ALICE

Geometry and Light

Bruno Lévy
ALICE Géométrie & Lumière
CENTRE INRIA Nancy Grand-Est
OVERVIEW

Part. 1. Research axes, Evolutions, Applications
Part. 2. From Graphics to Fabrication
Part. 3. From Geometry Processing to Applied Math.
Part. 4. Future Works
1
Research Axes
Evolution
Application Domains
Part. 1. Research Axes 2006 - 2012

Computer Graphics / Automatic Content Creation

Geometry Processing

\[
F_{L_p}^T = \int_T \| M_T(y - x_0) \|_p^p \, dy \\
= \frac{|T|}{(n+p)} \sum_{\alpha + \beta + \gamma = p} U_1^\alpha * U_2^\beta * U_3^\gamma \\
\]

where:

\[
U_1 = M_T(C_i - x_0) \\
V_1 * V_2 = [x_1, x_2, y_1, y_2, z_1, z_2]^T \\
V^\alpha = V * V * \ldots * V (\alpha \text{ times}) \\
\overline{V} = x + y + z
\]
Part. 1. Research Axes 2006 - 2012

Computer Graphics / Automatic Content Creation

Geometry Processing

Texture Mapping

\[
U_1 = M_C(C_i - x_0) \\
V_1 \times V_2 = [x_1, x_2, y_1, y_2, z_1, z_2]^T \\
V^\alpha = V \times V \times \ldots \times V (\alpha \text{ times}) \\
\bar{V} = x + y + z
\]
Part. 1. Research Axes 2013 - 2016

Into reality

Poppy – Inria project Flowers
Part. 1. Research Axes 2013 - 2016

Print your own “Poppy Robot” at home … not that easy !!!
Part. 1. Research Axes 2013 - 2016

Print a “scaffold” with the object
Part. 1. Research Axes 2013 - 2016

Into reality

Poppy – Inria project Flowers

Make it easy for everybody ("it" = object modeling, 3d printing …)
Part. 1. Research Axes 2013 - 2016

*Into reality*

*Poppy – Inria project Flowers*

*Into abstraction*
Part. 1. Research Axes 2013 - 2016

Into reality

Into abstraction

Poppy – Inria project Flowers

Simulating Reality – Realistic Numerical Models

3D mesh of a microstructure generated by IceSL
Part. 1. Application Domains

Inserting Shape Optimization into the loop
Part. 1. Application Domains
Inserting Shape Optimization into the loop
Part. 1. Application Domains
Inserting Shape Optimization into the loop
Part. 1. Application Domains
Inserting Shape Optimization into the loop
Part. 1. Application Domains
Inserting Shape Optimization into the loop
Part. 1. Application Domains
Reparative Surgery
Part. 1. Application Domains
Reparative Surgery
Part. 1. Application Domains
Reparative Surgery
Part. 1. Application Domains
Reparative Surgery
Part. 1. Application Domains
Computational Physics

Bose-Einstein Condensate

ANR BECASIM – cooperation with physicists and mathematicians
2

From Graphics to Fabrication
Part. 2 From Graphics to Fabrication

Coherent Parallel Hashing
Garcia, Lefebvre, Hornus, Lasram
SIGGRAPH Asia 2011

fish: $20.5M/8192^2$
A runtime cache for interactive procedural modeling
Reiner, Lefebvre, Diener, Garcia, Jobard, Dachsbacher
SMI 2012
Part. 2 From Graphics to Fabrication

Visualization of Bose-Einstein condensates with IceSL

ANR BECASIM – cooperation with physicists and mathematicians
Part. 2 From Graphics to Fabrication
Part. 2 From Graphics to Fabrication
Part. 2 From Graphics to Fabrication

Make it stand, Prevost, Whiting, Lefebvre, Sorkine, SIGGRAPH 2012
Part. 2 From Graphics to Fabrication

Make it stand, Prevost, Whiting, Lefebvre, Sorkine, SIGGRAPH 2012

Clean Color, Hergel, Lefebvre, Eurographics 2014
Part. 2 From Graphics to Fabrication

Make it stand, Prevost, Whiting, Lefebvre, Sorkine, SIGGRAPH 2012

Clean Color, Hergel, Lefebvre, Eurographics 2014

Bridge the gap, Dumas, Hergel, Lefebvre, SIGGRAPH 2014
Part. 2 From Graphics to Fabrication

Reparative Surgery – toy example
3

From Geometry Processing to Applied Mathematics
Part. 3 From Geometry Processing to Applied Math.

Exotic representation (Dexels)

Back to the standard modeling pipeline…
Finite Element Modeling?
How?
Part. 3 From Geometry Processing to Applied Math.

Optimize a Voronoi diagram from the point of view of sampling regularity (quantization noise power)

Minimize

\[ F = \sum_{i} \int_{\text{Vor}(i)} \left\| x_i - x \right\|^2 dx \]

Theorem: F is of class \( C^2 \) [Liu, Wang, L, Yan, Lu, ACM TOG 2008]
Part. 3 From Geometry Processing to Applied Math.

Theorem: F is of class $C^2$ [Liu, Wang, L, Yan, Lu, ACM TOG 2008]
Part. 3 From Geometry Processing to Applied Math.

Theorem: $F$ is of class $C^2$ [Liu, Wang, L, Yan, Lu, ACM TOG 2008]
Part. 3 From Geometry Processing to Applied Math.

Theorem: $F$ is of class $C^2$ [Liu, Wang, L, Yan, Lu, ACM TOG 2008]
Part. 3 From Geometry Processing to Applied Math.

Theorem: F is of class $C^2$ [Liu, Wang, L, Yan, Lu, ACM TOG 2008]
Anisotropic mesh:
* shape can vary
* size can vary
**Part. 3. Anisotropy**

*The input:* anisotropy field
Specifies shape and orientation

*Anisotropy:* An “alteration” of of distances and angles.

This is a circle!
\[ \{ q \mid \text{dist}(p,q) = 1 \} \]

anisotropic distance
Part. 3. Anisotropy

The dot product: a geometric tool

Anisotropic distance between \( p \) and \( q \) w.r.t. \( G \)

\[
d_G(p, q) = (\text{anisotropic}) \text{ length of shortest curve that connects } p \text{ with } q
\]

\[
l_G(C) = \int_{t=0}^{1} \sqrt{v(t)^t G(t) v(t)} \, dt
\]
Part. 3. Anisotropy

**The input:** anisotropy field

\[ G(x,y) = \begin{bmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y) \end{bmatrix} \]

\[ \{ q \mid d_G(p,q) = 1 \} \]
Part. 3. Anisotropy

The result: triangles are “deformed” by the anisotropy.
Part. 3. Anisotropy

The result: triangles are “deformed” by the anisotropy.

Q: How to compute an Anisotropic Centroidal Voronoi Tessellation?
Part. 3  Journey in the 6\textsuperscript{th} dimension

The key idea

\textbf{This example:}

Anisotropic mesh in 2d \hspace{1cm} \Longleftrightarrow \hspace{1cm} Isotropic mesh in 3d
Part. 3 Journey in the 6\textsuperscript{th} dimension

The key idea

This example:

Anisotropic mesh in 2d $\iff$ Isotropic mesh in 3d

Replace \textit{anisotropy} with \textit{additional dimensions}
Part. 3  Journey in the 6th dimension

The key idea

Replace \textit{anisotropy} with \textit{additional dimensions}

\textit{Note: more dimensions may be needed}
Part. 3  Journey in the 6th dimension

The key idea

Replace anisotropy with additional dimensions

Note: more dimensions may be needed

How many?

John Nash’s isometric embedding theorem:

Maximum: depending on desired smoothness

$C^1 : 2n$  \[\text{[Nash-Kuiper]}\]

$C^k : \text{bounded by } n(3n+11)/2$  \[\text{[Nash, Nash-Moser]}\]
Part. 3 Journey in the 6th dimension
A 6d embedding for curvature-adapted meshing

The Gauss map:

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} \rightarrow
\begin{bmatrix}
  N_x \\
  N_y \\
  N_z
\end{bmatrix}
\]
Part. 3 Journey in the 6th dimension

A 6d embedding for curvature-adapted meshing

Vorpaline meshing software
ERC “Proof of Concept”
Part. 3  Journey in the 6th dimension

Anisotropy through high-dim. embedding

3D anisotropic Voronoi diagram and anisotropic Vector Quantization

*Anisotropy represented by a background mesh embedded in 6D*

$$G_t = J_t^T J_t$$
Part. 3  Journey in the 6th dimension

New predicates

\[
\begin{align*}
\text{side}_1(p_1, p_2, q) &= \operatorname{Sign}(d^2(p_2, q) - d^2(p_1, q)) \\
\text{side}_2(p_1, p_2, p_3, q_1, q_2) &= \text{side}_1(p_1, p_2, q) \quad \text{where} \quad q = \Pi(p_1, p_3) \cap [q_1 q_2] \\
\text{side}_3(p_1, p_2, p_3, p_4, q_1, q_2, q_3) &= \text{side}_1(p_1, p_2, q) \quad \text{where} \quad q = \Pi(p_1, p_3) \cap \Pi(p_1, p_4) \cap \Delta(q_1 q_2 q_3) \\
\text{side}_4(p_1, \ldots, p_5, q_1, \ldots, q_4) &= \text{side}_1(p_1, p_2, q) \quad , \quad q = \Pi(p_1, p_3) \cap \Pi(p_1, p_4) \cap \Pi(p_1 p_5) \cap \text{tet}(q_1 q_2 q_3 q_4)
\end{align*}
\]
Part. 3  Journey in the 6th dimension

New predicates

Sign side2(
    point p0, point p1, point p2,
    point q0, point q1
) {

    scalar l1 = sq_dist(p1,p0) ;
    scalar l2 = sq_dist(p2,p0) ;

    scalar a10 = 2*dot_at(p1,q0,p0);
    scalar a11 = 2*dot_at(p1,q1,p0);
    scalar a20 = 2*dot_at(p2,q0,p0);
    scalar a21 = 2*dot_at(p2,q1,p0);

    scalar Delta = a11 - a10 ;
    scalar DeltaLambda0 = a11 - l1 ;
    scalar DeltaLambda1 = l1 - a10 ;
    scalar r =
        Delta*l2-a20*DeltaLambda0-a21*DeltaLambda1 ;

    Sign Delta_sign = sign(Delta) ;
    Sign r_sign      = sign(r) ;

    generic_predicate_result(Delta_sign*r_sign) ;

    begin_sos3(p0,p1,p2)
        sos(p0, Sign(Delta_sign*sign(Delta-a21+a20)))
        sos(p1, Sign(Delta_sign*sign(a21-a20)))
        sos(p2, NEGATIVE)
    end_sos
}
Part. 3  Journey in the 6th dimension

New predicates

Sign side2()
    point p0, point p1, point p2,
    point q0, point q1

) {

    scalar l1 = sq_dist(p1,p0);
    scalar l2 = sq_dist(p2,p0);

    scalar a10 = 2*dot_at(p1,q0,p0);
    scalar a11 = 2*dot_at(p1,q1,p0);
    scalar a20 = 2*dot_at(p2,q0,p0);
    scalar a21 = 2*dot_at(p2,q1,p0);

    scalar Delta = a11 - a10;
    scalar DeltaLambda0 = a11 - l1;
    scalar DeltaLambda1 = l1 - a10;
    scalar r =
        Delta*l2-a20*DeltaLambda0-a21*DeltaLambda1;

    Sign Delta_sign = sign(Delta);
    Sign r_sign = sign(r);

    generic_predicate_result(Delta_sign*r_sign);

    begin_sos3(p0,p1,p2)
        sos(p0, Sign(Delta_sign*sign(Delta-a21+a20)))
        sos(p1, Sign(Delta_sign*sign(a21-a20)))
        sos(p2, NEGATIVE)
    end_sos

}

Predicate Construction Toolkit [PCK] – make it easy for everybody
Part. 3 Optimal Transport
Gaspard Monge - 1784

Lorsqu'on doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de Déblai au volume des terres que l'on doit transporter, & le nom de
Part. 3 Optimal Transport – some references


Variational Principles for Minkowski Type Problems, Discrete Optimal Transport, and Discrete Monge-Ampere Equations
*Xianfeng Gu, Feng Luo, Jian Sun, S.-T. Yau*, ArXiv 2013

Minkowski-type theorems and least-squares clustering
*AHA! (Aurenhammer, Hoffmann, and Aronov)*, SIAM J. on math. ana. 1998

Topics on Optimal Transportation, 2003
Optimal Transport Old and New, 2008
*Cédric Villani*

Yann Brénier, Jean-David Benamou
Part. 3 Optimal Transport – Monge’s problem

Monge’s problem:

Find a transport map $T$ that minimizes $C(T) = \int_{\Omega} \| x - T(x) \|^2 \, d\mu(x)$
Part. 3 Optimal Transport – semi-discrete
The pre-images of the Diracs define a partition of $\Omega$. 
The pre-images of the Diracs define a partition of $\Omega$
This partition is a **power diagram** (more on this below)
Theorem [Aurenhammer, Hoffmann, Aronov 98], [Brenier91]:

given a measure μ with density, a set of points (S), a set of positive coefficients λ such that \( \sum \lambda_i = \int d\mu(x) \), it is possible to find the weights w such that the map \( T^w_S \) is an optimal transport map between \( \mu \) and \( v = \sum \lambda_i \delta(s_i) \).

Given the points (S), one can find the weights (w) such that \( \int_{\text{Pow}(s_i)} d\mu(x) = \lambda_i \)
Part. 3 Optimal Transport – the algorithm

The [AHA] paper summary:
- The optimal weights minimize a convex function
- The gradient of this convex function is easy to compute
Note: the weight $w(s)$ correspond to the Kantorovich potential $\psi(x)$ (solves a “discrete Monge-Ampere” equation)

The algorithm:

Summary:
The algorithm computes the weights $w_i$ such that the power cells associated with the Diracs correspond to the preimages of the Diracs.
The [AHA] paper summary:

- The optimal weights minimize a convex function
- The gradient of this convex function is easy to compute

Note: the weight $w(s)$ correspond to the Kantorovich potential $\psi(x)$
  (solves a “discrete Monge-Ampere” equation)

The algorithm:

Summary:

The algorithm computes the weights $w_i$ such that the power cells associated with the Diracs correspond to the preimages of the Diracs.
Part. 3 Optimal Transport – the algorithm

The [AHA] paper summary:

- The optimal weights minimize a convex function
- The gradient of this convex function is easy to compute

Note: the weight \( w(s) \) correspond to the Kantorovich potential \( \psi(x) \)
  (solves a “discrete Monge-Ampere” equation)

The algorithm:

**Summary:**
The algorithm computes the weights \( w_i \) such that the power cells associated with the Diracs correspond to the preimages of the Diracs.
The AHA paper summary:
• The optimal weights minimize a convex function
• The gradient of this convex function is easy to compute
Note: the weight $w(s)$ correspond to the Kantorovich potential $\psi(x)$
(solves a “discrete Monge-Ampere” equation)

The algorithm:

Summary:
The algorithm computes the weights $w_i$ such that the power cells associated with the Diracs correspond to the preimages of the Diracs.
The [AHA] paper summary:
- The optimal weights minimize a convex function
- The gradient of this convex function is easy to compute
Note: the weight \( w(s) \) correspond to the Kantorovich potential \( \psi(x) \)
  (solves a “discrete Monge-Ampere” equation)

The algorithm:

Summary:
The algorithm computes the weights \( w_i \) such that the power cells associated with the Diracs correspond to the preimages of the Diracs.
Wait a minute:

This means that one can move (and possibly deform) a power diagram simply by changing the weights?
Wait a minute:

This means that one can move (and possibly deform) a power diagram simply by changing the weights.

Reminder: Power diagram in 2d = intersection between Voronoi diagram in 3d and $IR^2$

$$h_i = \sqrt{w_{\text{max}} - w_i}$$

Height of point $i$  

Weight of point $i$
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport
Part. 3 Power Diagrams & Transport

Translating a Voronoi diagram:
1st Try: linear lifting
(Fail: scale by $1/\cos(x)$)
Part. 3 Power Diagrams & Transport

2nd Try: Curved lifting
"Converging beams" can compensate the $1/\cos(x)$ expansion by "re-concentrating" the points.
Part. 3  Power Diagrams & Transport

\[ d^2(p_i, q) - w_i \leq d^2(p_j, q) - w_j \quad \forall j \]

\[ d^2(p_i, q - T) \leq d^2(p_j, q - T) \quad \forall j \]

\[ (p_i - q + T)^2 \leq (p_j - q + T)^2 \quad \forall j \]

\[ d^2(p_i, q) + 2T \cdot (p_i - q) + T^2 \leq d^2(p_j, q) + 2T \cdot (p_j - q) + T^2 \quad \forall j \]

\[ d^2(p_i, q) + 2T \cdot p_i \leq d^2(p_j, q) + 2T \cdot p_j \]

\[ w_i \leq -2T \cdot p_i \]

\[ h_i = \sqrt{2T \cdot p_i - \min(T \cdot p_i)} \]

Translation d'un diagramme de Veconi: 
Secteurnel - Relativement en racine carrée.
Part. 3 Power Diagrams and Transport

- Voronoi diagram of B samples.
- How to "back displace" it onto A?

- Lifting on two "square root wings" translates both halves of B points into the two blobs of A.

- Solving for the OTM (T(x,y) vector field) is equivalent to solving for the "square root wings" (h(x,y) scalar function).
Numerical Experiment: *Splitting a disk*
Part. 3 Optimal Transport – 2D example
Numerical Experiment: Splitting a disk
Part. 3 Optimal Transport – 2D example
Numerical Experiment: *Splitting a disk*
Part. 3 Optimal Transport – 2D example

Numerical Experiment: Splitting a disk
Part. 3 Optimal Transport – 2D example
Numerical Experiment: *Splitting a disk*
Part. 3 Power Diagrams & Transport

C'est quoi l'équation en continu?

\[ \frac{1}{\sqrt{1 + \left( \frac{\partial h^2}{\partial x} + \frac{\partial h^2}{\partial y} \right)^2}} \mu(y) = \delta(x) \]

\[ T^{-1}(x) = \{ y \mid h^2(y) + d^2(x, y) \text{ minimum} \} \]

Se croise que:

\[ \langle n, \hat{z} \rangle \mu(y) = \delta(x) \]
Part. 3 Towards FWD (Fast Wasserstein Distance)
Part. 3 Towards FWD (Fast Wasserstein Distance)
Part. 3 Towards FWD (Fast Wasserstein Distance)
Part. 3 Towards FWD (Fast Wasserstein Distance)
Part. 3 Towards FWD (Fast Wasserstein Distance)
Part. 3 Relation with Vector Quantization

Observation 8. The quantization noise power $\hat{Q}(\hat{Y})$ computed in $\mathbb{R}^{d+1}$ corresponds to the term $f_{T_w}(W)$ of the function maximized by the weight vector that defines a semi-discrete optimal transport map plus the constant $w_M \mu(\Omega)$.

Proof.

\[
\hat{Q}(\hat{Y}) = \sum_i \int_{\text{Vor}(\hat{y}_i) \cap \mathbb{R}^d} \| \hat{x} - \hat{y}_i \|^2 d\mu
\]

\[
= \sum_i \int_{\text{Pow}_w(y_i)} \| x - y_i \|^2 - w_i + w_M d\mu
\]

\[
= f_{T_w}(W) + w_M \mu(\Omega)
\]
Part. 3 Self Organizing Optimal Transport Maps

Voxels

Splines
Future Works
Guiding principles:
(1) Make it easy for everybody!
(2) Integrate more and more fabrication constraints in modeling
Part. 4 Future Works in Fabrication

[ACM SIGGRAPH 2016]
Part. 4 Future Works in Applied Mathematics

Discrete Elements – from Equations to Programs
Short term: Hex-dominant meshing

LpCVT
[Lévy and Liu 2010]

PGP3d
[Ray, Sokolov, Untereiner, Lévy 2016]
Part. 4 Future Works in Applied Mathematics

Discrete Elements – from Equations to Programs

Short term: *Hex-dominant meshing*

Finite Elements function basis for non-conforming meshes (submitted)
Part. 4 Future Works in Applied Mathematics

Optimization of frame fields for hex-dominant meshing

*How to interpolate frame fields?*
Part. 4 Future Works in Applied Mathematics

Optimization of frame fields for hex-dominant meshing

*How to interpolate frame fields?*

A natural idea:
Frame field = 8 Dirac masses on the sphere
Optimal Transport for interpolation, barycenters …
Part. 4 Future Works in Applied Mathematics

Optimization of frame fields for hex-dominant meshing
*How to interpolate frame fields?*

**A natural idea:**
Frame field = 8 Dirac masses on the sphere
Optimal Transport for interpolation, barycenters …

**Not smooth enough**
Use “smoothed” version, with functions that has the same symmetries.
Symmetries of platonic solids reproduced with sums of Spherical Harmonics.
Part. 4 Future Works in Applied Mathematics

Optimization of frame fields for hex-dominant meshing

*How to interpolate frame fields?*

First results are encouraging
(scales-up well)

[ACM Transactions on Graphics 2016]
Longer term: from the principle of least action to optimal transport

JKO scheme (Jordan, Kinderlehrer, Otto)
Benamou, Carlier, Merigot, Oudet arXiv 1408.4536

EXPLORAGRAM project (INRIA exploratory project)
MAGA project (ANR project – submitted)
Part. 4 Future Works in Applied Mathematics

**Geometric Predicates:** How can we easily translate geometric predicates into computer programs? How can we certify their validity? Can we invent programming tools?

Source PCK file (using my current version)

```plaintext
Sign side2(
    point p0, point p1, point p2,
    point q0, point q1
) {
    scalar l1 = sq_dist(p1,p0);
    scalar l2 = sq_dist(p2,p0);
    scalar a10 = 2*dot_at(p1,q0,p0);
    scalar a11 = 2*dot_at(p1,q1,p0);
    scalar a20 = 2*dot_at(p2,q0,p0);
    scalar a21 = 2*dot_at(p2,q1,p0);
    scalar Delta = a11 - a10;
    scalar DeltaLambda0 = a11 - l1;
    scalar DeltaLambda1 = l1 - a10;
    scalar r = Delta*l2-a20*DeltaLambda0-a21*DeltaLambda1;
    Sign Delta_sign = sign(Delta);
    Sign r_sign = sign(r);
    generic_predicate_result(Delta_sign*r_sign);
    begin sos3(p0,p1,p2)
        sos(p0, Sign(Delta_sign*sign(Delta-a21-a20)))
        sos(p1, Sign(Delta_sign*sign(a21-a20)))
        sos(p2, NEGATIVE)
    end_sos
}
```

Source PCK file (using the tools that I plan to develop)

```plaintext
Sign side2(point p0, point p1, point p2, point q0, point q1) {
    scalar w0 = 0.0;
    scalar w1 = 0.0;
    scalar w2 = 0.0;
    sos_perturbation(w1, pi, pow(epsilon,1));
    Plane P1 = weighted_bisector(p0,w0,p1,w1);
    Point q = intersection(P1, segment(q0,q1));
    return Sign(sq_dist(q,p0) + w0 - sq_dist(q,p1) - w1);
}
```

`sqrt()`, `root_of()` … Voronoi diagram of Segments in 3d doable?
Highlights

ERC StG GOODSHAPE – Optimal Sampling

ERC PoC VORPALINE – Remeshing Software

ERC StG SHAPEFORGE – 3D printing made easy

ERC PoC ICEXL – 3D printing – scaling up

IceSL software – Fast CSG modeler, language, driver for 3d printers …

First algorithm that computes aniso. Voro. diagram and semi-discrete Optimal Transport in 3d (+ Predicate Cons. Kit)

Integration of research results in ALICE!

SHAPEFORGE - Dexels

GOODSHAPE/VORPALINE
Thank you!