

RESEARCH CENTRE

**Inria Nancy - Grand Est Center**

IN PARTNERSHIP WITH:

**Université de Lorraine, CNRS**

2022

ACTIVITY REPORT

Project-Team

PIXEL

## **Structure geometrical shapes**

IN COLLABORATION WITH: Laboratoire lorrain de recherche en informatique et ses applications (LORIA)

**DOMAIN**

**Perception, Cognition and Interaction**

**THEME**

**Interaction and visualization**

*Inria*

# Contents

<b>Project-Team PIXEL</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>2</b>
2.1 Scientific grounds . . . . .	3
<b>3 Research program</b>	<b>4</b>
3.1 Point clouds . . . . .	4
3.2 Parameterizations . . . . .	5
3.3 Hexahedral-dominant meshing . . . . .	6
<b>4 Application domains</b>	<b>7</b>
4.1 Geometric Tools for Simulating Physics with a Computer . . . . .	7
<b>5 Highlights of the year</b>	<b>8</b>
5.1 Awards . . . . .	8
5.2 Programming contests . . . . .	8
<b>6 New software and platforms</b>	<b>9</b>
6.1 New software . . . . .	9
6.1.1 stlbfgs . . . . .	9
6.1.2 ultimaille . . . . .	10
<b>7 New results</b>	<b>11</b>
7.1 Polycube hexahedral meshing . . . . .	11
7.2 Functional Maps . . . . .	11
7.3 Image analysis and synthesis . . . . .	14
<b>8 Bilateral contracts and grants with industry</b>	<b>14</b>
8.1 Bilateral contracts with industry . . . . .	14
<b>9 Dissemination</b>	<b>15</b>
9.1 Promoting scientific activities . . . . .	15
9.1.1 Conference . . . . .	15
9.1.2 Journal . . . . .	15
9.1.3 Research administration . . . . .	15
9.2 Teaching - Supervision - Juries . . . . .	15
9.2.1 Teaching . . . . .	15
9.2.2 Supervision . . . . .	17
9.2.3 Juries . . . . .	17
9.3 Popularization . . . . .	17
9.3.1 Interventions . . . . .	17
<b>10 Scientific production</b>	<b>18</b>
10.1 Major publications . . . . .	18
10.2 Publications of the year . . . . .	18
10.3 Cited publications . . . . .	19

## **Project-Team PIXEL**

*Creation of the Project-Team: 2020 March 01*

### **Keywords**

#### **Computer sciences and digital sciences**

- A5.5.1. – Geometrical modeling
- A5.5.2. – Rendering
- A6.2.8. – Computational geometry and meshes
- A8.1. – Discrete mathematics, combinatorics
- A8.3. – Geometry, Topology

#### **Other research topics and application domains**

- B3.3.1. – Earth and subsoil
- B5.1. – Factory of the future
- B5.7. – 3D printing
- B9.2.2. – Cinema, Television
- B9.2.3. – Video games

## 1 Team members, visitors, external collaborators

### Research Scientists

- Laurent Alonso [INRIA, Researcher]
- Etienne Corman [CNRS, Researcher]
- Nicolas Ray [INRIA, Researcher]

### Faculty Members

- Dmitry Sokolov [Team leader, UL, Associate Professor, HDR]
- Dobrina Boltcheva [UL, Associate Professor]

### Post-Doctoral Fellow

- David Lopez [INRIA, until Nov 2022]

### PhD Students

- Justine Basselin [RHINO TERRAIN, CIFRE]
- Guillaume Coiffier [UL]
- Yoann Coudert-Osmont [ENS DE LYON]
- David Desobry [INRIA]
- François Protais [INRIA, until Sep 2022]

### Administrative Assistant

- Emmanuelle Deschamps [INRIA]

### External Collaborators

- Hervé Barthélémy [RhinoTerrain]
- Jeanne Pellerin [TotalEnergies, HDR]

## 2 Overall objectives

PIXEL is a research team stemming from team ALICE founded in 2004 by Bruno Lévy. The main scientific goal of ALICE was to develop new algorithms for computer graphics, with a special focus on geometry processing. From 2004 to 2006, we developed new methods for automatic texture mapping (LSCM, ABF++, PGP), that became the de-facto standards. Then we realized that these algorithms could be used to create an abstraction of shapes, that could be used for geometry processing and modeling purposes, which we developed from 2007 to 2013 within the GOODSHAPE StG ERC project. We transformed the research prototype stemming from this project into an industrial geometry processing software, with the VORPALINE PoC ERC project, and commercialized it (TotalEnergies, Dassault Systems, + GeonX and ANSYS currently under discussion). From 2013 to 2018, we developed more contacts and cooperations with the “scientific computing” and “meshing” research communities.

After a part of the team “spun off” around Sylvain Lefebvre and his ERC project SHAPEFORGE to become the MFX team (on additive manufacturing and computer graphics), we progressively moved the center of gravity of the rest of the team from computer graphics towards scientific computing and computational physics, in terms of cooperations, publications and industrial transfer.

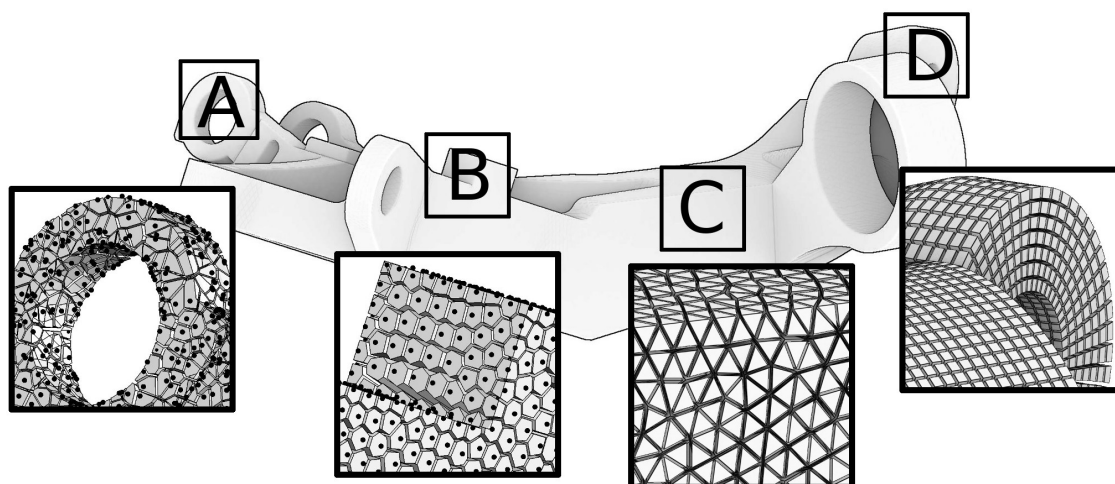


Figure 1: There is a wide range of possibilities to discretize a given domain. (A) Completely unstructured, white noise point sampling; (B) Blue noise point sampling exhibits some structure; (C) tetrahedral mesh; (D) hexahedral mesh.

We realized that *geometry* plays a central role in numerical simulation, and that “cross-pollination” with methods from our field (graphics) will lead to original algorithms. In particular, computer graphics routinely uses irregular and dynamic data structures, more seldom encountered in scientific computing. Conversely, scientific computing routinely uses mathematical tools that are not well spread and not well understood in computer graphics. Our goal is to establish a stronger connection between both domains, and exploit the fundamental aspects of both scientific cultures to develop new algorithms for computational physics.

## 2.1 Scientific grounds

Mesh generation is a notoriously difficult task. A quick search on the [NSF grant web page](#) with “mesh generation AND finite element” keywords returns more than 30 currently active grants for a total of \$8 million. **NASA indicates mesh generation as one of the major challenges for 2030 [43], and estimates that it costs 80% of time and effort in numerical simulation.** This is due to the need for constructing supports that match both the geometry and the physics of the system to be modeled. In our team we pay a particular attention to scientific computing, because we believe it has a world changing impact.

It is very unsatisfactory that meshing, i.e. just “preparing the data” for the simulation, eats up the major part of the time and effort. Our goal is to change the situation, by studying the influence of shapes and discretizations, and inventing new algorithms to automatically generate meshes that can be directly used in scientific computing. This goal is a result of our progressive shift from pure graphics (“Geometry and Lighting”) to real-world problems (“Shape Fidelity”).

Meshing is central in geometric modeling because it provides a way to represent functions on the objects being studied (texture coordinates, temperature, pressure, speed, etc.). There are numerous ways to represent functions, but if we suppose that the functions are piecewise smooth, the most versatile way is to discretize the domain of interest. Ways to discretize a domain range from point clouds to hexahedral meshes; let us list a few of them sorted by the amount of structure each representation has to offer (refer to Figure 1).

- At one end of the spectrum there are **point clouds**: they exhibit no structure at all (white noise point samples) or very little (blue noise point samples). Recent explosive development of acquisition techniques (e.g. scanning or photogrammetry) provides an easy way to build 3D models of real-world objects that range from figurines and cultural heritage objects to geological outcrops and entire city scans. These technologies produce massive, unstructured data (billions of 3D points per scene) that can be directly used for visualization purposes, but this data is not suitable for

high-level geometry processing algorithms and numerical simulations that usually expect meshes. Therefore, at the very beginning of the acquisition-modeling-simulation-analysis pipeline, powerful scan-to-mesh algorithms are required.

During the last decade, many solutions have already been proposed [39, 23, 34, 35, 29], but the problem of building a good mesh from scattered 3D points is far from being solved. Beside the fact that the data is unusually large, the existing algorithms are challenged also by the extreme variation of data quality. Raw point clouds have many defects, they are often corrupted with noise, redundant, incomplete (due to occlusions): *they all are uncertain*.

- **Triangulated surfaces** are ubiquitous, they are the most widely used representation for 3D objects. Some applications like 3D printing do not impose heavy requirements on the surface: typically it has to be watertight, but triangles can have an arbitrary shape. Other applications like texturing require very regular meshes, because they suffer from elongated triangles with large angles.

While being a common solution for many problems, triangle mesh generation is still an active topic of research. The diversity of representations (meshes, NURBS, ...) and file formats often results in a “Babel” problem when one has to exchange data. The only common representation is often the mesh used for visualization, that has in most cases many defects, such as overlaps, gaps or skinny triangles. Re-injecting this solution into the modeling-analysis loop is non-trivial, since again this representation is not well adapted to analysis.

- **Tetrahedral meshes** are the volumic equivalent of triangle meshes, they are very common in the scientific computing community. Tetrahedral meshing is now a mature technology. It is remarkable that still today all the existing software used in the industry is built on top of a handful of kernels, all written by a small number of individuals [27, 41, 47, 30, 40, 42, 28, 22].

Meshing requires a long-term, focused, dedicated research effort that combines deep theoretical studies with advanced software development. We have the ambition to bring this kind of maturity to a different type of mesh (structured, with hexahedra), which is highly desirable for some simulations, and for which, unlike tetrahedra, no satisfying automatic solution exists. In the light of recent contributions, we believe that the domain is ready to overcome the principal difficulties.

- Finally, at the most structured end of the spectrum there are **hexahedral meshes** composed of deformed cubes (hexahedra). They are preferred for certain physics simulations (deformation mechanics, fluid dynamics ...) because they can significantly improve both speed and accuracy. This is because (1) they contain a smaller number of elements (5-6 tetrahedra for a single hexahedron), (2) the associated tri-linear function basis has cubic terms that can better capture higher-order variations, (3) they avoid the locking phenomena encountered with tetrahedra [20], (4) hexahedral meshes exploit inherent tensor product structure and (5) hexahedral meshes are superior in direction dominated physical simulations (boundary layer, shock waves, etc). Being extremely regular, hexahedral meshes are often claimed to be The Holy Grail for many finite element methods [21], outperforming tetrahedral meshes both in terms of computational speed and accuracy.

Despite 30 years of research efforts and important advances, mainly by the Lawrence Livermore National Labs in the U.S. [46, 45], hexahedral meshing still requires considerable manual intervention in most cases (days, weeks and even months for the most complicated domains). Some automatic methods exist [33, 49], that constrain the boundary into a regular grid, but they are not fully satisfactory either, since the grid is not aligned with the boundary. The advancing front method [19] does not have this problem, but generates irregular elements on the medial axis, where the fronts collide. Thus, *there is no fully automatic algorithm that results in satisfactory boundary alignment*.

## 3 Research program

### 3.1 Point clouds

Currently, transforming the raw point cloud into a triangular mesh is a long pipeline involving disparate geometry processing algorithms:

- *Point pre-processing*: colorization, filtering to remove unwanted background, first noise reduction along acquisition viewpoint;
- *Registration*: cloud-to-cloud alignment, filtering of remaining noise, registration refinement;
- *Mesh generation*: triangular mesh from the complete point cloud, re-meshing, smoothing.

The output of this pipeline is a locally-structured model which is used in downstream mesh analysis methods such as feature extraction, segmentation in meaningful parts or building Computer-Aided Design (CAD) models.

It is well known that point cloud data contains measurement errors due to factors related to the external environment and to the measurement system itself [44, 38, 24]. These errors propagate through all processing steps: pre-processing, registration and mesh generation. Even worse, the heterogeneous nature of different processing steps makes it extremely difficult to know *how* these errors propagate through the pipeline. To give an example, for cloud-to-cloud alignment it is necessary to estimate normals. However, the normals are forgotten in the point cloud produced by the registration stage. Later on, when triangulating the cloud, the normals are re-estimated on the modified data, thus introducing uncontrollable errors.

We plan to develop new reconstruction, meshing and re-meshing algorithms, with a specific focus on the accuracy and resistance to all defects present in the input raw data. We think that pervasive treatment of uncertainty is the missing ingredient to achieve this goal. We plan to rethink the pipeline with the position uncertainty maintained during the whole process. Input points can be considered either as error ellipsoids [48] or as probability measures [31]. In a nutshell, our idea is to start by computing an error ellipsoid [50, 36] for each point of the raw data, and then to cumulate the errors (approximations) made at each step of the processing pipeline while building the mesh. In this way, the final users will be able to take the knowledge of the uncertainty into account and rely on this confidence measure for further analysis and simulations. Quantifying uncertainties for reconstruction algorithms, and propagating them from input data to high-level geometry processing algorithms has never been considered before, possibly due to the very different methodologies of the approaches involved. At the very beginning we will re-implement the entire pipeline, and then attack the weak links through all three reconstruction stages.

### 3.2 Parameterizations

One of the favorite tools we use in our team are parameterizations, and we have major contributions to the field: we have solved a fundamental problem formulated more than 60 years ago [2]. Parameterizations provide a very powerful way to reveal structures on objects. The most omnipresent application of parameterizations is texture mapping: texture maps provide a way to represent in 2D (on the map) information related to a surface. Once the surface is equipped with a map, we can do much more than a mere coloring of the surface: we can approximate geodesics, edit the mesh directly in 2D or transfer information from one mesh to another.

Parameterizations constitute a family of methods that involve optimizing an objective function, subject to a set of constraints (equality, inequality, being integer, etc.). Computing the exact solution to such problems is beyond any hope, therefore approximations are the only resort. This raises a number of problems, such as the minimization of highly nonlinear functions and the definition of direction fields topology, without forgetting the robustness of the software that puts all this into practice.

We are particularly interested in a specific instance of parameterization: hexahedral meshing. The idea [6, 5] is to build a transformation  $f$  from the domain to a parametric space, where the distorted domain can be meshed by a regular grid. The inverse transformation  $f^{-1}$  applied to this grid produces the hexahedral mesh of the domain, aligned with the boundary of the object. The strength of this approach is that the transformation may admit some discontinuities. Let us show an example: we start from a tetrahedral mesh (Figure 2, left) and we want to deform it in a way that its boundary is aligned with the integer grid. To allow for a singular edge in the output (the valency 3 edge, Figure 2, right), the input mesh is cut open along the highlighted faces and the central edge is mapped onto an integer grid line (Figure 2, middle). The regular integer grid then induces the hexahedral mesh with the desired topology.

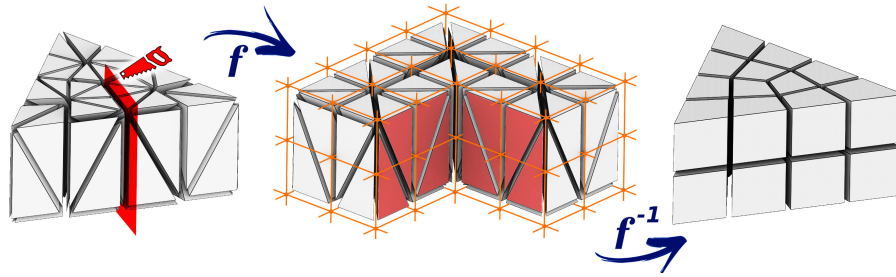


Figure 2: Hex-remeshing via global parameterization. **Left:** Input tetrahedral mesh. To allow for a singular edge in the center, the mesh is cut open along the red plane. **Middle:** Mesh in parametric space. **Right:** Output mesh defined by parameterization.

Current global parameterizations allow grids to be positioned inside geometrically simple objects whose internal structure (the singularity graph) can be relatively basic. We wish to be able to handle more configurations by improving three aspects of current methods:

- Local grid orientation is usually prescribed by minimizing the curvature of a 3D steering field. Unfortunately, this heuristic does not always provide singularity curves that can be integrated by the parameterization. We plan to explore how to embed integrability constraints in the generation of the direction fields. To address the problem, we already identified necessary validity criteria: for example, the permutation of axes along elementary cycles that go around a singularity must preserve one of the axes (the one tangent to the singularity). The first step to enforce this (necessary) condition will be to split the frame field generation into two parts: first we will define a locally stable vector field, followed by the definition of the other two axes by a 2.5D directional field (2D advected by the stable vector field).
- The grid combinatorial information is characterized by a set of integer coefficients whose values are currently determined through numerical optimization of a geometric criterion: the shape of the hexahedra must be as close as possible to the steering direction field. Thus, the number of layers of hexahedra between two surfaces is determined solely by the size of the hexahedra that one wishes to generate. In these settings, degenerate configurations arise easily, and we want to avoid them. In practice, mixed integer solvers often choose to allocate a negative or zero number of layers of hexahedra between two constrained sheets (boundaries of the object, internal constraints or singularities). We will study how to inject strict positivity constraints into these cases, which is a very complex problem because of the subtle interplay between different degrees of freedom of the system. Our first results for quad-meshing of surfaces give promising leads, notably thanks to *motorcycle graphs* [25], a notion we wish to extend to volumes.
- Optimization for the geometric criterion makes it possible to control the average size of the hexahedra, but it does not ensure the bijectivity (even locally) of the resulting parameterizations. Considering other criteria, as we did in 2D [32], would probably improve the robustness of the process. Our idea is to keep the geometry criterion to find the global topology, but try other criteria to improve the geometry.

### 3.3 Hexahedral-dominant meshing

All global parameterization approaches are decomposed into three steps: frame field generation, field integration to get a global parameterization, and final mesh extraction. Getting a full hexahedral mesh from a global parameterization means that it has positive Jacobian everywhere except on the frame field singularity graph. To our knowledge, there is no solution to ensure this property, but some efforts are done to limit the proportion of failure cases. An alternative is to produce hexahedral dominant meshes. Our position is in between those two points of view:

1. We want to produce full hexahedral meshes;



2. We consider as pragmatic to keep hexahedral dominant meshes as a fallback solution.

The global parameterization approach yields impressive results on some geometric objects, which is encouraging, but not yet sufficient for numerical analysis. Note that while we attack the remeshing with our parameterizations toolset, the wish to improve the tool itself (as described above) is orthogonal to the effort we put into making the results usable by the industry. To go further, our idea (as opposed to [37, 26]) is that the global parameterization should not handle all the remeshing, but merely act as a guide to fill a large proportion of the domain with a simple structure; it must cooperate with other remeshing bricks, especially if we want to take final application constraints into account.

For each application we will take as an input domains, sets of constraints and, eventually, fields (e.g. the magnetic field in a tokamak). Having established the criteria of mesh quality (per application!) we will incorporate this input into the mesh generation process, and then validate the mesh by a numerical simulation software.

## 4 Application domains

### 4.1 Geometric Tools for Simulating Physics with a Computer

Numerical simulation is the main targeted application domain for the geometry processing tools that we develop. Our mesh generation tools will be tested and evaluated within the context of our cooperation with Hutchinson, experts in vibration control, fluid management and sealing system technologies. We think that the hex-dominant meshes that we generate have geometrical properties that make them suitable for some finite element analyses, especially for simulations with large deformations.

We also have a tight collaboration with a geophysical modeling specialists via the RING consortium. In particular, we produce hexahedral-dominant meshes for geomechanical simulations of gas and oil reservoirs. From a scientific point of view, this use case introduces new types of constraints (alignments with faults and horizons), and allows certain types of nonconformities that we did not consider until now.

Our cooperation with RhinoTerrain pursues the same goal: reconstruction of buildings from point cloud scans allows to perform 3D analysis and studies on insolation, floods and wave propagation, wind and noise simulations necessary for urban planification.

## 5 Highlights of the year

### 5.1 Awards

Our spin-off company **Tessael** was awarded a prize at the **i-Lab innovation competition** organized by Bpifrance. The project is entitled “Subsurface2.0 (Tessael): New 3D mesh technology to de-risk the geological storage of CO<sub>2</sub>”.

Geological storage of CO<sub>2</sub> is now one of the key levers for containing global warming, but to have a significant impact, all geological actors need to commit to this approach. The transition to this new activity presents many challenges, including the lack of operational technologies to anticipate the risks of gas injection in geological strata, which can range from damage to surface equipment to the induction of microseisms. At Tessael, in order to accelerate this transition for operators of all sizes, we are developing an innovative 3D meshing software technology, which fits into the digital twin of the sub-soil allowing more reliable analyses of the environmental risks involved, as illustrated in Fig. 3.

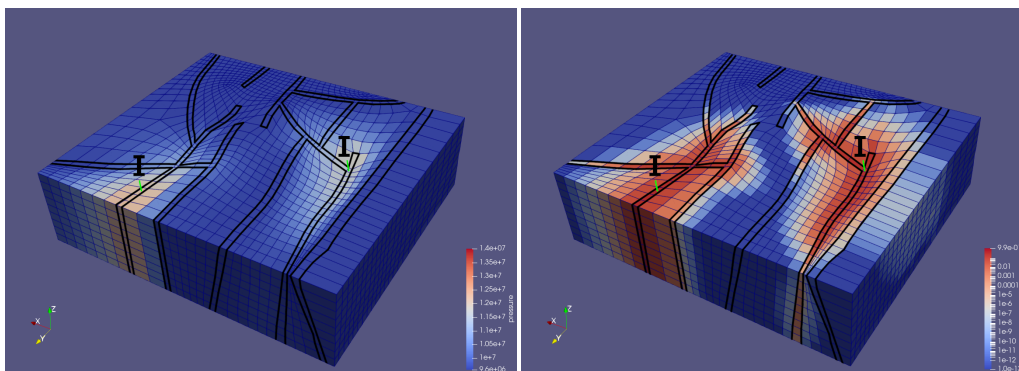


Figure 3: Flow simulation results on CO<sub>2</sub> injection inside fault corridors. Left image: pressure, right image: CO<sub>2</sub> saturation.

Bringing together research work by Pixel and relying on the long experience in the field of geological resource exploitation of the co-founders of Tessael, our geological meshing solution offers three benefits of:

- **Precision:** by a change of paradigm towards unstructured meshes, more flexible in the face of complex subsurface geology.
- **Optimization:** a single mesh for an integrated fluid and geomechanical study.
- **Simplicity:** a state-of-the-art technology compatible with standard software.

Thanks to the Subsurface2.0 project, with Tessael’s scientific and industrial partners, such as Inria, the University of Lorraine, TotalEnergies and IFP, the goal is to double the activity related to CO<sub>2</sub> storage. Through this significant increase in supply, it is expected to see a significant drop in the price of storage within five years, a first step towards large economies.

### 5.2 Programming contests

**Google Hash Code** Our PhD students David Desobry, Yoann Coudert–Osmont, Justine Basselin, and Guillaume Coiffier (Fig. 4, left) participated in the Google Hash Code 2022, an international contest where a dozen thousand teams confront their heuristics on a few instances of a NP-hard problem defined for the contest in a time limit of 4 hours. Our team succeeded in qualifying among the 40 teams to participate in the world final. Finishing **18th in the final**, the team obtained the best result for a 100% French team since 2016. The ranking and problem of the finals are available [on Google’s website](#).

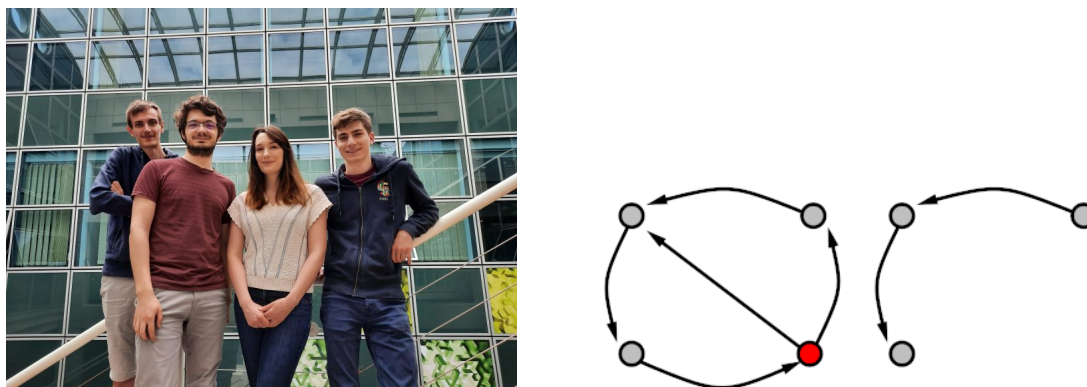


Figure 4: Left: Quiche LORIAinne team. Right: PACE challenge illustration.

**DFVS challenge** Yoann Coudert–Osmont and David Desobry participated to the challenge organized by the International Symposium on Parameterized and Exact Computation (IPEC). They have designed a heuristic solver for Directed Feedback Vertex Set. The problem is to remove a minimum number of vertices from a digraph such that the resulting digraph is acyclic (Fig. 4, right). The team proposed new graph reductions for this problem [7].

The solver was submitted to the 2022 edition of Parameterized Algorithms and Computational Experiments challenge and the team was **ranked 2nd out of 49** over a dataset composed of 200 graphs. DreyFVS first performs a guess on a reduced instance by leveraging the Sinkhorn-Knopp algorithm, to then improve this solution by pipelining two local search methods.

## 6 New software and platforms

### 6.1 New software

#### 6.1.1 stlbfgs

**Name:** C++ L-BFGS implementation using plain STL

**Keyword:** Numerical optimization

**Functional Description:** Many problems can be stated as a minimization of some objective function. Typically, if we can estimate a function value at some point and the corresponding gradient, we can descend in the gradient direction. We can do better and use second derivatives (Hessian matrix). There is an alternative to this costly option: L-BFGS is a quasi-Newton optimization algorithm for solving large nonlinear optimization problems [1,2]. It employs function value and gradient information to search for the local optimum. It uses (as the name suggests) the BFGS (Broyden-Goldfarb-Fletcher-Shanno) algorithm to approximate the inverse Hessian matrix. The size of the memory available to store the approximation of the inverse Hessian is limited (hence L- in the name): in fact, we do not need to store the approximated matrix directly, but rather we need to be able to multiply it by a vector (most often the gradient), and this can be done efficiently by using the history of past updates.

Suprisingly enough, most L-BFGS solvers that can be found in the wild are wrappers/transpilation of the original Fortran/Matlab codes by Jorge Nocedal and Dianne O’Leary.

Such code is impossible to improve if, for example, we are working near the limits of floating point precision, therefore meet stlbfgs: a from-scratch modern C++ implementation. The project has zero external dependencies, no Eigen, nothing, plain standard library.

This implementation uses Moré-Thuente line search algorithm [3]. The preconditioning is performed using the M1QN3 strategy [5,6]. Moré-Thuente line search routine is tested against data [3] (Tables 1-6), and L-BFGS routine is tested on problems taken from [4].

- [1] J. Nocedal (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35/151, 773-782. 3
- [2] Dong C. Liu, Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming* (1989).
- [3] Jorge J. Moré, David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software* (1994)
- [4] Jorge J. Moré, Burton S. Garbow, Kenneth E. Hillstom, "Testing Unconstrained Optimization Software", *ACM Transactions on Mathematical Software* (1981)
- [5] Gilbert JC, Lemaréchal C. The module M1QN3. INRIA Rep., version. 2006,3:21.
- [6] Gilbert JC, Lemaréchal C. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming* 45, pp. 407-435, 1989.

**URL:** <https://github.com/ssloy/stlbfgs>

**Contact:** Dmitry Sokolov

### 6.1.2 ultimaille

**Keyword:** Mesh

**Functional Description:** This library does not contain any ready-to-execute remeshing algorithms. It simply provides a friendly way to manipulate a surface/volume mesh, it is meant to be used by your geometry processing software.

There are lots of mesh processing libraries in the wild, excellent specimens are:

geogram

libigl

pmp

CGAL

We are, however, not satisfied with either of those. At the moment of this writing, Geogram, for instance, has 847 thousands (sic!) of lines of code. We strive to make a library under 10K loc able to do common mesh handling tasks for surfaces and volumes. Another reason to create yet-another-mesh-library is that we like explicit types and avoid auto as long as it reasonable. In practice it means that we cannot use libigl for the simple reason that we do not know what this data represents:

Eigen::MatrixXd V, Eigen::MatrixXi F, Is it a polygonal surface or a tetrahedral mesh? If surface, is it triangulated or is it a generic polygonal mesh? I simply can not tell... Thus, ultimaille provides several classes that allow to represent meshes:

PointSet PolyLine Triangles, Quads, Polygons Tetrahedra, Hexahedra, Wedges, Pyramids You can not mix tetrahedra and hexahedra in a single mesh, we believe that it is confusing to do otherwise. If you need a mixed mesh, create a separate mesh for each cell type: these classes allow to share a common set of vertices via a `std::shared_ptr`.

**Common principles** This library is meant to have a reasonable performance. It means that we strive to make it as rapid as possible as long as it does not deteriorate the readability of the source code. All the library is built around STL containers (mainly `std::vector<int>`), normally it will have no `malloc/free/new/delete` instructions. Despite that, there will be no `size_t` and `iterator::` in the code. An `int` is an `int`. Period. There will be as little templates as it is reasonable, the default data types are `int` and `double`.

**URL:** <https://github.com/ssloy/ultimaille>

**Contact:** Dmitry Sokolov

## 7 New results

**Participants:** Dmitry Sokolov, Nicolas Ray, Étienne Corman, Laurent Alonso, François Protais, David Lopez, Justine Basselin, Guillaume Coiffier, David Desobry, Yoann Coudert-Osmont.

### 7.1 Polycube hexahedral meshing

Polycube-maps are used as base complexes in various fields of computational geometry, including the generation of regular all-hexahedral meshes free of internal singularities. For our contract with the CEA, we focus on robust polycube generation enabling as coarse polycube hexahedral remeshing as possible. So, starting from a tetrahedral mesh, we want to compute a hexahedral mesh. Polycube methods are usually split into four steps (Fig. 5):

1. **Flagging** — The boundary of the volume is segmented into charts assigned to one of the six possible flags corresponding to six vectors aligned with a reference frame.
2. **Deformation** — A continuous, positive Jacobian mapping to the parametric space is then computed such that its distortion is minimized and the image of each chart is planar and perpendicular to the corresponding flag.
3. **Quantization** — Then we align the faces of the polycuboid with the integer grid to obtain a polycube. At this step, the volume can be filled by a unit grid.
4. **Inversion** — The polycube deformation is inverted to extract the final hexahedral mesh.

All four steps have robustness issues, and this year we have addressed the flagging and the deformation.

**Flagging** The strict alignment constraints behind polycube-based methods make their computation challenging for CAD models used in numerical simulation via finite element methods (FEM). We proposed a novel approach [13] based on an evolutionary algorithm to robustly compute polycube-maps in this context. We address the labelling problem, which aims to precompute polycube alignment by assigning one of the base axes to each boundary face on the input. Previous research has described ways to initialize and improve a labelling via greedy local fixes. However, such algorithms lack robustness and often converge to inaccurate solutions for complex geometries. Our proposed framework Evocube (Fig. 6) alleviates this issue by embedding labelling operations in an evolutionary heuristic, defining fitness, crossover, and mutations in the context of labelling optimization. We evaluate our method on a thousand smooth and CAD meshes, showing that it converges to accurate labellings on a wide range of shapes. The limitations of our method are also discussed thoroughly.

**Quantization** An important part of recent advances in hexahedral meshing focuses on the deformation of a domain into a polycube; the polycube deformed by the inverse map fills the domain with a hexahedral mesh. These methods are appreciated because they generate highly regular meshes. This year we have addressed a robustness issue that systematically occurs when a coarse mesh is desired: algorithms produce deformations that are not one-to-one, leading to collapse of large portions of the model when trying to apply the (undefined) inverse map (Fig. 7). The origin of the problem is that the deformation requires mixed integer optimization, where the difficulty to enforce the integer constraints is directly tied to the expected coarseness. Our solution [14] is to introduce sanity constraints preventing the loss of bijectivity due to the integer constraints.

### 7.2 Functional Maps

We continue to improve performance of our mapping algorithms. In particular, we aim at finding meaningful correspondences between 3D shapes. We build upon a representation of mappings between surfaces as linear operators acting on functions called "functional maps". This representation has proven

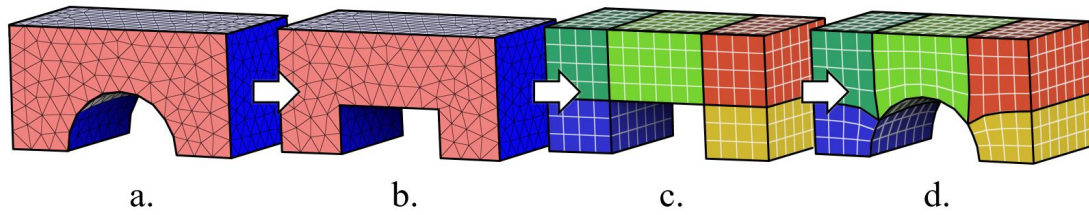


Figure 5: Generating polycube hexahedral meshes is done in 4 steps: (a.) the domain is first colored, (b.) and this coloring allows to deform it into a polycuboid. (c.) Then this polycuboid is mapped to an integer grid, thus providing us a polycube (a cluster of cubes). (d.) Applying the inverse deformation on the polycube gives us a hexahedral mesh of the original domain.

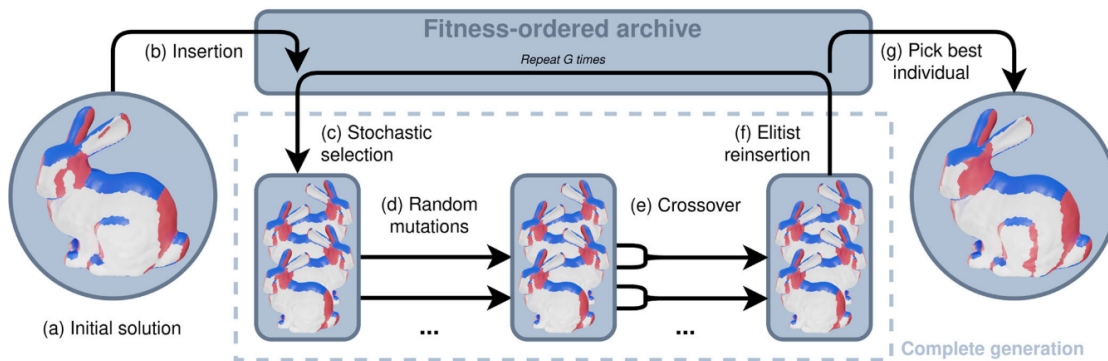


Figure 6: Architecture of our genetic optimization framework.

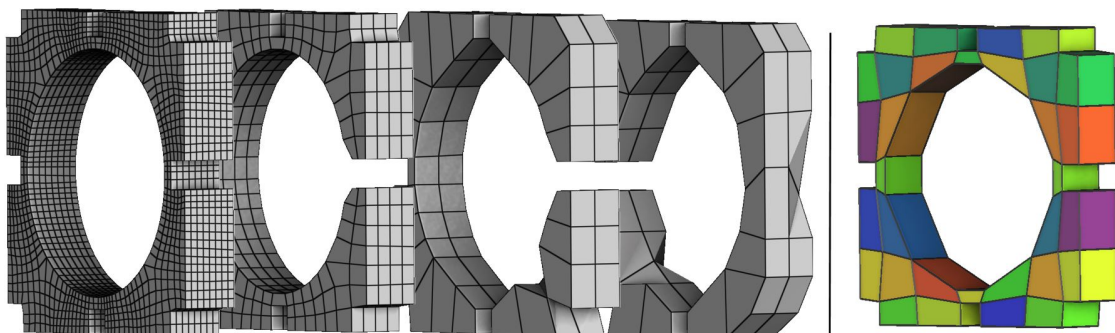


Figure 7: Polycube-based hexahedral meshing algorithms are able to produce very regular grids (left), but fail to capture important geometric features with a coarse grid (left → middle). Our method allows us to produce a coarse mesh preserving important geometric features (right).

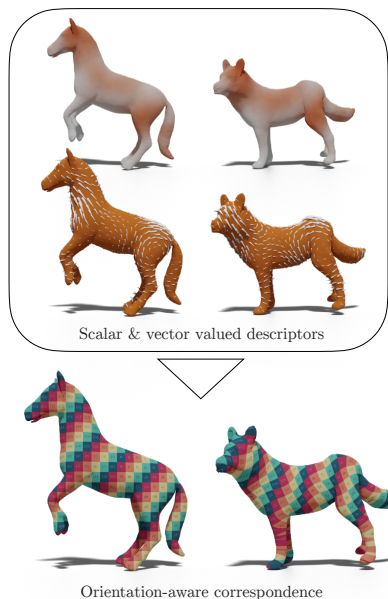


Figure 8: Our method aims at producing orientation-preserving maps for non-rigid 3D shape matching in a fully unsupervised setting through the estimation of descriptors whose gradients also align on source and target shapes.

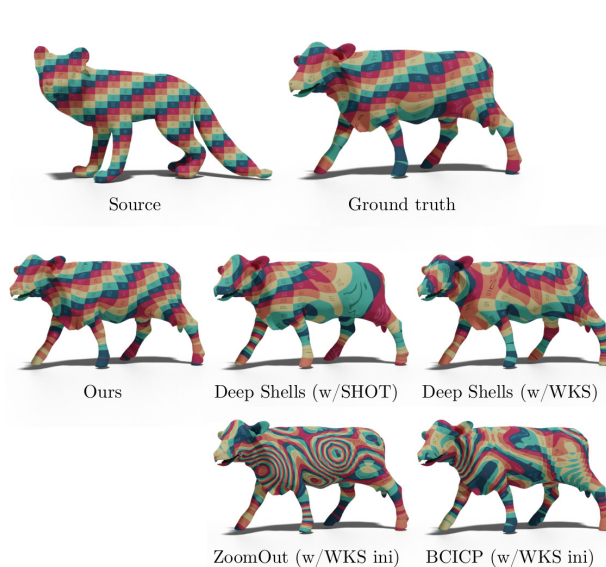


Figure 9: Qualitative results for baselines on the SMAL dataset. We see that our method yields a map very close to ground-truth, even on this challenging example with strong non-isometric distortions. Meanwhile, both axiomatic and learning-based baselines fail to predict accurate correspondences.

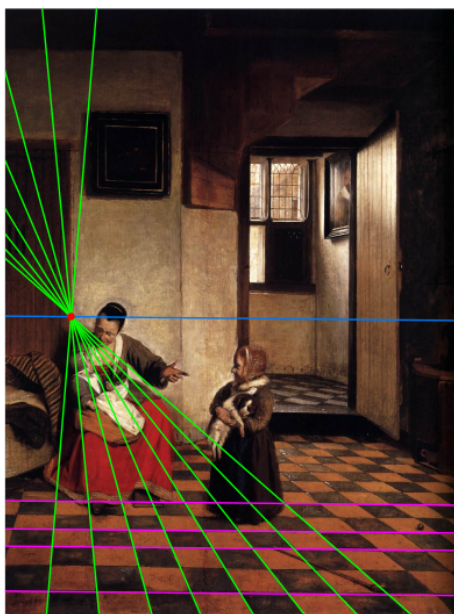


Figure 10: Extracted vanishing lines, vanishing point and horizon using the proposed method. The painting: A Woman With a Baby in Her Lap, Pieter de Hooch. Source: [Wikimedia Commons](#)



Figure 11: A closed space-filling curve inside a brain shape. In the left part, the trajectory is optimized to be boundary aligned while the right part the trajectory is optimized for isotropy. The number of transitions between the two color zones is also minimal with only two transitions. Minimizing the number of color transitions is very important for the quality of colored prints.

to be very flexible and efficient. The problem, however, as it mostly relies on geodesic distances, it is challenging to take the surface orientation into account. As a consequence, if a surface has internal symmetries, there are several equivalently good solutions to the shape matching problem.

This year, we introduced complex functional maps [11], which extend the functional map framework to conformal maps between tangent vector fields on surfaces. A key property of these maps is their orientation awareness (refer to Fig. 8). More specifically, we demonstrated that, unlike regular functional maps that link functional spaces of two surfaces, our complex functional maps establish a link between oriented tangent bundles, thus permitting robust and efficient transfer of tangent vector fields. By first endowing and then exploiting the tangent bundle of each shape with a complex structure, the resulting operations become naturally orientation aware, thus favoring orientation and angle preserving correspondence across shapes, without relying on descriptors or extra regularization.

Using complex functional map representation, we also proposed [12] a new deep learning approach to learn orientation-aware features in a fully unsupervised setting (refer to Fig 9). Our architecture is built on top of DiffusionNet, making it robust to discretization changes. Additionally, we introduce a vector field-based loss, which promotes orientation preservation without using (often unstable) extrinsic descriptors.

### 7.3 Image analysis and synthesis

In collaboration with Elmar Eisemann and Ricardo Marroqium from TU Delft, we proposed [9] a semi-automatic method to extract perspective lines from paintings (Fig. 10). Perspective cues play an important role in painting analysis as they may unveil important characteristics about the painter's techniques and creation process. Nevertheless, extracting perspective lines and their corresponding vanishing points is usually a laborious manual task. Moreover, small variations in the lines may lead to large variations in the vanishing points. The goal of this work is to mitigate the human variability factor and reduce the workload of this task.

In collaboration with the MFX team, we explored the optimization of closed space-filling curves under orientation objectives (Fig. 11). By solidifying material along the closed curve, solid layers of 3D prints can be manufactured in a single continuous extrusion motion. The control over orientation enables the deposition to align with specific directions in different areas, or to produce a locally uniform distribution of orientations, patterning the solidified volume in a precisely controlled manner. Our optimization framework [8] proceeds in two steps. First, we cast a combinatorial problem, optimizing Hamiltonian cycles within a specially constructed graph. We rely on a stochastic optimization process based on local operators that modify a cycle while preserving its Hamiltonian property. Second, we use the result to initialize a geometric optimizer that improves the smoothness and uniform coverage of the cycle while further optimizing for alignment and orientation objectives.

## 8 Bilateral contracts and grants with industry

**Participants:** Dmitry Sokolov, Nicolas Ray, Justine Basselin, François Protais, David Desobry.

### 8.1 Bilateral contracts with industry

- *Company:* CEA

*Duration:* 01/10/2019 – 30/09/2022

*Participants:* Dmitry Sokolov, Nicolas Ray and François Protais

*Abstract:* This project revolves around the generation of Polycubes guided by orientation fields. The first goal of the project is to define a new Polycube method that deforms an object along a previously generated 3D orientation field. Such a solution would overcome two major defects of Polycube methods, namely : (1) the possible absence of aligned mesh layers along the smooth edges; and (2) the poor treatment of sharp edges (which is very common on mechanical parts in CAD).



François defended his PhD [16] on October 21, 2022.

- *Company:* RhinoTerrain  
*Duration:* 01/12/2019 – 30/03/2024  
*Participants:* Dmitry Sokolov, Nicolas Ray and Justine Basselin  
*Abstract:* In this project, we are interested in the reconstruction phase in the context of LIDAR point clouds of city districts. These data are acquired via an airplane and contain every object present in the city: cars, trees, building roofs, roads and ground features, etc. Applications of these data, ranging from city visualization for tourism to flood and wind simulation, require the reconstruction of buildings as geometrical objects. As LIDAR point clouds are acquired from the sky, buildings are only represented by their roofs. Hence, determining the polygonal surfaces of roofs enables the reconstruction of the whole city by extrusion.  
 Justine defended her PhD [15] on December 14, 2022.

- *Company:* TotalEnergies  
*Duration:* 01/10/2020 – 30/03/2024  
*Participants:* Dmitry Sokolov, Nicolas Ray and David Desobry  
*Abstract:* The goal of this project is to improve the accuracy of rubber behavior simulations for certain parts produced by TotalEnergies, notably gaskets. To do this, both parties need to develop meshing methods adapted to the simulation of large deformations in non-linear mechanics. The Pixel team has a strong expertise in hex-dominant meshing, TotalEnergies has a strong background in numerical simulation within an industrial context. This collaborative project aims to take advantage of both expertises.

## 9 Dissemination

**Participants:** Dmitry Sokolov, Étienne Corman, Dobrina Boltcheva, Nicolas Ray, Justine Basselin, François Protais, Yoann Coudert-Osmont, David Desobry.

### 9.1 Promoting scientific activities

#### 9.1.1 Conference

**Reviewer** Members of the team were reviewers for Eurographics, SIGGRAPH, SIGGRAPH Asia, ISVC, Pacific Graphics, and SPM.

#### 9.1.2 Journal

**Reviewer - reviewing activities** Members of the team were reviewers for Computer Aided Design (Elsevier), Computer Aided Geometric Design (Elsevier), Transactions on Visualization and Computer Graphics (IEEE), Transactions on Graphics (ACM), Computer Graphics Forum (Wiley), Computational Geometry: Theory and Applications (Elsevier) and Computers & Graphics (Elsevier).

#### 9.1.3 Research administration

Dmitry Sokolov is responsible for the Industrial Club of GdR IG-RV (Groupement de recherche Informatique Géométrique et Graphique, Réalité Virtuelle et Visualisation).

### 9.2 Teaching - Supervision - Juries

#### 9.2.1 Teaching

Dmitry Sokolov is responsible for the 3d year of computer science license at the University of Lorraine. Members of the team have taught following courses:

- Master: Étienne Corman, Analysis and Deep Learning on Geometric Data, 12h, M2, École Polytechnique
- BUT 2 INFO : Dobrina Boltcheva, Advanced Object Oriented Programming & UML, 100h, 2A, IUT Saint-Dié-des-Vosges
- BUT 2 INFO : Dobrina Boltcheva, UML Modeling, 30h, 2A, IUT Saint-Dié-des-Vosges
- BUT 2 INFO : Dobrina Boltcheva, Advanced algorithmics, 30h, 2A, IUT Saint-Dié-des-Vosges
- BUT 2 INFO : Dobrina Boltcheva, Image Processing, 30h, 2A, IUT Saint-Dié-des-Vosges
- BUT 3 INFO : Dobrina Boltcheva, Computer Graphics, 20h, 3A, IUT Saint-Dié-des-Vosges
- Master : François Protais, Analyse et Conception de Logiciels, 11h, M1, University of Lorraine
- Master : François Protais, Représentations des Données Visuelles, 8, M1, University of Lorraine
- Master : François Protais, Fonctionnement d'un moteur du rendu 3D, 12h, M1, University of Lorraine
- License : Guillaume Coiffier, Automata and Langage theory, 16h, L2, University of Lorraine
- License : Guillaume Coiffier, Programming, 16h + 16h, L1, University of Lorraine
- License : Guillaume Coiffier, Optimization, 16h, L3, University of Lorraine
- License : David Desobry, Programmation avancée, 20h, L2, University of Lorraine
- License : David Desobry, Analyse de données, 16h, L3, University of Lorraine
- Master : Yoann Coudert–Osmont, Systèmes 2, 12h, L3, Université de Lorraine
- Master : Yoann Coudert–Osmont, Méthodologie de programmation et de conception avancée, 22h, L1, Université de Lorraine
- Master : Yoann Coudert–Osmont, Outils Système, 20h, L2, Université de Lorraine
- Master : Yoann Coudert–Osmont, NUMOC, 10h, L1, Université de Lorraine
- License : Dmitry Sokolov, Programming, 48h, 2A, University of Lorraine
- License : Dmitry Sokolov, Logic, 30h, 3A, University of Lorraine
- License : Dmitry Sokolov, Algorithmics, 25h, 3A, University of Lorraine
- License : Dmitry Sokolov, Computer Graphics, 16h, M1, University of Lorraine
- Master : Dmitry Sokolov, Logic, 22h, M1, University of Lorraine
- Master : Dmitry Sokolov, Computer Graphics, 30h, M1, University of Lorraine
- Master : Dmitry Sokolov, 3D data visualization, 15h, M1, University of Lorraine
- Master : Dmitry Sokolov, 3D printing, 12h, M2, University of Lorraine
- Master : Dmitry Sokolov, Numerical modeling, 12h, M2, University of Lorraine

## 9.2.2 Supervision

### Ongoing

- PhD in progress: David Desobry, Generation of hexahedral meshes for large deformation simulations, started September 2020, advisors: Dmitry Sokolov, Nicolas Ray, Jeanne Pellerin
- PhD in progress: Guillaume Coiffier, Global parametrization algorithms for hexahedral meshing, started September 2020, advisors: Étienne Corman, Dmitry Sokolov
- Yoann Coudert-Osmont, 2.5D Frame Fields for Hexahedral Meshing, started September 2021, advisors: Nicolas Ray, Dmitry Sokolov

**PhD defenses** This year two of our students defended their PhDs, both were in connection with bilateral contracts with industry (more details in section 8)

- François Protais has defended his PhD [16] on October 21, 2022. He is now working at Siemens on quadrilateral meshing for Simcenter STAR-CCM+.
- Justine Basselin has defended her PhD [15] on December 14, 2022. She is now working for Dassault Systèmes.

### 9.2.3 Juries

- Dmitry Sokolov participated in the PhD jury of Paul Viville (University of Strasbourg) as a reviewer.
- Dmitry Sokolov participated in the PhD jury of Igor Petranevskiy (ITMO, Saint Petersburg, Russia) as a reviewer.

## 9.3 Popularization

### 9.3.1 Interventions

We have participated to different science popularization events:

- *Chiche*
  - Guillaume Coiffier and Justine Basselin, Lycée Notre-Dame Saint-Joseph, Épinal, 14/1/2022, 21/1/2022, 14/3/2022
  - Guillaume Coiffier and Justine Basselin, Lycée de la Salle, Metz, 8/6/2022
- *Ma thèse en 180 secondes*, Guillaume Coiffier, University of Lorraine finals, 10/3/2022
- *Nuit européenne des chercheurs*, Guillaume Coiffier, Justine Basselin, Dmitry Sokolov, 29/9/2022
- *Experimentarium*, Guillaume Coiffier, Collège Jules Verne, Vittel and Médiathèque de Vittel, 12/10/2022
- Fête de la science, Guillaume Coiffier and Justine Basselin, Faculty of Science, 14 and 15/10/2022

Dmitry Sokolov was also invited to talk at the round table dedicated to teaching computer graphics in France during JFIG2022 (Bordeaux).

## 10 Scientific production

### 10.1 Major publications

- [1] J. Basselin, L. Alonso, N. Ray, D. Sokolov, S. Lefebvre and B. Lévy. ‘Restricted Power Diagrams on the GPU’. In: *Computer Graphics Forum* 40.2 (4th June 2021). DOI: [10.1111/cgf.142610](https://doi.org/10.1111/cgf.142610). URL: <https://hal.archives-ouvertes.fr/hal-03169575>.
- [2] V. Garanzha, I. Kaporin, L. Kudryavtseva, F. Protais, N. Ray and D. Sokolov. ‘Foldover-free maps in 50 lines of code’. In: *ACM Transactions on Graphics* Volume 40.issue 4 (19th July 2021), Article No.102, pp 1–16. DOI: [10.1145/3450626.3459847](https://doi.org/10.1145/3450626.3459847). URL: <https://hal.archives-ouvertes.fr/hal-03127350>.
- [3] N. Ray and D. Sokolov. ‘Robust Polyline Tracing for N-Symmetry Direction Field on Triangulated Surfaces’. In: *ACM Trans. Graph.* 33.3 (June 2014), 30:1–30:11. DOI: [10.1145/2602145](https://doi.org/10.1145/2602145). URL: <http://doi.acm.org/10.1145/2602145>.
- [4] N. Ray, D. Sokolov, S. Lefebvre and B. Lévy. ‘Meshless Voronoi on the GPU’. In: *ACM Trans. Graph.* 37.6 (Dec. 2018), 265:1–265:12. DOI: [10.1145/3272127.3275092](https://doi.org/10.1145/3272127.3275092). URL: <http://doi.acm.org/10.1145/3272127.3275092>.
- [5] N. Ray, D. Sokolov and B. Lévy. ‘Practical 3D Frame Field Generation’. In: *ACM Trans. Graph.* 35.6 (Nov. 2016), 233:1–233:9. DOI: [10.1145/2980179.2982408](https://doi.org/10.1145/2980179.2982408). URL: <http://doi.acm.org/10.1145/2980179.2982408>.
- [6] D. Sokolov, N. Ray, L. Untereiner and B. Lévy. ‘Hexahedral-Dominant Meshing’. In: *ACM Transactions on Graphics* 35.5 (2016), pp. 1–23. DOI: [10.1145/2930662](https://doi.org/10.1145/2930662). URL: <https://hal.inria.fr/hal-01397846>.

### 10.2 Publications of the year

#### International journals

- [7] G. Bathie, G. Berthe, Y. Coudert-Osmont, D. Desobry, A. Reinald and M. Rocton. ‘PACE solver description: DreyFVS’. In: *Leibniz International Proceedings in Informatics* (14th Dec. 2022). DOI: [10.4230/LIPIcs.IPEC.2022.31](https://doi.org/10.4230/LIPIcs.IPEC.2022.31). URL: <https://hal.inria.fr/hal-03920659>.
- [8] A. Bedel, Y. Coudert-Osmont, J. Martínez, R. I. Nishat, S. Whitesides and S. Lefebvre. ‘Closed space-filling curves with controlled orientation for 3D printing’. In: *Computer Graphics Forum* (2022). DOI: [10.1111/cgf.14488](https://doi.org/10.1111/cgf.14488). URL: <https://hal.inria.fr/hal-03185200>.
- [9] Y. Coudert-Osmont, E. Eisemann and R. Marroquim. ‘Semi-Automatic Perspective Lines from Paintings’. In: *Eurographics Workshop on Graphics and Cultural Heritage* (19th Dec. 2022). DOI: [10.2312/gch.20221219](https://doi.org/10.2312/gch.20221219). URL: <https://hal.inria.fr/hal-03920890>.
- [10] D. Desobry, F. Protais, N. Ray, E. Corman and D. Sokolov. ‘Frame Fields for CAD models’. In: *Lecture Notes in Computer Science* 13018 (1st Jan. 2022), pp. 421–434. DOI: [10.1007/978-3-030-90436-4\\_34](https://doi.org/10.1007/978-3-030-90436-4_34). URL: <https://hal.inria.fr/hal-03537852>.
- [11] N. Donati, E. Corman, S. Melzi and M. Ovsjanikov. ‘Complex Functional Maps : a Conformal Link Between Tangent Bundles’. In: *Computer Graphics Forum* 41.1 (1st Feb. 2022), pp. 317–334. URL: <https://hal.inria.fr/hal-03480605>.
- [12] N. Donati, E. Corman and M. Ovsjanikov. ‘Deep Orientation-Aware Functional Maps: Tackling Symmetry Issues in Shape Matching’. In: *CVPR '22 : IEEE Conference on Computer Vision and Pattern Recognition* (2022). DOI: [10.48550/arXiv.2204.13453](https://doi.org/10.48550/arXiv.2204.13453). URL: <https://hal.archives-ouvertes.fr/hal-03654308>.
- [13] C. Dumery, F. Protais, S. Mestrallet, C. Bourcier and F. Ledoux. ‘Evocube: a Genetic Labeling Framework for Polycube-Maps’. In: *Computer Graphics Forum* (31st Aug. 2022). DOI: [10.1111/cgf.14649](https://doi.org/10.1111/cgf.14649). URL: <https://hal.archives-ouvertes.fr/hal-03657779>.
- [14] F. Protais, M. Reberol, N. Ray, E. Corman, F. Ledoux and D. Sokolov. ‘Robust Quantization for Polycube Maps’. In: *Computer-Aided Design* Volume 150.103321 (1st Sept. 2022). DOI: [10.1016/j.cad.2022.103321](https://doi.org/10.1016/j.cad.2022.103321). URL: <https://hal.archives-ouvertes.fr/hal-03076711>.

### Doctoral dissertations and habilitation theses

- [15] J. Basselin. ‘Building recognition from 3D point clouds’. Université de Lorraine, 14th Dec. 2022. URL: <https://hal.inria.fr/tel-03943948>.
- [16] F. Protais. ‘Polycube-dominant meshing’. Université de Lorraine, 21st Oct. 2022. URL: <https://hal.archives-ouvertes.fr/tel-03775686>.

### Reports & preprints

- [17] G. Coiffier and E. Corman. *Seamless Global Parametrization in a Single Optimization*. 17th May 2022. URL: <https://hal.science/hal-03670525>.
- [18] V. Garanzha, I. Kaporin, L. Kudryavtseva, F. Protais, D. Desobry and D. Sokolov. *Lowest distortion mappings and equidistribution principle*. 2nd Nov. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03546749>.

## 10.3 Cited publications

- [19] T. C. Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte and C. Geuzaine. ‘A frontal approach to hex-dominant mesh generation’. In: *Adv. Model. and Simul. in Eng. Sciences* 1.1 (2014), 8:1–8:30. DOI: [10.1186/2213-7467-1-8](https://doi.org/10.1186/2213-7467-1-8). URL: <https://doi.org/10.1186/2213-7467-1-8>.
- [20] S. E. Benzley, K. Ernest Perry, B. C. Merkle and G. Sjaardema. ‘A Comparison of All-Hexahedral and All-Tetrahedral Finite Element Meshes for Elastic and Elasto-Plastic Analysis’. In: *International Meshing Roundtable conf. proc.* 1995.
- [21] T. Blacker. ‘Meeting the Challenge for Automated Conformal Hexahedral Meshing’. In: *9th International Meshing Roundtable*. 2000, pp. 11–20.
- [22] CGAL, *Computational Geometry Algorithms Library*. <http://www.cgal.org>.
- [23] *Cloud Compare*. <http://www.danielgm.net/cc/release/>.
- [24] Z. Du, M. Zhu, Z. Wu and J. Yang. ‘Measurement uncertainty on the circular features in coordinate measurement system based on the error ellipse and Monte Carlo methods’. In: *Measurement Science and Technology* 27.12 (2016), p. 125016. URL: <http://stacks.iop.org/0957-0233/27/i=12/a=125016>.
- [25] D. Eppstein, M. T. Goodrich, E. Kim and R. Tamstorf. ‘Motorcycle Graphs: Canonical Quad Mesh Partitioning’. In: *Proceedings of the Symposium on Geometry Processing*. SGP ’08. Copenhagen, Denmark: Eurographics Association, 2008, pp. 1477–1486. URL: <http://dl.acm.org/citation.cfm?id=1731309.1731334>.
- [26] X. Gao, W. Jakob, M. Tarini and D. Panozzo. ‘Robust Hex-Dominant Mesh Generation using Field-Guided Polyhedral Agglomeration’. In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36.4 (July 2017). DOI: [10.1145/3072959.3073676](https://doi.org/10.1145/3072959.3073676).
- [27] P. L. George, F. Hecht and E. Saltel. ‘Fully Automatic Mesh Generator for 3D Domains of Any Shape’. In: *IMPACT Comput. Sci. Eng.* 2.3 (Dec. 1990), pp. 187–218. DOI: [10.1016/0899-8248\(90\)90012-Y](https://doi.org/10.1016/0899-8248(90)90012-Y). URL: [http://dx.doi.org/10.1016/0899-8248\(90\)90012-Y](http://dx.doi.org/10.1016/0899-8248(90)90012-Y).
- [28] C. Geuzaine and J.-F. Remacle. ‘Gmsh: a three-dimensional finite element mesh generator’. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.
- [29] *GRAPHITE*. <http://alice.loria.fr/software/graphite/doc/html/>.
- [30] Y. Ito, A. M. Shih and B. K. Soni. ‘Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front Method’. In: *International Meshing Roundtable conf. proc.* 2004.
- [31] B. Lévy and E. Schwindt. ‘Notions of Optimal Transport theory and how to implement them on a computer’. In: *Computer and Graphics* (2018).
- [32] B. Lévy, S. Petitjean, N. Ray and J. Maillot. ‘Least squares conformal maps for automatic texture atlas generation’. In: *ACM transactions on graphics (TOG)*. Vol. 21. ACM, 2002, pp. 362–371.

- [33] L. Maréchal. 'A New Approach to Octree-Based Hexahedral Meshing'. In: *International Meshing Roundtable conf. proc.* 2001.
- [34] Meshlab. <http://www.meshlab.net/>.
- [35] MeshMixer. <http://www.meshmixer.com/>.
- [36] C. Mezian, B. Vallet, B. Soheilian and N. Paparoditis. 'Uncertainty Propagation for Terrestrial Mobile Laser Scanner'. In: *SPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2016.
- [37] M. Nieser, U. Reitebuch and K. Polthier. 'CubeCover - Parameterization of 3D Volumes'. In: *Computer Graphics Forum* 30.5 (2011), pp. 1397–1406. DOI: [10.1111/j.1467-8659.2011.02014.x](https://doi.org/10.1111/j.1467-8659.2011.02014.x). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.02014.x>.
- [38] J.-H. Park, Y.-D. Shin, J.-H. Bae and M.-H. Baeg. *Spatial Uncertainty Model for Visual Features Using a Kinect Sensor*. 2012.
- [39] Point Cloud Library. <http://www.pointclouds.org/downloads/>.
- [40] Schöberl. 'NETGEN An advancing front 2D/3D-mesh generator based on abstract rules'. In: *Computing and visualization in science* 1 (1 1997).
- [41] M. Shephard and M. Georges. 'Automatic three-dimensional mesh generation by the finite octree technique'. In: *International Journal for Numerical Methods in Engineering* 32 (4 1991).
- [42] H. Si. 'TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator'. In: *ACM Trans. on Mathematical Software* 41 (2 2015).
- [43] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, G. William, L. Elizabeth and D. Mavriplis. *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*. Tech. rep. NASA/CR-2014-218178, NF1676L-18332, 2014.
- [44] S. Soudarissanane, R. Lindenbergh, M. Menenti and P. Teunissen. 'Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points'. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.4 (2011), pp. 389–399. DOI: <https://doi.org/10.1016/j.isprsjprs.2011.01.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0924271611000098>.
- [45] M. L. Staten, S. J. Owen and T. Blacker. 'Unconstrained Paving & Plastering: A New Idea for All Hexahedral Mesh Generation'. In: *International Meshing Roundtable conf. proc.* 2005.
- [46] T. J. Tautges, T. Blacker and S. A. Mitchell. 'The whisker weaving algorithm'. In: *International Journal of Numerical Methods in Engineering* (1996).
- [47] N. P. Weatherill and O. Hassan. 'Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints'. In: *International Journal for Numerical Methods in Engineering* 37.12 (1994), pp. 2005–2039. DOI: [10.1002/nme.1620371203](https://doi.org/10.1002/nme.1620371203). URL: <http://dx.doi.org/10.1002/nme.1620371203>.
- [48] M. Ylimäki, J. Kannala and J. Heikkilä. 'Robust and Practical Depth Map Fusion for Time-of-Flight Cameras'. In: *Image Analysis*. Ed. by P. Sharma and F. M. Bianchi. Cham: Springer International Publishing, 2017, pp. 122–134.
- [49] Y. Zhang and C. Bajaj. 'Adaptive and Quality Quadrilateral/Hexahedral Meshing from Volumetric Data'. In: *International Meshing Roundtable conf. proc.* 2004.
- [50] D. Zhengchun, W. Zhaoyong and Y. Jianguo. 'Point cloud uncertainty analysis for laser radar measurement system based on error ellipsoid model'. In: *Optics and Lasers in Engineering* 79 (2016), pp. 78–84. DOI: <https://doi.org/10.1016/j.optlaseng.2015.11.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0143816615002675>.