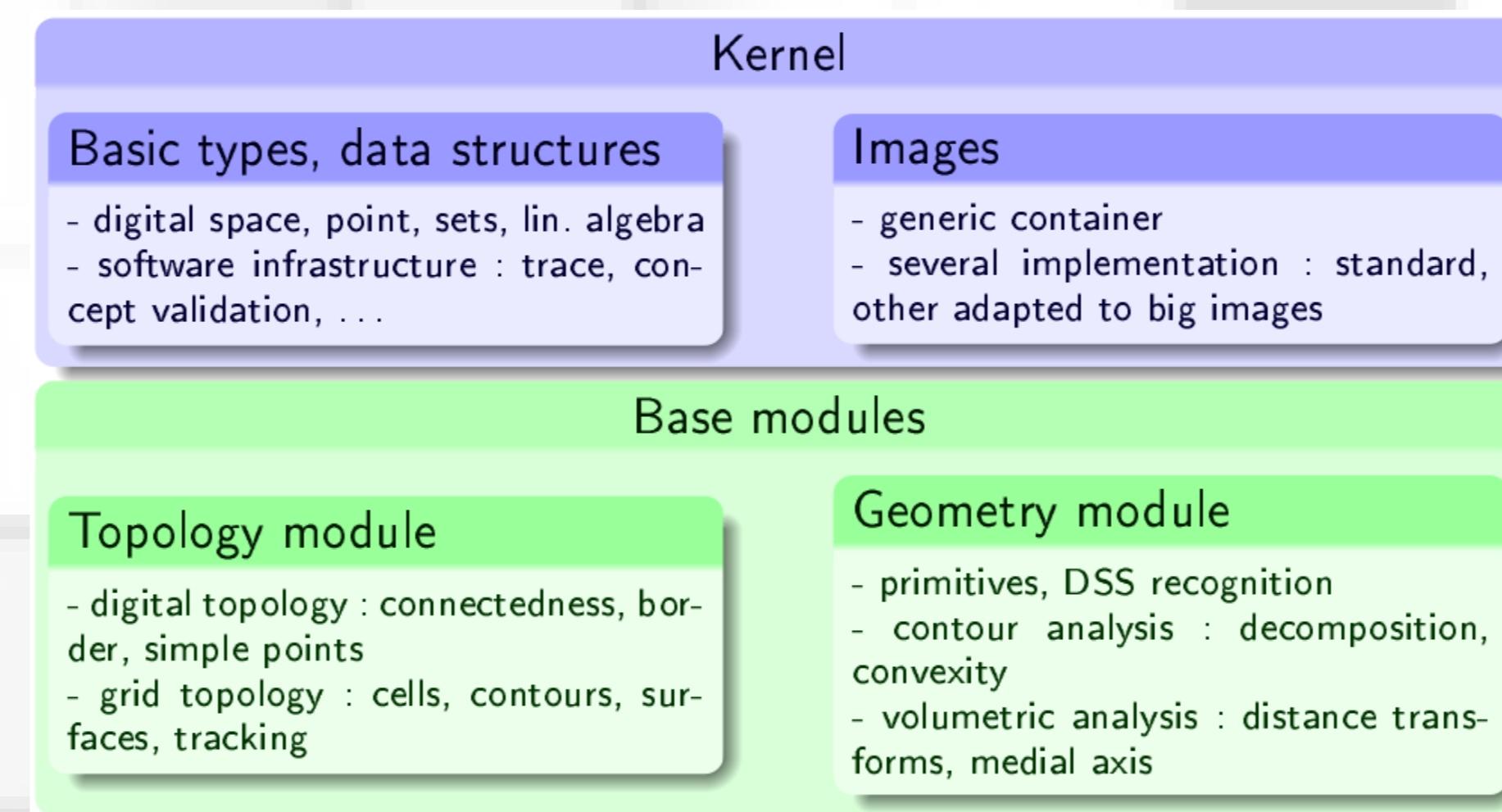
- **Genericity** and **efficiency**
- C++ library, concepts, generic programming with templates
- open-source, LGPL or GPL with restrictions
- user friendly, not necessarily kernel-developer friendly

1.4 A collaborative effort



1.5 The current DGtal team

- David Coeurjolly (LIRIS): infrastructure, kernel, images, volumetric geometry
- Jacques-Olivier Lachaud (LAMA): kernel, topology, 2D display (board)
- Bertrand Kerautret (LORIA): contours, 3D viewer
- Tristan Roussillon (LIRIS): 2D geometry
- Guillaume Damiand (LIRIS): kernel
- Sébastien Fourey (GREYC): kernel, board
- Isabelle Sivignon (Gipsa-lab): DSS



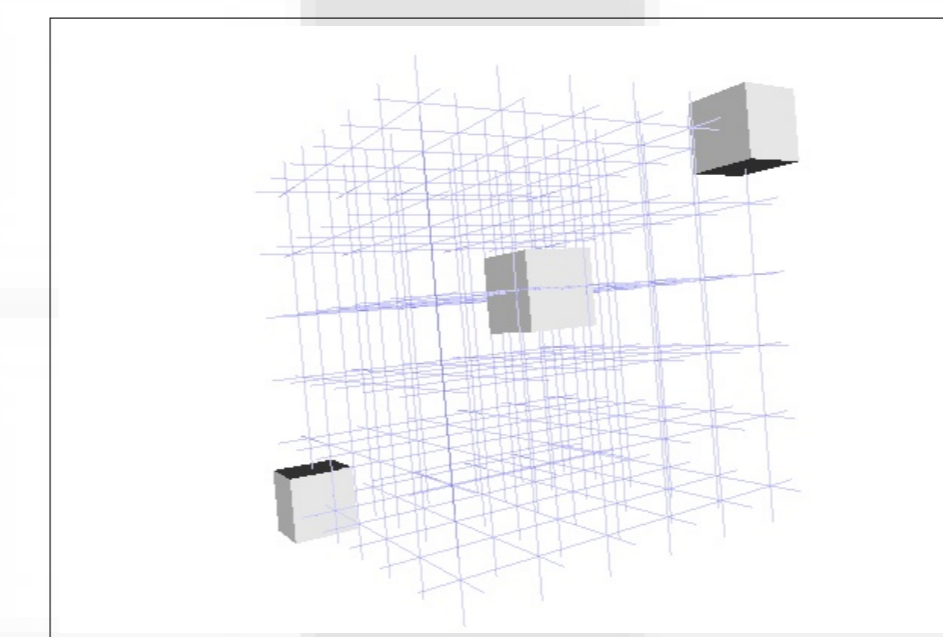
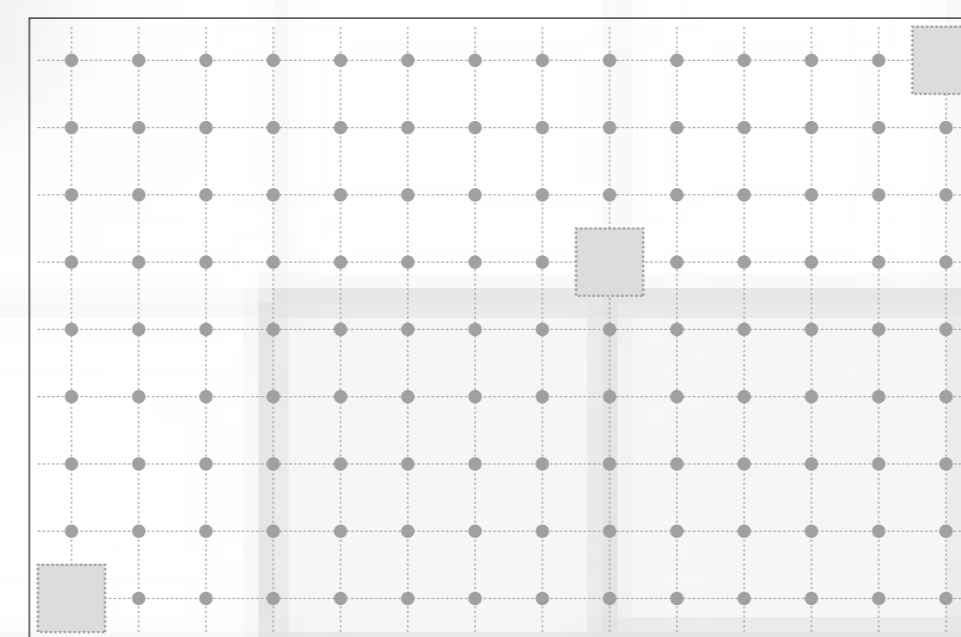
3 Features and examples

3.1 Generic spaces, domain, sets, etc

- arbitrary spaces, spanning iterators, adaptative type of sets

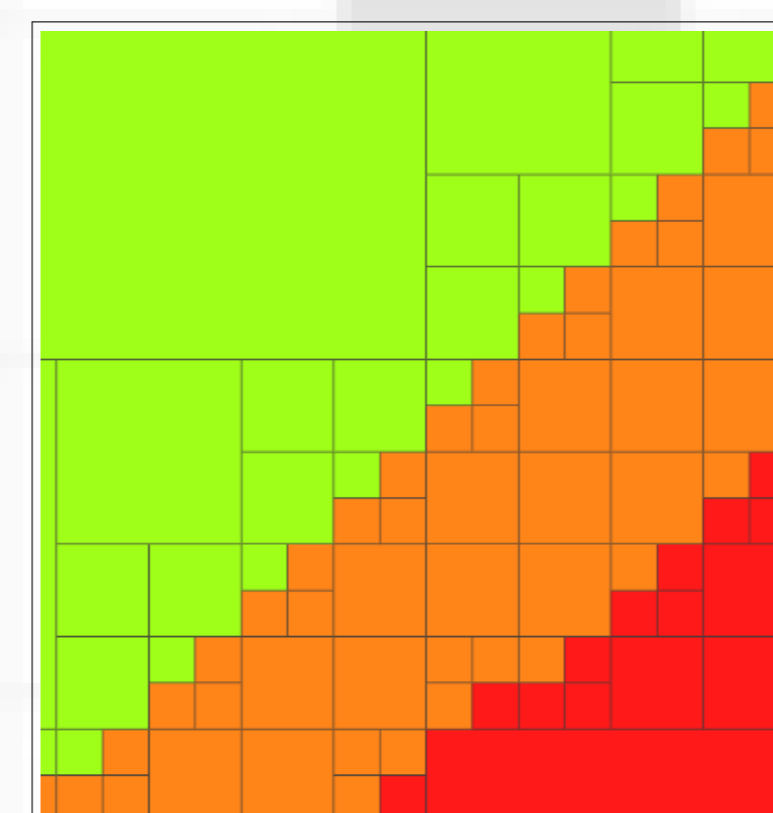
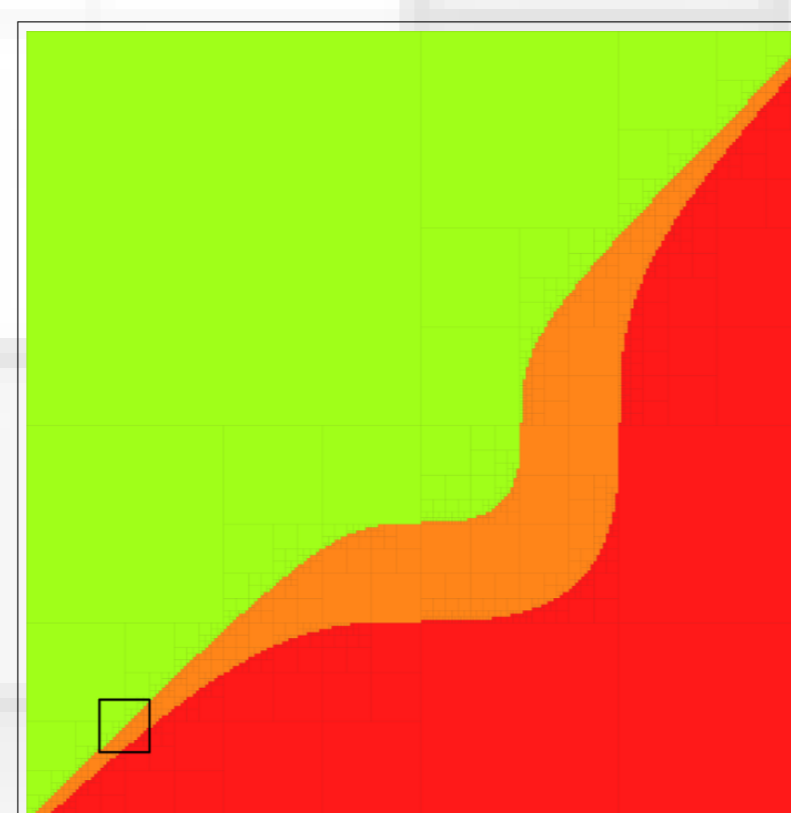
```
int main()
{
    typedef DGtal::SpaceND<2, DGtal::int32_t> MySpace;
    typedef MySpace::Point MyPoint;
    typedef HyperRectDomain<MySpace> MyDomain;
    MyPoint p1(-3,-4);
    MyPoint p2(10,4);
    MyPoint p3(5,1);
    MyDomain domain(p1,p2);
    DGtalBoard board; // for 2D display
    board << domain;
    board << p1 << p2 << p3;
    board.saveSVG("demo-kernel-1.svg");
    board.saveEPS("demo-kernel-1.eps");
    return 0;
}

int main( int argc, char** argv )
{
    QApplication application(argc,argv);
    typedef DGtal::SpaceND<3, DGtal::int32_t> MySpace;
    typedef MySpace::Point MyPoint;
    typedef HyperRectDomain<MySpace> MyDomain;
    MyPoint p1( 0, 0, 0 );
    MyPoint p2( 5, 5, 5 );
    MyPoint p3( 2, 3, 4 );
    MyDomain domain( p1, p2 );
    DGtalQGLViewer viewer; // for 3D visualization
    viewer.show();
    viewer << domain;
    viewer << p1 << p2 << p3;
    viewer<< DGtalQGLViewer::updateDisplay;
    return application.exec();
}
```



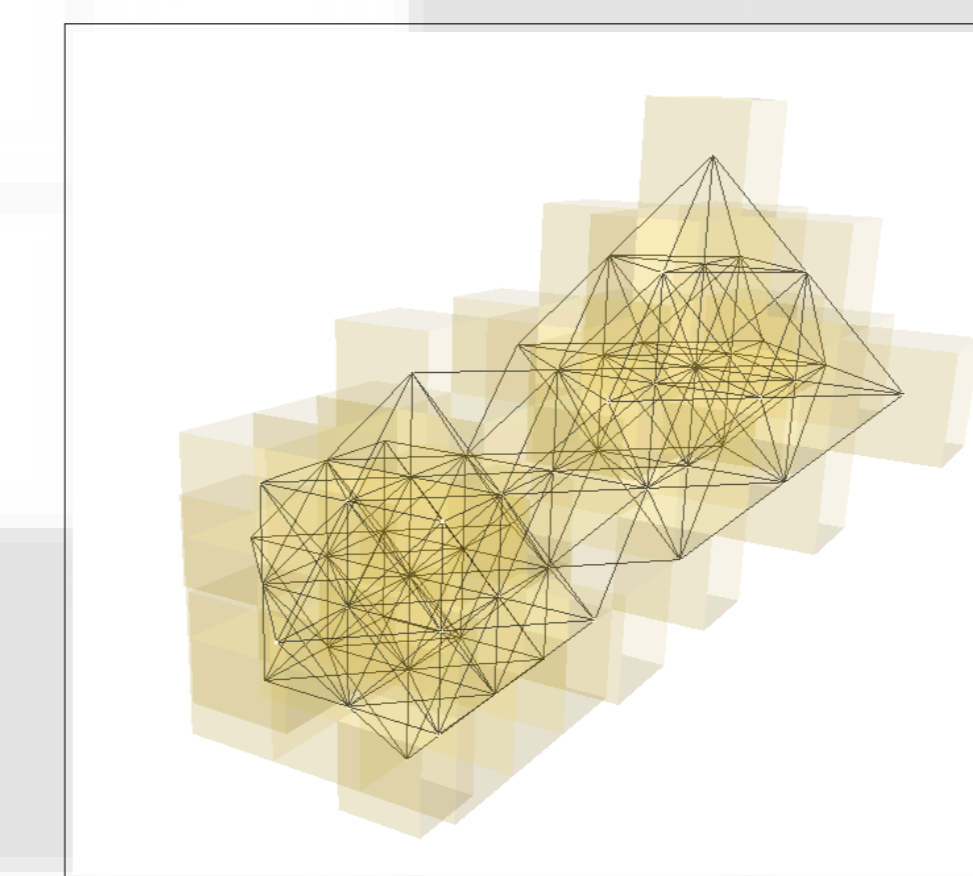
3.2 Generic images, adaptative containers

- several image containers (vector, hashtable), ITK backend

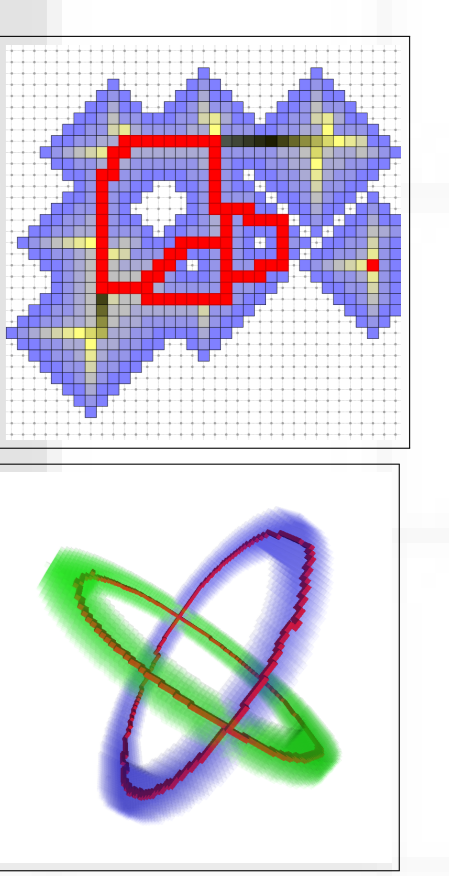


3.3 Digital (Rosenfeld's) topology

- digital topologies, connected components, borders, simple points, homotopic thinning

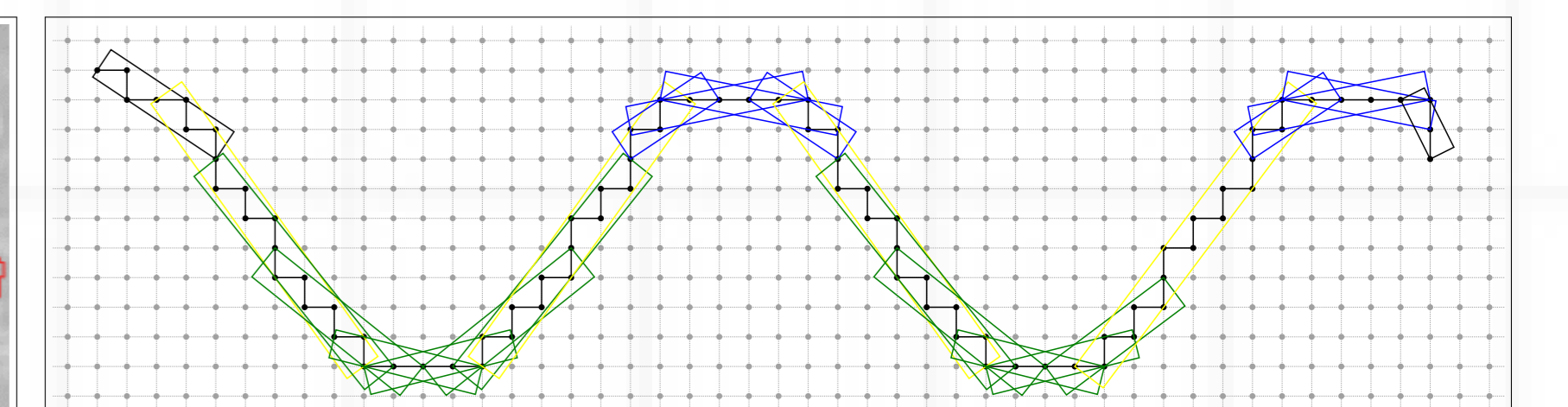
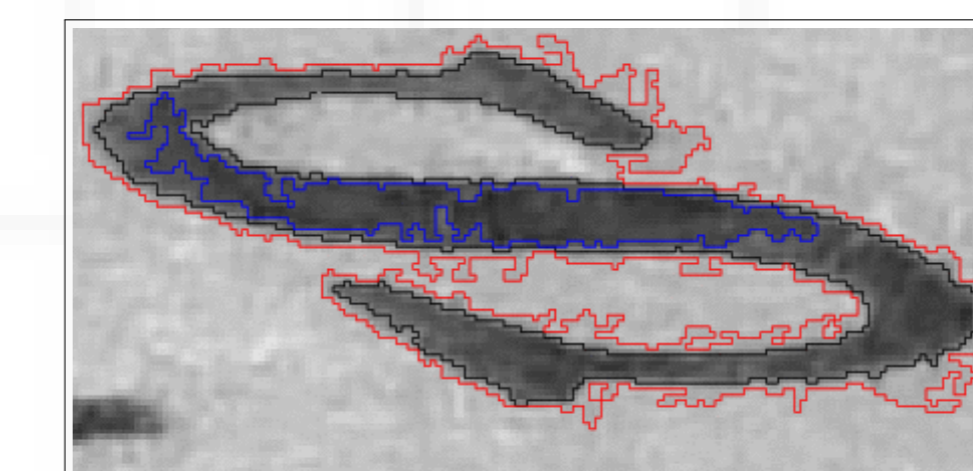


```
do
{
    DigitalSet & S = shape.pointSet();
    std::queue<DigitalSet::Iterator> Q;
    for ( DigitalSet::Iterator it = S.begin();
          it != S.end(); ++it )
        if ( shape.isSimple( *it ) )
            Q.push( it );
    nb_simple = 0;
    while ( ! Q.empty() )
    {
        DigitalSet::Iterator it = Q.front();
        Q.pop();
        if ( shape.isSimple( *it ) )
        {
            S.erase( *it );
            ++nb_simple;
        }
    }
} while ( nb_simple != 0 );
DigitalSet & S = shape.pointSet();
```

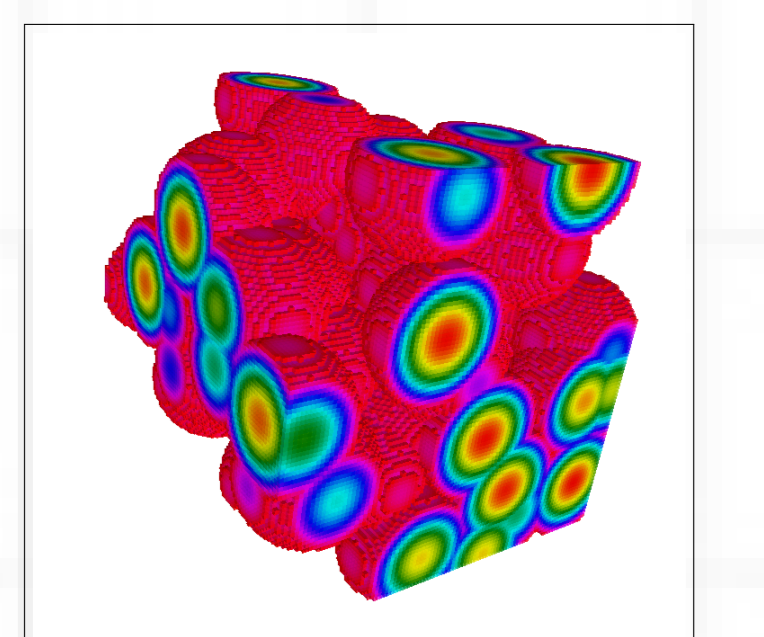
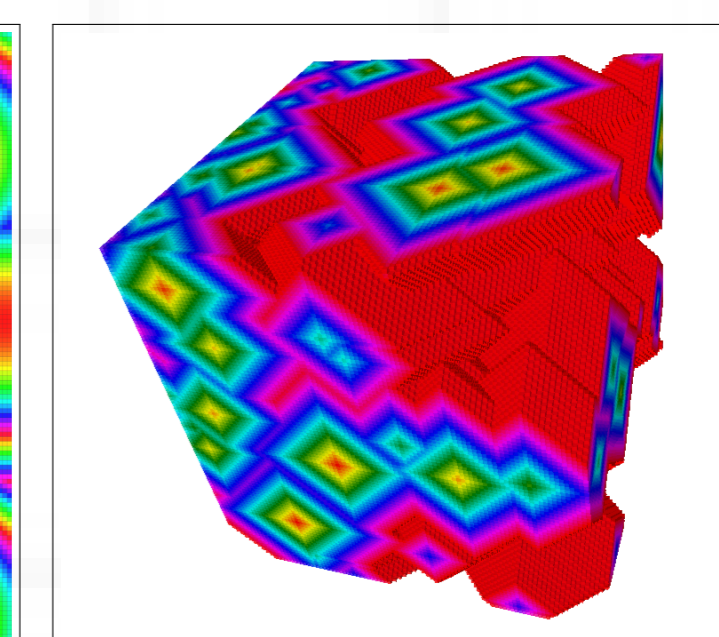
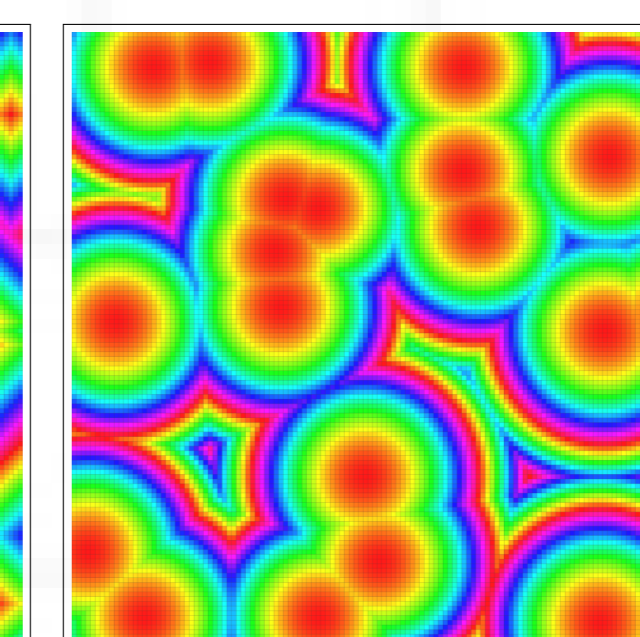
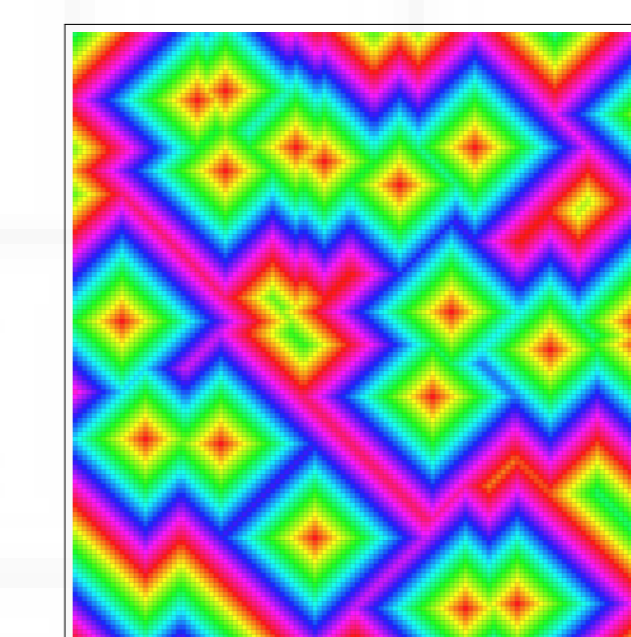


3.4 Geometry analysis

- 2d contours, primitives, DSL and DSS, decomposition, tangential cover

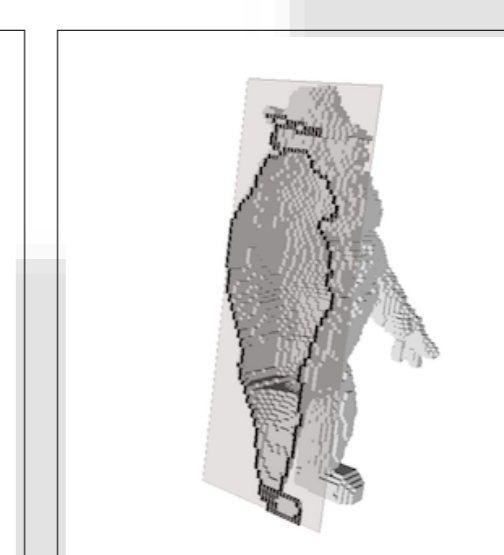
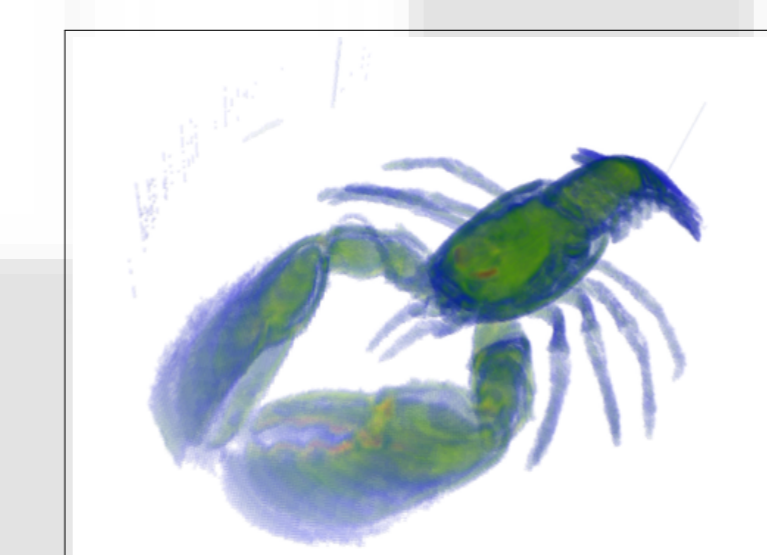


- nD volumetric distance transforms



3.5 Additional features

- simple 2D vector display and export with stream mechanism
- 3D viewer with stream mechanism based on QGLviewer (**New in 0.3**)
- image, volumes import/export
- grid or interpixel topology, cells, digital surfaces, surface tracking (**New in 0.3**)



- Forge (trac), unit tests, cmake/ctest/cdash, mailing lists
- User and developer documentation (doxygen)
- Cross-platform (Linux, MacOS, Windows)
- **Join DGtal !**