Optimal Transport Approximation of Measures

Frédéric de Gournay  Jonas Kahn  Léo Lebrat  Pierre Weiss

April 23, 2018

Abstract

We propose a fast and scalable algorithm to project a given density on a set of structured measures. The measures can be discrete or supported on curves for instance. The proposed principle and algorithm are a natural generalization of previous results revolving around the generation of blue-noise point distributions, such as Lloyd’s algorithm or more advanced techniques based on power diagrams. We provide a comprehensive convergence theory together with new approaches to accelerate the generation of point distributions. We also design new algorithms to project curves onto spaces of curves with bounded length and curvature or speed and acceleration. We illustrate the algorithm’s interest through applications in advanced sampling theory, non-photorealistic rendering and path planning.

1 Introduction

The aim of this paper is to approximate a target measure \( \mu \) with probability density function \( \rho : \Omega \to \mathbb{R}_+ \) with probability measures possessing some structure. This problem arises in a large variety of fields including finance (Pages and Wilbertz, 2012), computer graphics (Solomon et al., 2015), sampling theory (Boyer et al., 2016) or optimal facility location (Gastner and Newman, 2006). An example in non photo-realistic rendering is shown in Figure 1, where the target image in Fig. 1a is approximated by an atomic measure in Fig. 1b, by a smooth curve in Fig. 1c and by a set of segments in Fig. 1d. Given a set of admissible measures \( \mathcal{M} \) (i.e. atomic measures, measures supported on smooth curves or segments), the best approximation problem can be expressed as follows:

\[
\min_{\nu \in \mathcal{M}} D(\nu, \mu), \tag{1}
\]

where \( D \) is a distance between measures.

1.1 Contributions

Our main contributions in this article are listed below.

- Develop a few original applications for the proposed algorithm.
- Develop a fast numerical algorithm to solve problem (1), when \( D \) is the \( W_2 \) transportation distance and \( \Omega = [0,1]^2 \).
- Show its connections to existing methods such as Lloyd’s (1982) algorithm or optimal transport halftoning (De Goes et al., 2012).
- Design algorithms specific to the case where the space of admissible measures \( \mathcal{M} \) consists of measures supported on curves with geometric constraints (e.g. fixed length and bounded curvature).
- Generate a gallery of results to show the versatility of the approach.

In the next section, we put our main contributions in perspective.

1.2 Related works

1.2.1 Projections on measure spaces

To the best of our knowledge, the generic problem (1) was first proposed by Chauffert et al. (2017) with a distance \( D \) constructed through a convolution kernel. Formulation (1) covers a large amount of applications that are often not formulated explicitly as optimization problems. We review a few of them below.

Finitely supported measures A lot of approaches have been developed when \( \mathcal{M} \) is the set of finitely supported measures

\[
\mathcal{M}_{f,n} = \left\{ \nu = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}, (x_i) \in \Omega^n \right\}, \tag{2}
\]
where \( n \) is the support cardinality, or the set of atomic measures defined by:

\[
\mathcal{M}_{a,n} = \left\{ \nu(x,w) = \sum_{i=1}^{n} w[i] \delta_{x[i]}, (x_i) \in \Omega^n, w \in \Delta_{n-1} \right\},
\]

(3)

where \( \Delta_{n-1} = \{ \sum_{i=1}^{n} w[i] = 1, w[i] \geq 0, \forall i \} \) is the canonical simplex.

For these finitely supported measure sets, solving problem (1) yields nice stippling results, which is the process of approximating an image with a finite set of dots (see Fig. 1b). This problem has a long history and a large amount of methods were designed to find dots locations and radii that minimize visual artifacts due to discretization (Floyd, 1976; Lloyd, 1982; Ulichney, 1987; Balzer et al., 2009). Lloyd’s algorithm is among the most popular. We will see later that this algorithm is a solver of (1), with \( \mathcal{M} = \mathcal{M}_{a,n} \). Lately, explicit variational approaches (Schmaltz et al., 2010; De Goes et al., 2012) have been developed. The work of De Goes et al. (2012) is closely related to our paper since they propose solving (1), where \( D \) is the \( W_2 \) transportation distance and \( \mathcal{M} = \mathcal{M}_{f,n} \). This sole problem is by no means limited to stippling and it is hard to provide a comprehensive list of applications. Xin et al. (2016) give a few of them in their introduction.

**Best approximation with curves** Another problem that is met frequently is to approximate a density by a curve. This can be used for non photorealistic rendering of images or sculptures (Kaplan et al., 2005; Akleman et al., 2013). It can also be used to design trajectories of the nozzle of 3D printers (Chen et al., 2017). It was also used for the generation of sampling schemes (Boyer et al., 2016).

Apart from the last application, this problem is usually solved with methods that are not clearly expressed as an optimization problem.

**Best approximation with arbitrary objects** Problem (1) encompasses many other applications such as the optimization of networks (Gastner and Newman, 2006), texture rendering or non photorealistic rendering (Hertzmann, 2003; Hiller et al., 2003; Schlechtweg et al., 2005; Kim et al., 2009; Du, 2017), or sampling theory (Boyer et al., 2014).

Overall, this paper unifies many problems that are often considered as distinct with specific methods.

**1.2.2 Numerical optimal transport**

In order to quantify the distance between the two measures, we use transportation distances (Monge, 1781; Kantorovich, 1942; Villani, 2003). In our work, we are interested mostly in the semi-discrete setting, where one measure is a density and the other is discrete. In this setting, the most intuitive way to introduce this distance is via Monge’s transportation plan and allocation problems. Given an atomic measure \( \nu \in \mathcal{M}_{a,n} \) and a density \( \mu \), a transport plan \( T \in \mathcal{T}(x,w) \) is a mapping \( T : \Omega \rightarrow \{x[1], \ldots, x[n]\} \) such that \( \forall 1 \leq i \leq n, \mu(T^{-1}(x[i])) = w[i] \). In words, the mass at any point \( x \in \Omega \) is transported to point \( T(x) \). In this setting, the...
W_2\text{ transportation distance} is defined by:

\[ W^2_2(\mu, \nu) = \inf_{T \in T(\mathbf{x}, \mathbf{w})} \int_{\Omega} \|x - T(x)\|_2^2 \mu(x) \, dx, \quad (4) \]

and the minimizing mapping \( T \) describes the optimal way to transfer \( \mu \) to \( \nu \).

Computing the transport plan \( T \) and the distance \( W_2 \) is a challenging optimization problem. In the semi-discrete setting, Aurenhammer et al. (1998) have provided an efficient method based on “power diagram” or “Laguerre diagram”. This framework was recently further improved and analyzed (De Goes et al., 2012; Mériot, 2011; Lévy, 2015; Kitagawa et al., 2016). The idea is to optimize a convex cost function with second-order algorithms. We will make use of those results in the paper, and improve them by stabilizing them while keeping the second-order information.

1.2.3 Numerical projections on curve spaces

Projecting curves on admissible sets is a basic brick for many algorithms. For instance, mobile robots are subject to kinematic constraints (speed and acceleration), while steel wire sculptures have geometric constraints (length, curvature).

While the projection on kinematic constraints is quite easy, due to convexity of the underlying set (Chauffert et al., 2014), we believe that this is the first time projectors on sets defined through intrinsic geometry are designed. Similar ideas have been explored in the past. For instance, curve shortening with mean curvature motion (Evans et al., 1991) is a long-studied problem with multiple applications in computer graphics and image processing (Yezzi, 1998; Moisan, 1998; Tagliasacchi et al., 2012). The proposed algorithms allow exploring new problems such as curve lengthening with curvature constraints.

1.3 Paper outline

The rest of the paper is organized as follows. We first outline the overarching algorithm in Section 2. In Sections 3 and 4, we describe more precisely and study the theoretical guarantees of the algorithms used respectively for computing the Wasserstein distance, and for optimising the positions and weights of the points. We describe the relationships with previous models in Section 5. The algorithms in Sections 3 and 4 are enough for, say, halftoning, but do not handle constraints on the points. In Section 6, we add those constraints and design algorithms to make projections onto curves spaces with bounded speed and acceleration, or bounded length and curvature. Finally some application examples and results are shown in Section 7.

2 Global approach

In this section, we show how to numerically solve the best approximation problem:

\[ \inf_{\nu \in \mathcal{M}} W^2_2(\nu, \mu), \quad (5) \]

where \( \mathcal{M} \) is an arbitrary set of measures supported on \( \Omega = [0, 1]^2 \).

2.1 Discretization

Problem (5) is infinite-dimensional and first needs to be discretized to be solved using a computer. We propose to approximate \( \mathcal{M} \) by a subset \( \mathcal{M}_n \subseteq \mathcal{M}_{a,n} \) of the atomic measures with \( n \) atoms. The idea is to construct \( \mathcal{M}_n \) as

\[ \mathcal{M}_n = \{ \nu(\mathbf{x}, \mathbf{w}), \mathbf{x} \in \mathbf{X}_n, \mathbf{w} \in \mathbf{W}_n \}, \quad (6) \]

where the mapping \( \nu : (\Omega^n \times \Delta_{n-1}) \to \mathcal{M}_{a,n} \) is defined by

\[ \nu(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w[i] \delta_{\mathbf{x}[i]}. \quad (7) \]

The constraint set \( \mathbf{X}_n \subseteq \Omega^n \) describes interactions between points and the set \( \mathbf{W}_n \subseteq \Delta_{n-1} \) describes admissible weights.

Chauffert et al. (2017) have shown that for any subset \( \mathcal{M} \) of the probability measures, it is possible to construct a sequence of approximation spaces \( (\mathcal{M}_n)_{n \in \mathbb{N}} \) of type (6), such that the solution sequence \( (\nu^*_n)_{n \in \mathbb{N}} \) of the discretized problem

\[ \inf_{\nu \in \mathcal{M}_n} W^2_2(\nu, \mu), \quad (8) \]

converges weakly along a subsequence to a global minimizer \( \nu^* \) of the original problem (5). We will show explicit constructions of constraint sets \( \mathbf{X}_n \) and \( \mathbf{W}_n \) for measures supported on curves in Section 6.

The discretized problem (8) can now be rewritten in a form convenient for numerical optimization:

\[ \min_{\mathbf{x} \in \mathbf{X}, \mathbf{w} \in \mathbf{W}} F(\mathbf{x}, \mathbf{w}), \quad (9) \]
where we dropped the index \( n \) to simplify the presentation and where
\[
F(x, w) = \frac{1}{2} W_2^2(\nu(x, w), \mu).
\] (10)

2.2 Overall algorithm

In order to solve (9), we propose to use an alternating minimization algorithm: the problem is minimized alternately in \( x \) with a variable metric projected gradient descent and then in \( w \) with a direct method. Algorithm 1 describes the procedure in detail.

A few remarks are in order. First notice that we are using a variable metric descent algorithm with a metric \( \Sigma_k \succ 0 \). Hence, we need to use a projector defined in this metric by:
\[
\Pi_{\Sigma_k}^X(x_0) := \text{Argmin}_{x \in X} \| x - x_0 \|_{\Sigma_k}^2 \quad \text{with} \quad \| x - x_0 \|_{\Sigma_k}^2 = (\Sigma_k(x - x_0), (x - x_0)).
\]

Second, notice that \( X \) may be nonconvex. Hence, the projector \( \Pi_{\Sigma_k}^X \) on \( X \) might be a point-to-set mapping. In the \( x \)-step, the usual sign \( = \) is therefore replaced by \( \in \).

There are five major difficulties listed below to implement this algorithm:

- **\( \psi \)-step:** How to compute efficiently and robustly \( F(x, w) \)?
- **\( w \)-step:** How to compute \( \text{argmin}_{w \in W} F(x, w) \)?
- **\( x \)-step:** How to compute the gradients \( \nabla_x F \) and the metric \( \Sigma_k \)?
- **\( \Pi \)-step:** How to implement the projector \( \Pi_{\Sigma_k}^X \)?
- **Generally:** How to accelerate the convergence given the specific problem structure?

The next four sections provide answers to these questions.

Note that if there are no constraints like in halftoning or stippling, there is no projection and the \( \Pi \)-step is trivial: \( x_{k+1} = y_{k+1} \).

Algorithm 1 Alternating projected gradient descent to solve (1).

Require: Oracle that computes \( F \) \( \triangleright \psi\)-step.
Require: Projectors \( \Pi_X \) on \( X \).

1: Inputs:
2: Initial guess \( x_0 \)
3: Target measure \( \mu \)
4: Number of iterations \( N_{it} \).
5: Outputs:
6: An approximation \((\hat{x}, \hat{w})\) of the solution of (1).
7: for \( k = 0 \) to \( N_{it} - 1 \) do
8: \( w_{k+1} = \text{argmin}_{w \in W} F(x_k, w) \) \( \triangleright \) \( w\)-step
9: Choose a positive definite matrix \( \Sigma_k \).
10: \( y_{k+1} = x_k - \Sigma_k^{-1} \nabla_x F(x_k, w_{k+1}) \) \( \triangleright \) \( x\)-step
11: \( x_{k+1} \in \Pi_{\Sigma_k}^X(y_{k+1}) \) \( \triangleright \) \( \Pi\)-step
12: end for
13: Set \( \hat{x} = x_{N_{it}} \) and \( \hat{w} = w_{N_{it}} \).

3 Computing the Wassertein distance \( F : \psi\)-step

3.1 Semi-discrete optimal transport

In this paragraph, we review the main existing results about semi-discrete optimal transport and explain how this can be achieved. Finally we provide algorithms that proved to be more efficient than existing approaches.

We work under the following hypotheses.

**Assumption 1.**

- The space \( \Omega \) is a compact convex polyhedron, typically the hypercube.
- \( \mu \) is an absolutely continuous probability density function w.r.t. the Lebesgue measure.
- \( \nu \) is an atomic probability measure supported on \( n \) points.

Let us begin by a theorem stating the uniqueness of the optimal transport plan, which is a special case of Theorem 10.41 in the book by Villani (2008).

**Theorem 1.** Under Assumption 1, there is a unique optimal transportation plan \( T^* \), solution of problem (4).

Before further describing the structure of the optimal transportation plan, let us introduce a fundamental tool from computational geometry (Aurenhammer, 1991).
Definition 1 (Laguerre diagram). Let \( x \in \Omega^n \) denote a set of point locations and \( \psi \in \mathbb{R}^n \) denote a weight vector. The Laguerre cell \( L_i \) is a closed convex polygon set defined as

\[
L_i(\psi, x) = \{ x \in \Omega, \forall 1 \leq j \leq n, j \neq i, \|
\| x - x[i] \|_2^2 - \psi[i] \| \leq \|
\| x - x[j] \|_2^2 - \psi[j] \| \}. \tag{11}
\]

The Laguerre diagram generalizes the Voronoi diagram, since the latter is obtained by taking \( \psi = 0 \) in equation (11).

The set of Laguerre cells partitions \( \Omega \) in polyhedral pieces. It can be computed in \( O(n \log(n)) \) operations for points located in a plane (Aurenhammer, 1991). In our numerical experiments, we make use of the CGAL library to compute them (The CGAL Project, 2016).

We are now ready to describe the structure of the optimal transportation plan \( T^* \), see (Gangbo and McCann, 1996, Example 1.9).

**Theorem 2.** Under Assumption 1, there exists a vector \( \psi^* \in \mathbb{R}^n \), such that

\[
(T^*)^{-1}(x[i]) = L_i(\psi^*, x). \tag{12}
\]

In words, Theorem 2 states that each point \( x[i] \) is spread over a convex polygon. When transporting mass \( \mu \) to \( \nu \), the physical interpretation of \( \psi^*[i] \) is the cost of displacement of the mass at point \( x[i] \). From a numerical point of view, it allows transforming the infinite dimensional problem (4) into the following finite dimensional problem:

\[
W_2(\mu, \nu) = \max_{\psi \in \mathbb{R}^n} g(\psi, x, w), \tag{13}
\]

where

\[
g(\psi, x, w) = \sum_{i=1}^n \int_{L_i(\psi, x)} \left( \| x[i] - x \|^2 - \psi[i] \right) \, d\mu(x) + \sum_{i=1}^n \psi[i] w[i]. \tag{14}
\]

The last problem is to find vector \( \psi^* \). This is the subject of numerous recent papers, and we suggest an original approach in the next section.

3.2 Solving the dual problem

In this paragraph, we focus on the resolution of (13), i.e. computing the transportation distance numerically.

The following proposition summarizes the nice properties of the function \( g \). The derivative formula were already given in various papers (De Goes et al., 2012; Lévy, 2015). We refer the interested reader to the work by Kitagawa et al. (2016) for a rigorous proof of the result.

**Proposition 1.** Under Assumption 1, function \( g \) is concave with respect to variable \( \psi \), it is also twice differentiable and its derivatives are given by:

\[
\frac{\partial g}{\partial \psi_i} = w[i] - \mu(L_i(\psi, x)), \tag{15}
\]

\[
\frac{\partial^2 g}{\partial \psi_i \partial \psi_j} = \int_{\partial L_i(\psi, x) \cap \partial L_j(\psi, x)} \frac{d\mu(x)}{\| |x[i] - x[j]| \|} \text{ if } i \neq j. \tag{16}
\]

The formula for the diagonal term \( \frac{\partial^2 g}{\partial \psi_i \partial \psi_i} \) is given by the closure relation

\[
\forall 1 \leq i \leq n, \sum_{j=1}^n \frac{\partial^2 g}{\partial \psi_i \partial \psi_j} = 0. \tag{17}
\]

Proposition 1 suggests a way to compute the optimal vector \( \psi^* \): well-chosen first- or second-order ascent methods should converge to the global maximizer of problem (13), since the problem is concave. Many methods have been proposed in the literature, with the latest references providing strong convergence guarantees (Aurenhammer et al., 1998; De Goes et al., 2012; Mérigot, 2011; Lévy, 2015; Kitagawa et al., 2016). This may give the false impression that the problem has been fully resolved: in practice the conditions guaranteeing convergence are often unmet. For instance, it is well-known that the convergence of first-order methods depends strongly on the Lipschitz constant of the gradient (Nesterov, 2013b, Thm 2.1.7). Unfortunately, this Lipschitz constant may blow up depending on the geometry of the point set \( x \) and the regularity of the density \( \rho \), see Remark 1. On the other hand, second-order methods heavily depend on the Hölder regularity of \( g \) (Jarre and Toint, 2016; Grapiglia and Nesterov, 2017). Unfortunately, it can be shown that \( g \) is Hölder with respect to \( \psi \) only under certain circumstances. In particular, the mass of the Laguerre cells \( \mu(L_i(\psi, x)) \) should not vanish (Kitagawa et al., 2016, Remark 4.2). Hence, only first-order methods should be used in the early step of an optimization algorithm, and the initial guess should be well-chosen due to slow convergence. Then, second-order methods should be the method of choice. In this paper, we make use of a trust-region
replace by integrals of polynomials along the edges of the Laguerre diagram by using Green’s formula. Hence computing the cost function, the Hessian or the gradient all boil down to computing edge integrals.

Then, since the underlying density is piecewise polynomial, it is easy to see that only the first moments of the measure µ along the edges are needed to compute all formula. We pre-evaluate the moments by using exact quadrature formulas and then use linear combinations of the moments to finish the evaluation.

To the best of our knowledge, this is a novel lightweight procedure. It significantly speeds up the calculations compared to former works (Mérigot, 2011; De Goes et al., 2012), which enables discretization of the density ρ over an arbitrary 3D mesh. After finishing this paper, we realized that the idea of using Green formulas was already suggested by Xin et al. (2016), although not implemented. It is to be noted that this idea is particularly well suited to Cartesian grid discretization of the target density ρ. Indeed in this case, we take advantage of the fact that the intersection of the Laguerre cells and the grid can be computed analytically without search on the mesh.

4 Optimizing the weights and computing the gradient wrto the positions : w and x steps

4.1 Computing the optimal weights

In this section, we focus on the numerical resolution of the following subproblem

$$\arg\min_{w \in W} F(x, w).$$

4.1.1 Totally constrained w

When W = {w} is reduced to a singleton, the solution of (18) is obviously given by $w^* = w$.

4.1.2 Unconstrained minimization in w

When W is the simplex, the unconstrained minimization problem (18) can be solved analytically.

Proposition 2. If $W = \Delta_n - 1$, the solution $w^*$ of (18) is given for all $1 \leq i \leq n$ by

$$w^*[i] = \mu(L_i(0, x)),$$
that is the volume (w.r.t. the measure \( \mu \)) of the \( i \)-th Laguerre cell with zero cost \( \psi \), i.e. the \( i \)-th Voronoi cell.

Proof. In expression (14), the vector \( \psi \) can be interpreted as a Lagrange multiplier for the constraint

\[ \mu(T^{-1}(x[i])) = w[i]. \]

Since the minimization in \( w \) removes this constraint, the Lagrange multiplier might be set to zero. \( \square \)

4.2 Gradient \( \nabla_x F \) and the metric \( \Sigma_k \)

The following proposition allows to compute \( \nabla_x F(x, w) \). It can be found in (De Goes et al., 2012) for instance.

Proposition 3. Let \( \psi^* \) denote the maximizer of (13). The gradient \( \nabla_x F(x, w) \) is given by the following formula.

\[
\frac{\partial F(x, w)}{\partial x[i]} = w[i] (x[i] - b[i]) \tag{20}
\]

where \( b[i] \) is the barycenter of the \( i \)-th Laguerre cell \( L_i(\psi^*, x) \):

\[
b[i] = b(x)[i] = \frac{\int_{L_i(\psi^*, x)} x d\mu(x)}{\int_{L_i(\psi^*, x)} d\mu(x)}. \tag{21}
\]

Now, we discuss how to choose the metric (\( \Sigma_k \)) in Algorithm 1. This choice is critical but complex and we will only provide a good heuristic here. Notice that line searches should be used with caution here, since an evaluation of the cost function requires itself the resolution of a complicated convex programming problem described in paragraph 3.2.

Let us first recollect a few typical theorems about the convergence of first-order optimization algorithms (Nesterov, 2013a; Attouch et al., 2013).

Theorem 3. Let \( X \subset \mathbb{R}^n \) denote a closed set, \( \Sigma \in \mathbb{R}^{n \times n} \) denote a positive definite matrix and \( f : \mathbb{R}^n \to \mathbb{R} \) denote a \( C^1 \) function with Lipschitz continuous gradient:

\[
\forall (x_1, x_2) \in \mathbb{R}^n \times \mathbb{R}^n, \| \nabla f(x_1) - \nabla f(x_2) \|_{\Sigma^{-1}} \leq L \| x_1 - x_2 \|_{\Sigma}. \tag{22}
\]

Consider the following projected gradient descent

\[
x_{k+1} \in \Pi_X^\Sigma \left( x_k - \frac{1}{L} \Sigma^{-1} \nabla f(x_k) \right). \tag{23}
\]

Then the sequence \((x_k)_{k \in \mathbb{N}}\) converges along subsequences and satisfies

\[
\| x_{k+1} - x_k \|_2 = O \left( \frac{1}{k} \right), \tag{24}
\]

under either of the following additional assumptions.

- \( X \) is convex and compact.
- \( X = \mathbb{R}^d \), and \( f \) is coercive.
- \( f + \mathbb{1}_X \) is Kurdyka-Lojasiewicz, where \( \mathbb{1}_X \) is the indicator function of \( X \).

This theorem basically states that for a well-chosen constant metric dependent on the global Lipschitz constant, two consecutive iterates will vanish with the iterates. The step-size \( \frac{1}{L} \) together with the Lipschitz assumption is the key here. In particular, the sequence \((x_k)_{k \in \mathbb{N}}\) can diverge to \( \infty \) whenever the step-size is larger than \( \frac{1}{L} \). The last Kurdyka-Lojasiewicz assumption is very general. Unfortunately, it is unclear whether the Wasserstein distance in the semi-discrete setting satisfies it.

Theorem 3 shows that it is critical to evaluate the Lipschitz constant of \( \nabla_x F \). By equation (20), we need to evaluate the variations of the Laguerre cells centers of mass with respect to the Dirac mass locations. Unfortunately, the Lipschitz constant can be arbitrarily large for sites \( x \) in arbitrary position, or singular densities \( \rho \), see Remark 1. Hence, we can only hope for a local result describing the Lipschitz constant. Du et al. (1999) have studied a very closely related question, namely the variations of Laguerre cells with respect to the positions \( x \). This result together with Theorem 3 yields the following result.

Proposition 4. Assume that \( \nabla_x F(x^*) = 0 \), i.e. that \( x^*[i] = b(x^*)[i] \) for all \( i \). Set

\[
\Sigma = \text{diag}(\mu(L_i(\psi^*, x^*)))_{1 \leq i \leq n}). \tag{25}
\]

Then the mapping \( b(x) \) is locally Lipschitz (see definition (22)) at \( x^* \) with constant 1.

This proposition suggests that a variable metric gradient descent with a metric depending only on \( \mu(L_i(\psi^*, x^*)) \) may perform well in practice for \( X = \Omega^n \)

\[1\] We skip the technical definition of Kurdyka-Lojasiewicz functions and refer to the paper by Attouch et al. (2013) for more details.
5 Links with other models

5.1 Special cases of the framework

5.1.1 Lloyd’s algorithm

Lloyd’s (1982) algorithm is well-known to be a specific solver for problem (5), with $X = \Omega$ and $W = \Delta_{n-1}$, i.e. to solve the quantization problem with variable weights. We refer to the excellent review by Du et al. (1999) for more details. It is easy to check that Lloyd’s algorithm is just a special case of Algorithm 1, with the specific choice of metric

$$\Sigma_k = \text{diag}(\mu(\mathcal{L}_i(0, x))).$$

5.1.2 Blue noise through optimal transport

De Goes et al. (2012) has proposed to perform stippling by using optimal transport distance. This application can be cast as a special case of problem (5), with $X = \Omega$ and $W = \{\frac{1}{n}\}$. The algorithm proposed therein is also a special case of algorithm 1 with

$$\Sigma_k = \text{diag}(\mu(\mathcal{L}_i(\phi^*(x), x))) = \frac{1}{n}$$

and the step-size $\tau_k$ is optimized through a line search. Note however the extra cost of applying a line-search might not worth the effort, since a single function evaluation requires solving the dual problem (13).

5.2 Comparison with electrostatic halftoning

An alternative to the $W_2$ distance was proposed, implemented and studied (Schmaltz et al., 2010; Teuber et al., 2011; Fornasier et al., 2013; Chauffert et al., 2017). Namely, the distance $D$ in (1) is defined by

$$D(\nu, \mu) = \frac{1}{2}\|h \ast (\nu - \mu)\|_{L^2(\Omega)}^2,$$

(27)

where $h$ is a smooth convolution kernel and $\ast$ denotes the convolution product. This distance can be interpreted intuitively as follows: the measures are first blurred by a regularizing kernel to map them in $L^2(\Omega)$ and then compared using a simple $L^2$ distance.

In some cases, the two approaches are actually quite similar from a theoretical point of view. Indeed, it can be shown that the two distances are strongly equivalent under certain assumptions (Peyre, 2016).

The two approaches however differ significantly from a numerical point of view. Table 1 provides a quick summary of the differences between the two approaches. We detail this table below.

- The theory of optimization is significantly harder in the case of optimal transport since it is based on a subtle mix between first and second order methods.
- The convolution-based algorithms require the use of methods from applied harmonic analysis dedicated to particle simulations such as fast multiple methods (FMM) (Greengard and Rokhlin, 1987) or non uniform Fast Fourier Transforms (NUFFT) (Potts and Steidl, 2003). On their side, the optimal transport based approaches require the use of computational geometry tools such as Voronoi or Laguerre diagrams.
- The examples provided here are only two dimensional. Many applications in computer graphics require dealing with 3D problems or larger dimensional problems (e.g. clustering problems). In that case, the numerical complexity of convolution based problems seems much better controlled: it is only linear in the dimension $d$ (i.e. $O(dn \log(n))$), while the exact computation of Laguerre diagrams requires in $O([n^{d/2}])$ operations. Hence, for a large number of particles, the approach suggested here is mostly restricted to $d = 2$.
- In terms of computational speed for 2D applications, we observed that the optimal transport based approach was usually between 1 and 2 orders of magnitude faster.
- Finally, we did not observe significant differences in terms of approximation quality from a perceptual point of view.
Table 1: A comparison between convolution and optimal transport based approximation of measures.

6 Projections on curves spaces

In this section, we detail a numerical algorithm to evaluate the projector \( \Pi \mathbf{X} \), for spaces of curves with kinematic or geometric constraints.

6.1 Discrete curves

A discrete curve is a set of points \( \mathbf{x} \in \Omega^n \) with constraints on the distance between successive points. Let

\[
A^a_n : \mathbf{x} \mapsto \left( \begin{array}{c} x[2] - x[1] \\ \vdots \\ x[n] - x[n - 1] \\ x[1] - x[n] \end{array} \right)
\]

and

\[
A^b_n : \mathbf{x} \mapsto \left( \begin{array}{c} x[2] - x[1] \\ x[3] - x[2] \\ \vdots \\ x[n] - x[n - 1] \end{array} \right)
\]

denote the discrete first order derivatives operators with or without circular boundary conditions. From hereon, we let \( A_1 \) denote any of the two operators. In order to control the distance between two neighboring points, we will consider two types of constraints: kinematic ones and geometrical ones.

6.1.1 Kinematic constraints

Kinematic constraints typically apply to vehicles: a car for instance has a bounded speed and acceleration. Bounded speeds can be encoded through inequalities of type

\[
\| (A_1 \mathbf{x})[i] \|_2 \leq \alpha_1, \forall i. \tag{28}
\]

Similarly, by letting \( A_2 \) denote a discrete second order derivative, which can for instance be defined by \( A_2 = \partial_1^2 A_1 \), we may enforce bounded acceleration through

\[
\| (A_2 \mathbf{x})[i] \|_2 \leq \alpha_2, \forall i. \tag{29}
\]

The set \( \mathbf{X} \) is then defined by

\[
\mathbf{X} = \{ \mathbf{x} \in \Omega^n, \| A_1 \mathbf{x}\|_{\infty,2} \leq \alpha_1, \| A_2 \mathbf{x}\|_{\infty,2} \leq \alpha_2 \}, \tag{30}
\]

where, for \( \mathbf{y} = (y[1], \ldots, y[n]) \), \( \| \mathbf{y} \|_{\infty,p} = \sup_{1 \leq i \leq n} |y[i]|_p \).

6.1.2 Geometrical constraints

Geometrical constraints refer to intrinsic features of a curve such as its length or curvature. In order to control those quantities using differential operators, we need to parameterize the curve with its arc length. Let \( s : [0, T] \to \mathbb{R}^2 \) denote a \( C^2 \) curve with arc length parameterization, i.e. \( \| s(t) \|_2 = 1, \forall t \in [0, T] \). Its length is then equal to \( T \). Its curvature at time \( t \in [0, T] \) is equal to \( \kappa(t) = \| s'(t) \|_2 \).

In the discrete setting, constant speed parameterization can be enforced by imposing

\[
\| (A_1 \mathbf{x})[i] \|_2 = \alpha_1, \forall i. \tag{31}
\]

The total length of the discrete curve is then equal to \( (n - 1)\alpha_1 \).

Similarly, when (31) is satisfied, discrete curvature constraints can be captured by inequalities of type

\[
\| (A_2 \mathbf{x})[i] \|_2 \leq \alpha_2, \forall i. \tag{32}
\]

Indeed, at a index \( 2 \leq i \leq n - 1 \), we get:

\[
\| (A_2 \mathbf{x})[i] \|_2^2 = \| (\mathbf{x}[i] - \mathbf{x}[i - 1]) - (\mathbf{x}[i + 1] - \mathbf{x}[i]) \|_2^2
\]

\[
= \| (\mathbf{x}[i] - \mathbf{x}[i - 1]) \|_2^2 + \| (\mathbf{x}[i + 1] - \mathbf{x}[i]) \|_2^2 - 2 \langle \mathbf{x}[i] - \mathbf{x}[i - 1], (\mathbf{x}[i + 1] - \mathbf{x}[i]) \rangle
\]

\[
= 2\alpha^2_1 (1 - \cos(\theta_i)),
\]

where \( \theta_i = \angle (\mathbf{x}[i] - \mathbf{x}[i - 1], \mathbf{x}[i + 1] - \mathbf{x}[i]) \) is the angle between successive segments of the curve. Hence, by imposing (31) and (32), the angle \( \theta_i \) satisfies

\[
\theta_i \leq \arccos \left( 1 - \frac{\alpha^2_2}{2\alpha^2_1} \right). \tag{33}
\]

In order to fix the length and bound the curvature, we may thus choose the set \( \mathbf{X} \) as

\[
\mathbf{X} = \{ \mathbf{x} \in \Omega^n, \| (A_1 \mathbf{x})[i] \|_2 = \alpha_1, \| A_2 \mathbf{x}\|_{\infty,2} \leq \alpha_2 \}. \tag{34}
\]

Let us note already that this set is nonconvex, while (30) was convex.
6.1.3 Additional linear constraints

In applications, it may be necessary to impose other constraints such as passing at a specific location at a given time, closing the curve with \( x_1 = x_n \) or having a specified mean value. All those constraints are of form

\[
Bx = b,
\]

where \( B \in \mathbb{R}^{p \times 2n} \) and \( b \in \mathbb{R}^p \) are a matrix and vector describing the \( p \) linear constraints.

6.1.4 Summary

In this paper, we will consider discrete spaces of curves \( X \) defined as follows:

\[
X = \{ x \text{ such that } A_i x \in Y_i, 1 \leq i \leq m, Bx = b \},
\]

(36)

The operators \( A_i \) may be arbitrary, but in this paper, we will focus on differential operators of different orders. The set \( Y_i \) describes the admissible set for the \( i \)-th constraint. For instance, to impose a bounded speed (28), we may choose

\[
Y_1 = \{ y \in \mathbb{R}^{n \times 2}, \| y_i \|_2 \leq \alpha_1, \forall i \}.
\]

(37)

In all the paper, the set of admissible weights \( W \) will be either the constant \( \{1/n\} \) or the canonical simplex \( \Delta_{n-1} \).

6.2 Numerical projectors

The Euclidean projector \( \Pi_X : \mathbb{R}^n \rightarrow X \) is defined for all \( z \in \Omega^n \) by

\[
\Pi_X(z) = \underset{x \in X}{\text{Argmin}} \frac{1}{2} \| x - z \|_2^2
\]

\[
= \underset{A_i x \in Y_i, 1 \leq i \leq m}{\operatorname{Argmin}} \frac{1}{2} \| x - z \|_2^2
\]

(38)

When \( X \) is convex, \( \Pi_X(z) \) is a singleton. When it is not, there exists \( z \) such that \( \Pi_X(z) \) contains more than one element. The objective of this section is to design an algorithm to find critical points of (38).

The specific structure of (38) suggests using splitting based methods (Combettes and Pesquet, 2011), able to deal with multiple constraints and linear operators. The sparse structure of differential operator makes the Alternating Direction Method of Multipliers (ADMM; Glowinski, 2014), particularly suited for this problem. Let us turn (38) into a form suitable for the ADMM.

Let \( \gamma_1, \ldots, \gamma_m \) denote positive reals used as preconditioners. Define

\[
A = \begin{pmatrix} \gamma_1 A_1 \\ \vdots \\ \gamma_m A_m \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}
\]

(39)

and

\[
Y = \gamma_1 Y_1 \times \ldots \times \gamma_m Y_m.
\]

(40)

Problem (38) then becomes

\[
\Pi_X(z) = \underset{Bx = b}{\underset{x \in Y}{\text{Argmin}}} \frac{1}{2} \| x - z \|_2^2
\]

\[
= \underset{A_i x \in Y_i}{\text{Argmin}} \frac{1}{2} \| x - z \|_2^2
\]

(41)

where \( f_1(x) = \frac{1}{2} \| x - z \|_2^2 + \iota_Y(x), f_2(y) = \iota_Y(y), L = \{ x, Bx = b \} \) denotes the set of linear constraints and the indicator \( \iota_Y \) of \( Y \) is defined by:

\[
\iota_Y(y) = \begin{cases} 0 & \text{if } y \in Y, \\ +\infty & \text{otherwise}. \end{cases}
\]

(42)

The ADMM for solving (41) is given in Algorithm 2. Specialized to our problem, it yields Algorithm 3. The linear system can be solved with a linear conjugate gradient descent.

**Algorithm 2** Generic ADMM.

**Inputs:**

- functions \( f_1 \) and \( f_2 \), matrix \( A \), initial guess \( (x_0, \lambda_0) \), parameter \( \beta > 0 \).

1. while Stopping criterion not met do

   \[
   y_{k+1} = \underset{y}{\text{Argmin}} f_2(y) + \frac{\beta}{2} \| Ax - y_{k+1} + \lambda_k \|_2^2.
   \]

   \[
   x_{k+1} = \underset{x}{\text{Argmin}} f_1(x) + \frac{\beta}{2} \| Ax - y_{k+1} + \lambda_k \|_2^2.
   \]

   \[
   \lambda_{k+1} = \lambda_k + Ax_{k+1} - y_{k+1}.
   \]

2. end while

**Convergence issues** The convergence and rate of convergence of the ADMM is a complex issue that depends on the properties of functions \( f_1 \) and \( f_2 \) and on the linear transform \( A \). In the convex setting (30), the sequence \( (x_k) \) converges to \( \Pi_X(z) \) linearly (see Corollary 2 of Giselsson and Boyd (2017)). The behavior in a nonconvex setting (34) is still mostly open despite recent advances in Li and Pong (2015). Nevertheless, we report that we observed convergence empirically towards critical points of Problem (38).
Algorithm 3 ADMM to solve the projection problem.

Inputs:
Vector to project \( z \), initial guess \((x_0, \lambda_0)\), matrices \( A \) and \( B \), projector \((\Pi_Y)\), \( \beta > 0 \).

1: while Stopping criterion not met do
   \[ y_{k+1} = \Pi_Y (Ax_k + \lambda_k). \]
   Solve
   \[
   \begin{bmatrix}
   \beta A^T A + I & B^T \\
   B & 0
   \end{bmatrix}
   \begin{bmatrix}
   x_{k+1} \\
   \mu
   \end{bmatrix}
   = \begin{bmatrix}
   \beta A^T (y_{k+1} - \lambda_k) + z \\
   b
   \end{bmatrix}.
   \]
   \[ \lambda_{k+1} = \lambda_k + Ax_{k+1} - y_{k+1}. \]
2: end while

Choosing the coefficients \( \beta \) and \((\gamma_i)\) Despite recent advances (Nishihara et al., 2015), a theory to select good values of \( \beta \) and \((\gamma_i)\) still seems lacking. In this paper, we simply set \( \gamma_i = \|A_i\|_2 \), the spectral norm of \( A_i \). In practice, it turns out that this choice leads to stable results. The parameter \( \beta \) is set manually to obtain a good empirical behavior. Notice that for a given application, it can be tuned once for all.

6.3 Numerical examples

To illustrate the proposed method, we project the silhouette of a cat onto spaces of curves with fixed length and bounded curvature in Fig. 3. In the middle, we see how the algorithm simplifies the curve by making it smaller and smoother. On the right, we see how the method is able to make the curve longer, by adding loops of bounded curvature, while still keeping the initial cat’s shape.

6.4 Multi-resolution implementation

When \( X \) is a set of curves, the solution of (9) can be found more efficiently by using a multi-resolution approach. Instead of optimizing all the points simultaneously, it is possible to only optimize a down-sampled curve, allowing to get cheap warm start initialization for the next resolution.

In our implementation, we use a dyadic scaling. We up-sample the curve by adding mid-points in between consecutive samples. The weights from one resolution to the next are simply divided by a factor of 2.

7 Applications

7.1 Non Photorealistic Rendering with curves

In the following subsections we exhibit a few rendering results of images using different types of measures sets \( \mathcal{M} \).

7.1.1 Gray-scale images

A direct application of the proposed algorithm allows to approximate an arbitrary image with measures supported on curves. An example is displayed in Fig. 4 with curves satisfying different kinematic constraints.

7.1.2 Color images

There are different ways to render color images with the proposed idea. Wei (2010) and Chauffert et al. (2015) provide two different examples. In this section, we propose a simple alternative idea to give a color to the dots or curves. Given a target vectorial density \( \rho = (\rho_R, \rho_G, \rho_B) : \Omega \to [0, 1]^3 \), the algorithm we propose simply reads as follows:

1) We first construct a gray level image defined by:
   \[ \bar{\rho} = (\rho_R + \rho_G + \rho_B)/3. \] (43)

2) Then, we project the density \( \bar{\rho} \) onto the set of constraints \( \mathcal{M} \) with Algorithm 1. This yields a sequence of points \( x \in \Omega^n \).

3) Then, for each point \( x[i] \) of the discretized measure, we select a color as \( \rho(x[i]) \).

We use only saturated colors, explaining the division in step 3). The parallel for gray-scale images, is that we represent stippling results with disks taking only the maximal intensity. Then, the mean in step 1) is used to attract the curve towards the regions of high luminance of the image. An example of result of the proposed algorithm is shown in Figure 5.

7.1.3 Dynamic examples

The codes can also be used to approximate videos. The principle is simple: first we approximate the first sequence of the frame with our projection algorithm starting from an arbitrary initial guess. Then, the other
frames are obtained with the projection algorithm, taking as an initial guess, the result of the previous iteration. This ensures some continuity of the dots or curves between consecutive frames. Some videos are given in the supplementary material.

7.2 Path planning

In this section, we provide two applications of the proposed algorithm to path planning problems.

7.2.1 Videodrone

Drone surveillance is an application with increasing interest from cities, companies or even private individuals. In this paragraph, we show that the proposed algorithms can be used to plan the drone trajectories for surveillance applications. We use the criminal data provided by the City of Philadelphia (2017) to create a density map of crime in Philadelphia, see Fig. 6a. We give different weights to different types of crimes. By minimizing (1), we can design an optimal path, in the sense that it satisfies the kinematic constraints of the drone and passes close to dangerous spots more often than in the remaining locations. In this example, we impose a bounded speed, a maximal yaw angular velocity and also to pass at a given location at a given time to recharge the drone to satisfy autonomy constraints.

7.2.2 Railway design

In this example, we give an example of application of railway design. Assuming that trains drive at constant speed, it is necessary to bound the curvature of the railway. In addition, we would like the train to be pass nearby the most populated areas and to avoid some lo-
Figure 5: Examples of color curvling, 512k, $\approx 24^\circ$.

Figure 6: The data super imposed on a map of Philadelphia. A possible drone trajectory made. In this example, the drone passes 4 times to its recharging location, explaining the different colors of the trajectory. In this example, the trajectory was discretized with 8k points and optimized in 30”.

7.2.3 Sampling in MRI

Like Boyer et al. (2016), we propose to generate compressive sampling schemes in MRI (Magnetic Resonance Imaging), using the proposed algorithm.

In MRI, images are probed indirectly through their Fourier transform. Fourier transform values are sampled along curves with bounded speed and bounded ac-
acceleration, which exactly corresponds to the set of constraints defined in (30). The latest compressed sensing theories suggest that a good way of subsampling the Fourier domain, consists in drawing points independently at random according to a certain distribution \( \mu \), that depends on the image sparsity structure in the wavelet domain (Boyer et al., 2016; Adcock et al., 2017). Unfortunately, this strategy is impractical in MRI due to physical constraints. To simulate such a sampling scheme, we therefore propose to project \( \mu \) onto the set of admissible trajectories.

Let \( u : [0, 1]^2 \rightarrow \mathbb{R} \) denote a magnetic resonance image. The sampling process yields a set of Fourier transform values \( y[i] = \hat{u}(x[i]) \). Given this set of values, the image is then reconstructed by solving a nonlinear convex programming problem:

\[
\min_{v : v|x=y} \frac{1}{2} \| \hat{v}(x) - y \|_2^2 + \lambda \| \Psi u \|_1,
\]

where \( \Psi \) is a linear sparsifying transform, such as a redundant wavelet transform.

Figure 8: Example of sampling scheme generation and image reconstruction in MRI. The target density \( \mu \) is shown in 8a. The sampling scheme generated by our algorithm is shown in 8b. The background shows the Fourier transform of \( u \) in log-scale. It contains one fourth of the total number of Fourier transform values. The true image and the reconstructed one are shown in Fig. 8c and 8d.

Acknowledgments
The authors wish to thank Alban Gossard warmly for his help in designing numerical integration procedures.

References

Akleman, E., Xing, Q., Garigipati, P., Taubin, G.,


