Numer. Math. (2010) 116:357–381
DOI 10.1007/s00211-010-0305-8

Numerische
Mathematik

Derivatives with respect to metrics and applications: subgradient marching algorithm

F. Benmansour · G. Carlier · G. Peyré ·
F. Santambrogio

Received: 16 January 2009 / Revised: 29 March 2010 / Published online: 13 May 2010

Abstract This paper introduces a subgradient descent algorithm to compute a Riemannian metric that minimizes an energy involving geodesic distances. The heart of the method is the Subgradient Marching Algorithm to compute the derivative of the geodesic distance with respect to the metric. The geodesic distance being a concave function of the metric, this algorithm computes an element of the subgradient in $O(N^2 \log(N))$ operations on a discrete grid of N points. It performs a front propagation that computes a subgradient of a discrete geodesic distance. We show applications to landscape modeling and to traffic congestion. Both applications require the maximization of geodesic distances under convex constraints, and are solved by subgradient descent computed with our Subgradient Marching. We also show application to the inversion of travel time tomography, where the recovered metric is the local minimum of a non-convex variational problem involving geodesic distances.

Mathematics Subject Classification (2000) Primary 65K10 · 65M32 · 65D25; Secondary 91-08 · 35F21

F. Benmansour (✉) · G. Carlier · G. Peyré · F. Santambrogio
CEREMADE, UMR CNRS 7534, Université Paris-Dauphine,
Pl. de Lattre de Tassigny, 75775 Paris Cedex 16, France
e-mail: benmansour@ceremade.dauphine.fr

G. Carlier
e-mail: carlier@ceremade.dauphine.fr

G. Peyré
e-mail: peyre@ceremade.dauphine.fr

F. Santambrogio
e-mail: filippo@ceremade.dauphine.fr

F. Benmansour
Ecole Polytechnique Fédérale de Lausanne (EPFL),
Computer Vision Laboratory, CH-1015 Lausanne, Switzerland
e-mail: fethallah.benmansour@epfl.ch

1 Introduction

This paper is concerned with variational problems involving geodesic distances. We aim to find a Riemannian metric that optimizes an energy taking into account pairwise geodesic distances according to the metric.

The optimization of the metric is obtained using a gradient descent scheme, and the main contribution of the paper is an algorithm to compute the gradient of the geodesic distance according to the metric.

1.1 Variational problems with geodesic distances

An isotropic Riemannian metric ξ on a domain $\Omega \subset \mathbb{R}^d$ defines a weight $\xi(x)$ that penalizes a curve $\gamma(t)$ passing through a point $x = \gamma(t) \in \Omega$.

This paper considers the optimization of a metric ξ that solves general variational problems of the form

$$\min_{\xi \in \mathcal{C}} \mathcal{E}(\xi) = \sum_{s,t} \mathcal{E}_{s,t}(d_\xi(x_s, x_t)) + J(\xi). \tag{1.1}$$

where $\mathcal{C}$ is a convex set of constraints, for each pair of points $x_s, x_t \in \Omega$, $\mathcal{E}_{s,t}$ is an interaction functional, J is a convex regularization functional and d_ξ is the geodesic distance according to ξ.

The geodesic distance is the minimal length of rectifiable curves joining two points $x_s, x_t \in \Omega$

$$d_\xi(x_s, x_t) = \min_{\gamma(0)=x_s, \gamma(1)=x_t} L_\xi(\gamma). \tag{1.2}$$

where the length of a curve is defined as

$$L_\xi(\gamma) = \int_0^1 |\gamma'(t)| \xi(\gamma(t)) dt. \tag{1.3}$$

The mapping $\xi \mapsto d_\xi(x_s, x_t)$ is concave, as the minimum (1.2) of linear functions of ξ. The energy $\mathcal{E}$ is thus convex as long as each interaction functional $\mathcal{E}_{s,t}$ is convex and non-increasing. In this paper we consider two particular instances of (1.1) where the energy is convex, in which cases we find a global minimizer using our method. We also consider a non-convex problem for which we compute a local minimizer of $\mathcal{E}$.

1.2 Previous works

Geodesic distance computation. The estimation of geodesic distances $d_\xi(x_s, x_t)$ has been intensively studied in numerical analysis and can be approximated on a discrete grid of N points with the Fast Marching Method of Sethian [19], and Tsitsiklis [22] in $O(N\log(N))$ operations. This algorithm has opened the door to many applications in computer vision and medical image analysis where the minimal geodesic curves extract image features, see for instance [11,12,19]. Section 2 recalls the basics of the discretization of geodesic distances and Sect. 2.2 details the front propagation procedure underlying the Fast Marching method.

The optimization of the metric ξ according to a variational problem such as (1.1) is much less studied than the computation of geodesic distances. It is however an important problem in some specific fields, such as landscape design, traffic congestion and seismic imaging. In these applications, the metric ξ is optimized to meet certain criteria, or is recovered by optimization from a few geodesic distance measures.

We now describe some applications where such optimization problems naturally arise.

Convex geodesic distance maximization. The design of a landscape in a domain $\Omega \subset \mathbb{R}^d$ corresponds to the optimization of a metric $\xi(x)$ that describes locally the difficulty of passing through some point $x \in \Omega$. Buttazzo et al. consider in [8] a design criterion that corresponds to the maximization of geodesic distances between landmark points $\{x_s\}_{s=0}^{P-1}$, so that the interaction functionals in (1.1) are

$$\mathcal{E}_{s,t}(d) = -w_{s,t}d \tag{1.4}$$

where $w_{s,t} \geq 0$ are weights describing the interaction between the landmarks. In this application, we do not consider any regularization J. This criterion models agents located at the points $\{x_s\}_s$ and that are free to modify the landscape in order to defend themselves optimally from the other agents.

A continuous formulation of the problem is studied in [8] that proves existence of optimal solutions for certain sets of constraints $\mathcal{C}$. Section 4.1 shows numerical examples computed using our subgradient Marching algorithm.

Convex traffic congestion problem. A Wardrop equilibrium [23] defines a traffic density between points $\{x_s\}_{s=0}^{P-1}$ such that agents travel along geodesics for a metric ξ that reflects penalization of movements at points with high traffic density due to congestion effects.

A continuous generalization of this notion of equilibrium is proposed in [9]. It computes an equilibrium metric by solving a variational problem of the form (1.1) with linear interaction functionals (1.4) and a regularization of the form

$$J(\xi) = \int\limits_{\Omega} G(x, \xi(x))dx. \tag{1.5}$$

for a function G that is convex in ξ. We refer to [1] for more details about this variational formulation and how the metric ξ and the traffic intensity are related. Section 4.2 shows some numerical examples solved using our subgradient Marching algorithm.

Non-convex geodesic inversion problems. Seismic imaging computes an approximation of the underground from few surfaces measurements [10]. This corresponds to an ill-posed inverse problem that is regularized using smoothness prior information about the ground and simplifying assumptions about wave propagation.

For a pair (x_s, x_t) of emitter and receiver, denoted as $(s, t) \in \Gamma$, discarding multiple reflexions, the first arrival time of a pressure wave corresponds to the geodesic distance

$$\forall (s, t) \in \Gamma, \quad d_{s,t} = d_{\xi^0}(x_s, x_t) \tag{1.6}$$

for some unknown Riemannian metric ξ^0 that reflects the properties of the underground.

Travel time tomography recovers an approximation ξ of ξ^0 from few first time arrivals $(d_{s,t})_{(s,t)\in\Gamma}$. A least square recovery of ξ^0 involves the optimization of the geodesic distance through a non-convex variational problem of the form (1.1) with interaction functionals

$$\mathcal{E}_{s,t}(d) = \begin{cases} (d - d_{s,t})^2 & \text{if} \quad (s, t) \in \Gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{1.7}$$

The resulting energy $\mathcal{E}(\xi)$ is non convex, and optimization schemes compute a local minimizer of the energy.

Most methods perform an approximate recovery using ray tracing, that necessitates to compute many rays to cover the whole domain [2,3]. Eulerian methods compute the travel time distance by solving PDEs on a discrete grid, see [10] for a related tomography problem. An Eulerian travel time tomography method has been developed by Leung and Qian in [16]. It performs a regularized gradient descent of an energy similar to (1.1). Authors of [16] propose to compute the gradient of the functional using an adjoint state method, that requires, at each step of the descent, two computations of geodesic distances.

Section 4.3 proposes an alternative approach to compute the gradient of the functional to minimize. It might be slower than the method of [16] that only requires the evaluation of geodesic distances. Our method is however more general since it allows to tackle arbitrary energies involving geodesic distances. It might also be numerically more precise since we compute exactly the gradient of a discrete geodesic distance, while [16] discretizes a gradient defined for continuous PDEs. Our method also correctly accounts for the non-differentiability of the distance with respect to the metric by estimating an element of the subgradient of the functional.

It is however beyond the scope of this paper to compare the numerical complexity and precision of travel time tomography methods. Section 4.3 shows some examples of travel time tomography inversion computed using our subgradient Marching algorithm.

Non-convex geodesic shape regularization. Shape analysis in computer vision tackles the problem of comparing objects represented as surfaces or more general metric spaces.

A geodesic matching between two surfaces is obtained in [7] by minimizing an energy similar to (1.1) for interactions (1.7), but with respect to the sample locations $\{x_s\}_s$ for a fixed metric ξ that represents the geometry of the surface to be matched.

This approach is extended into a geodesic regularization in [13] by performing the minimization of (1.1) with respect to the metric ξ. In this work, the metric is represented as a triangulated 3D surface. This corresponds to an extension of our Subgradient Marching algorithm from a regular grid to a triangular grid. This extension makes use of the extension of the 2D grid Fast Marching [19] to triangulated meshes [15], see also [6,21].

1.3 Discrete subgradient descent

We propose a projected subgradient algorithm to solve a variational problem (1.1) involving geodesic distances. To that end we introduce the subgradient marching method, that computes the gradient of a discretized geodesic distance according to the metric.

In the sequel, we will refer to $\delta_\xi(x_s, x_t)$ as a subgradient of the concave mapping $\xi \mapsto d_\xi(x_s, x_t)$ instead of a supergradient. This slight abuse of terminology should not create confusion however.

Projected subgradient descent. This paper proposes to find a solution to (1.1) using a projected subgradient descent. Starting with some initial metric $\xi^{(0)}$, one iterates

$$\xi^{(k+1)} = \Pi_{\mathcal{C}} \left(\xi^{(k)} - \rho_k \nabla_{\xi^{(k)}} \mathcal{E} \right) \tag{1.8}$$

where $\rho_k > 0$ is a decaying sequence of gradient step size and $\Pi_{\mathcal{C}}$ is the orthogonal projection on the set of constraints $\mathcal{C}$, and where the gradient of the energy $\mathcal{E}$ is

$$\nabla_\xi \mathcal{E} = \sum_{s,t} \delta_\xi(x_s, x_t) \mathcal{E}'_{s,t}(d_\xi(x_s, x_t)) + \nabla_\xi J(\xi),$$

where $\delta = \delta_\xi(x_s, x_t)$ is the subgradient at ξ of the mapping $\xi \mapsto d_\xi(x_s, x_t)$. For any location $y \in \Omega$, $\delta(y)$ tells how much the geodesic distance between x_s and x_t is sensitive to variations on $\xi(y)$.

We also consider cases where the interactions $\mathcal{E}_{s,t}$ do not lead to a convex energy $\mathcal{E}$, in which case (1.8) is only guaranteed to converge to a local minimum of $\mathcal{E}$ if $\xi^{(k)}$ does not encounter a point of non-differentiability during the gradient descent.

Discretization difficulties and proposed method. The main bottleneck to compute the solution of (1.1) using a gradient method such as (1.8) is to compute subgradients $\delta_\xi(x_s, x_t)$. Furthermore, this computation should be performed on a discrete grid.

A small perturbation $\xi_\varepsilon = \xi + \varepsilon h$ defines a distance $d_{\xi_\varepsilon}(x_s, x_t)$ between x_s and x_t, that can be differentiated with respect to ε at $\varepsilon = 0$

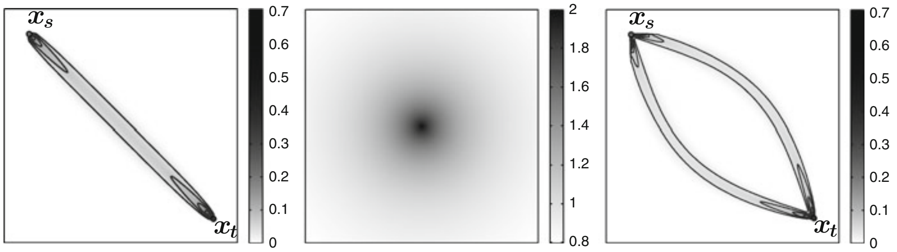


Fig. 1 On the *left*, $\delta_\xi(x_s, x_t)$ and some of its iso-levels for $\xi = 1$. In the *middle*, a non constant metric $\xi(x) = 1/(1.5 - \exp(-\|c - x\|))$, where c is the center of the domain. On the *right*, an element of the superdifferential of the geodesic with respect to the metric shown in the *middle*

$$\frac{d}{d\varepsilon} d_{\xi_\varepsilon}(x_s, x_t)\Big|_{\varepsilon=0} = \int\limits_\gamma h \, d\mathcal{H}^1 = \int\limits_0^1 h(\gamma(t))|\gamma'(t)|dt, \tag{1.9}$$

where the curve γ is the geodesic curve between x_s and x_t according to the metric ξ, which is a curve with minimal length

$$L_\xi(\gamma) = d_\xi(x_s, x_t).$$

If γ is unique, this shows that $\xi \mapsto d_{\xi_\varepsilon}(x_s, x_t)$ is differentiable at ξ, and that its gradient $\delta_\xi(x_s, x_t)$ is a measure supported along the curve γ. In the case where this geodesic is not unique, this quantity may fail to be differentiable. Yet, the map $\xi \mapsto d_\xi(x_s, x_t)$ is anyway concave (as an infimum of linear quantities in ξ) and for each geodesic we get an element of the super-differential through Equation (1.9).

The extraction of geodesics is quite unstable, especially for metrics such that x_s and x_t are connected by many curves of length close to the minimum distance $d_\xi(x_s, x_t)$. It is thus unclear how to discretize in a robust manner the gradient of the geodesic distance directly from the continuous definition (1.9). We propose in this paper an alternative method, where $\delta_\xi(x_s, x_t)$ is defined unambiguously as a subgradient of a discretized geodesic distance. Furthermore, this discrete subgradient is computed with a fast Subgradient Marching algorithm.

Figure 1 shows two examples of subgradients, computed with the algorithm detailed in Sect. 3. Near a degenerate configuration, we can see that the subgradient $\delta_\xi(x_s, x_t)$ might be located around several minimal curves.

Anisotropic metrics. The geodesic distance and its subgradient can be defined for more general Riemannian metric ξ that depends both on the location $\gamma(t)$ of the curve and on its local direction $\gamma'(t)/|\gamma'(t)|$. The algorithm presented (Tables 1, 2) in this paper extends to this more general setting, thus allowing to design arbitrary anisotropic Riemannian metric. This requires to use more advanced Fast Marching methods, such as the ones developed in [5,20], see also [13] for a related extension of our method to 3D meshes. We decided however to restrict our attention to the isotropic case, that has many practical applications.

Table 1 Fast Marching algorithm.

```
Initialization: $\mathcal{U}_{x_s} = 0, \mathcal{S}_{x_s} = Trial, \quad \forall (i, j) \neq x_s,\ \mathcal{S}_{i,j} = Far, \mathcal{U}_{i,j} = +\infty$.
repeat
    Select point: $(i, j) \longleftarrow \underset{(i',j'), \mathcal{S}_{i',j'} = Trial}{\text{argmin}} \mathcal{U}_{i',j'}$.
    Tag: $\mathcal{S}_{i,j} \leftarrow Known$.
    for $(i', j') \in \mathcal{N}(i, j)$ do
        if $\mathcal{S}_{i',j'} = Trial$ or $Far$ then
            $\mathcal{S}_{i',j'} \leftarrow Trial$
            Update the value of $\mathcal{U}_{i',j'} = u$ by solving (2.3).
until $\{(i, j)\ ;\ \mathcal{S}_{i,j} = Trial\} = \emptyset$;
```

Table 2 Subgradient Marching algorithm.

```
Initialization: $\mathcal{U}_{x_s} = 0$, $\mathcal{S}_{x_s} = Trial$, $\nabla_\xi \mathcal{U}_{x_s} = \mathbf{0}$ the null vector. $\forall (i, j) \neq x_s,\ \mathcal{S}_{i,j} = Far$, $\mathcal{U}_{i,j} = +\infty$.
repeat
    Select point: $(i, j) \longleftarrow \underset{(i',j'), \mathcal{S}_{i',j'} = Trial}{\text{argmin}} \mathcal{U}_{i',j'}$.
    Tag: $\mathcal{S}_{i,j} \leftarrow Known$.
    for $(i', j') \in \mathcal{N}(i, j)$ do
        if $\mathcal{S}_{i',j'} = Trial$ or $Far$ then
            $\mathcal{S}_{i',j'} \leftarrow Trial$
            Update the value of $\mathcal{U}_{i',j'}$, using either (2.5), (2.6) or (2.7).
            Update the value of $\nabla_\xi \mathcal{U}_{i',j'}$, using either (3.1), (3.2) or (3.3).
until $\{(i, j)\ ;\ \mathcal{S}_{i,j} = Trial\} = \emptyset$;
```

1.4 Contributions

This paper proposes a projected subgradient descent (1.8) to minimize variational problems that are discretized versions of (1.1). The key ingredient and the main contribution of the paper is the Subgradient Marching Algorithm, detailed in Sect. 3, that computes an element of the subgradient of the geodesic distance with respect to the metric. This algorithm follows the optimal ordering used by the Fast Marching, making the overall process only $O(N^2 \log(N))$ to compute subgradients of the maps $\xi \mapsto d_\xi(x_s, x_t)$ for a fixed x_s and for all the grid points x_t.

2 Discrete geodesic distances

Our approach to minimize variation problems such as (1.1) first defines a discrete geodesic distance $d_\xi(x_s, x_t)$ as the solution of a discretized partial differential equation. A discrete subgradient $\delta_\xi(x_s, x_t)$ of the map $\xi \mapsto d_\xi(x_s, x_t)$ is then defined to solve exactly discrete variational problems involving geodesic distances. This is a general framework that could be extended to a larger class of non-linear partial differential equations.

2.1 Discretization

Eikonal equation. To define a discrete geodesic distance and the corresponding subgradient, we consider a fixed starting point x_s, and define the distance map to this point

$$\mathcal{U}^{\xi}(x) = d_{\xi}(x_s, x). \tag{2.1}$$

Note that we have dropped the dependancy on the starting point x_s, that is assumed to be known and fixed.

As shown in [17], this distance map is the unique viscosity solution of the Eikonal non-linear PDE

$$\begin{cases} \|\nabla \mathcal{U}^{\xi}(x)\| = \xi, \\ \mathcal{U}^{\xi}(x_s) = 0. \end{cases} \tag{2.2}$$

The computation of $\mathcal{U}^{\xi}(x)$ thus requires the discretization of (2.2) so that a numerical scheme captures the viscosity solution of the equation.

Upwind discretization. In the following, we describe the computation in 2D of the geodesic distance and assume that the domain is $\Omega = [0, 1]^2$, although the scheme carries over for an arbitrary domain in any dimension. We consider regular grid discretization, although our method extends to more complicated discretizations, such as for instance triangulations (see [13]).

We will also drop the dependence on ξ and x_s of the distance map $\mathcal{U} = \mathcal{U}^{\xi}$ to ease the notations. The geodesic distance map $\mathcal{U}^{\xi}$ is discretized on a grid of $N = n \times n$ points, so that $\mathcal{U}_{i,j}$ for $0 \leq i, j < n$ is an approximation of $\mathcal{U}^{\xi}(ih, jh)$ where the grid step is $h = 1/n$. The metric ξ is also discretized so that $\xi_{i,j} = \xi(ih, jh)$.

Classical finite difference schemes do not capture the viscosity solution of (2.2). Upwind derivative should be used instead

$$\begin{aligned} D_1\mathcal{U}_{i,j} &:= \max\{(\mathcal{U}_{i,j} - \mathcal{U}_{i-1,j}), (\mathcal{U}_{i,j} - \mathcal{U}_{i+1,j}), 0\}/h, \\ D_2\mathcal{U}_{i,j} &:= \max\{(\mathcal{U}_{i,j} - \mathcal{U}_{i,j-1}), (\mathcal{U}_{i,j} - \mathcal{U}_{i,j+1}), 0\}/h. \end{aligned}$$

As proposed by Rouy and Tourin [18], the discrete geodesic distance map $\mathcal{U} = (\mathcal{U}_{i,j})$ is found as the solution of the following discrete non-linear equation that discretizes (2.2)

$$D\mathcal{U} = \xi \quad \text{where} \quad D\mathcal{U}_{i,j} = \sqrt{(D_1\mathcal{U}_{i,j})^2 + (D_2\mathcal{U}_{i,j})^2}. \tag{2.3}$$

Rouy and Tourin [18] showed that this discrete geodesic distance $\mathcal{U}$ converges to $\mathcal{U}^{\xi}$ when h tends to 0.

Figure 2 shows an example of a discrete geodesic distance map $\mathcal{U}$. The metric ξ takes lower values along a black curve than the background, so that the geodesic curves tends to follow this feature. An example of geodesic curve γ between x_s and x_t is

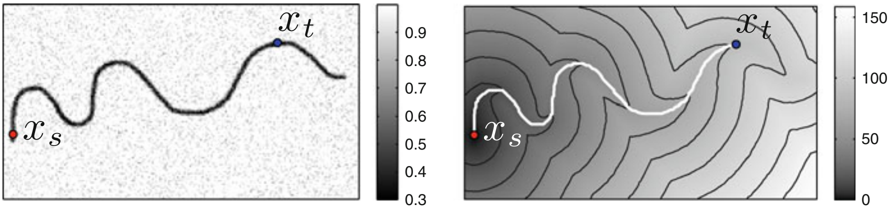


Fig. 2 Example of the minimal path computation using the Fast Marching algorithm. *On the left* the metric ξ. *On the right* The distance map $\mathcal{U}$ and the minimal path linking x_s to x_t

shown on the right, that is obtained by solving the ordinary differential equation

$$\frac{\mathrm{d}\gamma}{\mathrm{d}t} = -\nabla\mathcal{U}(\gamma(t)) \quad \text{and} \quad \gamma(0) = x_s.$$

2.2 Fast marching propagation

The Fast Marching algorithm, introduced by Sethian in [19] and Tsitsiklis in [22], allows to solve (2.3) in $O(N\log(N))$ operations using an optimal ordering of the grid points. This greatly reduces the numerical complexity with respect to iterative methods, because grid points are only visited once.

We recall the basic ideas underlying this algorithm, because our Subgradient Marching algorithm detailed in Sect. 3 makes use of the same ordering process.

The values of $\mathcal{U}$ may be regarded as the arrival times of wavefronts propagating from the source point x_s with velocity $1/\xi$. The central idea behind the Fast Marching method is to visit grid points in an order consistent with the way wavefronts propagates.

In the course of the algorithm, the state $\mathcal{S}_{i,j}$ of a grid point (i, j) passes successively from $\mathcal{S}_{i,j}$ =*Far* (no estimate of $\mathcal{U}_{i,j}$ is available) to $\mathcal{S}_{i,j}$ =*Trial* (an estimate of $\mathcal{U}_{i,j}$ is available, but it might not be the solution of (2.2)) to $\mathcal{S}_{i,j}$ =*Known* (the value of $\mathcal{U}_{i,j}$ is fixed and solves (2.2)). The set of *Trial* points forms an interface between *Known* points (initially the point x_s alone) and the *Far* points. The Fast Marching algorithm progressively propagates this front of *Trial* points so that all grid points are visited (see Fig. 3).

At each iteration of the algorithm, a point (i, j) is tagged as $\mathcal{S}_{i,j}$ =*Known* so that $\mathcal{U}_{i,j}$ is the solution of (2.2). The value of $\mathcal{U}$ at the neighboring points $(i', j') \in \mathcal{N}(i, j) = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$ such that (i', j') is not Known yet are updated by solving (2.3), using only the values of $\mathcal{U}$ that are *Known*.

The decision of moving the state of a point from $\mathcal{S}_{i,j}$ =*Trial* to $\mathcal{S}_{i,j}$ =*Known* is made by selecting the *Trial* point with minimum value of $\mathcal{U}$. It can be shown that updating the value of $\mathcal{U}$ by solving (2.3) can only increase the value of $\mathcal{U}_{(i',j')}$ for a trial point (i', j'), so that the values of *Known* points are ensured to solve (2.2). A heap data structure allows one to locate this minimum point in at most $\log(N)$ operations, so that the overall complexity of the algorithm is $O(N\log(N))$ operations. This is

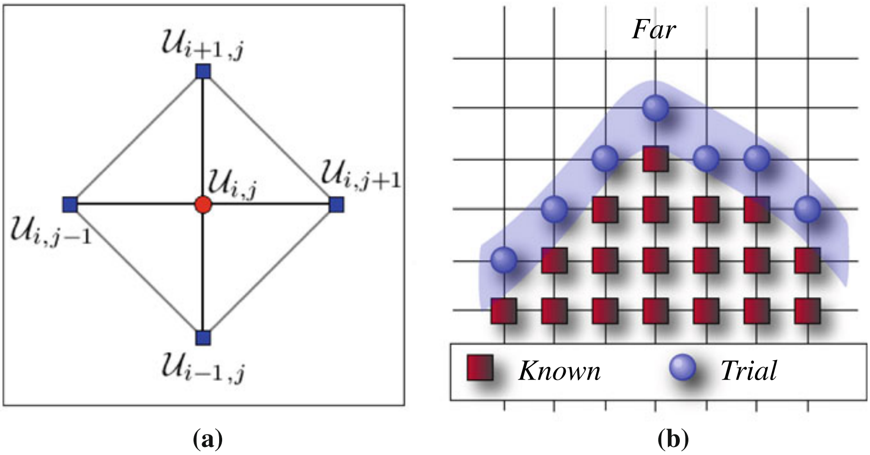


Fig. 3 Fast Marching propagation on a regular grid. **a** Neighborhood $\mathcal{N}(i, j)$, **b** Points states during the propagation

similar to Dijkstra's algorithm for computing shortest paths on graphs [14], the main difference is the expression of the local contribution to the weighted distance.

Table 1 details the steps of the Fast Marching algorithm.

2.3 Update of the geodesic distance map and parental relations

Each step of the Fast Marching (FM) requires the resolution of (2.3) to update the value of $u = \mathcal{U}_{i,j}$ for a small set of points (i, j). One thus needs to compute the solution u of

$$\max\{(u - \mathcal{U}_{i-1,j}), (u - \mathcal{U}_{i+1,j}), 0\}^2 + \max\{(u - \mathcal{U}_{i,j-1}), (u - \mathcal{U}_{i,j+1}), 0\}^2 = (h\xi_{i,j})^2. \tag{2.4}$$

This computation deserves special attention because our Subgradient Marching requires the computation of the derivative of the obtained distance $\mathcal{U}_{i,j} = u$. Thanks to the recursive structure of the FM procedure, the value $\mathcal{U}_{i,j}$ will only depend upon the values at some of its neighbors, which we will call *parents* of (i, j).

In order to solve (2.4) we first detect which of the values $\mathcal{U}_{i-1,j}, \mathcal{U}_{i+1,j}$ is smaller (notice that one or both of these values could a priori be $+\infty$), let $a \in \{-1, +1\}$ be such that $\mathcal{U}_{i+a,j} = \min\{\mathcal{U}_{i-1,j}, \mathcal{U}_{i+1,j}\}$. Analogously, we choose $b \in \{-1, +1\}$ such that $\mathcal{U}_{i,j+b} = \min\{\mathcal{U}_{i,j-1}, \mathcal{U}_{i,j+1}\}$. In case of equality in these minimization we choose a pair (a, b) according to any previously chosen conventional priority rule so as to avoid ambiguities.

We now concentrate on the three points $\{(i, j), (i + a, j), (i, j + b)\}$ (see Fig. 3a) and we try to find the solution u of (2.4): depending on whether the neighboring point are *Known* or not, different possibilities may occur.

- In the case where $\mathcal{U}_{i+a,j} \neq +\infty, \mathcal{U}_{i,j+b} \neq +\infty$ and $|\mathcal{U}_{i+a,j} - \mathcal{U}_{i,j+b}| < h\xi_{i,j}$, then no $u \leq \max\{U_{i+a,j}, \mathcal{U}_{i,j+b}\}$ can be a solution of (2.4), and, since the left-hand side of such an equation is increasing in u, there is only one solution, which is the maximum real solution of the quadratic equation

$$(u - \mathcal{U}_{i+a,j})^2 + (u - \mathcal{U}_{i,j+b})^2 = (h\xi_{i,j})^2. \tag{2.5}$$

  In this case we say that both points $(i + a, j)$ and $(i, j + b)$ are parents of (i, j).
- In the case where $\mathcal{U}_{i+a,j} \neq +\infty$ and $\mathcal{U}_{i,j+b} = +\infty$, then one defines

$$u = \mathcal{U}_{i+a,j} + h\xi_{i,j} \tag{2.6}$$

  and the point $(i + a, j)$ is the only parent of (i, j).
- In the case where $\mathcal{U}_{i+a,j} = +\infty$ and $\mathcal{U}_{i,j+b} \neq +\infty$, then one takes

$$u = \mathcal{U}_{i,j+b} + h\xi_{i,j}. \tag{2.7}$$

  and in this case it is the point $(i, j + b)$ which is the only parent of (i, j).
- If $\mathcal{U}_{i+a,j} \neq +\infty, \mathcal{U}_{i,j+b} \neq +\infty$ but $|\mathcal{U}_{i+a,j} - \mathcal{U}_{i,j+b}| > h\xi_{i,j}$, then no value of $u \geq \max\{U_{i+a,j}, \mathcal{U}_{i,j+b}\}$ can be a solution of (2.4). In such a case take

$$u = \min\{\mathcal{U}_{i+a,j}, \mathcal{U}_{i,j+b}\} + h\xi_{i,j} \tag{2.8}$$

  and the point which realizes the minimum between $\mathcal{U}_{i+a,j}$ and $\mathcal{U}_{i,j+b}$ (which is unique, in this case) will be the only parent of (i, j).

Notice that, when a point (i, j) is tagged as $\mathcal{S}_{i,j}$ =*Known* by the algorithm, its value $\mathcal{U}_{i,j}$ solves the discrete Eikonal equation (2.3), so that this value only depends on the value of its parent(s), which is (are) necessarily *Known*.

The Fast Marching algorithm, during the propagation, stores this parental relationship $(i, j) \rightarrow (i + a^\star, j)$ and/or $(i, j) \rightarrow (i, j + b^\star)$, since each point except x_s has exactly one or two parents. This defines a directed graph structure without cycle, that stores the dependencies induced by the resolution of the discrete Eikonal equation. We obviously define the set of *ascendants* of (i, j) as the set composed of (i, j) itself and of those points which are before (i, j) in this parental relation, i.e. which are either the parents of (i, j), or parents of parents. Figure 4 shows examples of such a graph of dependencies for two different metrics.

3 Subgradient marching algorithm

This section details our Subgradient Marching algorithm that computes an element $\delta_\xi(x_s, x_t)$ of the subgradient of the mapping $\xi \mapsto d_\xi(x_s, x_t)$. Since ξ is discretized on a regular grid of N points, it is represented as a vector $\xi \in \mathbb{R}^N$ of dimension N. A subgradient $\delta_\xi(x_s, x_t) \in \mathbb{R}^N$ is thus also a vector of N components.

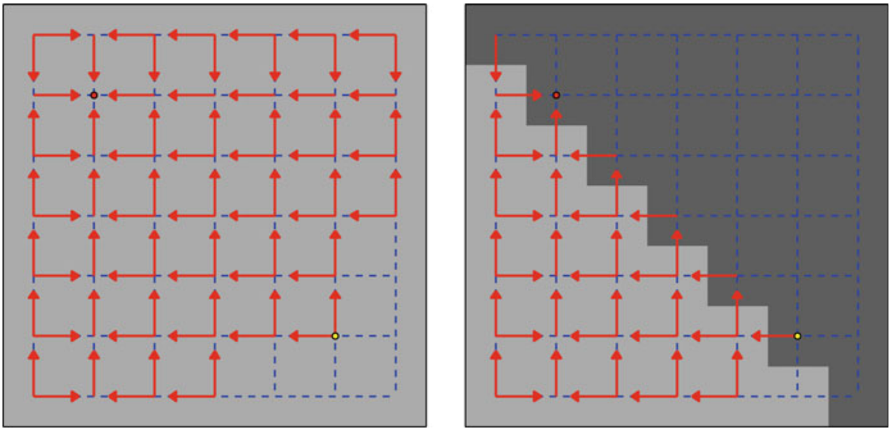


Fig. 4 Graphs of dependencies of the Fast Marching computations. *Left* uniform metric, *right* varying metric

For a fixed starting point x_s, our algorithm in fact computes for all grid points $x_t = (i, j)$,

$$\delta_\xi(x_s, x_t) = \nabla_\xi \mathcal{U}_{i,j} \in \mathbb{R}^N$$

which is a subgradient of the mapping $\xi \mapsto \mathcal{U}$, where $\mathcal{U}$ is the discrete geodesic distance to x_s, computed by solving (2.3) as detailed in Sect. 2.

3.1 Concavity of the geodesic distance

To solve variational problems involving the geodesic distance $d_\xi(x_s, x)$, for $x = (ih, jh)$, one would like to differentiate with respect to ξ the discrete distance map $\mathcal{U}^\xi_{i,j}$, obtained by solving (2.3). Actually, this is not always possible, since the mapping $\xi \mapsto \mathcal{U}^\xi_{i,j}$ is not necessary smooth. The following proposition proves that $\mathcal{U}^\xi_{i,j}$ is a concave function of ξ and this allows for superdifferentiation (the correspondent of subdifferential for concave functions instead of convex).

Proposition 3.1 *For a given point* (i, j), *the functional* $\xi \mapsto \mathcal{U}^\xi_{i,j}$ *is concave.*

Proof In the following we drop the dependence on (i, j) and note $\mathcal{U}^\xi = \mathcal{U}^\xi_{i,j}$. Thanks to the homogeneity, it is sufficient to prove super-additivity. We want to prove the inequality

$$\mathcal{U}^{\xi_1+\xi_2} \geq \mathcal{U}^{\xi_1} + \mathcal{U}^{\xi_2}.$$

Thanks to the comparison principle of Lemma 3.2 below, it is sufficient to prove that $\xi_1 + \xi_2 \geq D(\mathcal{U}^{\xi_1} + \mathcal{U}^{\xi_2})$, where the operator D is defined in (2.3). This is easily done if we notice that the operator D is convex (as it is a composition of the function

$(s,t) \mapsto \sqrt{s^2+t^2}$, which is convex and increasing in both s and t, and the operator D_1 and D_2, which are convex since they are produced as a maximum of linear operators) and 1−homogeneous, and hence it is subadditive, i.e. it satisfies $D(u+v) \le Du+Dv$.

Lemma 3.2 *If* $\xi \le \eta$, *then* $\mathcal{U}^{\xi} \le \mathcal{U}^{\eta}$.

Proof Let us suppose at first a strict inequality $\xi < \eta$. Take a minimum point for $\mathcal{U}^{\eta} - \mathcal{U}^{\xi}$ and suppose it is not the starting point x_s. Computing D and using sub-additivity we have

$$\eta = D\mathcal{U}^{\eta} \le D(\mathcal{U}^{\eta} - \mathcal{U}^{\xi}) + D\mathcal{U}^{\xi} = D(\mathcal{U}^{\eta} - \mathcal{U}^{\xi}) + \xi,$$

which gives $D(\mathcal{U}^{\eta} - \mathcal{U}^{\xi}) \ge \eta - \xi > 0$. Yet, at minimum points we should have $D(\mathcal{U}^{\eta} - \mathcal{U}^{\xi}) = 0$ and this proves that the minimum is realized at x_s, which implies $\mathcal{U}^{\eta} - \mathcal{U}^{\xi} \ge 0$.

To handle the case $\xi \le \eta$ without a strict inequality, juste replace η by $(1+\varepsilon)\eta$ and notice that the map $\eta \mapsto \mathcal{U}^{\eta}$ is continuous.

3.2 Recursive subdifferentiation

Proposition 3.1 proved that for a fixed point (i,j) and a fixed source x_0 the functional $\xi \mapsto \mathcal{U}^{\xi}_{i,j}$ is concave. For a metric $\xi > 0$, one can thus consider an element $\nabla_{\xi}\mathcal{U}_{i,j}$ of the subdifferential of this functional.

The value of $\mathcal{U}_{i,j}$ at a point (i,j) depends only on the values of its parents $(i+a^{\star},j)$ and/or $(i,j+b^{\star})$ through quadratic or linear equations (2.5), (2.6) or (2.7). The sub-differential $\nabla_{\xi}\mathcal{U}_{i,j}$ thus also depends on the subdifferentials $\nabla_{\xi}\mathcal{U}_{i+a^{\star},j}$ and/or $\nabla_{\xi}\mathcal{U}_{i,j+b^{\star}}$.

One has to consider several cases, depending on the number of parents of (i,j).

- For the special case $(i,j) = x_s$, the value of $\mathcal{U}_{i,j}$ is zero and does not depend on ξ. Thus, $\nabla_{\xi}\mathcal{U}_{i,j} = \mathbf{0}$, the null vector.
- If (i,j) has two parents, differentiating (2.5) with respect to ξ leads to

  $$\alpha(\nabla_{\xi}\mathcal{U}_{i,j} - \nabla_{\xi}\mathcal{U}_{i+a^{\star},j}) + \beta(\nabla_{\xi}\mathcal{U}_{i,j} - \nabla_{\xi}\mathcal{U}_{i,j+b^{\star}}) = h^2\xi_{i,j}\mathbb{1}_{i,j}$$

  where $\alpha = \mathcal{U}_{i,j} - \mathcal{U}_{i+a^{\star},j} \in \mathbb{R}$ and $\beta = \mathcal{U}_{i,j} - \mathcal{U}_{i,j+b^{\star}} \in \mathbb{R}$. Since $\xi > 0$, one has $\alpha+\beta > 0$. The subgradient at the point (i,j) is thus the vector $\nabla_{\xi}\mathcal{U}_{i,j}$ defined as

  $$\nabla_{\xi}\mathcal{U}_{i,j} = \frac{1}{\alpha+\beta}\left(h^2\xi_{i,j}\mathbb{1}_{i,j} + \alpha\nabla_{\xi}\mathcal{U}_{i+a^{\star},j} + \beta\nabla_{\xi}\mathcal{U}_{i,j+b^{\star}}\right), \qquad (3.1)$$

  where $\mathbb{1}_{i,j}$ is the Dirac vector

  $$\mathbb{1}_{i,j}(i',j') = \begin{cases} 1 & \text{if} \quad (i,j) = (i',j'), \\ 0 & \text{otherwise.} \end{cases}$$

- If only $(i+a^\star, j)$ is a parent of (i, j), differentiating (2.6) with respect to ξ leads to

$$\nabla_\xi \mathcal{U}_{i,j} = \nabla_\xi \mathcal{U}_{i+a^\star,j} + h\mathbb{1}_{i,j}. \tag{3.2}$$

- If only $(i, j+b^\star)$ is a parent of (i, j), differentiating (2.7) with respect to ξ leads to

$$\nabla_\xi \mathcal{U}_{i,j} = \nabla_\xi \mathcal{U}_{i,j+b^\star} + h\mathbb{1}_{i,j}. \tag{3.3}$$

Applying these rules during the Fast Marching propagation allows one to compute the value of the subgradient $\nabla_\xi \mathcal{U}_{i,j}$ at all grid points (i, j). The corresponding Subgradient Marching algorithm is detailed in Table 2.

Each vector $\nabla_\xi \mathcal{U}_{i,j}$ stores at most N non-zero coefficients, so that the overall computation takes $O(N^2 \log(N))$ operations and has a space complexity of $O(N^2)$.

Figure 1 shows two examples of subgradients $\nabla_\xi \mathcal{U}_{i,j}$ computed with the Subgradient Marching algorithm. For the metric $\xi = 1$, the subgradient is concentrated closely along the geodesic, which is a straight line. The second example shows a configuration for which the subgradient is located around two geodesic curves.

Notice that every vector $\nabla_\xi \mathcal{U}_{i,j}$ is a vector whose entries corresponding to points that are not *ascendant* of (i, j) vanish and whose non-zero entries are smaller than h. This may be easily proven by a recursive argument thanks to (3.1), (3.2), (3.3), using the inequality $\alpha + \beta \geq \sqrt{\alpha^2 + \beta^2} = h\xi_{i,j}$ (in Figure 1 the values have been scaled with $h = 1$).

3.3 Validity of the subgradient marching algorithm

The following theorem ensures the validity of the Subgradient Marching algorithm.

Theorem 3.3 *For $\xi > 0$, a given starting point x_0 and a given point (i, j), the vector $\nabla_\xi \mathcal{U}_{i,j}$ computed with Subgradient Marching belongs to the subdifferential set of the functional $\xi \mapsto \mathcal{U}^\xi_{i,j}$.*

Proof Consider the set of metrics $\xi \in (\mathbb{R}_+)^N$ which give distinct values to the action map at every point, i.e. such that $\mathcal{U}_{i,j} \neq \mathcal{U}_{i',j'}$ for $(i, j) \neq (i', j')$. In this case the expression of $\mathcal{U}_{i,j}$ is given by the recursive algebraic formula (3.1), (3.2) or (3.3) involving the values of the parents. Each parent (both in the case of one parent and in the case of two parents) is defined with no ambiguity and the same tree of parental dependence would stay valid even if ξ was changed by small perturbations. It means that there exists a neighborhood of ξ such that, for all other metrics $\tilde{\xi}$ in such a neighborhood, all the parental relations between points are the same. Hence, for $\tilde{\xi}$ in this neighborhood the value of $\mathcal{U}^{\tilde{\xi}}_{i,j}$ is given by the same algebraic and recursive expression. The vector $\nabla_\xi \mathcal{U}_{i,j}$ is exactly the differential of this expression.

Now we suppose that ξ is such that there are possible ex-aequo entries in the vector $u(i, j) = \mathcal{U}^\xi_{i,j}$. One can slightly perturb this function by a sequence of function

u_n satisfying the same strict inequalities satisfied by u (i.e., $u(i,j) < u(i',j') \Rightarrow u_n(i,j) < u_n(i',j')$) and replacing equalities by inequalities according to the order convention $\tilde{<}$ on the points of the grid: this means

$$u(i,j) = u(i',j'),\ (i,j) \tilde{<} (i',j') \Rightarrow u_n(i,j) < u_n(i',j').$$

This is possible by small perturbations, so that $u_n(i,j) \to u(i,j)$ for each (i,j). Moreover in this way the parental relation is left unchanged for ξ_n and for ξ. Then define ξ_n by $\xi_n = Du_n$ according to (2.3). Obviously we have $\xi_n \to \xi$.

Moreover, for every (i,j) and every n the vectors $\nabla_\xi \mathcal{U}^{\xi_n}_{i,j}$ belongs to the superdifferential of the map $\xi \mapsto \mathcal{U}^{\xi}_{i,j}$ (since the function is concave and this vector is the gradient). Since the graph of the superdifferential is closed, the limit of this sequence of vectors must belong to the superdifferential at ξ. This limit actually exists and is given by $\nabla_\xi \mathcal{U}^{\xi}_{i,j}$, because of the continuity of the formulas that we used to compute all of these vectors. This is possible because the approximation ξ_n was chosen in order not to change the parental relations. □

4 Applications

This section describes some applications of the Subgradient Marching algorithm to the resolution of variational problems of the form (1.1). We consider discretized metrics $\xi \in \mathbb{R}^N$, and the geodesic distances $d_\xi(x_s, x_t)$ are discrete distances computed as detailed in Sect. 2.1.

4.1 Landscape design

For the design of a landscape in a domain $\Omega \subset \mathbb{R}^d$, the interaction functional are linear (1.4), so that the problem corresponds to the maximization of geodesic distances.

Constrained distance maximization. The optimization of $\mathcal{E}$ should be done under additional constraints on the set of admissible metrics, in order to avoid degenerate solutions. We consider here a local constraint

$$\forall i,j, \quad 0 < \underline{\xi}_{i,j} \leq \xi_{i,j} \leq \overline{\xi}_{i,j}, \tag{4.1}$$

that accounts for the maximal concentration of material allowed. We also consider a global constraint

$$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \xi_{i,j} \leq \lambda, \tag{4.2}$$

where $|\Omega|$ is the number of grid points in Ω, that accounts for the total amount of ground material available. The constant λ satisfies necessarily

$$\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\underline{\xi}_{i,j}\leq\lambda\leq\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\overline{\xi}_{i,j}.$$

We note that the maximization of $\mathcal{E}$ under the pointwise constraint (4.1) alone would be saturated everywhere $\xi_{i,j}=\overline{\xi}_{i,j}$.

Taking into account that the constraint (4.2) is obviously saturated, the landscape design problem is written as

$$\xi^\star\in\underset{\xi\in\mathcal{C}}{\operatorname{argmin}}\ \mathcal{E}(\xi)=-\sum_{s,t}w_{s,t}d_\xi(x_s,x_t) \tag{4.3}$$

where

$$\mathcal{C}=\left\{\xi\in\mathbb{R}^N\ ;\ \underline{\xi}_{i,j}\leq\xi_{i,j}\leq\overline{\xi}_{i,j},\ \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\xi_{i,j}=\lambda\right\}, \tag{4.4}$$

and we note that this minimum might be non-unique.

To solve (4.3), we use the projected gradient descent (1.8), with no regularization $J=0$. In this case, the gradient of $\mathcal{E}$ simply reads

$$\nabla_\xi\mathcal{E}=-\sum_{s,t}w_{s,t}\nabla_\xi\mathcal{U}_s^\xi(x_t). \tag{4.5}$$

where each $\mathcal{U}_s^\xi(x)=d_\xi(x_s,x)$ is the distance map to a landmark x_s. Each subgradient vector $\nabla_\xi\mathcal{U}_s^\xi(x_t)$ is computed by the subgradient descent algorithm, Table 2, starting the front propagation from the point x_s.

Thanks to the following proposition, the projection $\Pi_\mathcal{C}$ is easily computed. A simple dichotomy is used to find the value of α that satisfies (4.7)

Proposition 4.1 *For $\underline{\xi}\leq\lambda\leq\overline{\xi}$, one has*

$$\Pi_\mathcal{C}(\xi)=\mathcal{P}_{\underline{\xi}}^{\overline{\xi}}(\xi+\alpha),\quad\textit{where}\quad\mathcal{P}_{\underline{\xi}}^{\overline{\xi}}(\xi)_{i,j}=\max(\min(\xi_{i,j},\overline{\xi}_{i,j}),\underline{\xi}_{i,j}) \tag{4.6}$$

where $\alpha\in\mathbb{R}$ is such that

$$\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\Pi_\mathcal{C}(\xi+\alpha)_{i,j}=\lambda. \tag{4.7}$$

Proof The projected metric $\Pi(\xi)$ satisfies

$$\Pi_\mathcal{C}(\xi)=\min_{\tilde{\xi}\in\mathcal{C}}\left\|\tilde{\xi}-\xi\right\|^2. \tag{4.8}$$

For a given Lagrange multiplier $\alpha \in \mathbb{R}$, this corresponds to minimize

$$\min_{\underline{\xi} \leq \tilde{\xi} \leq \overline{\xi}} \|\tilde{\xi} - \xi\|^2 - 2\alpha \sum_{(i,j)\in\Omega} \tilde{\xi}_{i,j}.$$

By developing the square terms, one sees that this amounts to project $\xi + \alpha$ on the set of constraints $\underline{\xi} \leq \tilde{\xi} \leq \overline{\xi}$. This is obtained by truncation with the operator $\mathcal{P}_{\underline{\xi}}^{\overline{\xi}}$, as written in (4.6).

Since the function

$$\alpha \mapsto \sum_{(i,j)\in\Omega} \mathcal{P}_{\underline{\xi}}^{\overline{\xi}}(\xi + \alpha)_{i,j}$$

is an increasing continuous function, there exists $\alpha \in \mathbb{R}$ satisfying conditions (4.7).

The following theorem ensures the convergence of the projected gradient descent.

Theorem 4.2 *For $\rho_k = 1/k$, the sequence $(\xi^{(k)})_k$ converges to a minimizer $\xi^\star$ of* (4.3).

Proof As stated for instance in [4], the convergence of a subgradient descent is ensured if

$$\sum_k \rho_k = +\infty \quad \text{and} \quad \sum_k \rho_k^2 < +\infty.$$

and if the sequence $(\nabla_{\xi^{(k)}}\mathcal{E})_k$ stay bounded. Since for each landmarks (x_s, x_t), the mapping $\xi \mapsto \mathcal{U}_s^\xi(x_t)$ is concave and 1-homogeneous, it is Lipschitz continuous and hence its subgradients are bounded. □

Numerical examples. We first consider $p = 2$ agents located at two points x_0, x_1 in the corners of a square domain, as shows in Fig. 5. The constraints are set to $\underline{\xi} = 0.1$, $\overline{\xi} = 1$ and $\lambda = 0.2\,|\Omega|$. The domain Ω is sampled on a square grid of 100×100 points. Since the landmarks are close to the boundary of Ω, hills appear between each x_i and the boundary. This phenomena is explained by the fact that it is less costly to *build* these short hills and it makes a bypass behind the defender more difficult.

Figure 6 left, shows the decay of the error between the iterates $\xi^{(k)}$ and an optimal metric $\xi^\star$. Figure 6, right, shows the increase of the energy, which is not strictly monotonic because of the non-smooth nature of the problem to optimize.

Figure 7, left and middle, shows the influence of the total mass parameter λ. Decreasing the value of λ causes the optimal metric $\xi^\star$ to be more concentrated around the landmark positions. By decreasing value of $\underline{\xi}$ toward 0, these regions approach circular shape (see Figure 7 right). One can note that the Gradient Marching algorithm is able to compute a subgradient of $\mathcal{E}$ as soon as $\xi > 0$, but the convergence tends to become slower when $\underline{\xi}$ approaches 0.

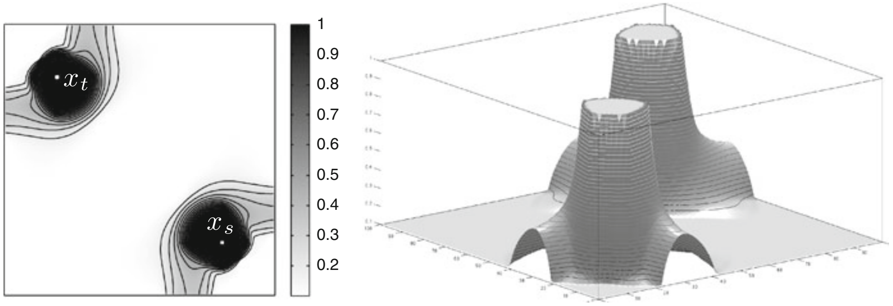


Fig. 5 2D and 3D display of the optimal metric $\xi^\star$

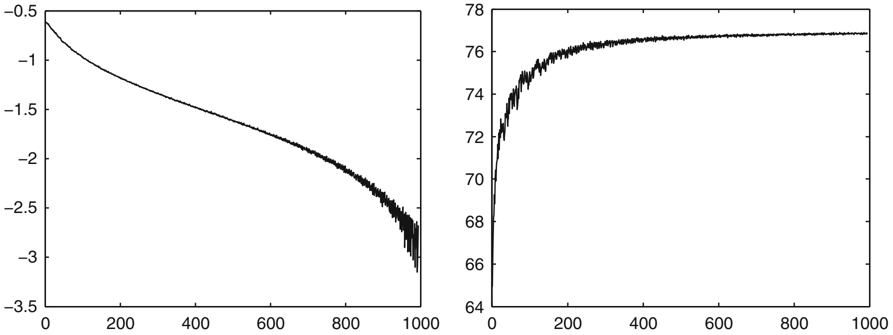


Fig. 6 *Left* decrease of the error $\log_{10}(\|\xi^{(k)} - \xi^\star\|/\|\xi^\star\|)$, *right* increase of the energy $\mathcal{E}(\xi^{(k)})$

One can indeed prove that if $\underline{\xi}_{i,j} = 0$ and $\overline{\xi}_{i,j} = c$, in a continuous setting, the unique optimal metric $\xi^\star$ is given by $\xi_{i,j} = c$ for (i, j) in two disks around x_0 and x_1 and $\xi_{i,j} = 0$ elsewhere, if λ is small enough so that the two disks fit inside Ω.

Figure 8, left, shows an example of spatially varying constraints. To prevent the agent located in x_1 to modify the metric, we enforce

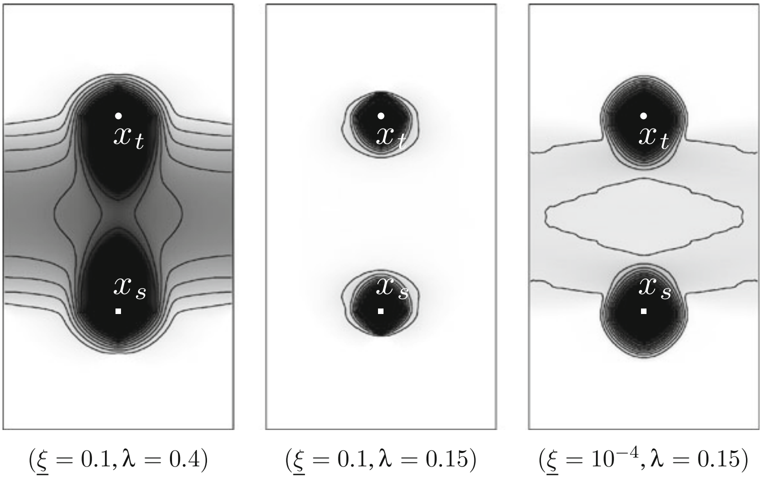
$$\forall (i, j) \in \Omega_1, \quad \underline{\xi}_{i,j} = \overline{\xi}_{i,j} = 0.1,$$

where Ω_1 is a region surrounding x_1, whereas outside Ω_1 we set

$$\forall (i, j) \notin \Omega_1, \quad \underline{\xi}_{i,j} = 0.1, \quad \overline{\xi}_{i,j} = 1.$$

The metric is thus only optimized in $\Omega \setminus \Omega_1$, as shown on Fig. 8 left.

Figure 8, right, shows an example of optimal metric $\xi^\star$ computed with $P = 8$ landmarks. The weights between the landmarks are set to $w_{s,t} = 1$ and $\underline{\xi}_{i,j} = 0.1$, $\overline{\xi}_{i,j} = 6$. Figure 9 shows the iterations of the algorithm for a domain Ω with a hole.



$(\underline{\xi} = 0.1, \lambda = 0.4)$ $(\underline{\xi} = 0.1, \lambda = 0.15)$ $(\underline{\xi} = 10^{-4}, \lambda = 0.15)$

Fig. 7 Dependence on parameters λ and $\underline{\xi}$ of the optimal metric ξ. In all examples $\overline{\xi} = 1$

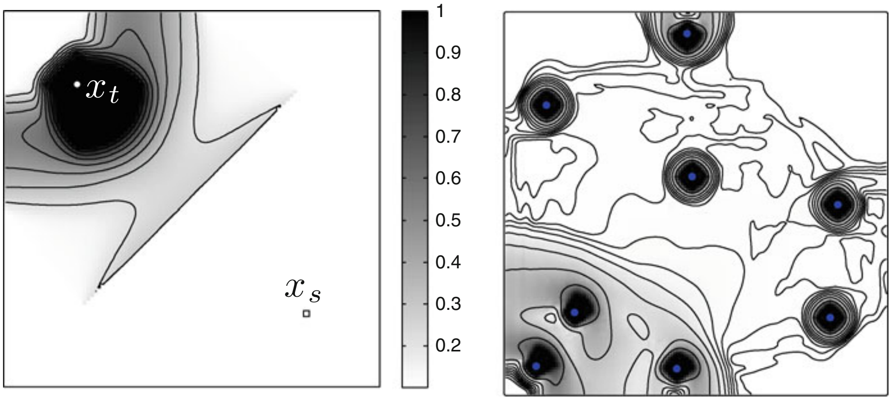
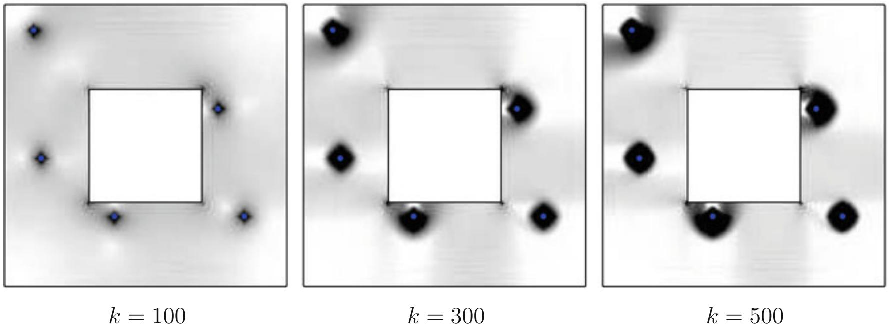


Fig. 8 *Left* spatially vayring constraint $\overline{\xi}_{i,j}$ with $P = 2$ landmarks, *right* constant constraint with $P = 8$ landmarks



$k = 100$ $k = 300$ $k = 500$

Fig. 9 Iterations $\xi^{(k)}$ computed for a domain Ω with a hole and with $P = 5$ landmarks

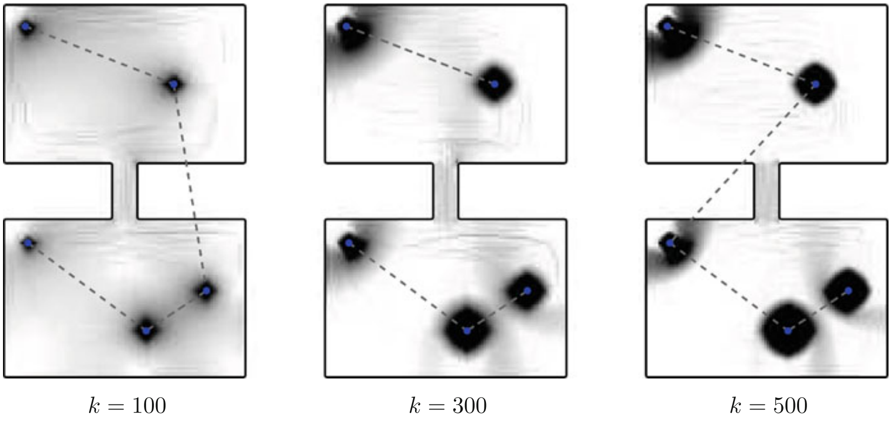


Fig. 10 Iteration $\xi^{(k)}$ computed during the subgradient descent. The *dashed line* corresponds to the connexion $s \to t(s)$ of nearest neighbors points

Extension of the model. It is possible to modify the energy $\mathcal{E}$ defined in (4.3) to mix differently the distances between the points $\{x_s\}_s$. One can for instance minimize

$$\mathcal{E}_{\min}(\xi) = -\sum_s \min_{t \neq s} d_\xi(x_s, x_t).$$

This functional is the opposite of the minimum of concave functions, and hence $\mathcal{E}_{\min}$ is a convex function. The minimization of the energy $\mathcal{E}_{\min}$ forces each landmark to be maximally distant from its closest neighbors.

The subgradient of $\mathcal{E}_{\min}$ is computed as

$$\nabla_\xi \mathcal{E} = -\sum_s \nabla_\xi \mathcal{U}_s^\xi(x_{t(s)}).$$

where, for each landmark x_s, $x_{t(s)}$ is the closest landmark according to the metric ξ

$$t(s) = \operatorname*{argmin}_{t \neq s} d_\xi(x_s, x_t).$$

The projected gradient descent (1.8) converges to a minimum of $\mathcal{E}_{\min}$.

Figure 10, left and center, shows how the metric $\xi^{(k)}$ evolves during the iterations of a projected gradient descent. The graph connecting each x_s to its nearest neighbor $x_{t(s)}$ is overlaid. The points x_s are clustered on two sides of the domain, so that during the first iterations, the graph connects points on each side of the domain. During the iterations, the nearest neighbor connexion $s \to t(s)$ evolves, until reaching a stable configuration, as shown on Fig. 10.

4.2 Traffic congestion equilibria

The simulation of a static traffic congestion is the computation of a Riemannian metric so that agents travel along geodesics between points $\{x_s\}_s$. The metric is the unique minimizer of (1.1) for linear interactions (1.4). The regularization J is a discretization of (1.5)

$$J(\xi) = \frac{1}{3} \sum_{(i,j)} |\xi_{i,j}|^3$$

Each weight $w_{s,t}$ is the strength of the traffic between two landmarks x_s and x_t. The only constraint is that metrics should be positive

$$\mathcal{C} = \left\{ \xi \in \mathbb{R}^N \ ; \ \xi_{i,j} \geq 0 \right\}.$$

The projection on this constraint set is simply

$$\Pi_{\mathcal{C}}(\xi) = \max(0, \xi),$$

and the subgradient descent (1.8) is guaranteed to converge to the solution of (1.1) if one uses for instance $\rho_k = 1/k$.

For this application, the subgradient of $\mathcal{E}$ is

$$\nabla_\xi \mathcal{E} = j(\xi) - \sum_{s,t} w_{s,t} \nabla_\xi \mathcal{U}_s(x_t), \tag{4.9}$$

where

$$j(\xi) = (\xi_{i,j}^2)_{(i,j)} \in \mathbb{R}^N.$$

The subgradient $\nabla_\xi \mathcal{U}_s(x_t)$ at ξ of the mapping $\xi \mapsto d_\xi(x_s, x_t)$ is computed with the Subgradient Marching algorithm, Table 2, starting the front propagation at the point x_s.

Numerical example. Figure 11 shows an example of congested metric with a complex domain Ω and four landmarks. The two landmarks x_{s_1}, x_{s_2} are sources of traffic and x_{t_1}, x_{t_2} are targets, so that $w_{s_1,s_2} = w_{t_1,t_2} = 0$ and we choose the other weights

$$w_{s_1,t_1} + w_{s_1,t_2} = 2(w_{s_2,t_1} + w_{s_2,t_2}) \quad \text{and} \quad \frac{w_{s_1,t_1}}{w_{s_1,t_2}} = \frac{w_{s_2,t_1}}{w_{s_2,t_2}},$$

so that the traffic intensity going out from x_{s_1} is twice the one from x_{s_2}. One can note the two hollows on each side of the river appearing because of the inter-sides and intra-sides crossed traffics.

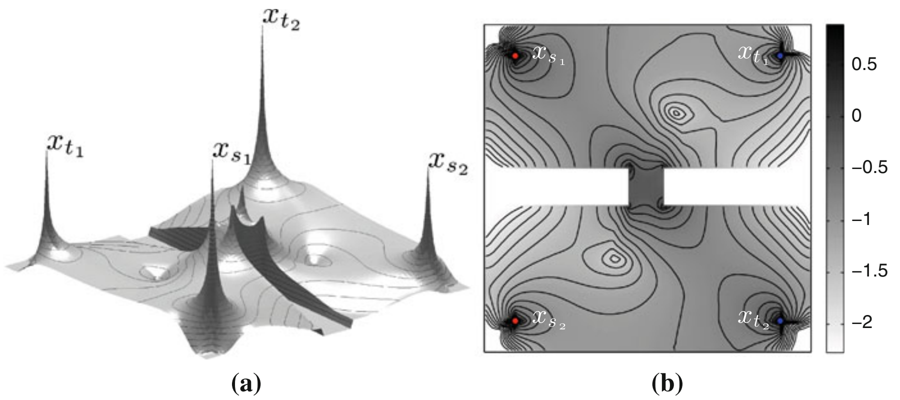


Fig. 11 Two sources and two targets, with a river and a bridge on a symmetric configuration and an asymmetric traffic weights. **a** 3D view of ξ^*, **b** Flat view of $\log(\xi^*)$

4.3 Travel time tomography

A simple model of seismic data acquisition assumes that geodesic distances are collected between points $(x_s)_s$. Pairs of emitter and receiver are denoted as (x_s, x_t) for $(s, t) \in \Gamma$.

In our simplified tomography setting, the data acquisition (1.6) computes geodesic distances $d_{s,t}$ for each pair $(s, t) \in \Gamma$ according to an unknown Riemannian metric ξ^0.

Geodesic tomography inversion. The recovery is obtained by solving (1.1) for a data fidelity term (1.7) as interaction functionals.

We use a discrete Sobolev regularization

$$J(\xi) = \frac{\mu}{2}\|\text{grad}\xi\|^2 = \frac{\mu}{2}\sum_{(i,j)\in\Omega}\|\text{grad}_{i,j}\xi\|^2$$

where the operator grad is a finite difference discretization of the 2D gradient

$$\text{grad}_{i,j}\xi = (\xi_{i+1,j} - \xi_{i,j}, \xi_{i,j+1} - \xi_{i,j}),$$

with Neumann condition on the boundary $\partial\Omega$ of the domain. The parameter μ controls the strength of the regularization and should be adapted to the number $|\Gamma|$ of measurements and the smoothness of ξ^0.

We use the local and global constraints (4.4), where

$$\underline{\xi} = \min_{(i,j)\in\Omega}\xi^0_{i,j}, \quad \overline{\xi} = \max_{(i,j)\in\Omega}\xi^0_{i,j}, \quad \text{and} \quad \lambda = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\xi^0_{i,j}.$$

The travel time recovery is thus obtained by minimizing

$$\underset{\xi \in \mathcal{C}}{\operatorname{argmin}} \; \frac{1}{2} \sum_{(s,t)\in\Gamma} (d_\xi(x_s, x_t) - d_{s,t})^2 + \frac{\mu}{2} \|\mathrm{grad}\xi\|^2 . \tag{4.10}$$

Subgradient descent recovery. The minimization problem (4.10) is non-convex, but a local minimizer $\xi^\star$ can be computed using the projected gradient descent (1.8). In this case, the gradient of the energy $\mathcal{E}$ at a metric ξ reads

$$\nabla_\xi \mathcal{E} = \mu \Delta \xi - \sum_{s,t} (d_\xi(x_s, x_t) - d_{s,t}) \nabla_{\xi^{(k)}} \mathcal{U}_s(x_t).$$

where $\Delta = -\mathrm{grad}^* \circ \mathrm{grad}$ is the Laplacian. The subgradient $\nabla_\xi \mathcal{U}_s(x_t)$ of the mapping $\xi \mapsto d_\xi(x_s, x_t)$ is computed with the Subgradient Marching algorithm, Table 2, starting the propagation from the point x_s.

The subgradient descent (1.8) converges to a local minimum $\xi^\star$ of the problem (4.10).

Numerical examples. Figure 12 shows two examples of smooth metrics ξ^0 recovered from travel time tomography measurements. In each case, we set $\overline{\xi}/\underline{\xi} = 1.3$ and we use $\xi^{(0)}_{i,j} = \lambda$ as an initial flat metric.

For the first example, we use $P = 100$ points distributed evenly on the boundary of a square Ω, discretized at $N = 150 \times 150$ points, and each points acts both as emitter and sensor.

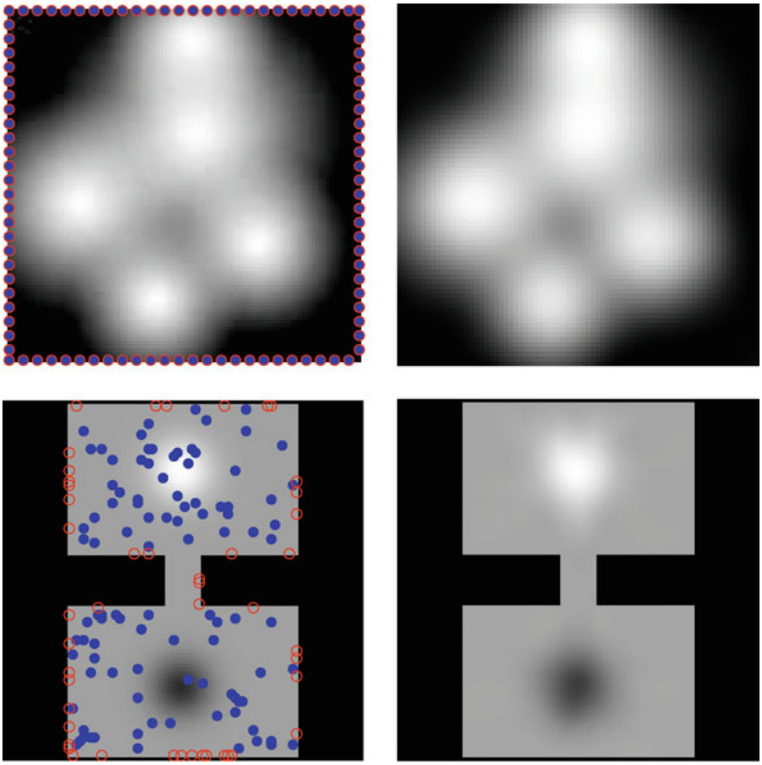
For the second example, we use 50 emitter points $\{x_s\}_{s=0}^{49}$ distributed evenly on the boundary of a complicated domain Ω, and 150 sensors $\{x_t\}_{s=50}^{199}$ distributed randomly within the domain. Each emitter is connected to all the sensors, so that $(s,t) \in \Gamma$ if and only if $s < 50$ and $t \geq 50$.

We enforce the smoothness of the solution by setting a large enough regularization parameter $\mu = 0.1$. Figure 13 shows the decay of the energy $\mathcal{E}$ and the error $\left\|\xi^{(k)} - \xi^\star\right\|$.

The recovery error $\left\|\xi^0 - \xi^\star\right\| / \left\|\xi^0\right\|$ is 2.5×10^{-2} for the first example and 7×10^{-3} for the second example. Both examples show that for moderately complicated tomography problems (smooth medium and low contrast), a good approximation can be obtained by Subgradient Marching descent. These synthetic examples are however quite simple, and a detailed analysis of the method and the properties of the minimizer $\xi^\star$ is desirable but beyond the scope of this paper.

5 Conclusion

We have presented a new projected gradient descent algorithm to optimize a metric with respect to an energy involving geodesic distances. The heart of our approach is the Subgradient Marching algorithm, which computes the derivative of a discrete geodesic distance with respect to the metric. Up to our knowledge, this is the first



ξ^0 $\xi^\star$

Fig. 12 Examples of travel time tomography recovery

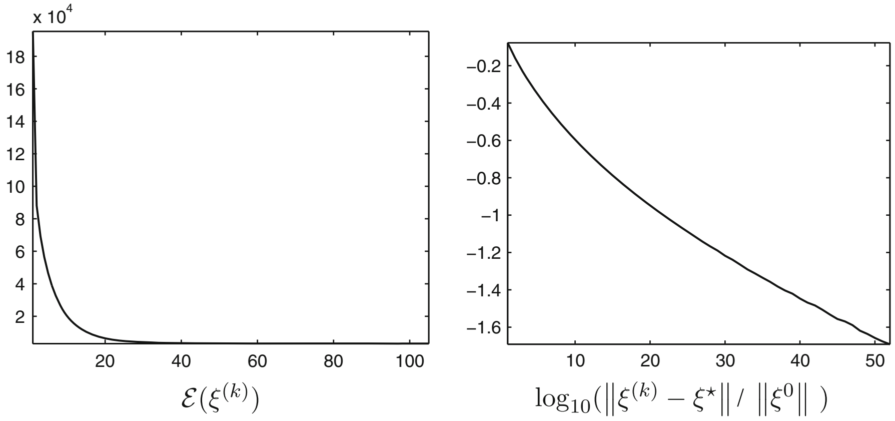


Fig. 13 Decay of the energy and the error for the first example shown in Fig. 12

time that a consistent numerical tool has been introduced to solve discrete variational problems that take into account geodesic distances between points. Three representative applications illustrate the practical use of Subgradient Marching. Landscape design and traffic congestion lead to the minimization of a convex functional, and can be solved efficiently with a projected gradient descent. Recovery of geodesic inverse problems such as travel time tomography is obtained by computing a local minimizer of a non-convex problem.

References

1. Benmansour, F., Carlier, G., Peyré, G., Santambrogio, F.: Numerical approximation of continuous traffic congestion equilibria. Netw. Heterog. Media **4**(3), 605–623 (2009)
2. Berryman, J.G.: Stable iterative reconstruction algorithm for nonlinear traveltime tomography. Inverse Probl. **6**(1), 21–42 (1990)
3. Bishop, T.N., Bube, K.P., Cutler, R.T., Langan, R.T., Love, P.L., Resnick, J.R., Shuey, R.T., Spindler, D.A., Wyld, H.W.: Tomographic determination of velocity and depth in laterally varying media. Geophysics **50**(6), 903–923 (1985)
4. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.: Numerical Optimization, 2nd edn. Springer, Heidelberg (2006)
5. Bornemann, F., Rasch, C.: Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle. Comput. Vis. Sci. **9**(2), 57–69 (2006)
6. Bronstein, A., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer, Berlin (2007)
7. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proc. Natl. Acad. Sci. (PNAS) **103**(5), 1168–1172 (2006)
8. Buttazzo, G., Davini, A., FragalÍ, I., Maciá, F.: Optimal Riemannian distances preventing mass tranfer. J. Reine Angew. Math. **575**, 157–171 (2004)
9. Carlier, G., Jimenez, C., Santambrogio, F.: Optimal transportation with traffic congestion and wardrop equilibria. SIAM J. Control Opt. **47**(3), 1330–1350 (2008)
10. Cavalca, M., Lailly, P.: Accounting for the definition domain of the forward map in traveltime tomography–application to the inversion of prismatic reflections. Inverse Probl. **23**(1), 139–164 (2007)
11. Cohen, L.D.: Minimal paths and fast marching methods for image analysis. In: Paragios, N., Chen, Y., Faugeras, O. (eds.) Handbook of Mathematical Methods in Computer Vision, Springer, Berlin (2005)
12. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: a minimal path approach. Int. J. Comput. Vis. **24**(1), 57–78 (1997)
13. Devir, Y., Rosman, G., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: On reconstruction of non-rigid shapes with intrinsic regularization. In: Proceedings of the Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA) (2009)
14. Dijkstra, E.W.: A note on two problems in connection with graphs. Numer. Math. **1**, 269–271 (1959)
15. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. **95**(15), 8431–8435 (1998)
16. Leung, S., Qian, J.: An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals. Commun. Math. Sci. **4**(1), 249–266 (2006)
17. Lions, P.-L.: Generalized Solutions of Hamilton–Jacobi Equations. Pitman, Boston (1982)
18. Rouy, E., Tourin, A.: A viscosity solution approach to shape from shading. SIAM J. Numer. Anal. **29**, 867–884 (1992)
19. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press, London (1999)
20. Sethian, J.A., Vladimirsky, A.: Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms. SIAM J. Numer. Anal. **41**(1), 325–363 (2003)
21. Sethian, J.A., Vladimirsky, A.: Fast Methods for the Eikonal and related Hamilton–Jacobi equations on unstructured meshes. Proc. Natl. Acad. Sci. USA **97**(11), 5699–5703 (2000)
22. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Trans. Automat. Contr. **40**, 1528–1538 (1995)
23. Wardrop, J.G.: Some theoretical aspects of road traffic research. Proc. Inst. Civ. Eng. **2**(2), 325–378 (1952)