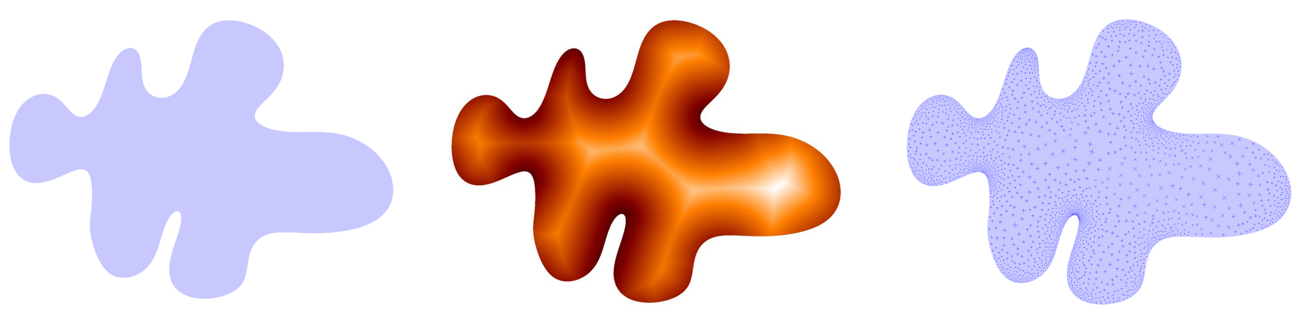
Isotropic 2D Quadrangle Meshing with Size and Orientation Control

Bertrand Pellenard[1], Pierre Alliez[1] and Jean-Marie Morvan[2-3]; [1]: Inria Sophia Antipolis - Méditerranée, [2]: Université Lyon 1/CNRS, [3]: King Abdullah University of Science and Technology

Abstract: we propose an approach for automatically generating isotropic 2D quadrangle meshes from arbitrary domains with a fine control over sizing and orientation of the elements. At the heart of our algorithm is an optimization procedure that, from a coarse initial tiling of the 2D domain, enforces each of the desirable mesh quality criteria (size, shape, orientation, degree, regularity) one at a time, in an order designed not to undo previous enhancements. Our experiments demonstrate how well our resulting quadrangle meshes conform to a wide range of input sizing and orientation fields.

INPUT



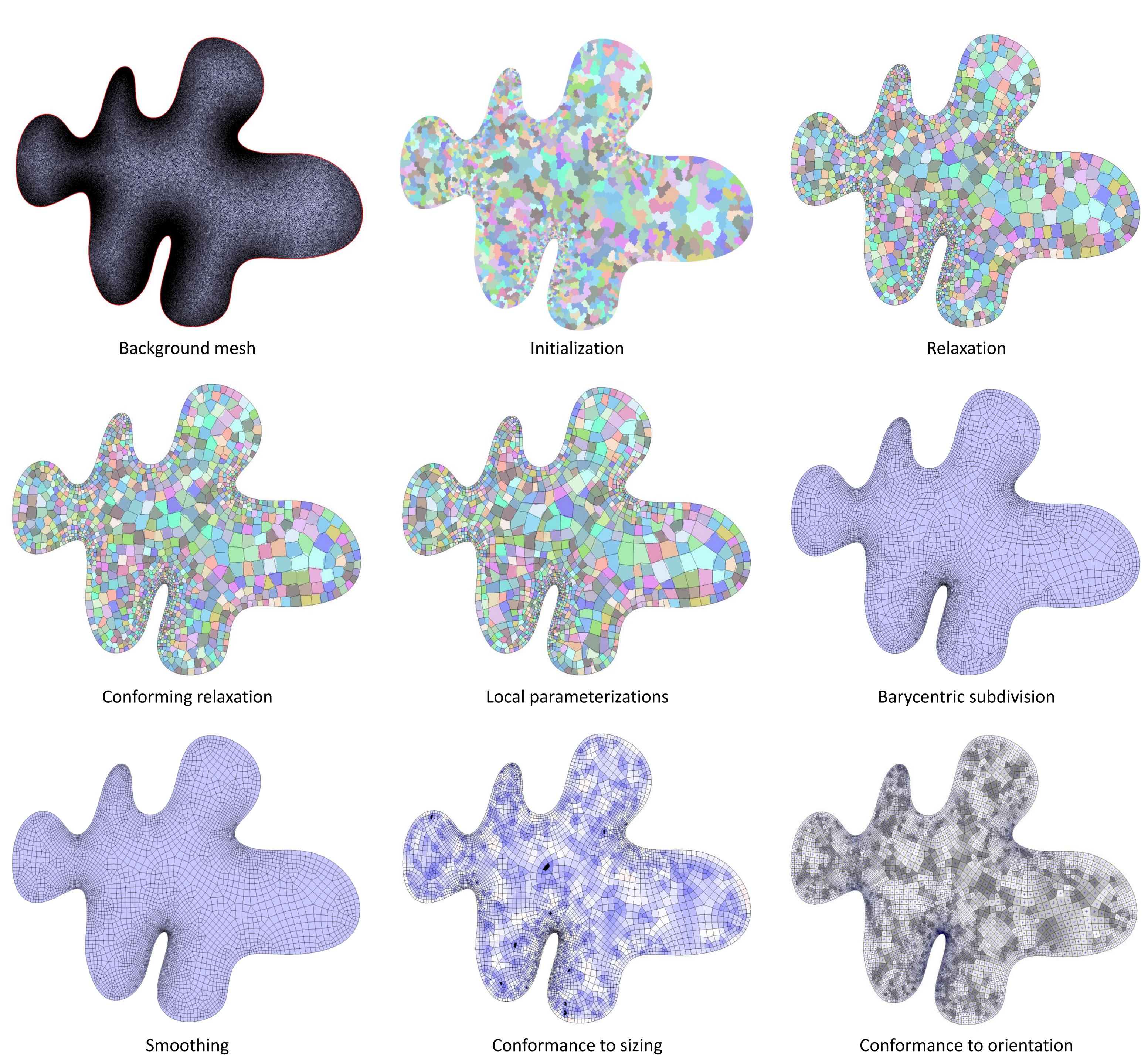
Domain — Sizing field — Cross field

ALGORITHM

STEPS	Size	Shape	Orientation	Degree	Regularity
1 Initialization	●	◐			
2 Relaxation	○	●	●		
3 Conforming relaxation	○	○	○	◐	◐
4 Local parameterizations	○	○	○	◐	●
5 Barycentric subdivision	○		○	●	○
6 Smoothing	○	●	○	○	○

● met ○ preserved ◐ partially met

OVERVIEW



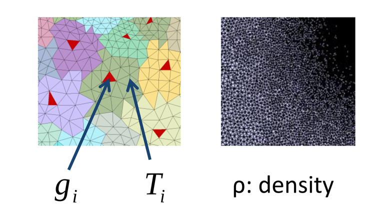
2 $\mathcal{L}_\infty$ relaxation

Minimize (continuous/discrete)

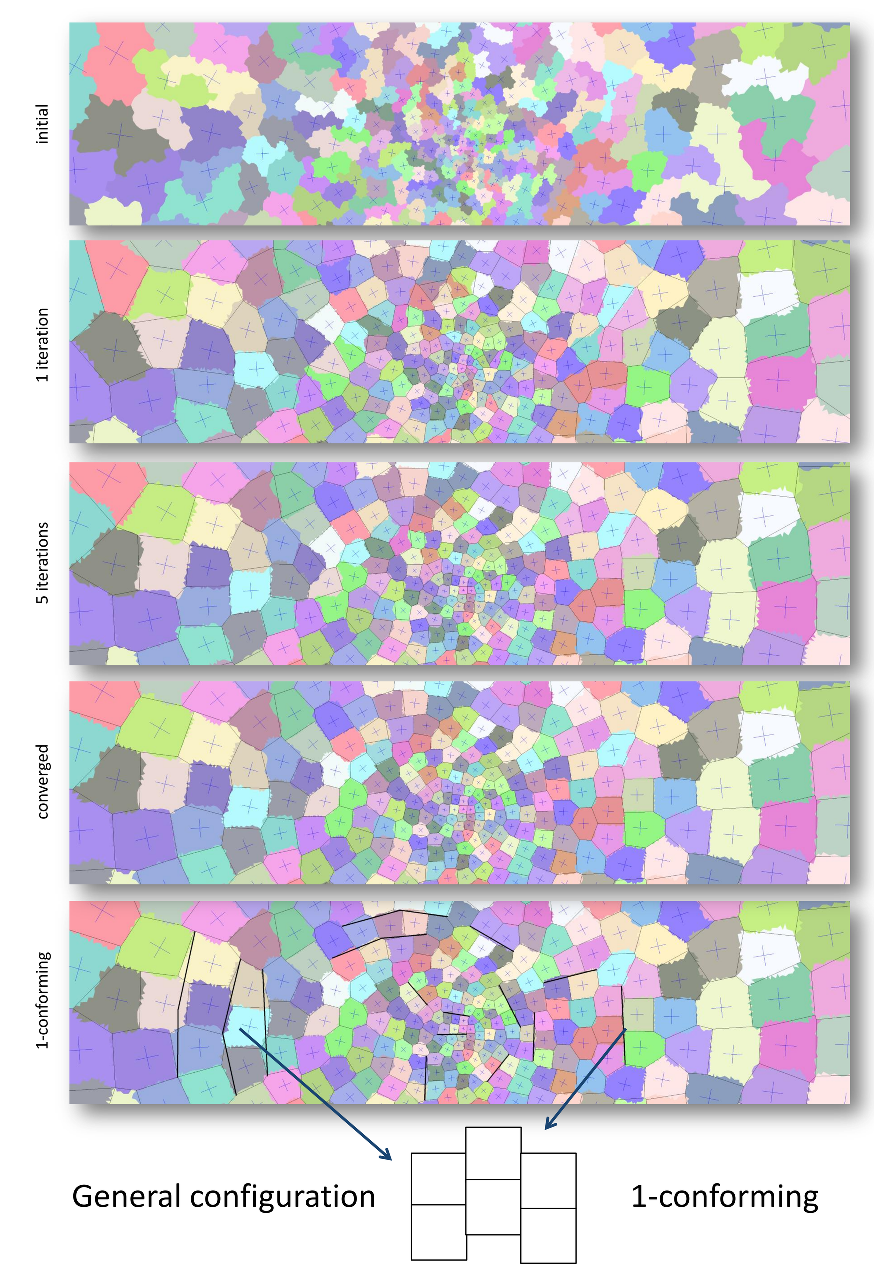
$$E(g_1,\dots,g_n,V_1,\dots,V_n)=\sum_{i=0}^{n}\int_{V_i}\rho(x)d_\infty(x,g_i)dx$$

$$\mathcal{H}\left(\{g_i\}_{i=1}^n,\{T_i\}_{i=1}^n\right)=\sum_{i=1}^{n}\left(\sum_{t_j\in T_i}\rho(c(t_j))\text{area}(t_j)d_\infty^{\mathcal{R}}(c(t_j),c(\mu(T_i)))\right)$$

$\{g_i\}_{i=1}^n$: triangle generators
$\{T_i\}_{i=1}^n$: tiling
$c(t)$: triangle t centroid
μ : triangle containing tile centroid

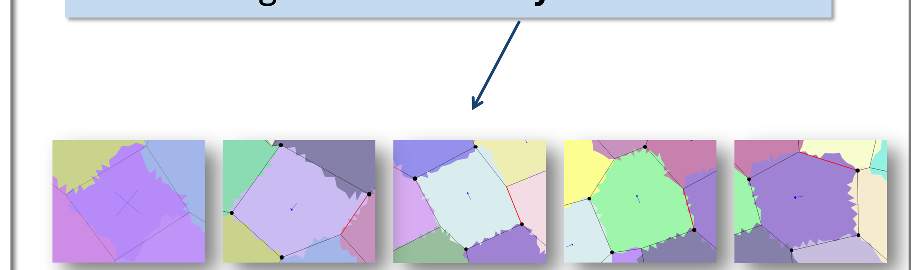


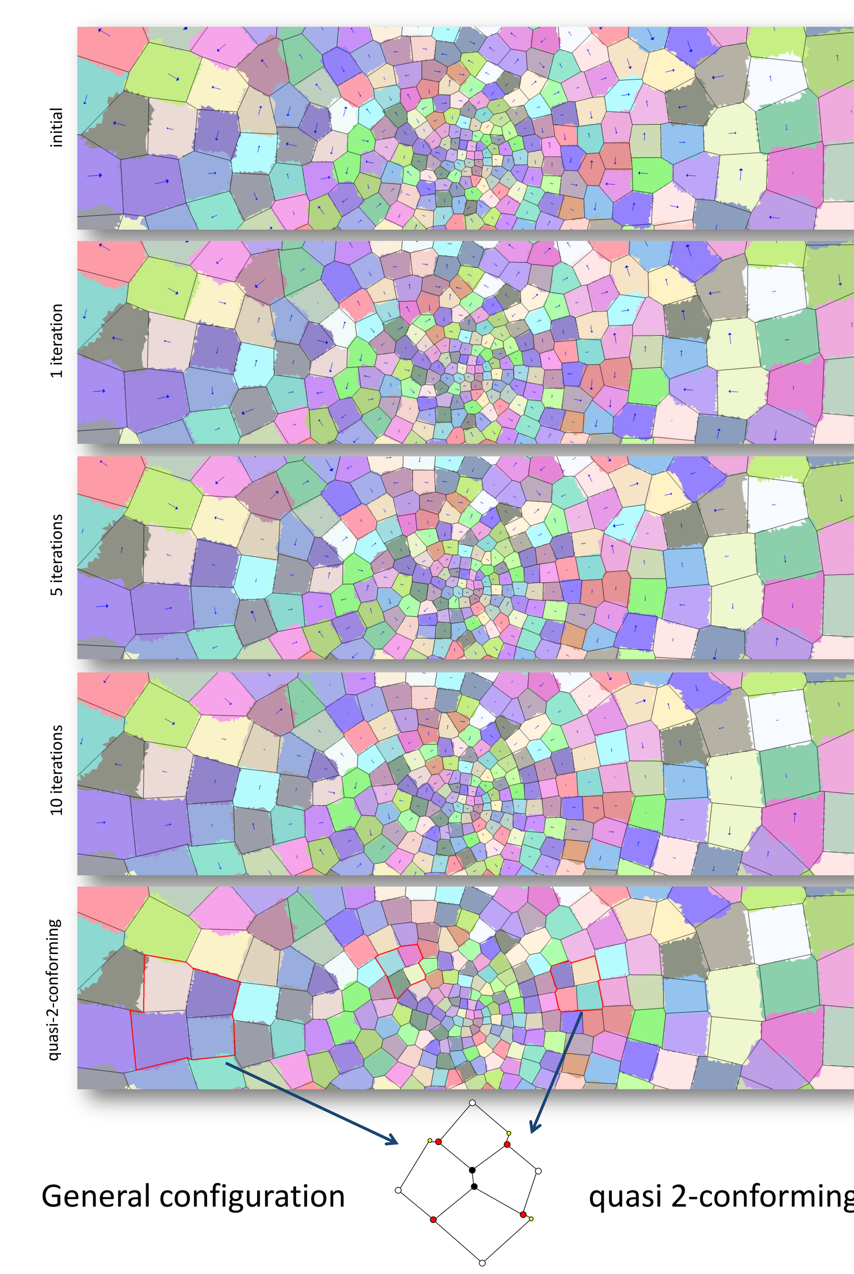
While no convergence **do**
Discrete partitioning
Relocate generators to centroids



3 $\mathcal{L}_\infty$ conforming relaxation

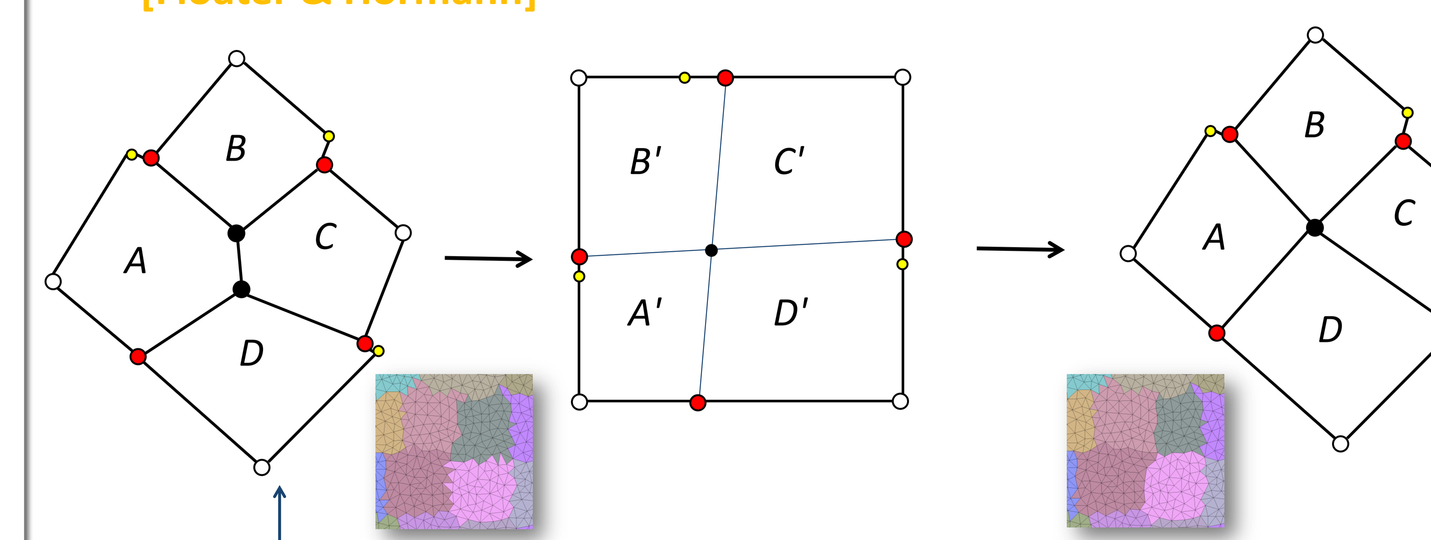
While no convergence **do**
Discrete partitioning
Relocate generators to ***shifted*** centroids





4 Local conformal parameterizations

[Floater & Hormann]



For all configurations :
Parameterize four tiles over unit square
Re-index tile triangles through Dijkstra shortest path in parameter space

Parameterization : solve AN = Ñ, where:
N: column vector of unknown interior vertex parameter coordinates
Ñ: column vector that depends on boundary vertex parameter coordinates
A: sparse matrix with Laplace coefficients