A Numerical Algorithm for L-2 Optimal Transport in 3D

Bruno Lévy

Mathematical Modelling and Numerical Analysis Will be set by the publisher
Modélisation Mathématique et Analyse Numérique

A NUMERICAL ALGORITHM FOR L_2 SEMI-DISCRETE OPTIMAL TRANSPORT IN 3D

Bruno Lévy[1]

Abstract. This paper introduces a numerical algorithm to compute the L_2 optimal transport map between two measures μ and ν, where μ derives from a density ρ defined as a piecewise linear function (supported by a tetrahedral mesh), and where ν is a sum of Dirac masses.

I first give an elementary presentation of some known results on optimal transport and then observe a relation with another problem (optimal sampling). This relation gives simple arguments to study the objective functions that characterize both problems.

I then propose a practical algorithm to compute the optimal transport map between a piecewise linear density and a sum of Dirac masses in 3D. In this semi-discrete setting, Aurenhammer et.al [*8th Symposium on Computational Geometry conf. proc.*, ACM (1992)] showed that the optimal transport map is determined by the weights of a power diagram. The optimal weights are computed by minimizing a convex objective function with a quasi-Newton method. To evaluate the value and gradient of this objective function, I propose an efficient and robust algorithm, that computes at each iteration the intersection between a power diagram and the tetrahedral mesh that defines the measure μ.

The numerical algorithm is experimented and evaluated on several datasets, with up to hundred thousands tetrahedra and one million Dirac masses.

Résumé. Cet article décrit un algorithme numérique pour calculer l'application de transport optimal L_2 entre deux mesures μ et ν, où μ dérive d'une densité ρ linéaire par morceaux (supportée par un maillage tétraédrique), et où ν est une somme de masses de Dirac.

Je donne tout d'abord une présentation élémentaire de quelques résultats connus sur le transport optimal, et observe ensuite une relation avec un autre problème (l'échantillonage optimal). Cette relation fournit des arguments simples pour étudier les fonctions objectifs caractérisant les deux problèmes.

Je propose ensuite un algorithme pratique pour calculer le transport optimal entre une densité linéaire par morceaux et une somme de masses de Dirac en 3D. Dans ce cas semi-discret, Aurenhammer et.al [*8th Symposium on Computational Geometry conf. proc.*, ACM (1992)] ont montré que l'application de transport optimal est déterminée par les poids d'un diagramme de puissance. Les poids optimaux sont calculés en minimisant une fonction objectif convexe à l'aide d'une méthode quasi-Newton. Pour évaluer cette fonction objectif et son gradient, je propose un algorithme efficace et robuste, qui calcule à chaque itération l'intersection entre un diagramme de puissance et le maillage tétrahédrique qui définit la mesure μ.

L'algorithme numérique est expérimenté et évalué sur plusieurs jeux de données, comportant jusqu'à plusieurs centaines de milliers de tétraèdres et un million de masses de Dirac.

1991 Mathematics Subject Classification. 49M15, 35J96, 65D18.

04/01/2014.

Keywords and phrases: optimal transport, power diagrams, quantization noise power, Lloyd relaxation

[1] Inria Nancy Grand-Est and LORIA, rue du Jardin Botanique, 54500 Vandoeuvre, France

Introduction

Optimal Transportation, initially studied by Monge [30], is a very general problem formulation that can be used as a model for a wide range of applications domains. In particular, it is a natural formulation for several fundamental questions in Computer Graphics [7, 26, 27]

This article proposes a practical algorithm to compute the optimal transport map between two measures μ and ν, where μ derives from a density ρ defined as a piecewise linear function (supported by a tetrahedral mesh), and where ν is a sum of Dirac masses. Possible applications comprise measuring the (approximated) Wasserstein distance between two shapes and deforming a 3D shape onto another one (3D morphing).

I first review some known results about optimal transport in Section 1, its relation with power diagrams [4,27] in Section 1.4 and observe some connections with another problem (optimal sampling [13, 24]). The structure of the objective function minimized by both problems is very similar, this allows reusing known results for both functions. This gives a simple argument to easily compute the gradient of the quantization noise power minimized by optimal sampling, and this gives the second order continuity of the objective function minimized in semi-discrete optimal transport (see Section 1.6).

I then propose a practical algorithm to compute the optimal transport map between a piecewise linear density and a sum of Dirac masses in 3D (Section 2). This means determining the weights of a power diagram, obtained as the unique minimizer of a convex function [4]. Following the approach in [27], to optimize this function, I use a quasi-Newton solver combined with a multilevel algorithm. Adapting the approach to the 3D setting requires an efficient method to compute the intersection between a power diagram and the tetrahedral mesh that defines the density μ.

To compute these intersections, the algorithm presented here simultaneously traverses the tetrahedral mesh and the power diagram (Section 2.1). The required geometric predicates are implemented in both standard floating point precision and arbitrary precision, using arithmetic filtering [28], expansion arithmetics [36] and symbolic perturbation [15]. Both predicates and power diagram construction algorithm are available in PCK (Predicate Construction Kit) part of my publically available "GEOGRAM" programming library[1].

The algorithm was experimented and evaluated on several datasets (Section 3).

1. Optimal Transport: an Elementary Introduction

This section, inspired by [37], [35], [10] and [1], presents an introduction to optimal transport. It stays at an elementary level that corresponds to what I have understood and that keeps computer implementation in mind.

1.1. The initial formulation by Monge

The problem of Optimal Transport was first introduced and studied by Monge [30]. With modern notations, it can be stated as follows :

$$(M) \qquad \begin{array}{l} \text{given } \Omega \text{ a Borel set and two measures } \mu \text{ and } \nu \text{ on } \Omega \text{ such that } \mu(\Omega) = \nu(\Omega), \\ \text{find } T : \Omega \to \Omega \text{ such that } \left\{ \begin{array}{ll} (C1) & \nu = T\sharp\mu \\ (C2) & \int_\Omega c(x, T(x)) d\mu \text{ is minimal} \end{array} \right. \end{array} \tag{1}$$

where c denotes a convex distance function. In the first constraint $(C1)$, $T\sharp\mu$ denotes the pushforward of μ by T, defined by $T\sharp\mu(X) = \mu(T^{-1}(X))$ for any Borel (i.e. measurable) subset X of Ω. In other words, the constraint $(C1)$ means that T should preserve the mass of any measurable subset of Ω. The functional in $(C2)$ has a non-symmetric structure, that makes it difficult to study the existence for problem (M).

[1] http://gforge.inria.fr/projects/geogram/

The non-symmetry comes from the constraint that T should be a map. It makes it possible to merge mass but not to split mass. This problem occurs whenever the source measure μ has mass concentrated on a $d-1$ dimensional manifold [25].

1.2. **The relaxation of Kantorovich for Monge's problem**

To overcome this difficulty, Kantorovich proposed a relaxation of problem (M) where mass can be both split and merged. The idea consists of manipulating measures on $\Omega \times \Omega$ as follows :

$$(K) \qquad \min \left\{ \int_{\Omega\times\Omega} c(x,y) d\gamma \mid \gamma \in \Pi(\mu,\nu) \right\} \tag{2}$$
$$\text{where } \Pi(\mu,\nu) = \{\gamma \in \mathcal{P}(\Omega\times\Omega) \mid (P_1)\sharp\gamma = \mu \ ; \ (P_2)\sharp\gamma = \nu\}$$

where (P_1) and (P_2) denote the two projections $(x,y) \in \Omega\times\Omega \mapsto x$ and $(x,y) \in \Omega\times\Omega \mapsto y$ respectively.

The pushforwards of the two projections $(P_1)\sharp\gamma$ and $(P_2)\sharp\gamma$ are called the marginals of γ. The probability measures γ in $\Pi(\mu,\nu)$, i.e. that have μ and ν as marginals, are called *transport plans*. Among the transport plans, those that are in the form $(Id\times T)\sharp\mu$ correspond to a transport map T :

Observation 1. *If $(Id\times T)\sharp\mu \in \pi(\mu,\nu)$, then T pushes μ to ν.*

Proof. $(Id\times T)\sharp\mu$ belongs to $\pi(\mu,\nu)$, therefore $(P_2)\sharp(Id\times T)\sharp\mu = \nu$, or $((P_2)\circ(Id\times T))\,\sharp\mu = \nu$, thus $T\sharp\mu = \nu$ □

With this observation, for transport plans of the form $\gamma = (Id\times T)\sharp\mu$, (K) becomes

$$\min\left\{ \int_{\Omega\times\Omega} c(x,y) d\left((Id\times T)\sharp\mu\right) \right\} \quad = \quad \min\left\{ \int_{\Omega} c(x,T(x)) d\mu \right\}$$

At this point, a standard approach to tackle the existence problem is to find some regularity in both the functional and space of admissible transport plans, i.e. proving that the functional is smooth enough and finding a compact set of admissible transport plans. Since the set of admissible transport plans contains at least the product measure $\mu\otimes\nu$, it is non-empty, and existence can be proved using a topological argument that exploits the smoothness of the functional and the compactness of the set. Once the existence of a transport plan is proved, an interesting question is whether there exists a transport map that corresponds to this transport plan. Unfortunately, problem (K) does not directly exhibit the properties required by this path of reasoning. However, one can observe that (K) is a linearly constrained optimization problem. This calls for studying the dual formulation, as done by Kantorovich. This dual formulation has a nice structure, that allows answering the questions above (existence of a transport plan, and whether there is a transport map that corresponds to this transport plan when it exists).

1.3. **The dual formulation of Kantorovich**

The dual formulation can be stated as follows[2] :

$$(D) \qquad \max\left\{ \int_\Omega \phi d\mu + \int_\Omega \psi d\nu \mid \begin{array}{ll} (C1) & \phi \in L^1(\mu); \psi \in L^1(\nu); \\ (C2) & \phi(x)+\psi(y) \le c(x,y) \forall (x,y) \in \Omega\times\Omega \end{array} \right\} \tag{3}$$

Following the classical image that gives some intuition about this formula, imagine now that you are hiring a transport company to do the job for you. The company has a special way of calculating the price: the function $\phi(x)$ corresponds to what they charge you for loading at x, and $\psi(y)$ what they charge for unloading at y. The

[2]Showing the equivalence with problem (K) requires some care, the reader is referred to [37] chapter 5. Note that [37] uses a slightly different definition (with $\phi-\psi$ instead of $\phi+\psi$), that makes the detailed argument simpler but that breaks symmetry between ϕ and ψ. Since I stay at an elementary level, I prefer to keep the symmetry.

company tries to maximize profit (therefore is looking for a max instead of a min), but they cannot charge you more than what it will cost you if you do the job yourself $(C2)$.

The existence for (D) is difficult to study, since the class of admissible functions that satisfy $(C1)$ and $(C2)$ is non-compact. However, more structure in the problem can be revealed by referring to the notion of *c-transform*, that exhibits a class of admissible functions with regularity :

Definition 1. *Given a function $\mathcal{X} : \Omega \to \bar{\mathbb{R}}$, the c-transform $\mathcal{X}^c$ is defined by :*

$$\mathcal{X}^c := \inf_{x\in\Omega} c(x,y) - \mathcal{X}(x)$$

- *If for a function ϕ there exists a function $\mathcal{X}$ such that $\phi = \mathcal{X}^c$, then ϕ is said to be* c-concave*;*
- *$\mathbf{\Psi}_c(\Omega)$ denotes the set of c-concave functions on Ω.*

It is now possible to make two observations, that allow us to restrict ourselves to the class of c-concave functions for the possible choices for ϕ and ψ :

Observation 2. *If (ϕ,ψ) is admissible for (D), then (ϕ,ϕ^c) is also admissible.*

Proof.

$$\begin{cases} \forall (x,y)\in\Omega\times\Omega, \phi(x)+\psi(y)\le c(x,y) \\ \phi^c(y) = \inf\limits_{x\in\Omega} c(x,y)-\phi(x) \end{cases}$$

$$\begin{array}{rcl} \phi(x)+\phi^c(y) & = & \phi(x)+\inf_{x'\in\Omega}\left(c(x',y)-\phi(x')\right) \\ & \le & \phi(x)+c(x,y)-\phi(x) \\ & \le & c(x,y) \end{array}$$

□

Observation 3. *If (ϕ,ψ) is admissible for (D), then a* better candidate *can be found by replacing ψ with ϕ^c :*

Proof.

$$\begin{cases} \phi^c(y) & = & \inf\limits_{y\in\Omega} c(x,y)-\phi(x) \\ \forall x\in\Omega, \psi(y) & \le & c(x,y)-\phi(x) \end{cases} \quad \Rightarrow \quad \psi(y)\le\phi^c(y)$$

□

Therefore, we have $\min(K) = \max\limits_{\psi\in\mathbf{\Psi}_c(\Omega)} \int\limits_\Omega \psi d\mu + \int\limits_\Omega \psi^c d\nu$

I will not detail here the proof for the existence, the reader is referred to [37], Chapter 4. The idea is that we are now in a much better situation, since the class of admissible functions $\mathbf{\Psi}_c(\Omega)$ is compact (provided that we fix the value of Ψ at one point of Ω to remove the translational invariance degree of freedom of the problem).

Since we have computer implementation in mind, our goal is to find a numerical algorithm to compute an optimal transport map T. At first sight, though the values of the functionals match at a solution of (K) and (D), it seems to be difficult to deduce T from a solution to the dual problem (D). However, there is a nice relation between the dual problem (D) and the initial Monge's problem (M), detailed in [37], chapters 9 and 10. The main result characterizes the pairs of points (x,y) that are connected by the transport plan :

Theorem 1.

$$\forall (x,y)\in\partial_c\psi, \nabla\psi(x)-\nabla_x c(x,y)=0$$

where $\partial_c\psi = \{(x,y)|\phi(x)+\psi(y)=c(x,y)\}$ denotes the so-called c-subdifferential *of ψ.*

Proof. See [37] chapter 10.
I summarize the heuristic argument given at the beginning of the same chapter, that gives some intuition :
Consider a point (x,y) on the c-subdifferential $\partial_c\psi$, that satisfies $\phi(y)+\psi(x)=c(x,y)$ (1).

By definition, $\phi(y) = \psi^c(y) = \inf_x c(x,y) - \psi(x)$, thus $\forall \tilde{x}, \phi(y) \leq c(\tilde{x},y) - \psi(\tilde{x})$, or $\phi(y) + \psi(\tilde{x}) \leq c(\tilde{x},y)$ (2).
By substituting (1) into (2), one gets $\psi(\tilde{x}) - \psi(x) \leq c(\tilde{x},y) - c(x,y)$ for all $\tilde{x}$.
Imagine now that $\tilde{x}$ follows a trajectory parameterized by ϵ and starting at x. One can compute the gradient along an arbitrary direction w by taking the limit when ϵ tends to zero in the relation $\frac{\psi(\tilde{x})-\psi(x)}{\epsilon} \leq \frac{c(\tilde{x},y)-c(x,y)}{\epsilon}$. Thus we have $\nabla\psi(x)\cdot w \leq \nabla_x c(x,y)\cdot w$. The same derivation can be done with $-w$ instead of w, and one gets: $\forall w, \nabla\psi(x)\cdot w = \nabla_x c(x,y)\cdot w$, thus $\forall (x,y) \in \partial_c\psi, \nabla\psi(x) - \nabla_x c(x,y) = 0$.
Note: the derivations above are only formal ones and do not make a proof. The proof requires a much more careful analysis, using generalized definitions of differentiability and tools from convex analysis. □

In the L_2 case, i.e. $c(x,y) = 1/2\|x-y\|^2$, we have $\forall (x,y) \in \partial_c\psi, \nabla\psi(x) + y - x = 0$, thus, whenever the optimal transport map T exists, we have $T(x) = x - \nabla\psi(x) = \nabla(\|x\|^2/2 - \psi(x))$. Not only this gives an expression of T, but also it allows characterizing T as the gradient of a *convex* function, which is an interesting property since it implies that two "transported particles" $x_1 \mapsto T(x_1)$ and $x_2 \mapsto T(x_2)$ cannot collide, as shown below :

Observation 4. *If $c(x,y) = 1/2\|x-y\|^2$ and $\psi \in \mathbf{\Psi}_c(\Omega)$, then $\bar{\psi} : x \mapsto \bar{\psi}(x) = \|x\|^2/2 - \psi(x)$ is convex (it is an equivalence if $\Omega = \mathbb{R}^d$).*

Proof.

$$\begin{aligned}
\psi(x) &= \inf_y \tfrac{|x-y|^2}{2} - \phi(y) \\
&= \inf_y \tfrac{\|x\|^2}{2} - x\cdot y + \tfrac{\|y\|^2}{2} - \phi(y) \\
-\bar{\psi}(x) &= \phi(x) - \tfrac{\|x\|^2}{2} = \inf_y -x\cdot y + \left(\tfrac{\|y\|^2}{2} - \phi(y)\right) \\
\bar{\psi}(x) &= \sup_y x\cdot y - \left(\tfrac{\|y\|^2}{2} - \phi(y)\right)
\end{aligned}$$

The function $x \mapsto x\cdot y - \left(\frac{\|y\|^2}{2} - \phi(y)\right)$ is linear in x, therefore the graph of $\bar{\psi}$ is the upper envelope of a family of hyperplanes, thus $\bar{\psi}$ is convex. □

Observation 5. *Consider the trajectories of two particles parameterized by $t \in [0,1]$, $t \mapsto (1-t)x_1 + tT(x_1)$ and $t \mapsto (1-t)x_2 + tT(x_2)$. If $x_1 \neq x_2$ and for $0 < t < 1$ the particles cannot collide.*

Proof. By contradiction, suppose that you have $t \in (0,1)$ and $x_1 \neq x_2$ such that:

$$\begin{aligned}
&(1-t)x_1 + tT(x_1) &&= (1-t)x_2 + tT(x_2) \\
&(1-t)x_1 + t\nabla\bar{\psi}(x_1) &&= (1-t)x_2 + t\nabla\bar{\psi}(x_2) \\
&(1-t)(x_1-x_2) + t(\nabla\bar{\psi}(x_1) - \nabla\bar{\psi}(x_2)) &&= 0 \\
&\forall v, (1-t)v\cdot(x_1-x_2) + tv\cdot(\nabla\bar{\psi}(x_1) - \nabla\bar{\psi}(x_2)) &&= 0 \\
&\text{take } v = (x_1 - x_2) \\
&(1-t)\|x_1-x_2\|^2 + t(x_1-x_2)\cdot(\nabla\bar{\psi}(x_1) - \nabla\bar{\psi}(x_2)) &&= 0
\end{aligned}$$

which is a contradiction since this quantity is the sum of two strictly positive numbers (recalling the definition of the convexity of $\bar{\psi}$: $\forall x_1 \neq x_2, (x_1-x_2)\cdot(\nabla\bar{\psi}(x_1) - \nabla\bar{\psi}(x_2)) > 0$).

□

At this point, we know that when the optimal transport map exists, it can be deduced from the function ψ using the relation $T(x) = \nabla\bar{\psi} = x - \nabla\psi$. We now consider some ways of finding the function ψ.

The classical change of variable formula gives:

$$\forall B, \int_B \mu(x)d\mu = \mu(B) = \nu(T(B)) = \int_B |\det JT(x)|\, T(x)d\nu \tag{4}$$

where JT denotes the Jacobian matrix of T.

If μ and ν both have a density u and v (i.e. $\forall B, \mu(B) = \int_B u(x)dx$ and $\nu(B) = \int_B v(x)dx$), then one can (formally) consider (4) in a pointwise manner :

$$\forall x \in \Omega, u(x) = |\det JT(x)|\, v(T(x)) \; ; \tag{5}$$

injecting $T = \nabla\bar{\psi}$ and $JT = H\bar{\psi}$ in (5) gives:

$$\forall x \in \Omega, u(x) = \left|\det H\bar{\psi}(x)\right| v(\nabla\bar{\psi}(x)) \tag{6}$$

where $H\bar{\psi}$ denotes the Hessian of $\bar{\psi}$. Equation 6 is known as the *Monge-Ampère* equation. It is a highly non-linear equation, and its solution when it exists often has singularities. It is similar to the eikonal equation that characterizes the distance function and that has a singularity on the medial axis. Note that the derivations above are only formal, studying the solutions of the Monge-Ampère equation requires using more elaborate tools, and several types of weak solutions can be defined (viscosity solutions, solutions in the sense of Brenier, ...).

Still keeping computer implementation in mind, one may consider three different problem settings :

- **continuous:** if μ and ν have a density u and v, it is possible to numerically solve the Monge-Ampère equation, as done in [5] and [34];
- **discrete:** if both μ and ν are discrete (sums of Dirac masses), then finding the optimal transport plan becomes an assignment problem, that can be solved with some variants of linear programming techniques (see the survey in [9]);
- **semi-discrete:** if μ has a density and ν is discrete (sum of Dirac masses), then an optimal transport map exists. It has interesting connections with notions of computational geometry. The remainder of this paper considers this problem setting.

1.4. **The semi-discrete case**

I now consider that μ has a density u, and that $\nu = \sum_{i=1}^{k} \nu_i \delta_{p_i}$ is a sum of k Dirac masses, that satisfies $\nu(\Omega) = \sum_{i=1}^{k} \nu_i = \mu(\Omega)$. Whenever T exists, the pre-images of the Dirac masses $T^{-1}(p_i)$ partition Ω up to a negligible set (the common boundaries of the parts). This subsection reviews the main results in [4], showing that this partition corresponds to a geometrical structure called a power diagram. Note that the same result can be obtained as a direct consequence of Brenier's polar factorization theorem [8]. However, the arguments in [4] can be used to define a practical numerical algorithm, as experimented in 2D in [27] and in 3D further in this paper.

Definition 2. *Given a set P of k points p_i in $\mathbb{R}^d$ and a set W of k real numbers w_i, the* Voronoi diagram *$Vor(P)$ and the* power diagram *$Pow_W(P)$ are defined as follows :*

- *The Voronoi diagram $Vor(P)$ is the partition of $\mathbb{R}^d$ into the subsets $Vor(p_i)$ defined by :*
  $Vor(p_i) := \{x | \|x - p_i\|^2 < \|x - p_j\|^2 \quad \forall j \neq i\}$;
- *the power diagram $Pow_W(P)$ is the partition of $\mathbb{R}^d$ into the subsets $Pow_W(p_i)$ defined by :*
  $Pow_W(p_i) := \{x | \|x - p_i\|^2 - w_i < \|x - p_j\|^2 - w_j \quad \forall j \neq i\}$;
- *the map T_W defined by $\forall i, \forall p \in Pow_W(p_i), T_W(p) = p_i$ is called the* assignment defined by the power diagram *$Pow_W(P)$.*

It can be shown that the assignment defined by a power diagram is an optimal transport map (the main argument of the proof is sketched further). Then one needs to determine - when it is possible - the parameters of this power diagram (i.e. the weights) that realize the optimal transport towards a *given* discrete target measure ν. Intuitively, a power diagram may be thought-of as a generalization of the Voronoi diagram, with additional "tuning buttons" represented by the weights w_i. Changing the weight w_i associated with a point p_i influences the area and the measure $\mu(\mathrm{Pow}_W(p_i))$ of its power cell (the higher the weight, the larger the power

cell). Though the relation between the weights and the measures of the power cells is non-trivial[3], it is well behaved, and as shown below, one can prove the existence and uniqueness of a set of weights such that the measure of each power cell $\mu(\text{Pow}_W(p_i))$ matches a prescribed value ν_i. In this case, the prescribed measures ν_i are referred to as *capacity constraints*, and the power diagram is said to be *adapted* to the capacity constraints. At this point, since we already know that the assignment defined by a power diagram is an optimal transport map, then we are done (i.e. the assignment defined by the power diagram is the optimal transport map that we are looking for). I shall now give more details about the proofs of the two parts of the reasoning.

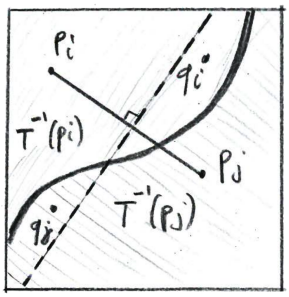

FIGURE 1. Illustration of the (by-contradiction) argument that the common boundary between the pre-images of p_i and p_j is contained by a straight line orthogonal to $[p_i, p_j]$.

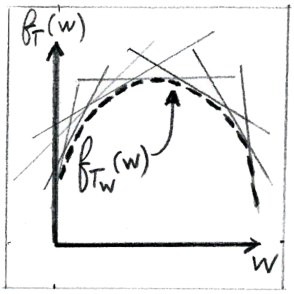

FIGURE 2. The weight vector that defines an optimal transport map can be found as the maximizer of a convex function, defined as the lower envelope of a family of linear functions.

Theorem 2. *Given a set of points P and a set of weights W, the assignment $T_{P,W}$ defined by the power diagram is an optimal transport map.*

Proof. I give here the main idea of the proof (see [4] for the complete one). The main argument is that if T is an optimal transport map, then the common boundary of the pre-images $T^{-1}(p_i)$ and $T^{-1}(p_j)$ of two Dirac masses is a straight line orthogonal to the segment $[p_i, p_j]$. The argument, obtained by contradiction, is illustrated in Figure 1. Suppose that the common boundary between the pre-images $T^{-1}(p_i)$ and $T^{-1}(p_j)$ is not a straight line (thick curve in the figure), then one can find a straight line orthogonal to the segment $[p_i, p_j]$ that has an intersection with the common boundary (dashed line in the figure), and two points q_i and q_j located as shown in the figure. Then, it is clear (by the Pythagorean theorem) that re-assigning q_j to $T^{-1}(p_i)$ and q_i to $T^{-1}(p_j)$ lowers the transport cost, which contradicts the initial assumption. It is then possible to establish that the pre-images correspond to power cells, by invoking some properties of power diagrams [3]. □

Theorem 3. *Given a measure μ with density, a set of points (p_i) and prescribed masses ν_i such that $\sum \nu_i = \mu(\Omega)$, there exists a weights vector W such that $\mu(Pow_W(p_i)) = \nu_i$.*

Proof. Consider the function $f_T(W) = \int_\Omega \|x - T(x)\|^2 - w_{T(x)} d\mu$, where $T : \Omega \to P$ is an *arbitrary* assignment. One can observe that:

- If the assignment T is fixed, $f_T(W) = \int_\Omega \|x - T(x)\|^2 d\mu - \sum_{i=1}^k w_i \mu(T^{-1}(p_i))$ is affine in W. In Figure 2, the graph of $f_T(W)$ for a fixed assignment T corresponds to one of the straight lines (note that in the figure, the "W axis" symbolizes k coordinates);
- we now consider a fixed value of W and different assignments T. Among all the possible T's, it is clear that $f_T(W)$ is minimized by T_W, the assignment defined by the power diagram with weights W (the definition of the power cell minimizes at each point of Ω the integrand in the equation of $f_T(W)$).

[3]Misleadingly, the term 'weight' seems similar to 'mass', but both notions are not directly related.

Now take $T = T_W$ in $f_T(W)$, in other words, consider the function $f_{T_W}(W)$. Its graph, depicted as a dashed curve in Figure 2, is the lower envelope of a family of hyperplanes, thus it is a concave function, with a single maximum. For the next steps of the proof, we now need to compute the gradient $\nabla_W f_{T_W}(W)$. Note that when computing the variations of $f_{T_W}(W)$, both the argument W of f and the parameter T_W change, making the computations quite involved. When T_W changes, the power cells change, and one needs to compute integrals over varying domains. However, it is possible to drastically simplify computations by using the *envelope theorem* [29]: Given a parameterized family of functions $f_T(W)$ (in our case, the parameter is T), whenever the gradient of $\nabla_W f_{T_W}(W)$ exists, it is equal to the gradient $\nabla_W f_{T^*}(W)$ computed at the minimizer T^* (f_{T_W} in our case) [4]. In other words, when computing the gradients, one can directly use the expression of $f_T(W)$ and ignore the variations of T in function of W. In Figure 2, it means that the tangent to f_{T_W} at W corresponds to the (linear) graph of $f_T(W)$ with a fixed $T = T_W$. Note that in our case, the so-called *choice set*, i.e. where T is chosen, is the set of all the assignments between Ω and P. This requires a special version of the envelope theorem that works for such a general choice sets [29].

One can see that the components of the gradient correspond to the (negated) measures of the power cells :

$$\begin{aligned}\frac{\partial f_{T_W}(W)}{\partial w_i} &= \nabla_W \left(\underbrace{\int_\Omega \|x - T(x)\|^2 d\mu}_{\text{constant}(W)} - \sum_{i=1}^{k} w_i \mu(T_W^{-1}(p_i)) \right) \\ &= -\mu(T_W^{-1}(p_i)) = -\mu(\text{Pow}_W(p_i))\end{aligned}$$

We are now in a very good situation to establish the existence and uniqueness of the weight vector W that realizes the optimal transport map. The idea is to use f_{T_W} to construct a function g that has a global maximum realized at a weight vector such that the measures of the power cells match the prescribed measures. Consider the function g defined by $g(W) = f_{T_W}(W) + \sum_i \nu_i w_i$. The components of the gradient of g are given by $\partial g/\partial w_i = -\mu(\text{Pow}_W(p_i)) + \nu_i$. This function is also concave (it is the sum of a concave function plus a linear one), therefore it has a unique global maximum where the gradient is zero. Therefore, at the maximum of g, for each power cell, the measure $\mu(\text{Pow}_W(p_i))$ matches the prescribed measure ν_i. □

Besides showing the existence of a semi-discrete transport map and characterizing it as the assignment defined by a power diagram, the proof in Theorem 3 directly leads to a numerical algorithm, as shown in [27], described in Section 2 further. A similar algorithm can be obtained by starting from a discrete version of the Monge-Ampere equation and the characterization of T as the gradient of a piecewise linear convex function [16].

1.5. **Relation with Kantorovich's dual formulation**

It is interesting to see the relation between the proof of Aurenhammer et.al that does not use the formalism of optimal transport, and the dual formulation of optimal transport. Interestingly, one can remark that the same argument (lower envelope of hyperplanes) is used to establish the concavity of f_{T_W} in Theorem 3 and the convexity of $\bar{\psi}$ in Observation 4. The relation between both formulations can be further explained if we link the Kantorovich potential ϕ and the weights w_i with the relation $\phi(y_i) = 1/2 w_i$. For instance, injecting $\phi(y_i) = 1/2 w_i$ and $c(x, y) = 1/2\|x - y\|^2$ into $\psi(x) = \phi^c(x) = \inf_y c(x, y) - \phi(y)$ gives $\psi(x) = 1/2 \inf_i \|x - y_i\|^2 - w_i$. This corresponds to the definition of the power cells (intuitively, the inf in the definition of ϕ^c is the same as the inf in the definition of the power cell). Now consider $T(x) = x - \nabla\psi(x)$. Still using the expression of $\psi(x)$ above, we get $T(x) = x - 1/2\nabla_x(\|x - y_i\|^2 - w_i) = y_i$. This connects the characterization of T as the solution of $\nabla\phi(x) - \nabla_x c(x, y) = 0$ (Theorem 1) with the characterization of T as the assignment defined by the power diagram (Theorem 3). This corresponds to the point of view developed in [16].

[4] In terms of convex analysis, the same result can also be obtained by considering properties of the subgradient

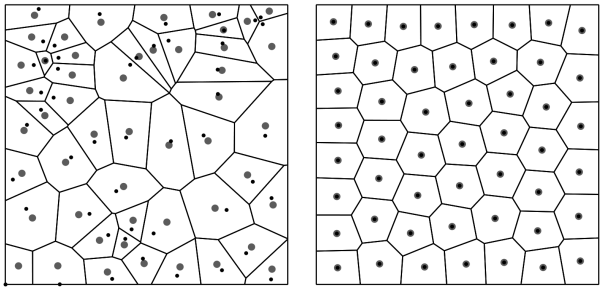

FIGURE 3. Left: random points (black), Voronoi diagram and cell centroids (gray); Right: a centroidal Voronoi diagram.

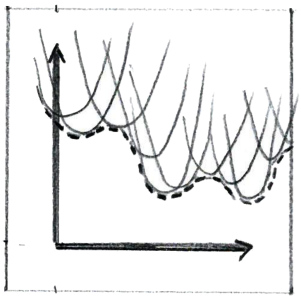

FIGURE 4. The quantization noise power Q minimized in vector quantization is the lower envelope of a family of quadratic functions Q_T.

1.6. **Relation with optimal sampling**

In this section, I exhibit some relations between semi-discrete optimal transport and another problem referred to as *optimal sampling* (or *vector quantization*). Given a compact set $\Omega \subset \mathbb{R}^d$, a measure μ with a density, and a set of k points Y in $\mathbb{R}^d$, the *quantization noise power* of Y is defined as :

$$Q(Y) := \int_\Omega \min_i \|x - y_i\|^2 d\mu = \sum_{i=1}^k \int_{\mathrm{Vor}(y_i)\cap\Omega} \|x - y_i\|^2 d\mu \tag{7}$$

The quantization noise power measures how good Y is at "sampling" Ω (the smaller, the better), see the survey in [13]. The *vector quantization problem* consists in minimizing $Q(Y)$ (i.e. finding the poinset Y that best samples Ω). This notion comes from signal processing theory, and was used to find the optimal assignment of frequency bands for multiplexing communications in a single channel [24]. Designing a numerical algorithm that optimizes Q requires to evaluate the gradient of Q. This requires computing integrals over varying domains (since the Voronoi cells of the y_i's depend on the y_i's), which requires several pages of careful derivations, as done in [13, 17]. At the end, most of the terms cancel-out, leaving a simple formula (see below). One can note the similarity between the quantization noise power (Equation 7) and the objective function maximized by the weight vector in semi-discrete optimal transport (proof of Theorem 3). This suggests using the same type of argument (envelope theorem) to directly obtain the gradient of Q :

Observation 6. *The function Q is of class C^1 (in general it is smoother, see Theorem 4 below) and the components of its gradient relative to one of the point y_i are given by:*

$$\nabla_{y_i} Q(Y) = 2m_i(y_i - g_i)$$

where $m_i = \mu(Vor(y_i)) = \int_{Vor(y_i)} d\mu$ denotes the mass of the Voronoi cell $Vor(y_i)$ and $g_i = 1/m_i \int_{Vor(y_i)} x d\mu$ denotes the centroid of the Voronoi cell $Vor(y_i)$.

Proof. Consider the function $Q_T(Y) := \int_\Omega \|x - T(x)\|^2 d\mu$, parameterized by an assignment $T : \Omega \to Y$. We are in a setting similar to semi-discrete optimal transport (Section 1.4), except that the function $Q_T(Y)$ is quadratic (see Figure 4), whereas $F_T(W)$ is linear (Figure 2). We have :

- $Q(Y) = Q_{T_{\mathrm{Vor}}}(Y)$;
- for a given Y, T_{Vor} is the unique affectation that minimizes $Q_T(Y)$.

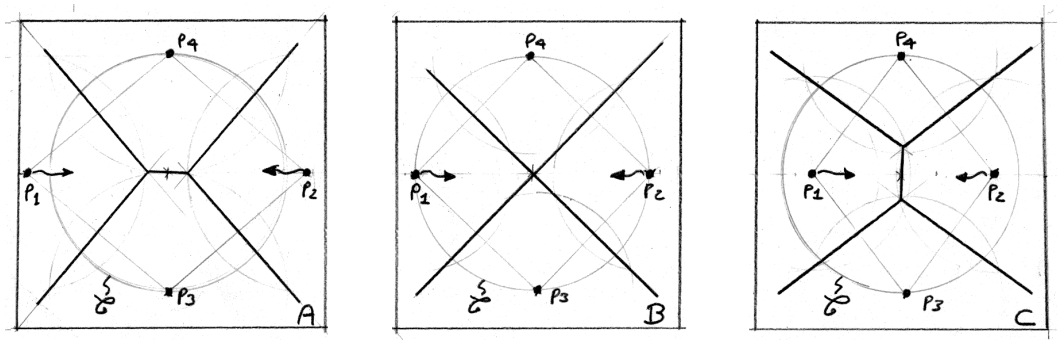

FIGURE 5. Combinatorial change in a Voronoi diagram.

By the envelope theorem [29], we have:

$$\begin{aligned}
\nabla Q(Y) &= \nabla Q_{T_{\mathrm{Vor}}}(Y) = \nabla \textstyle\sum_i \int\limits_{\mathrm{Vor}(y_i)} (x^2 - 2x \cdot y_i + y_i^2) d\mu \\
&= \textstyle\sum_i \left(\nabla \int\limits_{\mathrm{Vor}(y_i)} x^2 d\mu - 2\nabla \int\limits_{\mathrm{Vor}(y_i)} x \cdot y_i d\mu + \nabla \int\limits_{\mathrm{Vor}(y_i)} y_i^2 d\mu \right) \\
\nabla_{y_i} Q(y) &= -2y_i \int\limits_{\mathrm{Vor}(y_i)} x d\mu + 2y_i \int\limits_{\mathrm{Vor}(y_i)} d\mu \\
&= -2m_i g_i + 2m_i y_i = 2m_i (y_i - g_i)
\end{aligned}$$

□

This directly gives the expression of the gradient of Q and explains why most of the terms cancel out in the derivations conducted in [17]. I mention that the same result can be obtained in a more general setting with Reynold's transport theorem [33] (that deals with functions integrated over varying domains).

The regularity of the quantization noise power Q can be further characterized [23]:

Theorem 4. *Given a measure μ with a C^2 density u supported by a compact set $\Omega \subset \mathbb{R}^d$, the quantization noise power $Q(Y) = \sum_i \int_{Vor(i)} \|x - y_i\|^2 u(x) dx$ is of class C^2 whenever the points Y are in generic position and C^1 otherwise. The points are in generic position iff for each pair of points $(y_i, y_j), j \neq i$ we have :*

- *$y_j \neq y_i$;*
- *the intersection $\Pi(i,j) \cap \partial\Omega$ between the bisector $\Pi(i,j) = \{x | d(x, y_i) = d(x, y_j)\}$ and the boundary of Ω is of dimension at most $d - 2$).*

Proof. The main argument of the proof in [23] can be summarized as follows : the expression of $Q(Y)$ depends on the combinatorics of the Voronoi diagram of the points Y. Over subsets where the combinatorics is constant, Q is a rational fraction, of class C^∞. Therefore, to study the regularity of Q, one only needs to examine the configurations of Y that correspond to combinatorial changes. Figure 5 shows an example of such a combinatorial change, that occurs when the points p_1 and p_2 cross the circle $\mathcal{C}$. A combinatorial change occurs whenever Y is in degenerate configuration, with at least one set of at least $d + 1$ co-spherical points (Figure 5-B). In such a configuration, the Delaunay triangulation of the points is non-unique, and each possibility yields a different expression of Q. In the example shown in Figure 5-A, there are two expressions of Q, that correspond to the combinatorics shown in Figure 5-A and 5-C. Analyzing the smoothness of Q means proving that these expressions have C^2 contact at the degenerate configuration. It can be done by considering their Taylor expansions, and showing that they match up to the second order term. □

The smoothness of Q has a practical importance. It suggests using second order methods to optimize Q. In addition, we conjecture that this result still holds when u is C^0, as suggested by our empirical studies in [23]. However, to compute the second order derivatives, the envelope argument cannot be used to compute the Hessian of Q, and the structure of the formulas [14, 17, 23] do not suggest that direct computation can be avoided for them. Note also that Q is the lower envelope of a family of parabola (instead of a family of hyperplanes), therefore the concavity argument does not hold, and the graph of Q has many local minima (as depicted in Figure 4). The local minima of Q, i.e. the point sets Y such that $\nabla Q = 0$, satisfy $\forall i, y_i = g_i$, in other words, the position at each point y_i corresponds to the centroids of the Voronoi cell associated with y_i. For this reason, a stationary point of Q is called a *centroidal Voronoi tessellation*. To compute a centroidal Voronoi tessellation, it is possible to iteratively move each point towards the centroid of its Voronoi cell (Lloyd relaxation [24]), which is equivalent to minimizing Q with a gradient descent method [13]. It is also possible to minimize Q with Newton-type methods [23] that show faster convergence.

More relations between semi-discrete optimal transport and vector quantization can be exhibited by considering a power diagram as the intersection between a $d+1$ Voronoi diagram and $\mathbb{R}^d$:

Observation 7. *The d-dimensional power diagram $Pow_W(Y)$ corresponds to the intersection between the $d+1$ dimensional Voronoi diagram $Vor(\hat{Y})$ and $\mathbb{R}^d$, where the $\mathbb{R}^{d+1}$ lifting $\hat{y}_i$ of y_i is defined by :*

$$\hat{y_i} = \begin{pmatrix} y_{i,1} \\ y_{i,2} \\ \vdots \\ y_{i,d} \\ h_i = \sqrt{w_M - w_i} \end{pmatrix}$$

where $y_{i,j}$ denotes the j-th coordinate of point y_i, and where w_M denotes the maximum of all weights $Max(w_i)$.

Proof.

$$\begin{aligned}
\mathrm{Vor}(\hat{y}_i) \cap \mathbb{R}^d &= \{x \quad | \quad \|\hat{x} - \hat{y}_i\|^2 < \|\hat{x} - \hat{y}_j\|^2 \ \forall j \neq i\} \\
&= \left\{ x \quad | \quad \left\| \begin{bmatrix} x \\ 0 \end{bmatrix} - \begin{bmatrix} y_i \\ \sqrt{w_M - w_i} \end{bmatrix} \right\|^2 < \left\| \begin{bmatrix} x \\ 0 \end{bmatrix} - \begin{bmatrix} y_j \\ \sqrt{w_M - w_j} \end{bmatrix} \right\|^2 \ \forall j \neq i \right\} \\
&= \{x \quad | \quad \|x - y_i\|^2 - w_i + w_M < \|x - y_j\|^2 - w_j + w_M \ \forall j \neq i\} \\
&= \{x \quad | \quad \|x - y_i\|^2 - w_i < \|x - y_j\|^2 - w_j \ \forall j \neq i\} \\
&= \mathrm{Pow}_W(y_i)
\end{aligned}$$

□

We can now see a relation between vector quantization and semi-discrete optimal transport. We consider now the function $\hat{Q}(\hat{Y})$ defined by :

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

where $\hat{Y}$ denotes a set of points in $\mathbb{R}^{d+1}$. The quantity $\hat{Q}(\hat{Y})$ measures the "quality of the sampling" of the measure μ realized by $\hat{Y}$.

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.

$$\begin{aligned}\hat{Q}(\hat{Y}) &= \sum_i \int_{\mathrm{Vor}_{(\hat{y}_i)\cap\mathbb{R}^d}} \|\hat{x}-\hat{y}_i\|^2 d\mu \\ &= \sum_i \int_{\mathrm{Pow}_W(y_i)} \|x-y_i\|^2 - w_i + w_M d\mu \\ &= f_{T_W}(W) + w_M\mu(\Omega)\end{aligned}$$

□

The quantization noise power Q is already known to be of class C^2 almost everywhere (Theorem 4). As a consequence of this observation, since the function $f_{T_W}(W)$ can be obtained through the change of variable $h_i = \sqrt{w_M - w_i}$, it is also of class C^2 almost everywhere. This gives more justification for using a quasi-Newton method to find the maximum of g as done in [27] and in this paper (but note that a complete justification would require to find some bounds on the eigenvalue of the Hessian).

Another consequence of this observation is that given $\Omega \subset \mathbb{R}^d$, a measure μ with a density u supported by Ω and a pointset $\hat{Y}$ in $\mathbb{R}^{d+1}$, optimizing the quantization noise power $\hat{Q}(\hat{Y}) = \sum_i \int_{\mathrm{Vor}_{\hat{y}_i}\cap\Omega} \left\| \begin{bmatrix} x \\ 0 \end{bmatrix} - \hat{y}_i \right\|^2 u(x)dx$ for the first d coordinates of the $\hat{y}_i$'s moves the points y_i in a way that optimizes the sampling of (Ω,μ), and optimizing for the $d+1$-th coordinate computes the weights of a power diagram that defines an assignment that transports μ to the points. Interestingly, the first problem has multiple local minima, whereas the second one admits a global maximum.

2. Numerical Algorithm

I shall now explain how to use the results in Section 1.4 and turn them into an efficient numerical algorithm. The algorithm is a variation of the one in [27]. Besides generalizing it to the 3d case, I make some observations that improve the efficiency of the multilevel optimization method.

The input of the algorithm is a measure μ, supported by a simplicial complex M (i.e. an interconnected set of tetrahedra in 3D), a set Y of k points y_i and k masses ν_i such that $\sum \nu_i = \mu(M)$ where $\mu(.)$ is defined as follows : For a set $B \subset \mathbb{R}^3$, the measure $\mu(B)$ corresponds to the volume of the intersection between the tetrahedra of M and B. Optionally, M can have a density linearly interpolated from its vertices. In this setting, the measure of B corresponds to the integral of the linearly interpolated density on the intersection between B and the tetrahedra of M.

The weight vector that realizes the optimal transport can be obtained by maximizing the function $g(W)$ using different numerical methods. The single-level version of the algorithm in [27] is outlined in Algorithm 1 :

To facilitate reproducing the results, I give more details about each step of the algorithm: (1): note that the components of the gradient of g correspond to the difference between the prescribed measures ν and the measures of the power cells. This gives an interpretation of the norm of the gradient of g, and helps choosing a reasonable ϵ threshold. In the experiments below, I used $\epsilon = 0.01 * \mu(M)/\sqrt{k}$. (2): the algorithm that computes the intersection between a power diagram and a tetrahedral mesh is detailed further (Algorithm 2). (3),(4): once the intersection $\mathrm{Pow}_W(Y) \cap M$ is computed, the terms $g(W)$ and $\nabla g(W)$ are obtained by summing the contributions of each intersection (grayed area in Figure 6). (5): To maximize g, as in [27], I use the L-BFGS numerical optimization method [21]. An implementation of L-BFGS is available in [22].

2.1. **Computing the intersection between a tetrahedral mesh and a power diagram**

To adapt the 2d algorithm in [27] to the 3d case, the only required component is a method that computes the intersection between a tetrahedral mesh and a power diagram (step (2) in Algorithm 1) :

Data: A tetrahedral mesh M, a set of points Y and masses ν_i such that $\mu(M) = \sum \nu_i$
Result: The weight vector W that determines the optimal transport map T from M to $\sum \nu_i \delta_{y_i}$

$W \leftarrow 0$
(1) **while** $\|\nabla g(W)\|^2 < \epsilon$ **do**
  (2) Compute $\mathrm{Pow}_W(Y) \cap M$
  (3) Compute $g(W) = \sum_i \int_{\mathrm{Pow}_W(y_i) \cap M} \|x - y_i\|^2 - w_i d\mu + \sum_i \nu_i w_i$
  (4) Compute $\nabla g(W) = -\mu(\mathrm{Pow}_W(y_i)) + \nu_i$
  (5) update W with L-BFGS
end

Algorithm 1: Semi-discrete optimal transport (single-level algorithm)

Data: A tetrahedral mesh M, a set of points Y and a weight vector W
Result: The intersection $\mathrm{Pow}_W(Y) \cap M$

(1) S: Stack(couple(tet index, point index))
foreach *tetrahedron* $t \in M$ **do**
  (2) **if** t is not marked **then**
    (3) $i \leftarrow i \mid \mathrm{Pow}_W(y_i) \cap t \neq \emptyset$
    (4) Mark(t,i)
    (5) Push(S, (t,i))
    while S is not empty **do**
      (6) (t,i) $\leftarrow$ Pop(S)
      (7) P: Convex $\leftarrow \mathrm{Pow}_W(y_i) \cap t$
      (8) Accumulate(P)
      (9) **foreach** j neighbor of i in P **do**
        if (t, j) is not marked **then**
          (10) Mark(t, j)
          (11) Push(S, (t, j))
        end
      end
      (12) **foreach** t' neighbor of t in P **do**
        if (t', i) is not marked **then**
          (13) Mark(t', i)
          (14) Push(S, (t', i))
        end
      end
    end
  end
end

Algorithm 2: Computing $\mathrm{Pow}_W(Y) \cap M$ by propagation

The algorithm works by propagating simultaneously over the tetrahedra and the power cells. It traverses all the couples (t, i) such that the tetrahedron t has a non-empty intersection with the power cell of y_i. I give below more details on the different steps of the algorithm :

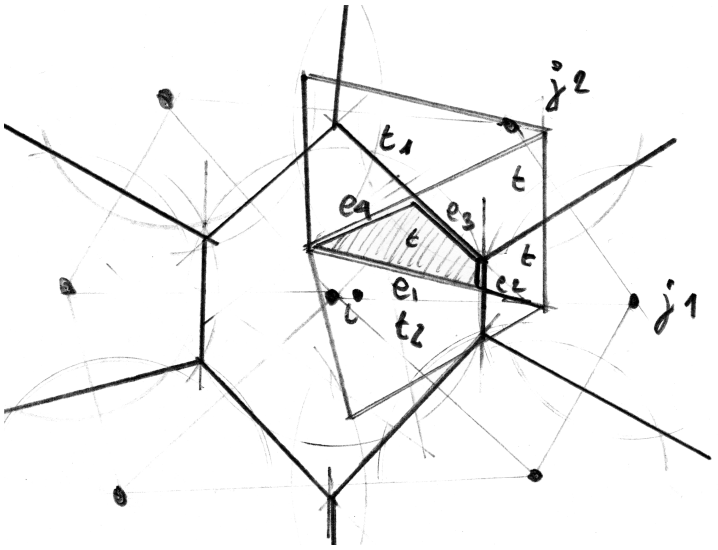

FIGURE 6. Computing the intersection between a power diagram and a tetrahedral mesh by propagation.

(1): To keep track of the tetrahedra and power cells that remain to be traversed, the algorithm uses a *Stack* data structure (e.g., the class `std::stack` of the Standard Template Library). The *Push* operation appends a couple (t, i) to the top of the stack (5), (11), (14), and the *Pop* operation removes the couple (t, i) from the top of the stack (6).

(2): The algorithm keeps track of all the couples (t, i) that were already processed, where t denotes a tetrahedron index and i a point index. A couple (t, i) that was already visited is said to be "marked". A tetrahedron t that has at least a marked couple (t, i) is said to be marked. This can be implemented using a *Set* data structure (e.g. `std::set`). Some more efficient implementations that avoid the time and memory cost of the Set data structure are also described in [32] and [38].

(3): Propagation is initialized by starting from an arbitrary tetrahedron t and a point y_i that has a non-empty intersection between its power cell and t. I use the point y_i that minimizes its power distance $\|y_i - .\|^2 - w_i$ to one of the vertices of t.

(7): a tetrahedron t and a power cell $\mathrm{Pow}_W y_i$ can be both described as the intersection of half-spaces, as well as the intersection $t \cap \mathrm{Pow}_W y_i$. To compute the intersection, I use re-entrant clipping (each half-space is removed iteratively). I implemented two version of the algorithm, a non-robust one that uses floating point arithmetics, and a robust one [19], that uses arithmetic filters [28], expansion arithmetics [36] and symbolic perturbation [15]. Both predicates and power diagram construction algorithm are available in PCK (Predicate Construction Kit) part of my publicly available "GEOGRAM" programming library[5].

(8) the contribution of each intersection $P = t \cap \mathrm{Pow}_W y_i$ is added to g and ∇g. The convex P is illustrated in the (2d) figure 6 as the grayed area (in 3d, P is a convex polyhedron). The algorithm then propagates to both neighboring tetrahedra and points.

(9): each portion of a facet of t that remains in P triggers a propagation to a neighboring tetrahedron t'. In the 2d example of Figure 6, this corresponds to edges e_1 and e_4 that trigger a propagation to triangles t_2 and

[5] `http://gforge.inria.fr/projects/geogram/`

nb masses k	1000	2000	5000	10000	30000	50000	100000
nb iter	146	200	328	529	1240	1103	1102
time (s)	2.8	6.4	21	65	232	568	847

TABLE 1. Statistics for a simple translation scenario with the single-level algorithm. The threshold for $\|\nabla g\|^2$ is set to $\epsilon = 0.01 * \mu(M)/\sqrt{k}$.

t_1 respectively.

(12): each facet of P generated by a power cell facet triggers propagation to a neighboring point. In the 2d example of the figure, this corresponds to edges e_2 and e_3 that trigger propagation to points y_{j_1} and y_{j_2} respectively.

This algorithm is parallelized, by partitioning the mesh M into M_1, M_2, $\dots M_{nb\_cores}$ and by computing in each thread $M_{thrd} \cap \mathrm{Pow}_W(Y)$.

I conducted a simple experiment, where M is a tessellated sphere with 2026 tetrahedra, and Y a sampling of the same sphere shifted by a translation vector of three times the radius of the sphere. The statistics in Table 1 obtained with a standard PC[6] show that the single-level algorithm does not scale-up well with the number of points and starts taking a significant time for processing 10K masses and above. This confirms the observation in [27]. This is because at the initial iteration, all the weights are zero, and the power diagram corresponds to the Voronoi diagram of the points y_i. At this step, only some points y_i on the border of the pointset have a Voronoi cell that "see" the mesh M (i.e. that have a non-empty intersection with it). It takes many iteration to compute the weights that "shift" the concerned power cells onto M and allow inner points to see M. It is only once all the points of Y "see" M that the numerical method can capture the trend of g around the maximum (and then it takes a small number of iterations to the algorithm to balance the weights). Intuitively, Y is "peeled" only one layer of points at a time. The bad effect on performances is even more important than in [27], because in the 3d setting, the proportion of "inner" points relative to the number of points on the border of the pointset is larger than in 2d.

2.2. **Multi-level algorithm**

To improve performances, I follow the approach in [27], that uses a multilevel algorithm. The idea consists in "bootstrapping" the algorithm on a coarse sub-sampling of the pointset. The "peeling" effect mentioned in the previous paragraph is limited since we have a small number of points. Then the algorithm is run with a larger number of points, using the previously computed weights as an initialization. The set of points can be decomposed into multiple level of increasing resolution. The complete algorithm is detailed below :

In my implementation, for step (1), the ratio between the number of points in a level and in the rest of the points is set to 0.125. For the spatial sort in step (2), the algorithm, available in "GEOGRAM"[7], was inspired by the variant of the Hilbert sort implemented in [12]. (3): Before computing the optimal transport maps, since the number of points changes at each level, the masses of the points need to be updated. At step (4), to further improve the speed of convergence, I initialize the the weight of a new point w_i using linear least squares with 10 nearest neighbors for degree 1 and quadratic least squares with 20 nearest neighbors for degree 2 : In Section 1.5, we remarked that the weights w_i corresponds to the potential ϕ evaluated at y_i (with a 1/2 factor). For a translation, we know that $T^{-1}(y) = y - V = y - \nabla\phi$, therefore $\phi(y) = V \cdot y$ where V denotes the translation vector. In more general settings, ϕ is still likely to be quite regular (except on its singularities where T is discontinuous). When initializing a level from the previous one, this suggests initializing the new

[6]experiments done with a 2.8 GHz Intel Core i7-4900MQ CPU with an implementation of Algorithm 2 that uses 8 threads.

[7]`http://gforge.inria.fr/projects/geogram/`

Data: A tetrahedral mesh M, a set of points Y and masses ν_i such that $\mu(M) = \sum \nu_i$
Result: The weight vector W that determines the optimal transport map T from M to $\sum \nu_i \delta_{y_i}$

Apply a random permutation to the points Y
(1) Partition the interval of indices $[1, k]$ of Y into n_l intervals $[b_l, e_l]$ of increasing size
foreach level l **do**
  (2) Sort the points $y_{b_l} \dots Y_{e_l}$ spatially
  (3) **For each** i, $\nu_i \leftarrow |M|/e_l$
  (4) Interpolate the weights $w_{b_l} \dots w_{e_l}$ from the already computed weights $w_1 \dots w_{b_l - 1}$
  Optimize the weights using Algorithm 1
end

Algorithm 3: Semi-discrete optimal transport (multi-level algorithm)

nb masses	1000	2000	5000	10000	30000	50000	100000
deg. 0 time (s)	2.5	6	19	38	184	356	959
deg. 1 time (s)	1	2	6	14	54	103	172
deg. 2 time (s)	1.4	2.2	6	16	58	138	172
BRIO/deg. 2 time (s)	1	1.65	3.4	9	26	62	106
single level time (s)	2.8	6.4	21	65	232	568	847

TABLE 2. Statistics for a simple translation scenario with the multi-level algorithm. The mesh M has 61233 tetrahedra. Timings are in seconds. Each level is initialized from the previous one with regressions of different degrees.

w_i's from a regression of their nearest neighbors computed at the previous level. Table 2 shows the statistics for initialization with the nearest neighbor (deg. 0), linear regression with 10 nearest neighbors (deg. 1) and quadratic regression with 20 nearest neighbors (deg. 2). As can be seen, initializing with linear regression results in a significant speedup. In this specific case though, quadratic regression does not gain anything. It is not a big surprise since we know already that $\phi(y) = V \cdot y$ is linear in this specific case, but it can slightly improve performances in more general settings, as shown further.

The influence of the degree of the regression is evaluated in Table 3 for a configuration where a sphere is splitted into two spheres (first row in Figure 9). Unlike in the previous translation case, in this configuration the potential ϕ is non-linear (see the deformations of the spheres), and a higher degree regression slightly improves the speed of convergence for a large number of points, since it captures more variations of ϕ and better initializes W.

Finally, it is possible to gain another x2 speedup factor : the BRIO (Biased Randomized Insertion Order) algorithm that we use to compute the power diagrams [2] sorts the points with a multilevel spatial reordering method, that makes it very efficient. In short, BRIO computes a power diagram using a multi-level method. A coarse power diagram is first created from a randomly chosen subset of the points and then it is refined using successive levels. It is possible to use the same multilevel spatial ordering for both the numerical optimization and for computing the power diagrams (the *levels* used by the multilevel optimal transport computation and by the multilevel power diagram algorithm are then the same). Statistics are reported in the row (BRIO/deg. 2) in the table. Since only the weights change during the iterations, this order needs to be computed once only, at the beginning of the algorithm. Note the overall 8x acceleration factor as compared to the single-level algorithm in Table 1 (repeated in the last row of Table 2 to ease comparison).

nb masses	1000	2000	5000	10000	30000	50000	100000
BRIO/deg. 1 time (s)	1	1.7	3.5	9.8	25	61.7	122
BRIO/deg. 2 time (s)	0.9	1.6	3.5	8.4	28.3	61.4	112

TABLE 3. Statistics for splitting a sphere into two spheres with the multi-level algorithm. Timings are in seconds. Each level is initialized from the previous one with regressions of different degrees.

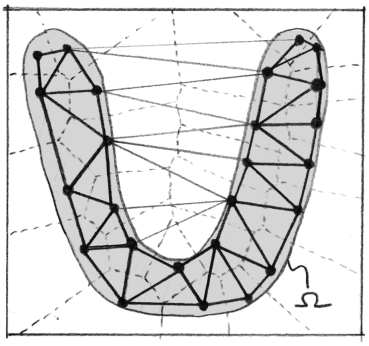

FIGURE 7. Example of a Delaunay triangulation restricted to a domain Ω (thick triangles).

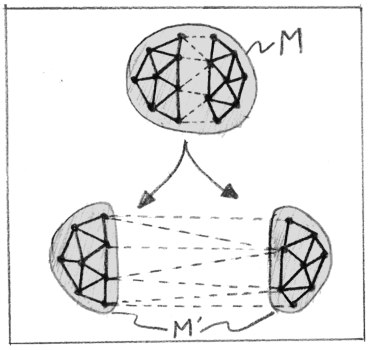

FIGURE 8. The triangles that do not appear in both restricted Delaunay triangulations are discarded (dashed).

2.3. Using semi-discrete transport to approximate the transport between two tetrahedral meshes

I now consider the case where the input is a pair of tetrahedral meshes M and M', with the goal of generating a sequence of tetrahedral meshes that realize an approximation of the optimal transport between M and M'. The algorithm will generate a mesh G with k vertices and a pair of points p_i^0 and p_i^1 attached to each vertex. Transport is parameterized by time $t \in [0,1]$ with $p_i(t) = (1-t)p_i^0 + tp_i^1$. To compute G, we need to determine the following elements :

(1) the vertices positions (p_i^0) at time $t = 0$;
(2) the vertices positions (p_i^1) at time $t = 1$;
(3) the tetrahedra of G. Each tetrahedron connects four vertices (p_i, p_j, p_k, p_l)

This requires to introduce the notions of *restricted Voronoi diagram* and *restricted Delaunay triangulation* (as well as their weighted counterparts, restricted power diagram and restricted regular triangulation) :

Definition 3. *Given a set P of k points p_i in $\mathbb{R}^d$ and a set $\Omega \subset \mathbb{R}^d$, the* restricted Voronoi diagram $Vor(P)|_\Omega$ *and the* restricted Delaunay triangulation $Del(P)|_\Omega$ *are defined as follows :*

- *The restricted Voronoi diagram $Vor(P)|_\Omega$ is the partition of Ω into the subsets $Vor(p_i)|_\Omega$ defined by : $Vor(p_i)|_\Omega := \{x \in \Omega \mid \|x - p_i\|^2 < \|x - p_j\|^2 \quad \forall j \neq i\} \quad = \quad Vor(p_i) \cap \Omega$;*
- *The restricted Delaunay triangulation $Del(P)|_\Omega$ is the abstract simplicial set $\Sigma = \Sigma^1 \cup \Sigma^2 \cup \ldots \Sigma^k$ where $\Sigma^i = \left\{ \{ Vor(p_{j_1}), Vor(p_{j_2}), \ldots Vor(p_{j_i}) \} \mid \overline{Vor(p_{j_1})} \cap \ldots \overline{Vor(p_{j_i})} \neq \emptyset \right\}$; $\overline{Vor(p_i)} = Vor(p_i) \cup \partial Vor(p_i)$.*

Similarly, one can define the restricted power diagram $Pow_W(P)|_\Omega$ and the associated restricted regular triangulation $Reg_W(P)|_\Omega$:

- *The restricted power diagram $Pow_W(P)|_\Omega$ is the partition of Ω into the subsets $Pow_W(p_i)|_\Omega$ defined by : $Pow_W(p_i)|_\Omega := \{x \in \Omega \mid \|x - p_i\|^2 - w_i < \|x - p_j\|^2 - w_j \quad \forall j \neq i\} \quad = \quad Pow_W(p_i) \cap \Omega$;*

- *The restricted regular triangulation* $Reg_W(P)|_\Omega$ *is the abstract simplicial set* $\Sigma_W = \Sigma_W^1 \cup \Sigma_W^2 \cup \dots \Sigma_W^k$ *where* $\Sigma_W^i = \left\{ \{Pow_W(p_{j_1}), Pow_W(p_{j_2}), \dots Pow_W(p_{j_i})\} \mid \overline{Pow_W(p_{j_1})} \cap \dots \overline{Pow_W(p_{j_i})} \neq \emptyset \right\}$.

An example of restricted Voronoi diagram and restricted Delaunay triangulation is depicted in Figure 7. The restricted Voronoi diagram corresponds to the partition of Ω (gray area) realized by the Voronoi cells (dashed lines). The restricted Delaunay triangulation is symbolized by thick triangles. The thin triangles do not belong to the restricted Delaunay triangulation, because the Voronoi vertices they correspond to do not belong to Ω (intersections of dashed lines).

Equipped with these notions, we can now find a way of computing the elements that determine G. We start from a semi-discrete optimal transport computed by the algorithm in the previous section, i.e. a set of k points Y and a weight vector W, such that the application that maps each restricted power cell $\mathrm{Pow}_W(y_i)|_M$ to y_i is an optimal transport map. The mesh G will be constructed by connecting the vertices Y with tetrahedra and determining the original position at time $t = 0$ of the vertices :

(1) vertices positions (p_i^1) at time $t = 1$: I use the sampling Y of M' ($p_i^1 = y_i$);
(2) vertices positions (p_i^0) at time $t = 0$: once the weight vector W is computed, each point y_i is back-mapped to a restricted power cell $\mathrm{Pow}_W(y_i)|_M$ (that corresponds to the set of points of M transported to the Dirac mass located at y_i). To determine y_i^0, I use the centroid of $\mathrm{Pow}_W(y_i)|_M$;
(3) tetrahedra of G: we now imagine that the weight vector W varies over time, and that it is linearly interpolated between $W(0) = W$ and $W(1) = [0 \dots 0]$. At time $t = 0$, the power diagram $\mathrm{Pow}_{W(0)}(Y)$ partitions M into power cells (that correspond to the pre-images of the y_i's). At time $t = 1$, since all the weights are zero, the power diagram $\mathrm{Pow}_{W(1)}(Y)$ corresponds to the Voronoi diagram of the y_i's, that sample M'. Therefore, we can define two meshes : $\mathrm{Reg}_{W(0)}(Y)|_M$ and $\mathrm{Reg}_{W(1)}(Y)|_{M'} = \mathrm{Del}(Y)|_{M'}$. Since topology changes may occur between M and M', in general, the two meshes $\mathrm{Reg}_W(Y)|_M$ and $\mathrm{Del}(Y)|_{M'}$ differ. The idea is then to only keep the tetrahedra that remain stable during the transport (i.e., that both appear in $\mathrm{Reg}_W(Y)|_M$ and $\mathrm{Del}(Y)|_{M'}$). Figure 8 shows an example where M is splitted into two components of M'. Some triangles (dashed) do not belong to the restricted Delaunay triangulation of M'.

The algorithm can be summarized as follows :

Data: Two tetrahedral meshes M and M', and k the desired number of vertices in the result
Result: A tetrahedral mesh G with k vertices and a pair of points p_i^0 and p_i^1 attached to each vertex. Transport is parameterized by time $t \in [0, 1]$ with $p_i(t) = (1-t)p_i^0 + tp_i^1$.

(1) Sample M' with a set Y of k points
(2) Compute the weight vector W that realizes the optimal transport between M and Y (Algorithm 3)
(3) Compute $E = \mathrm{Del}(Y)|M'$ and $F = \mathrm{Pow}_W(Y)|M$; Tets(G) $\leftarrow E \cap F$
(4) **Foreach** $i \in [1 \dots k]$, $(p_i)^0 \leftarrow \mathrm{centroid}(\mathrm{Pow}_W(y_i) \cap M)$; $(p_i)^1 \leftarrow y_i$

Algorithm 4: Approximated optimal transport between two tetrahedral meshes

At step (1), to compute a homogeneous sampling, I initialize Y with a centroidal Voronoi tessellation (see Section 1.6).

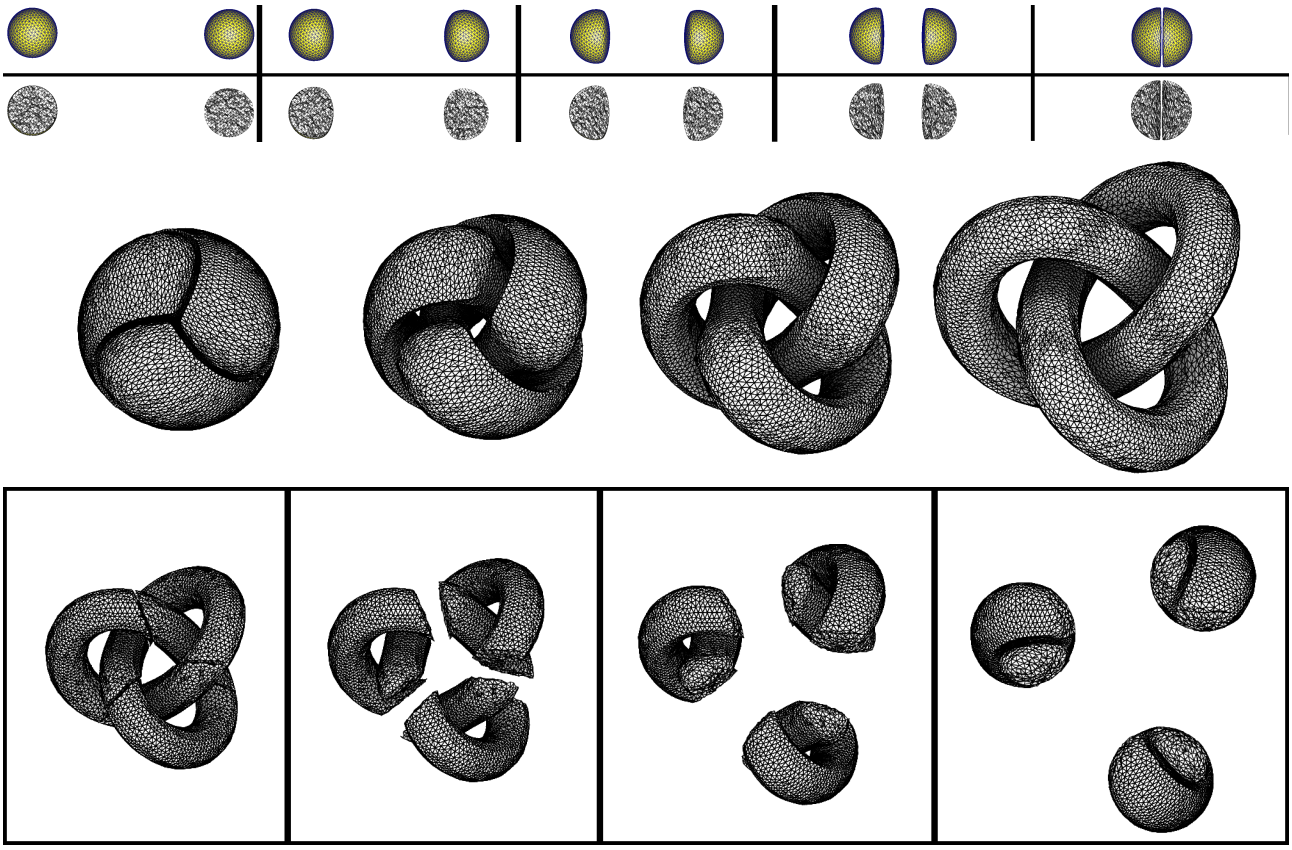

FIGURE 9. Some examples of semi-discrete optimal transport with topology changes. The two-spheres to one-sphere transport (first row) is also shown in cross-section to display the evolution of the tetrahedral mesh inside the spheres.

nb masses	1000	2000	5000	10000	30000	50000	10^5	3×10^5	5×10^5	10^6
time (s)	1.45	3.2	7.3	17.3	55	154	187	671	1262	2649

TABLE 4. Statistics for the Armadillo $\rightarrow$ sphere optimal transport with varying number of masses (see third row of Figure 10). Timings are given in seconds. The multi-level algorithm with BRIO pre-ordering and degree 2 regressions is used.

3. Results and conclusions

Several results are shown in Figures 9 and 10. Note that when the volume of M and M' differ, using $\nu_i = |M|/k$ changes the "density" of M' and preserves the total mass. The intermediary steps are generated by using $p_i = (1-t)p_i^0 + tp_i^1$ for the locations at the vertices of G. As can be seen, the combinatorial criterion that selects the stable tetrahedra successfully finds the discontinuities. The third row of Figure 10 demonstrates some potential applications in computer graphics. In the bottom row, the obtained deformation looks "natural" and "visually pleasing" (as far as I can judge, but my own judgment may be biased ...). However, a "user" would probably prefer to rotate the star in the center column of Figure 10 rather than splitting and merging the branches, but optimal transport "does not care" about preserving topology. Results with a spatially varying density are shown in Figures 11 and 12.

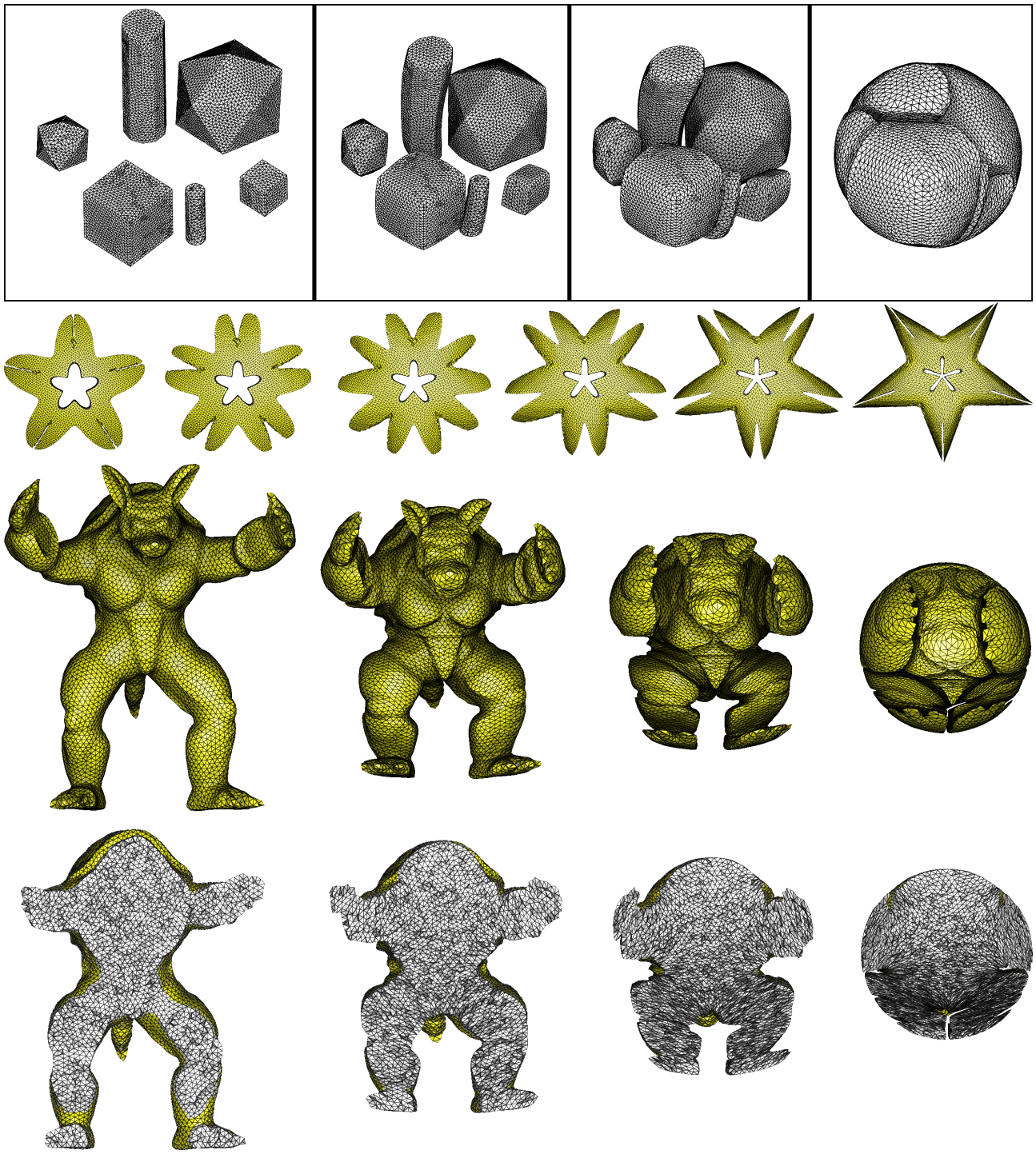

FIGURE 10. More examples of semi-discrete optimal transport. Note how the solids deform and merge to form the sphere on the first row, and how the branches of the star split and merge on the second row. The Armadillo-to-sphere transport (third row) is shown in cross-section (fourth row) to display the evolution of the tetrahedral mesh.

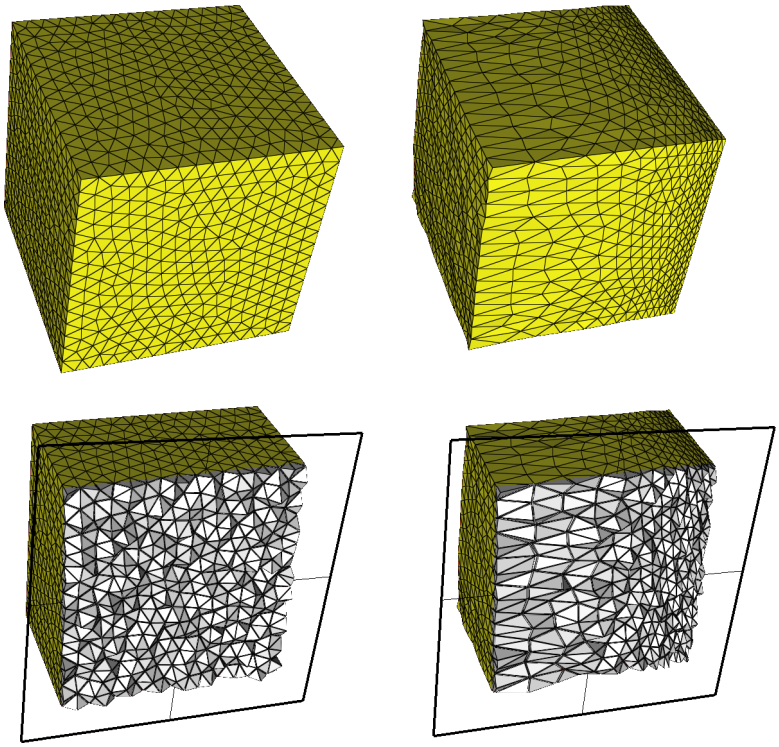

FIGURE 11. Transport from a uniform density (sampled with 5000 Dirac masses) to a varying density of $1.0 + x^2$. The second row shows the mesh evolution in cross-section. Computation takes 28 seconds, using 3 levels.

Timings for the Armadillo $\rightarrow$ sphere optimal transport are given in Table 4. The algorithm scales up reasonably well, and computes the optimal transport from a tetrahedral mesh to 300K Dirac masses in 10 minutes. It scales-up to 1 million Dirac masses (but it nearly takes 45 minutes).

The numerical experiment tend to confirm that this algorithm can be used as a practical way of computing the Wasserstein distance. It scales up to one million Dirac masses, which is out of reach of discrete combinatorial algorithms, such as the Hungarian algorithm [31] that operates in $O(n^3)$. Note also that the continuous Benamou-Brenier formulation [5] would require to solve a series of $4d$ Poisson problems. As suggested in [6], our algorithm may be used to implement the Jordan-Kinderlehrer-Otto scheme [18] to solve Monge-Ampere type PDEs. This will require to compute the gradient of the Wasserstein distance, which can be done using derivations similar to the ones in Section 1.6 and derivatives of Voronoi vertices in [20].

To conclude, I mention that the main limitation of Algorithm 4 is that the discontinuities are sampled at the precision of the initial sampling, that does not takes them into account. As a consequence, this leaves a gap that has a width of one tetrahedron in the result. This gap corresponds to the tetrahedra that are not stable through transport (see Figure 8). One can clearly see this gap in the figures. Moreover, when the shape undergoes strong deformations, flipping may occur, making the concerned pairs of tetrahedra disappear in the result (for instance, one can observe some holes in the legs of the armadillo in Figure 10). With a better representation of discontinuity, one may obtain a more precise representation of the transport. By exploiting characterization of the singular set where the mapping is discontinuous [11], it may be possible to design a better algorithm. In our setting, this leads to the following questions :

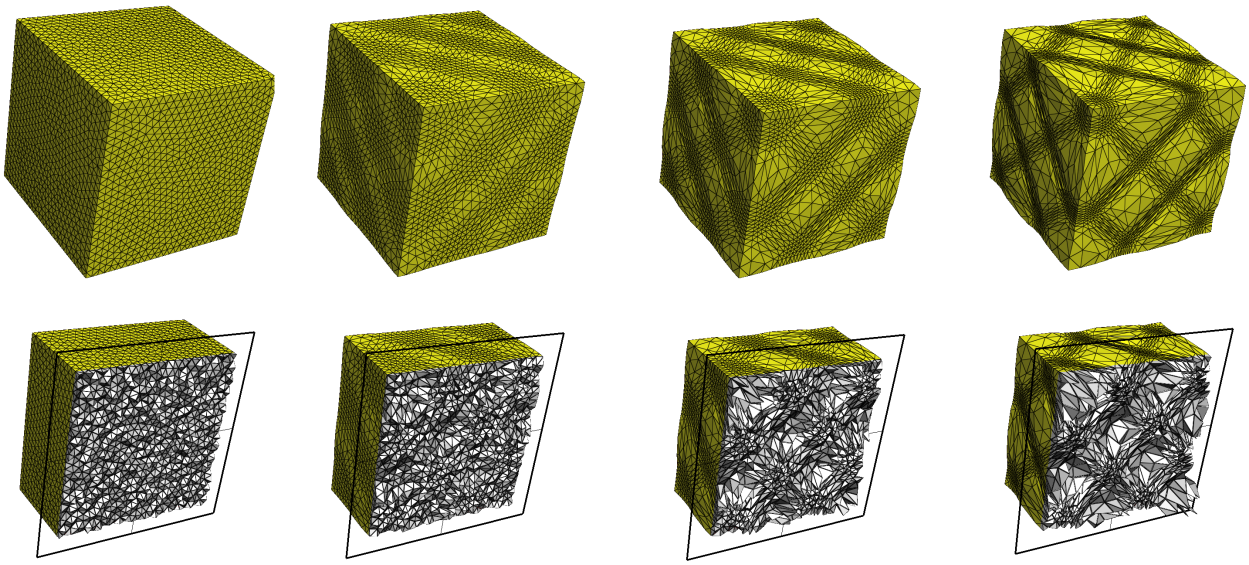

FIGURE 12. Transport from a uniform density (sampled with 20000 Dirac masses) to a varying density of $2.0 + \sin(x)\sin(y)\sin(z)$. The second row shows the mesh evolution in cross-section. Computation takes 102 seconds, using 4 levels.

(1) Given two measures μ and ν supported by tetrahedral meshes M and M', is it possible to invent an algorithm that generates a faithful representation of the singular set ?
(2) What does the singular set looks like if M and M' both have a density linearly interpolated over the tetrahedra ?
(3) What does the singular set looks like if μ and ν are supported by two different set of spheres ?

I wish to thank Nicolas Bonneel for many discussions and for proofreading an early version of this article, Jean-David Benamou, Quentin Mérigot and Gabriel Peyré for discussions, and the anonymous reviewer for detailed comments and suggestions that helped improving this article.

REFERENCES

[1] L. AMBROSIO AND N. GIGLI, *A users guide to optimal transport*, Modelling and Optimisation of Flows on Networks, Lecture Notes in Mathematics, (2013), pp. 1–155.
[2] N. AMENTA, S. CHOI, AND G. ROTE, *Incremental constructions con brio*, in Proceedings of the Nineteenth Annual Symposium on Computational Geometry, SCG '03, New York, NY, USA, 2003, ACM, pp. 211–219.
[3] F. AURENHAMMER, *Power diagrams: Properties, algorithms and applications*, SIAM J. Comput., 16 (1987), pp. 78–96.
[4] F. AURENHAMMER, F. HOFFMANN, AND B. ARONOV, *Minkowski-type theorems and least-squares partitioning*, in Symposium on Computational Geometry, 1992, pp. 350–357.
[5] J.-D. BENAMOU AND Y. BRENIER, *A computational fluid mechanics solution to the monge-kantorovich mass transfer problem*, Numerische Mathematik, 84 (2000), pp. 375–393.
[6] J.-D. BENAMOU, G. CARLIER, Q. MÉRIGOT, AND E. OUDET, *Discretization of functionals involving the monge-ampère operator*, arXiv, (2014). [math.NA] http://arxiv.org/abs/1408.4336.
[7] N. BONNEEL, M. VAN DE PANNE, S. PARIS, AND W. HEIDRICH, *Displacement interpolation using lagrangian mass transport*, ACM Trans. Graph., 30 (2011), p. 158.
[8] Y. BRENIER, *Polar factorization and monotone rearrangement of vector-valued functions*, Communications on Pure and Applied Mathematics, 44 (1991), pp. 375–417.
[9] R. BURKARD, M. DELL'AMICO, AND S. MARTELLO, *Assignment Problems*, SIAM, 2009.

[10] L. Caffarelli, *The monge-ampère equation and optimal transportation, an elementary review*, Optimal transportation and applications (Martina Franca, 2001), Lecture Notes in Mathematics, (2003), pp. 1–10.
[11] G. De Philippis and A. Figalli, *Partial regularity for optimal transport maps*, Publications mathématiques de l'IHES, (2014), pp. 1–32.
[12] C. Delage and O. Devillers, *Spatial sorting*, in CGAL User and Reference Manual. CGAL Editorial Board, 2011. 3.9 edition.
[13] Q. Du, V. Faber, and M. Gunzburger, *Centroidal voronoi tessellations: Applications and algorithms*, SIAM Rev., 41 (1999), pp. 637–676.
[14] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: applications and algorithms*, SIAM Review, 41 (1999), pp. 637–676.
[15] H. Edelsbrunner and E. P. Mücke, *Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms*, ACM TRANS. GRAPH, 9 (1990), pp. 66–104.
[16] X. Gu, F. Luo, J. Sun, and S.-T. Yau, *Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations*, arXiv, (2013). [math.PR] http://arxiv.org/abs/1302.5472.
[17] M. Iri, K. Murota, and T. Ohya, *A fast Voronoi-diagram algorithm with applications to geographical optimization problems*, in Proc. IFIP, 1984, pp. 273–288.
[18] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the fokker-planck equation*, SIAM J. Math. Anal, 29 (1999), pp. 1–17.
[19] B. Lévy, *Restricted voronoi diagrams for (re)-meshing surfaces and volumes*, in Curves and Surfaces conference proceedings, 2014.
[20] B. Lévy and Y. Liu, *Lp centroidal voronoi tesselation and its applications*, ACM Transactions on Graphics (SIGGRAPH conference proceedings), (2010).
[21] D. C. Liu and J. Nocedal, *On the limited memory bfgs method for large scale optimization*, Math. Program., 45 (1989), pp. 503–528.
[22] Y. Liu, *HLBFGS, a hybrid l-bfgs optimization framework which unifies l-bfgs method, preconditioned l-bfgs method, preconditioned conjugate gradient method.* http://research.microsoft.com/en-us/um/people/yangliu/software/HLBFGS/.
[23] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang, *On centroidal Voronoi tessellation—energy smoothness and fast computation*, ACM Transactions on Graphics, 28 (2009), pp. 1–17.
[24] S. P. Lloyd, *Least squares quantization in pcm*, IEEE Transactions on Information Theory, 28 (1982), pp. 129–137.
[25] R. J. McCann, *Existence and uniqueness of monotone measure-preserving maps*, Duke Mathematical Journal, 80 (1995), pp. 309–323.
[26] F. Mémoli, *Gromov-wasserstein distances and the metric approach to object matching*, Foundations of Computational Mathematics, 11 (2011), pp. 417–487.
[27] Q. Mérigot, *A multiscale approach to optimal transport*, Comput. Graph. Forum, 30 (2011), pp. 1583–1592.
[28] A. Meyer and S. Pion, *FPG: A code generator for fast and certified geometric predicates*, in Real Numbers and Computers, Santiago de Compostela, Espagne, 2008, pp. 47–60.
[29] P. Milgrom and I. Segal, *Envelope Theorems for Arbitrary Choice Sets*, Econometrica, 70 (2002), pp. 583–601.
[30] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, Histoire de l'Acadmie Royale des Sciences (1781), (1784), pp. 666–704.
[31] J. Munkres, *Algorithms for the assignment and transportation problems*, Journal of the Society of Industrial and Applied Mathematics, 5 (1957), pp. 32–38.
[32] V. Nivoliers, *Echantillonage pour l'approximation de fonctions sur des maillages*, PhD thesis, IAEM, Université de Lorraine, November 2012.
[33] V. Nivoliers and B. Lévy, *Approximating functions on a mesh with restricted voronoi diagrams*, in ACM/EG Symposium on Geometry Processing / Computer Graphics Forum, 2013.
[34] N. Papadakis, G. Peyré, and E. Oudet, *Optimal Transport with Proximal Splitting*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 212–238.
[35] F. Santambrogio, *Introduction to Optimal Transport Theory*, in Optimal Transport, Theory and Applications, August 2014. [math.PR] http://arxiv.org/abs/1009.3856.
[36] J. R. Shewchuk, *Robust adaptive floating-point geometric predicates*, in Symposium on Computational Geometry, 1996, pp. 141–150.
[37] C. Villani, *Optimal transport : old and new*, Grundlehren der mathematischen Wissenschaften, Springer, Berlin, 2009.
[38] D. Yan, W. Wang, B. Lévy, and Y. Liu, *Efficient computation of clipped voronoi diagram*, Computer-Aided Design Journal, (2011).