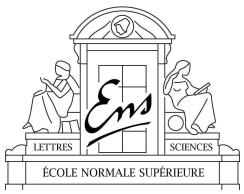
Shortest paths calculations, and applications to medical imaging

Mickaël Péchaud



École Doctorale de Sciences Mathématiques de Paris Centre
Certis (ENS / ENPC)

Doctorat

Informatique

Shortest paths calculations, and applications to medical imaging

Mickaël Péchaud

Under the supervision of Renaud KERIVEN

Defended on September 25th 2009

JURY

Rachid Deriche	
Michel Habib	
Renaud Keriven	Advisor
Ron Kimmel	Reviewer
Xavier Pennec	Reviewer
Emmanuel Prados	

Remerciements

Je tiens tout d'abord à remercier Renaud pour avoir été à l'origine de mon intérêt pour la vision par ordinateur, et pour les nombreuses discussions scientifiques et musicales que nous avons eues au cours de ces années.

Un grand merci à Patou, Pierre-Momo, Shaggy et Alex pour l'ambiance au bureau, et nos nombreuses discussions plus ou moins techniques.

Je remercie également les chercheurs avec qui j'ai été amené à travailler durant ces années, en particulier Max et Gabriel.

J'adresse toute ma gratitude à Joëlle et Brigitte pour leur disponibilité, leur gentillesse et leur compétence.

Merci à tous les gens qui m'ont permis de continuer à mener une existence équilibrée durant ces quelques années – avec une pensée particulière pour les musiciens de Samarkand, et maistre Oliver.

Un remerciement spécial à Vanya pour m'avoir soutenu au fil de ma thèse, en particulier pendant la période de rédaction...

Je souhaiterais enfin dédier ce manuscrit à mes parents, ainsi qu'à Nanette, Nanine et Papillon.

Table des matières

Introduction

L'informatique est la science du traitement automatique de l'information. La vision par ordinateur en est une branche, dont l'objectif est le traitement automatique d'informations de nature visuelle. Elle est née dans les années 70 comme une branche de l'intelligence artificielle. Son projet initial était de doter des ordinateurs ou des robots d'une vision similaire à la vision biologique – une application typique étant de permettre à un robot muni d'unes ou plusieurs caméras de se déplacer de façon autonome dans son environnement. Parmi les problèmes auxquels la vision par ordinateur s'intéresse figurent donc la reconstruction tridimensionnelle de l'environnement à partir d'une ou plusieurs images, la séparation d'une image en composantes pertinentes – par exemple un objet et son arrière-plan – appelée segmentation, puis l'obtention d'information de plus haut niveau concernant l'environnement – par exemple reconnaitre tel ou tel objet – en vue d'interagir avec celui-ci.
Ce programme général est loin d'être résolu, mais en empruntant à de nombreuses disciplines, telles les mathématiques, le traitement du signal et l'apprentissage, la vision par ordinateur a néanmoins développé un grand nombre d'outils permettant l'analyse et le traitement d'images ou de séquences d'images, ouvrant la voie à des applications dans de nombreux champs.
En parallèle d'applications dont le bénéfice social est sujet à caution – applications militaires et videosurveillance – la vision par ordinateur a permis des progrès considérables dans le cadre de *l'imagerie médicale*. L'émergence récente de nouvelles modalités d'imagerie (IRM et ses variantes, MEEG, PET...) a permis des avancées importantes en terme de diagnostic de pathologies et de compréhension du fonctionnement des être vivants, mais a également créé un besoin d'outils permettant d'analyser des données de plus en plus volumineuses, dont le traitement manuel par un expert peut s'avérer

très couteux en temps, voire impossible.

Ce travail de thèse propose quelques applications du formalisme des plus courts chemins à la segmentation de structures anatomiques dans des images médicales issues de modalités diverses.

Outline

Chapter I proposes a general viewpoint of shortest paths problems in discrete or continuous spaces, and mentions some applications of this formalism in computer science, as well as in other domains. It introduces some notions about shortest paths in Riemannian manifolds, and in spaces equipped with a *potential*, i.e. in which displacement speed is not necessarily constant in the whole space.

Chapter II details some algorithms to compute shortest paths. The exposition focuses on Dijkstra algorithm in the discrete case, and on *Fast-Marching* in the continuous case. We propose a unified presentation of those two algorithms. A new convergence proof of *Fast-Marching* is proposed in the case of a bidimensional space equipped with an isotropic potential and discretized on a regular grid. Our formalism is extended to more and more general spaces. Finally, we show convergence of *Fast-Marching* on a Riemannian manifold equipped with an anisotropic potential, provided the discretization satisfies some condition we will detail in the sequel.

The next chapter are dedicated to applications of this algorithm to analysis of medical images. A central idea of our work is to compute shortest paths in abstract spaces – derived from the image space – but which contain more information, typically concerning the orientation of the anatomical structures we wish to segment.

Chapter III shows how such a formalism can be used to segment tubular structure in bidimensional images – typically blood vessels, but we will show that it can also be applied to road segmentation in satellite images. Our main contribution is to use a four-dimensional space which takes into account orientation and radius of the vessels. We will show several advantages to use such a space.

We will also apply this framework to the segmentation of cortical images from a blood flow analysis, and propose an extension to an iterative method for the segmentation of a network of tubular structures.

Chapter IV is an application of shortest paths to the analysis of diffusion MRI with high angular resolution data. We will use a space of dimension five to perform this task.

The first appendix tackles problems related with shortest paths computations. It tackles the computation of shortest paths in the presence of a mask which forbids a part of space, as well as the computation of some connectivity measures.
The second appendix consists in an independent work about the semi-automatic labelling of electrodes in Electroencephalography (EEG). This work is a part of a not-yet developed system to quickly obtain tridimensional calibration of electrodes during EEG experiments.

All this work has given rise to publications in computer vision and medical imaging conferences. Chapter III is adapted from the article *Extraction of Tubular Structures over an orientation domain* published in the conference *Computer Vision and Pattern Recognition 2009*[167], with Gabriel Peyré and Renaud Keriven, and of *SIFT-based Sequence Registration and Flow-based Cortical Vessel Segmentation applied to High Resolution Optical Imaging Data*[168], published in *International Symposium on Biomedical Imaging 2008* with Thomas Deneux, Ivo Vanzetta and Renaud Keriven. The end of the chapter is published as a research report, and is currently under review in *Medical Image Analysis*. A part of the work exposed in chapter IV was published in *Medical Image Computing and Computer Assisted Intervention 2009*, with Maxime Descoteaux and Renaud Keriven.
Finally, appendix B corresponds to an independent work published in the *Medical Image Computing and Computer Assisted Intervention 2007*[166] with Renaud Keriven, Théodore Papadopoulo and Jean-Michel Badier.

Implementations were done mainly in *C++*, using the *CertisLib* library, developed by the CERTIS team. Visualisation and analysis of data were performed using *Matlab*, *Paraview* and *BrainVizu* for the work presented in chapter IV.

Chapitre I

Shortest paths

> « Le chemin le plus court d'un point à un autre est la ligne droite,
> à condition que les deux points soient bien en face l'un de l'autre. »
>
> ("The shortest path from one point to another is the straight line,
> provided that the two points are squarely in front of each other")
>
> (Pierre Dac, Francis Blanche)

Introduction

Computing shortest path is a specific instance of optimisation problem, and a major human concern : each time we ask ourselves if we would better take motorway rather than trunk road, which route to follow in order to go from Saint-Malo to Pointe-à-Pitre, how to solve a Rubik's Cube in a minimal number of moves, how to get our knight from c3 to e5, or if we should take line 6 then 13 rather than 8 then 13 to go from Daumesnil to Varenne, we are attempting to solve a shortest path problem – or at least to compare several paths which have the same extremities – typically the place where we are, and the place where we want to go.

In view of the diversity of these problem, the commonplace that the shortest path between two points is a straight line is clearly not sufficient. Short does it mean short in space or in time ? What is a straight line in a Rubik's Cube ? In short, what are we talking about exactly ?

- In which *space* are we moving ? Is it a tridimensional Euclidean space ? A relativist space ? A discrete space ? Are there obstacles ?
- What is a *path* in that space ?
- What is the *cost* associated with this path ? For example, do we want to find the shortest or the quickest path ?

Now come the more alarming questions :

- is there a (one only) shortest path ?

And, if the answer is yes,

- how to compute it ?

Section I.1 describes a general framework for shortest paths, and enunciates some basic properties. Section I.2 focuses on discrete shortest paths. Finally, section I.3 describes many framework for continuous shortest paths problems, details some applications, and gives some mathematical properties of the considered spaces.

Contents

I.1 Generalities on shortest paths

Let E be a set.

In this work, we are interested in a specific class of shortest path problems. In particular, we will impose that :

- a path has a starting point and an ending point – if we denote by $\mathcal{C}_{st}$ the set of paths from $s \in E$ to $t \in E$, the set $\{\mathcal{C}_{st} \,|(s,t) \in E^2\}$ is a partition (which can contain the empty set) of the set of all the paths $\mathcal{C}$,
- the paths can be concatenated if they are compatibles : if $C_1 \in \mathcal{C}_{XY}$ and $C_2 \in \mathcal{C}_{YZ}$, then $C_1@C_2 \in \mathcal{C}_{XZ}$: @ is a partial associative binary operation on the set of paths $\mathcal{C}$,
- a cost function c from $\mathcal{C}$ to F is defined – where F is an ordered set equipped with a binary operation $+$ which is compatible with the order relation. We furthermore impose that $c(C_1@C_2) = c(C_1) + c(C_2)$ for all compatible paths.
- for all $s \in E$, there exist a path in $\mathcal{C}_{ss}$ with null cost, and neutral for @.

Let us also introduce the notion of *subpath* :

Definition I.1.0.1 (Subpath)
Let $C \in \mathcal{C}_{st}$ be a path from s to t. C' is a subpath of C if and only if there exists two paths C_1 and C_2 such that $C = C_1@C'@C_2$.

In the sequel, we will focus on the $F = \mathbb{R}^+$ case, in which the cost can be naturally interpreted in terms of length (or duration) of the path.
We then define the *distance* between two points s and t by :

$$d(s,t) \stackrel{\text{def.}}{=} \begin{cases} \inf\limits_{\gamma \in \mathcal{C}_{st}} c(\gamma) & \text{if } \mathcal{C}_{st} \neq \emptyset \\ +\infty & \textit{otherwise} \end{cases} \tag{I.1.1}$$

We then have

Proposition I.1.0.1
d satisfies the triangular inequality

Proof : Let us consider s, t and $u \in E$. If $\mathcal{C}_{st} = \emptyset$ or $\mathcal{C}_{tu} = \emptyset$, we clearly have $d(s,u) \leq d(s,t) + d(t,u)$. Otherwise, we chose $\epsilon > 0$. By definition of $d(s,t)$ and $d(t,u)$, there exists two paths $\gamma_1 \in \mathcal{C}_{st}$ and $\gamma_2 \in \mathcal{C}_{tu}$ such that $c(\gamma_1) \leq d(s,t) + \epsilon/2$ and $c(\gamma_2) \leq d(t,u) + \epsilon/2$. Then, $\gamma_1@\gamma_2 \in \mathcal{C}_{st}$ and $c(\gamma_1@\gamma_2) \leq d(s,t) + d(t,u) + \epsilon$, hence $d(s,u) \leq d(s,t) + d(t,u)$.

□

If we fix a point $s \in E$, and if the distance from s to any other point in E is finite, we get the following function :

$$\mathcal{U}_s \overset{\text{def.}}{=} \left\{ \begin{array}{ccc} E & \to & \mathbb{R}^+ \\ t & \mapsto & d(s,t) \end{array} \right. \tag{I.1.2}$$

called *distance map* from s.

We call *shortest path* between two points s and t any path of $\mathcal{C}_{st}$ with length $d(s,t)$:

$$\gamma^*(s,t) \overset{\text{def.}}{=} \underset{\gamma \in \mathcal{C}_{st}}{\operatorname{argmin}}\ c(\gamma) \tag{I.1.3}$$

Existence or uniqueness of shortest paths are not guaranteed, and strongly depends on the properties on the space E.
Equipped with this nutshell formalism, we already can enunciate the following property :

Proposition I.1.0.2
A subpath of a shortest path is a shortest path.

Proof : Let $C \in \mathcal{C}_{st}$ be a shortest path, and $C' \in \mathcal{C}_{uv}$ a subpath of C. Let C_1 and C_2 be two paths such that $C = C_1 @ C' @ C_2$. We have $c(C) = c(C_1) + c(C') + c(C_2)$. Let us assume that C' is not a shortest path from u to v. Then there exists a path $C'' \in \mathcal{C}_{uv}$ such that $c(C'') < c(C')$. Then, $C_1 @ C'' @ C_2 \in \mathcal{C}_{st}$ and $c(C_1 @ C'' @ C_2) = c(C_1) + c(C'') + c(C_2) < c(C)$, which is absurd.

□

Shortest paths between sets The shortest path notion can be generalised to starting and ending sets of points.
If $S \subset E$ and $T \subset E$, we can define the set of all path from S to T as

$$\mathcal{C}_{ST} \overset{\text{def.}}{=} \bigcup_{\substack{s \in S \\ t \in T}} \mathcal{C}_{st} \tag{I.1.4}$$

and then the distance between those sets as

$$d(S,T) \overset{\text{def.}}{=} \left\{ \begin{array}{cc} \inf\limits_{\gamma \in \mathcal{C}_{ST}} c(\gamma) & \text{if } \mathcal{C}_{st} \neq 0 \\ +\infty & \textit{otherwise} \end{array} \right. = \inf_{\substack{s \in S \\ t \in T}} d(s,t) \tag{I.1.5}$$

along with the distance map from S :

$$\mathcal{U}_S \overset{\text{def.}}{=} \begin{cases} E & \rightarrow & \mathbb{R}^+ \\ t & \mapsto & d(S,\{t\}) \end{cases} \tag{I.1.6}$$

A shortest path between S and T is a path reaching the distance (if it exists) :

$$\gamma^*(S,T) \overset{\text{def.}}{=} \underset{\gamma \in \mathcal{C}_{ST}}{\operatorname{argmin}}\ c(\gamma) \tag{I.1.7}$$

In the sequel, we will handle shortest paths and distance maps problems in which the entire space is know *a priori* – not discovered progressively during computation, which is often the case for motion planning problems in robotics.
Such problems can be classified in two main classes, depending on the continuous or discrete character of the space E.

I.2 Discrete Shortest Paths

Most of the discrete shortest paths problems can be recast in graph theory terms. A very good introduction to graph theory and its algorithms can be found in [3].

I.2.1 Directed graphs

Let (S, A, w) be a graph in which S is a finite set of vertices, $A \subset S \times S$ is the set of edges linking the vertices and $w : A \rightarrow \mathbb{R}$ is a weight function defined on the edges.
We call *path* from $s \in S$ to $t \in S$ any succession $(s_0, a_0, s_1, \ldots, a_{m-1}, s_m)$ $m \in \mathbb{N}$ of edges and vertices such that

- $s_0 = s$
- $s_m = t$
- $\forall i \in [0, m-1]$ $a_i = (s_i, s_{i+1})$.

We thus define the concatenation of two compatible paths in the following way :

$$\begin{aligned}(s_0, a_0, \dots a_{m-1}, s_m)@(t_0, b_0, \dots b_{n-1}, t_n) = \\ (s_0, a_0, \dots a_{m-1}, s_m = t_0, b_0, \dots b_{n-1}, t_n)\end{aligned} \tag{I.2.1}$$

and the *length* of the path (figure 1.1) $\gamma = (s_0, a_0, s_1, \dots, a_m, s_m)$ as

$$c(\gamma) \stackrel{\text{def.}}{=} \sum_{i=1}^{m} w(a_i) \tag{I.2.2}$$

In particular, (s_0) is a path from s_0 to itself, of null length.

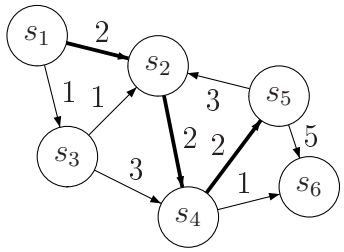


FIG. 1.1 – An example of graph. $(s_1, (s_1, s_2), s_2, (s_2, s_4), s_4, (s_4, s_5), s_5)$ is a path from s_1 to s_5 of length 6.

I.2.2 Undirected Graph

Shortest paths problems on undirected graphs is a specific case of the previous problem. To any undirected graph, we can associate a directed graph by replacing every edge by two opposite edges of same weight.

I.2.3 Existence and uniqueness of shortest paths

The existence of a shortest path is not guaranteed on a graph :

- there can be no path between two vertices (figure 1.2, left).
- there can a path but no minimal path (figure 1.2, right).

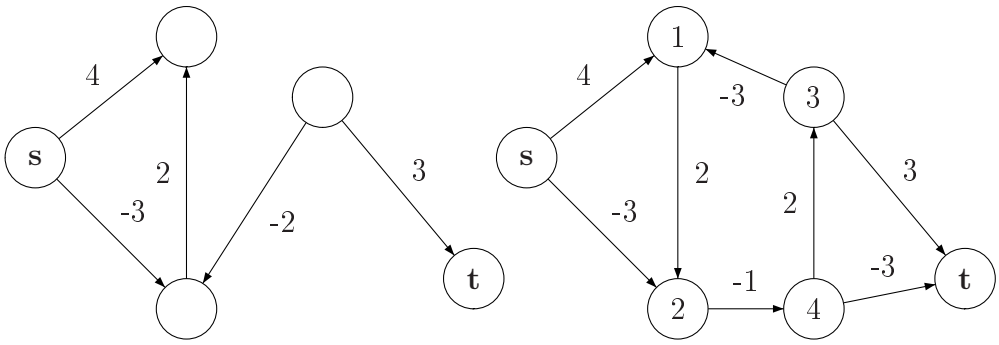


FIG. 1.2 – *Left :* a graph without path from s to t. *Right :* a graph without shortest path from s to t. The negative length loop $(1,(1,2),2,(2,4),4,(4,3),3,(3,1),1$ allows to find arbitrary small paths between the points.

However, we have the following property :

Proposition I.2.3.1
Let (S,A,w) be a graph, such that $w : A \to \mathbb{R}^+$. Let s and $t \in S$.
If $\mathcal{C}_{st} \neq \emptyset$, then a shortest path exists from s to t.

Proof :

- First, notice that for all path in $\mathcal{C}_{st}$, there exist a shortest path without loop in $\mathcal{C}_{st}$,
- then notice that there is a finite number of paths without loop from s to t, hence the existence of a minimal length path, which is also a shortest path in $\mathcal{C}_{st}$.

□

In particular, in a strongly connected graph with positive weights, shortest paths exist between any pair of vertices.

Here is an interesting property of shortest paths between ont vertex s and all other vertices.

Proposition I.2.3.2
Let (S, A, w) be a graph, with $w : A \to \mathbb{R}^+$. Let $s \in S$. Then, there exist a tree $\mathcal{A}$ build from S such that

- *s is the root of the tree ;*
- *t is a node of the tree if and only if there exists a shortest path from s to t ;*
- *if the paternity relation in $\mathcal{A}$ is denoted by p, $(s \dots p(t), (p(t), t), t)$ is a shortest path from s to t.*

Proof :
This tree is simply a set of edges which contains a shortest path from s to all accessible vertex t, and which is minimal for inclusion.

□

Such a tree is called *shortest paths tree* (figure 1.3).

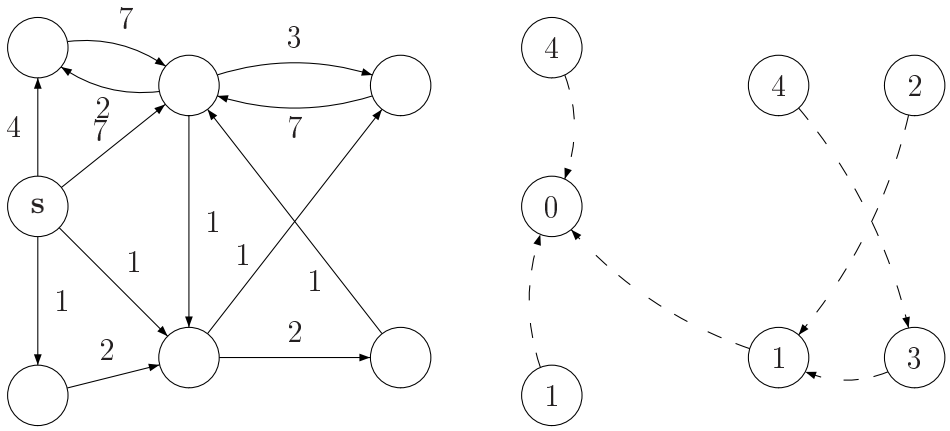


FIG. 1.3 – Distance map and shortest path tree from vertex s

In section II.1.1, we will explain how to compute such distance maps and shortest path trees.

I.2.4 Applications

A huge amount of problems can be recast in this framework. It is pointless to try and draw up a complete map of possible applications. Let us cite some classical problems.

- A classical application of shortest paths on graph is the computation of trajectories over transportation networks. Edges correspond to portions of roads, and vertices to intersections. Weights account for the time to travel along a portion of road.
- This framework is used to compute routing in electronic data networks [205]. Vertices represent routers (or other nodes), and edges represent connexions between routers. Weights depend on the available bandwidth.
- The computation of knight moves we mentioned earlier can be casted in a search of shortest path on a graph (figure 1.4). More generally, for all system with a finite number of states, and transitions between those state, finding paths between two states can be done by computing shortest paths in a graph with constant weights.

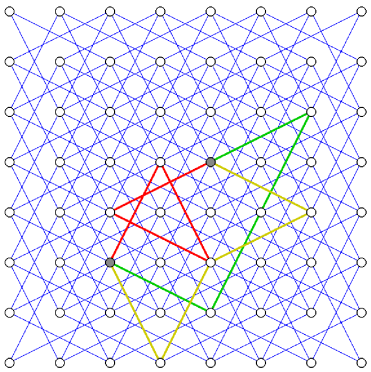


FIG. 1.4 – Undirected graph corresponding to the possible moves of a knight over a chessboard. Each vertex corresponds to a square, each edge to a possible move. In red, yellow, and green : three paths from e3 to c5 in a minimal number of moves.

- More generally, all dynamic programming problems can be formulated in terms of shortest paths problem in a graph. [131].
- Some linear programming problem can be recast in discrete shortest paths

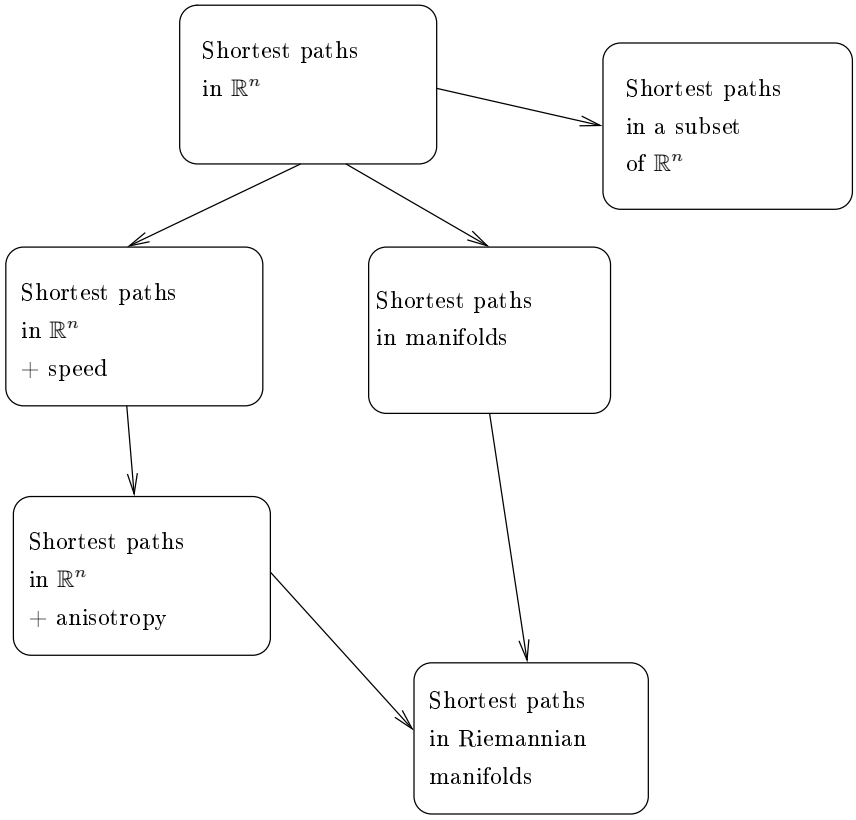


FIG. 1.5 – Different continuous shortest paths problems

computation [40].

- Many motion planning problems can also be formulated in this framework [113].

I.3 Continuous shortest paths and distance maps

I.3.1 Different frameworks for continuous shortest paths

In this section, we will present some shortest paths problems which occur in different domains, and require different theoretical frameworks. Figure 1.5 synthesises these frameworks.

Let us consider a continuous space E – for practical purposes, E is generally a subset of $\mathbb{R}^n$ or a finite dimension manifold.

We define a path as a function $\mathcal{C}^0$ and piecewise-$\mathcal{C}^1$ from $[0,1]$ to E – this

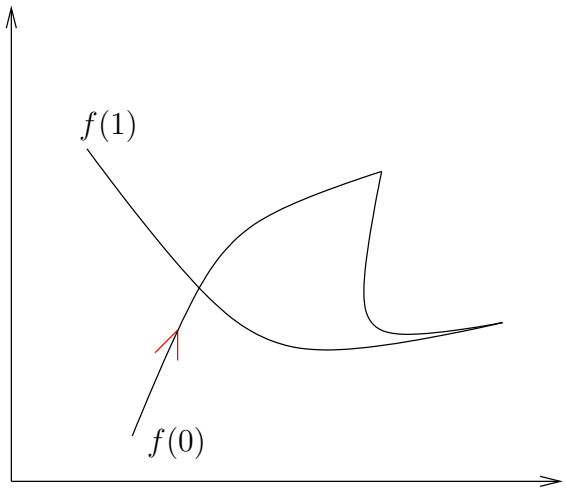


FIG. 1.6 – A path in $\mathbb{R}^2$

condition being sufficient in order to define the length of such a curve [1]. If f is such a path, the *extremities* of f are $f(0)$ and $f(1)$ (figure 1.6).
The concatenation of two paths $f_1 : [0,1] \to E$ and $f_2 : [0,1] \to E$ is defined as

$$f_1 @ f_2 \stackrel{\text{def.}}{=} \begin{cases} t \in [0,1/2] & \mapsto & f_1(2t) \\ t \in]1/2,1] & \mapsto & f_2(2t-1) \end{cases} \tag{I.3.1}$$

I.3.1.1 Shortest paths in $\mathbb{R}^n$: the straight line strikes back

Let us define $E = \mathbb{R}^n$, with $n \geq 1$.
The length of a curve f is defined by

$$\mathcal{L}(f) \stackrel{\text{def.}}{=} \int_0^1 \|f'(t)\| dt \tag{I.3.2}$$

Notice that this quantity is invariant when one reparametrise the curve. In particular, if we use the arc length, we get

$$\mathcal{L}(f) = \int_0^{\mathcal{L}(f)} \|f'(s)\| ds \tag{I.3.3}$$

[1] still, it is not necessary : it is possible to define a length for a more general class of curves, namely rectifiable curves

with $\|f'(s)\| = 1$.
In this framework, we can show the following property :

Proposition I.3.1.1
Let x and $y \in E$. Then, the shortest path from x to y is the straight line, i.e. $f^* : t \mapsto x + t\frac{y-x}{\|y-x\|}$

Proof :
If f is a curve with x and y extremities, we have $\mathcal{L}(f^*) = \int_0^1 \|f^{*\prime}(t)\|dt = \|x-y\| = \|\int_0^1 f'(t)dt\| \leq \int_0^1 \|f'(t)\|dt = \mathcal{L}(f)$

□

The distance between two points corresponds with Euclidean distance, *i.e.* $\mathcal{U}_s(t) = \|t-s\|_2$. In particular, level sets of distance map are circles in 2D, and spheres in 3D (figure 1.7).

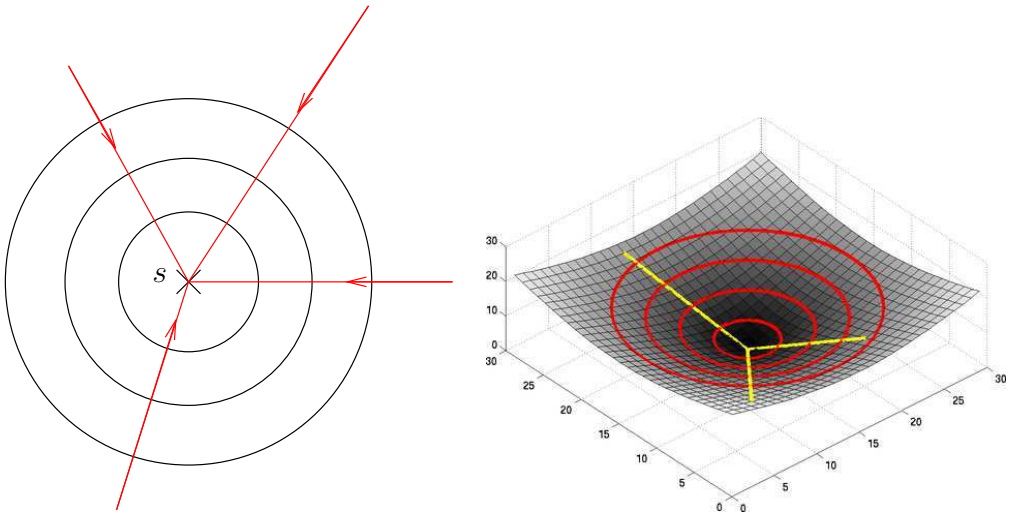


FIG. 1.7 – Shortest paths in $\mathbb{R}^2$. Level sets of distance map are circles, and shortest paths are segments.

Things are becoming more interesting when one consider a set S of starting points instead of one point. If S is a closed set, a compacity argument easily shows the existence of a shortest path from S to any point of E. This shortest path is still a straight line. Figures 1.8 and 1.9 demonstrate this on two examples.

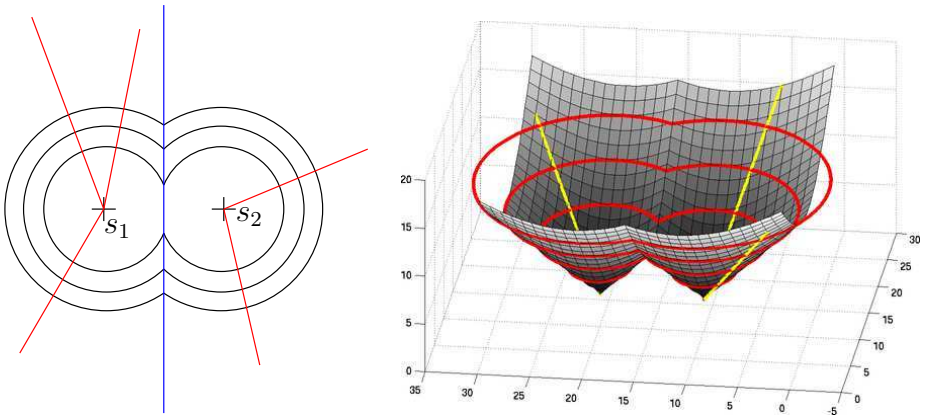


FIG. 1.8 – Shortest paths to 2 points s_1 and s_2. Level sets are represented in black, and shortest paths in red. The blue line correspond to points which are equidistant from s_1 and s_2, *i.e.* to the shock points of the fronts emanating from s_1 and s_2.

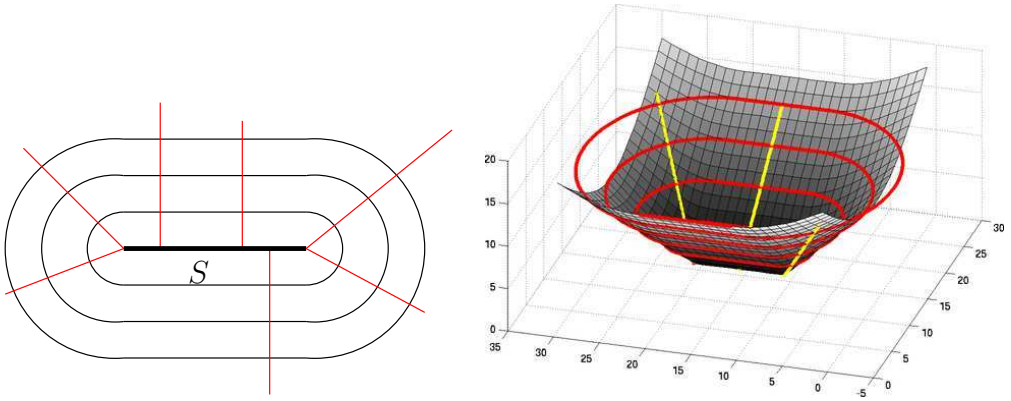


FIG. 1.9 – Shortest paths to a segment. Level sets are represented in black, and shortest paths in red.

These results can be interpreted in terms of front propagation [188] : let us consider a starting set S, and a front propagating outside S with a constant speed 1. The $\mathcal{U}_S(t) = \alpha$ level set corresponds to the front position after a time α. A classical analogy is the propagation of a forest fire propagating at constant speed from an initial hearth [30].

Applications The interpretation in terms of front is extremely productive. As illustrated figure 1.8, the meeting points between two fronts correspond

to points that are equidistant from starting points in S. This results can be generalised to an arbitrary number of starting points, which paves the way toward two applications : the calculation of Voronoi diagrams [188] if S is discrete, and skeletization of objects Ω by using $S = \delta\Omega$ [204, 173, 78].
Another application is shape offsetting, which consists in finding level sets of distance functions to a given set [188] – with applications to growing of obstacles in motion planning.
Computing the distance function to a closed curve is also a classical step of *level sets* [188] implementations – in order to obtained a smooth function which zero level set correspond to the curve.
This formalism was also used in images denoising in 2D or 3D, leading to algorithms that can guarantee topological properties of the object to be denoised, such that homotopy to a sphere [108, 202, 13].
A list of other applications can also be found in [57].

I.3.1.2 Shortest paths on a subset of $\mathbb{R}^n$

From a mathematical point of view, things become much more complex as soon as the considered space is a strict subset of $\mathbb{R}^n$.
As an example, let us consider the plane without the origin. There is no shortest path from $(1, 0)$ to $(0, 1)$: paths with a length arbitrarily close to 2 exist, but no path of length 2(figure 1.10).

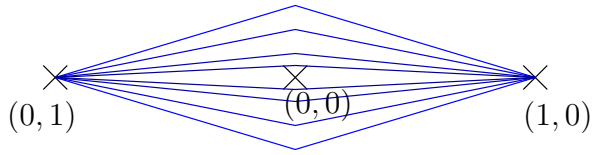


FIG. 1.10 – Existence of shortest path is not guaranteed in a subset of the plane

However existence of shortest paths is guaranteed for several specific cases.

Applications. An important application of this formalism is again motion planning : this calculation allow to plan trajectories of robots in an environment consisting of obstacles, or in which some position are forbidden.

Approximate numerical methods exists for general spaces [99, 96, 68, 69]. However, it is often advantageous to take advantage of the shape of obstacles in order to obtain dedicated exact algorithms.
For example, if the obstacles are open polygons, it is possible to compute an exact solution in polynomial time. Indeed, this problem can be reduces to a calculation of discrete shortest path in a *visibility graph*, or by decomposing space into connected cells [138]. An interesting introduction to this topic can be found in [113].

I.3.1.3 Local speed

Let E be a subset of $\mathbb{R}^n$, and let us consider a mapping $\rho : E \to \mathbb{R}^{+*}$ – called *potential* over E.
Keeping the curve definition above, we can define the length of a curve with respect to this potential as

$$\mathcal{L}_\rho(f) \stackrel{\text{def.}}{=} \int_0^1 \rho(f(t))\|f'(t)\|dt \tag{I.3.4}$$

If we parametrise the curve by arc length, we have

$$\mathcal{L}_\rho(f) = \int_0^{\mathcal{L}(f)} \rho(f(s))ds \tag{I.3.5}$$

hence

$$\frac{\mathcal{L}_\rho(f)}{\mathcal{L}(f)} =< \rho >_f \tag{I.3.6}$$

where $< \rho >_f$ is the average value of ρ along the curve. If we interpret $\mathcal{L}_\rho(f)$ as a *travel time* to go from $f(0)$ to $f(1)$, $\mathcal{L}(f)$ being the Euclidean length of the curve, ρ can be seen as the inverse of a local *speed* of displacement.
For practical purposes, as soon as the considered potential map is non-trivial, there is no analytical form for the shortest paths. Solution will not reside anymore in exact algorithms, but on numerical methods leading to approximate solutions – one of these methods will be thoroughly detailed in II.3.
However, exact algorithms exists in some specific cases, for example if the 2D space is partitioned into polygons in which speed is constant [137].

From a theoretical point of view, there is no general guarantee of the existence of shortest paths. As an example, let us consider the space $E = [0,1]^2$ with potential

$$\rho \overset{\text{def.}}{=} (x,y) \rightarrow \begin{cases} 1 & \text{si } y < 1/2 \\ 2 & \text{otherwise} \end{cases}$$

Then, there is no shortest path from $(0,1/2)$ to $(1,1/2)$.
Nevertheless, we will see an existence theorem for shortest paths in a more general framework, which guarantees in particular the existence of shortest paths in $[0,1]^n$ when ρ is continuous.

Applications

This problem is particularly important from an historical point of view in geometrical optics [110]. The *refractive index* of a medium is defined as the ratio between light celerity in void over light celerity in that medium, *i.e.* $n \overset{\text{def.}}{=} \frac{c}{v}$. The *Fermat's principle* enunciates that the trajectories followed by light rays are of extremal duration. If we consider a medium E and an index $n(x)$ for each point, the duration of light journey along a trajectory f is thus given by

$$\mathcal{L}(f) = \frac{1}{c} \int_0^1 n(f(t)) \|f'(t)\| dt \tag{I.3.7}$$

Shortest paths for this length correspond to possible trajectories of light rays – and in particular we find that lights propagates along straight lines in an homogeneous medium.
The refraction laws (or Snell-Descartes' laws), which describe the behaviour of light rays at the interface between two homogeneous media can also be retrieved from this equation (figure 1.11).
This formalism is also applied in image analysis to contour segmentation either as an alternative to active contours [38] or in the framework of segmentation by region growing [127, 132, 49]. In [65, 95], criteria similar to I.3.6 were also analysed and used to perform contour segmentation. The basic idea – which we will detail further in chapter II – is to compute shortest paths or distance maps in the image plane, setting the potential such that shortest

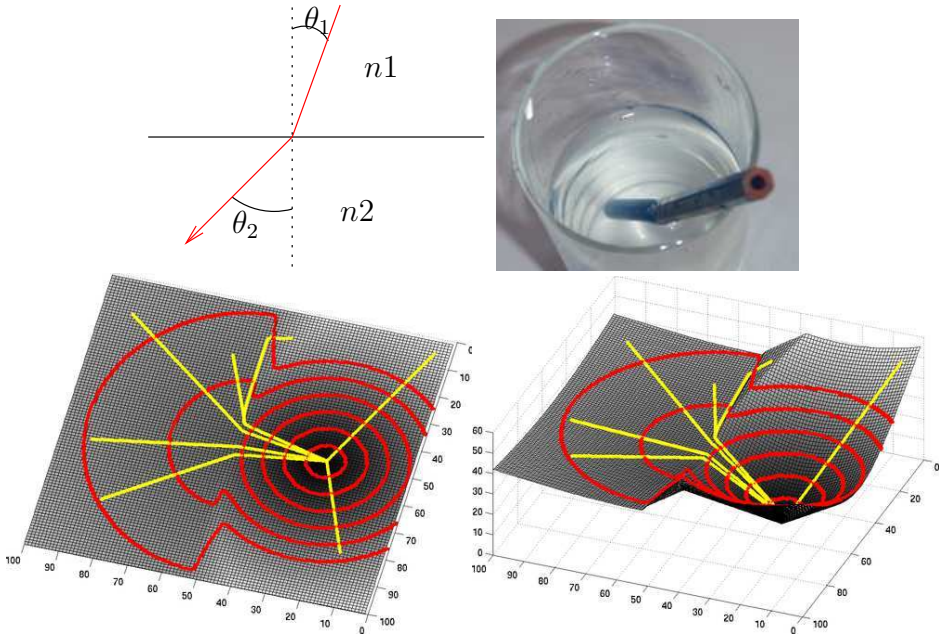


FIG. 1.11 – *Left :* refraction of a light ray at the interface of two homogeneous media of index n_1 and n_2 with $n_2 < n_1$. The ray follows the shortest path, and its trajectory tends to remain longer in the half-plane with smaller index. More precisely, we have $\sin(\theta_1)n_1 = \sin(\theta_2)n_2$. *Right :* illustration of this phenomenon at the interface between air and water – water index being approximately 1.3 times bigger. *Bottom :* distance map and shortest paths in a plane separated in two domains of indices 1 and 4 – s being in the area of bigger index.

paths or level sets of distance maps follow interesting curves in the image – e.g. blood vessels or objects contours.
This kind of methods can be paired with watershed algorithms in order to segment cluster of objects [144].
Distance map and front calculation is also used to estimate arrival times for seismic waves in Earth's mantle [189], or to model propagation of electric signals in the framework of human heart electrophysiological modelling [184].
In [64], the same formalism is used in order to compute correspondences between curves.
Finally, the *shape from shading* – which consists in reconstructing a tridimensional shape from its illumination – needs the use of a formalism close to the one of distance maps [94, 174, 99, 162]. It is also the case for the problem of reconstructing a depth map from the normals of the surface [81].

I.3.1.4 Anisotropy

Here is another interpretation of shortest paths computation in a space with a potential. Let E and ρ be defined as previously. Let us define for all $x \in E$ and for all $\mathbf{v} \in \mathbb{R}^n$

$$\|\mathbf{v}\|_x \stackrel{\text{def.}}{=} \rho(x)\|\mathbf{v}\|. \tag{I.3.8}$$

If we consider equation (I.3.4), we have,

$$\mathcal{L}_\rho(f) = \int_0^1 \|f'(t)\|_{f(t)} dt \tag{I.3.9}$$

We meet again the definition of length proposed in (I.3.2), but in a space equipped with a different metric.
From now, it is easy to generalise these definitions to anisotropic metrics, for which the potential depends not only on the location, but also on the orientation of the curve.
Let E be a subset of $\mathbb{R}^n$, and let us consider a mapping $g : E \to \mathcal{S}_n^+(\mathbb{R})$, where $\mathcal{S}_n^+(\mathbb{R})$ is the set of symmetric positive definite matrices of size $n \times n$.
For every point $x \in E$, it defines a metric : for all $\mathbf{v} \in \mathbb{R}^n$, we define

$$\|\mathbf{v}\|_{g(x)} \stackrel{\text{def.}}{=} \sqrt{\mathbf{v}^T g(x) \mathbf{v}} \tag{I.3.10}$$

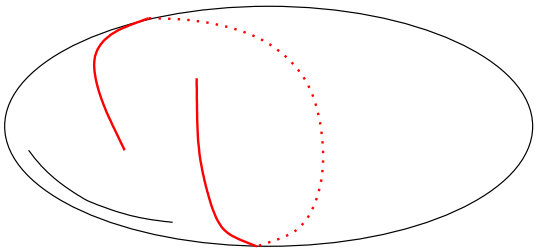


FIG. 1.12 – A path on an ellipsoid

Then, we can define again the length of a curve as follows :

$$\mathcal{L}_g(f) \overset{\text{def.}}{=} \int_0^1 \|f'(t)\|_{g(f(t))} dt \tag{I.3.11}$$

(I.3.8) is therefore a specific case of this equation when $g(x) = \rho(x)^2 I_n$ is an homothety.

Applications

This formalism was mainly used in medical imaging in order to model electrophysiological phenomena [185], or to reconstruct fibers if diffusion tensor imaging [87, 145].

I.3.1.5 Shortest paths in a manifold

Let $E = V$ be a manifold [176] of dimension k, embedded in $\mathbb{R}^n$ for some $n \in \mathbb{N}$.

We can define the length of a curve on this manifold as

$$\mathcal{L}(f) \overset{\text{def.}}{=} \int_0^1 \|f'(t)\| dt \tag{I.3.12}$$

where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^n$.

Figure 1.12 shows an example of path on an ellipsoid embedded in $\mathbb{R}^3$.

Shortest paths problems on manifolds appear in trajectories issues on the surface of the Earth – which is of capital interest to allow navigators to reach one point from another as fast as possible.

It is possible to show that shortest paths on this surface are portions of great circles on the sphere (figure 1.13). This result is known at least from Aristotle.

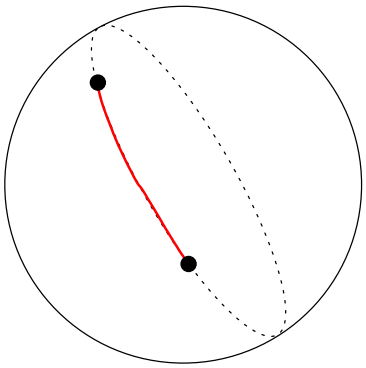


FIG. 1.13 – Shortest paths on a sphere are portions of great circles

A geometrical demonstration of this result is proposed in [124], as well as results concerning shortest paths on cylinder or cones.
Leonhard Euler [55] was the first one to tackle this problem on a general surface. For a convex surface, Euler solves it by noticing that a shortest path between two points corresponds to a tighten thread on the surface going through those points.
The shortest path notion on a manifold is related with the notion of *geodesic*. A geodesic is defined a a curve with normal acceleration on the manifold. It is possible to show that a shortest path on a manifold is a geodesic, and that a geodesic is *locally* a shortest path [66]. These to notions are generally mixed up in the computer vision community, and we will use either of the two words in the sequel.

Applications
Several geometric problems are based on shortest paths computations on manifolds : parametrisation of surfaces [182], sampling of surfaces [155], Voronoi diagrams calculation on surfaces [188, 99].
Shortest path calculation on polyhedra was also studied, an is applied to problem in motion planning [2, 158, 157].

I.3.1.6 Shortest paths on a Riemannian manifold

Let V be a manifold of dimension k embedded in $\mathbb{R}^n$ for some $n \in \mathbb{N}$. For all $x \in V$, let us denote by $T_x(V)$ the *tangent space* of V at x. It is a vector

space of dimension k such that for any curve f on the manifold, and for all point $f(t) \in V$ on the curve, $f'(t) \in T_{f(t)}(V)$.

A *Riemannian manifold* (V, g) is defined as a manifold V such that for all point $x \in V$, $T_x(V)$ is equipped with a symmetric positive definite bilinear form $g(x) : T_x(V) \times T_x(V) \rightarrow \mathbb{R}^+$ called *potential*. Usually, a continuity constraint is imposed for g. Excellent introduction to the study of Riemannian manifolds can be found in [116] and [66].

$$\|\mathbf{v}\|_x \stackrel{\text{def.}}{=} \sqrt{g(x)(\mathbf{v}, \mathbf{v})} \tag{I.3.13}$$

We can now define the length of a curve on V by

$$\mathcal{L}_g(f) \stackrel{\text{def.}}{=} \int_0^1 \|f'(t)\|_{f(t)} dt \tag{I.3.14}$$

g can be interpreted as the inverse of a speed tensor.

Therefore, shortest paths on manifolds is a specific case of this framework, in which $g(x)(\mathbf{v}, \mathbf{v})$ corresponds with the Euclidean norm of the embedding space.

Applications When the potential is isotropic, this formalism can be used to segment targeted curves on surfaces. In [203, 10], the authors proposed to use it to segment sulci on cortical surface. It was also used to segment surfaces in tridimensional images – viewed as an union of shortest paths belonging to that surface [6, 7].

I.3.2 Theoretical aspects

It is possible to prove existence of geodesics and to demonstrate properties of the distance maps in the most general case we have mentioned. Complete proofs come under non-trivial mathematics, and will not be detailed here. However, we will give intuitive proofs in a few specific cases.

I.3.2.1 Distance

Proposition I.3.2.1
Let (V, g) be a connected complete Riemannian manifold, equipped with a continuous metric. We define the length $\mathcal{L}$ of a curve as previously.
Then, the mapping d induced by $\mathcal{L}$ (I.1.1) is a distance function.

Proof :

We present a sketch of proof. A complete one can be found in [66].

Triangular inequality holds from I.1.0.1.

Symmetry is derived from the possibility of travelling on the curve in both directions : if f is a path from a to b, then $t \to f(1-t)$ is a path from b to a of same length. Symmetry is obtained by considering the infimum of length of all paths from a to b.

The definite character is more difficult to show. Let us consider two distinct points s and t. We will prove that $d(s,t) > 0$. Let us embed the manifold in some space $\mathbb{R}^n$ and let us consider the compact set $B_s \overset{\text{def.}}{=} V \cap B(s, \frac{\|s-t\|}{2})$. By continuity of the metric, there exist $\epsilon \in \mathbb{R}$ such that for all $x \in B_s$ and for all $v \in T_x(V)$ $g(x)(v,v) > \epsilon\|v\|$. Let us consider a path from s to t, and denote $t_0 \overset{\text{def.}}{=} \inf_{t\in[0,1]} \{t \in [0,1] \mid f(t) \notin B_s\}$. Then, $\mathcal{L}_\rho(f) \geq \int_0^{t_0} \|f'(t)\|_{f(t)} dt \geq \epsilon \int_0^{t_0} \|f'(t)\| \geq \epsilon | \int_0^{t_0} f'(t)| = \epsilon \frac{\|s-t\|}{2}$. Therefore, we have $d(s,t) \geq \epsilon \frac{\|s-t\|}{2} > 0$

□

I.3.2.2 Geodesics

The Hopf-Rinow theorem [83] guarantees the existence of geodesics for a large class of Riemannian manifolds.

Theorem I.3.2.2 (Hopf-Rinow)
For any complete connected Riemannian manifold, and for any couple of points (s,t) of the manifold, there exists a geodesic of minimal length between s and t.

I.3.2.3 Distance map properties

Recall that the distance map $\mathcal{U}(S)$ to a starting set is defined as $\mathcal{U}(S,t) \overset{\text{def.}}{=} d(S,\{t\})$ for all $t \in V$. We can check easily that

Proposition I.3.2.3
Let (V,g) be a complete connected Riemannian manifold, equipped with a continuous metric. Let $S \subset V$ be a compact set.
Then $\mathcal{U}(S)$ is continuous.

In particular, the distance map to a single point is continuous.

However, even in the simplest cases, the distance map if not differentiable. As an example, in the case of distance map to a point in $\mathbb{R}^2$ (figure 1.7), $\mathcal{U}_s$ is not differentiable at point s. In the case of distance map to two points (figure 1.8), the distance map is also not differentiable at points which are equidistant from the two starting points.
Yet, we have the following property.

Proposition I.3.2.4
If $\mathcal{U}_s$ is differentiable at t, then $|\nabla_t \mathcal{U}_s|_{[g(t)]^{-1}} = 1$.

Proof :
We provide a proof when $E = \mathbb{R}^n$, equipped with a potential ρ. It can be extended to any Riemannian space, but this requires technical tools we will not develop here.
If $\mathcal{U}_s$ is differentiable at t, we can write

$$\mathcal{U}_s(t + \mathbf{dt}) = \mathcal{U}_s(t) + \nabla_t \mathcal{U}_s.\mathbf{dt} + \mathrm{o}(|\mathbf{dt}|).$$

Furthermore, $\mathcal{U}_s(t + \mathbf{dt})$ being the length of the shortest path from s to $t + \mathbf{dt}$, and the norm being continuous, we have

$$\mathcal{U}_s(t + \mathbf{dt}) \leq \mathcal{U}_s(t) + \rho(t)|\mathbf{dt}| + \mathrm{o}(|\mathbf{dt}|).$$

In particular, if we set $\mathbf{dt} = \epsilon \nabla_t \mathcal{U}_s$, and decrease ϵ toward 0, we have

$$|\nabla_t \mathcal{U}_s| \leq \rho(t)$$

Now let us consider a shortest path γ from s to t. We set $\mathbf{dt} = \gamma(1) - \gamma(1-\epsilon)$.
Let us define $U(x) \overset{\text{def.}}{=} \mathcal{U}_s(\gamma(x))$. We thus have $U'(x) = \nabla_{\gamma(x)} \mathcal{U}_s \gamma'(x)$, and $x = 1$, $U'(1) = \nabla_t \mathcal{U}_s.\gamma'(1) \leq |\nabla_t \mathcal{U}_s||\gamma'(1)|$.
By the way, we have

$$U(1) - U(1-\epsilon) = \epsilon|\gamma'(t)|\rho(t) + o(\epsilon)$$

hence

$$U'(1) = |\gamma'(t)|\rho(t)$$

and

$$|\nabla_t \mathcal{U}_s|_t \geq \rho(t)$$

hence the result.
On the way, we also proved that $\gamma'(t)$ and $\nabla_t \mathcal{U}_s$ are collinear.

□

Definition I.3.2.1
We call Eikonal equation *the following partial derivative equation :*

$$\|\nabla_x \mathcal{U}_S\|_{g^{-1}(x)} = 1 \quad with \quad \forall s \in S \;\; \mathcal{U}_S(s) = 0 \tag{I.3.15}$$

The previous proposition enunciates that if the distance map is differentiable at some point, it is solution of the Eikonal equation at that point. It would be interesting to obtain a converse of this results, which would characterise $\mathcal{U}_s$ globally as a solution of Eikonal equation. This is a tough problem, since, as we saw, $\mathcal{U}_s$ is not differentiable at any point.
[43] introduced the notion of viscosity solution for a large class of partial differential equations, allowing to circumvent this issue (figure 1.14).

Definition I.3.2.2
We call u *a* viscosity solution *of the Eikonal equation if and only if for any mapping* $\varphi \in \mathcal{C}^1(V)$ *and for all* $x_0 \in V$ *local minimum of* $u - \varphi$ *we have* $\|\nabla_{x_0}\varphi\|_{g^{-1}(x_0)} = 1$

This definition disposes of the differentiability constraint on u. Some physical insight of this notion are detailed in [186].
A specific case of results proved in [43] can be enunciated as follow

Theorem I.3.2.5
Let (V, g) *be a Riemannian manifold, and* $S \subset V$ *a compact set.*
Then, $\mathcal{U}_S$ *is the unique viscosity solution of the Eikonal equation* (I.3.15).

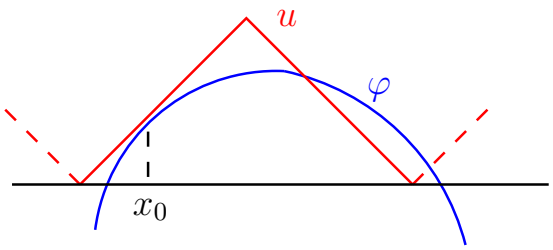


FIG. 1.14 – Illustration of the viscosity solution definition in dimension 1. u is the distance function to two initial points. φ is a $\mathcal{C}^1(\mathbb{R})$ mapping. x_0 is a local minimum $u - \varphi$. Then the equality $|\nabla_{x_0}\varphi| = 1$ holds.

The following theorem also holds.

Theorem I.3.2.6

Let (V, g) be a Riemannian manifold, and s, $t \in V$.
Let γ be a geodesic between s and t. Then, up to parametrisation, γ is solution of the following differential equation

$$\gamma'(t) = -\frac{g(\gamma(t))^{-1}\nabla_{\gamma(t)}\mathcal{U}_s}{\|g(\gamma(t))^{-1}\nabla_{\gamma(t)}\mathcal{U}_s\|} \quad \textit{with} \quad \gamma(0) = t. \tag{I.3.16}$$

Proof :
In the case of $\mathbb{R}^n$ equipped with a potential, we must show that

$$\gamma'(t) = -\frac{\nabla_{\gamma(t)}\mathcal{U}_s}{\|\nabla_{\gamma(t)}\mathcal{U}_s\|} \quad \text{with} \quad \gamma(0) = t. \tag{I.3.17}$$

The proof is immediately derived from collinearity of $\gamma'(t)$ and $\nabla_{\gamma(t)}\mathcal{U}_s$ we observed during the proof of I.3.2.4.
This result can be generalised to Riemannian manifolds.

□

In the case of $\mathbb{R}^n$ equipped with a potential, this implies the orthogonality between shortest paths and level sets of distance map (figures 1.7 1.8 1.9).
The last two results are extremely important for practical computation of shortest paths and distance maps. Computing a distance map is reduced to the problem of approximating the solution of a partial differential equation (we will detail a method to do so in section II.3), and computing a shortest

path boils down to performing a gradient descent on the obtained map – or more precisely a descent along the *characteristics* of the solution [56] – which aligns with with gradient direction in the case of an isotropic potential.

I.4 Conclusion

Shortest paths naturally appear in the modelling of several problems, either in the discrete case (shortest paths in a graph) or in the continuous one (shortest path in $\mathbb{R}^n$ or in a manifold). The explicit calculation of shortest paths is thus of primary interest for the resolution of numerous problems.
The next chapter details some methods allowing to compute exact or approximate solutions to these problems.

Chapitre II

Shortest paths computation

Introduction

This chapter is an attempt to propose a clear presentation of algorithms to compute shortest paths – in particular *Fast-Marching*.
We will present Dijkstra algorithm for computation of shortest paths on graphs (section II.1). Then we will show an fruitless attempt to use this algorithm in a continuous framework (section II.2). We will thus present the state-of-the art solution to this problem – i.e. *Fast-Marching*. . Section II.3 will consist in a full exposition of the method – our formalism being different than the classical one, which will allow both to have a point-of-view unified with Dijkstra algorithm, and to perform easy generalisations. A proof of convergence will be proposed in this case. We will therefore show how to extend this algorithm to any dimension, and to anisotropic potentials with principal components aligned with the grid. Finally, in section II.4 we will detail the algorithm and give a proof in the most general framework – i.e. shortest paths on Riemannian spaces. This presentation, while keeping a geometrical point-of-view, is a generalisation of results indicated in [186] and [30].

Contents

II.1 Discrete shortest paths computation

Depending on the targeted application, numerous methods exist in order to compute discrete shortest paths.
We will restrict ourselves to the the problem of computing distance maps and shortest paths from a fixed set of initial vertices (notice that it is possible to compute distances between any couple of points using algorithms such that Floyd-Warshall and Johnson [181] .)

If negative values are permitted in the graph, there is no guarantee of the existence of shortest paths between two vertices (figure 1.2). In this context, finding a shortest path between to vertices is a NP-complete problem [92]. When the graph does not contain any loop of negative length, one can prove that shortest paths exist, and they can be found using polynomial algorithms such that Bellman-Ford algorithm [123, 1, 139].

In the sequel, we will only consider graphs with positive weights. In this case, our problem can be solved in polynomial time by using Dijkstra algorithm – which we are going to detail.
Notice also that if all weights in the graph are equal to 1, dedicate algorithms exists to compute shortest paths[19].

II.1.1 Dijkstra algorithm

In this section, we will focus on methods to compute distance maps from one given vertex s – along with shortest paths from any other vertex to s. In the sequel, a graph will be denoted as (S, A), where S represents the vertices, and A the edges. Furthermore we will denote $n \overset{\text{def.}}{=} |S|$ and $m \overset{\text{def.}}{=} |A|$. $w : A \to \mathbb{R}^+$ is a weight defined on the edges of the graph. $\mathcal{N}(s) \subset S$ represents the neighbors of s in the graph, and $p \overset{\text{def.}}{=} \max_{s \in S}\{|\mathcal{N}(s)|\}$ is the maximal connectivity for a vertex.
We have the following fundamental property for the distance map $\mathcal{U}_s$ from a vertex s on a graph.

Proposition II.1.1.1
$\mathcal{U}_s(t) = \min_{v \in \mathcal{N}(t)} \mathcal{U}_s(v) + w(v, t)$

Proof :
For any neighbor v of t, we have $\mathcal{U}_s(t) \leq \mathcal{U}_s(v) + w(v, t)$.
Furthermore, let us consider a shortest path from s to t, namely $(s, \dots u, (u, t), t)$ (cf. I.1.0.2). Then, the sub-path $(s, \dots u)$ is a shortest path from s to u. We then have $\mathcal{U}_s(u) = l(s, \dots u)$, and $\mathcal{U}_s(t) = l((s, \dots u, (u, t), t)) = \mathcal{U}_s(u) + w(u, t)$.

□

The proof also shows that if $(s, \dots u, (u, t), t)$ is a shortest path, the minimum in (II.1.1.1) is reached for $v = u$.
Some vertex s being chosen, Dijkstra algorithm [54, 119] allows to compute the distance map $\mathcal{U}_s$ as long as a shortest paths tree in $O(n(log(n)+p))$ time. At any stage, the algorithm keeps up an estimate d of $\mathcal{U}_s$. It is based on a local update routine derived from II.1.1.1. This routine allows to estimate d for a vertex t, knowing values of d for its neighbors. Furthermore, it updates the father $p(t)$ of t in the shortest paths tree.

Algorithm 1 update(t)

INPUT: A vertex t.

for all $v \in \mathcal{N}(t)$ **do**
  if $d(v) + w((v,t)) < d(t)$ **then**
    $d(t) \leftarrow d(v) + w((v,t))$
    $p(t) \leftarrow v$
  end if
end for

Dijkstra algorithm travels all over the graph, and sequentially performs such update steps. The order in which to perform these operations is critical.
A first attempt is to iteratively perform them on all the vertices. One gets the algorithm described in 2.

Algorithm 2 Iterative algorithm for shortest paths computations

INPUT: A graph (S, A), $s \in S$

OUTPUT: $\forall t \in V \;\; d(t) = \mathcal{U}_s(t)$

Initialization:
Set $d(s) = 0$ and $d(t) = +\infty$ for all $t \neq s$.
$p(s) \leftarrow 0$
while convergence is not reached **do**
  for all $t \in S$ **do**
    update(t)
  end for
end while

It is possible to show that n iterations are sufficient to reach convergence. This algorithm thus runs in $O(n^2 p)$ time.
However, it is possible to improve this complexity. Refining II.1.1.1, we obtain :

Proposition II.1.1.2

$$\mathcal{U}_s(t) = \min_{\substack{v \in \mathcal{N}(t) \\ \mathcal{U}_s(v) < \mathcal{U}_s(t)}} \mathcal{U}_s(v) + w(v,t)$$

This means that $\mathcal{U}_s$ value for a given vertex only depends on values of neigh-

boring vertices with lower values – in particular, the *update* routine is performing useless operations.

This property – which introduces a *causality* or *upwinding* notion in $\mathcal{U}_s$ – allows one to design a new dynamic-programming-like algorithm to compute shortest paths : one can compute $\mathcal{U}_s$ for vertices " close" from s, and then extend the computation to further vertices. In the case when weights are all 1 in the graph, this corresponds to a breadth first exploration.

Three disjoint sets of vertices are kept up :

- $\mathcal{A}$ (*alive*) : the set of vertices for which $d = \mathcal{U}_s$.
- $\mathcal{T}$ (*trial*) : the set of vertices for which an estimation d of $\mathcal{U}_s$ is available – i.e. points being considered.
- $\mathcal{F}$ (*far*) : the set of vertices for which no estimation d of $\mathcal{U}_s$ is available

At every iteration, the algorithm selects a vertex $t \in \mathcal{T}$ with minimal $d(t)$ estimation. One can show that $d(t) = \mathcal{U}_s(t)$ for such a vertex. This vertex is transferred in $\mathcal{A}$. Its neighbors are transferred in $\mathcal{T}$, and their estimated distance is updated by using the value found for d The algorithm is synthesised in 3 and 4.

Algorithm 3 update(v,t)

INPUT: A vertex v. A neighboring vertex t.

```
if d(t) + w((t, v)) < d(v) then
    d(v) ← d(t) + w((t, v))
    p(v) ← t
end if
```

Figure 2.1 shows an iteration of the algorithm. At anytime during the computation, $\mathcal{T}$ can be seen as a front propagating from s.

Dijkstra algorithm can be easily generalised to a set of starting vertices S : one just need to replace $\mathcal{T} \leftarrow \{s\}$ with $\mathcal{T} \leftarrow S$, and by setting $\forall s \in S$ $d(s) = 0$ during the initialisation.

Proof of correctness Let us prove the correctness of Dijkstra algorithm. We want to prove that for any vertex t, we have $d(t) = \mathcal{U}_s(t)$ after execution of the algorithm.

Algorithm 4 Dijkstra algorithm

INPUT: A graph (S, A), $s \in S$

OUTPUT: $\forall t \in V \;\; d(t) = \mathcal{U}_s(t)$

Initialization:

Set $d(s) = 0$ and $d(t) = +\infty$ for all $t \neq s$. Set $\mathcal{A} = \emptyset$, $\mathcal{T} = \{s\}$ and $\mathcal{F} = V \backslash \{s\}$.

while there exists $t \in \mathcal{T}$ **do**

Select $t \in \mathcal{T}$ such $d(t)$ is minimal.

$\mathcal{T} \leftarrow \mathcal{T} \backslash \{t\}$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{t\}$

for all $v \in \mathcal{N}(t) \backslash \mathcal{A}$ **do**

if $v \in \mathcal{F}$ **then**

$\mathcal{F} \leftarrow \mathcal{F} \backslash \{v\}$, $\mathcal{T} \leftarrow \mathcal{T} \cup \{v\}$

end if

$update(v, t)$

end for

end while

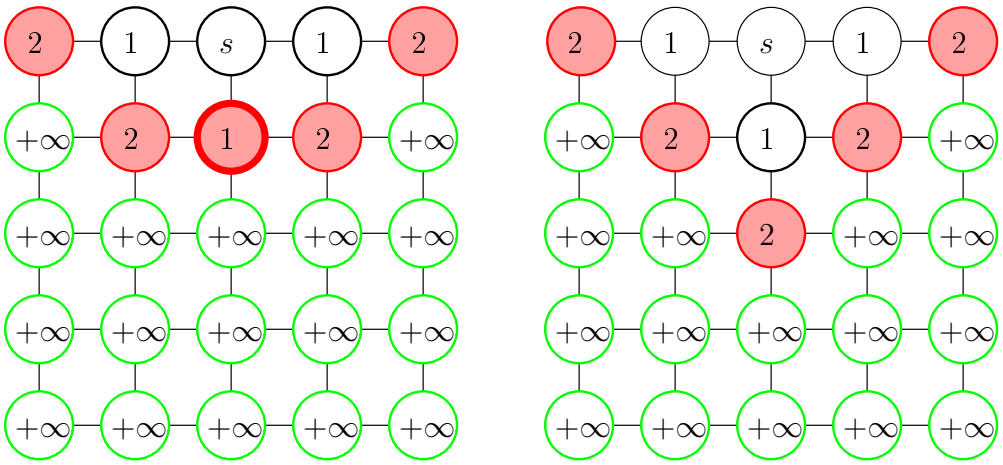


FIG. 2.1 – One iteration of Dijkstra algorithm. Black vertices : $\mathcal{A}$. Red vertices : $\mathcal{T}$. Green vertices : $\mathcal{F}$. Vertex of $\mathcal{T}$ with minimal weight (in bold) is selected, and transferred to $\mathcal{A}$. The vertex under it is transferred from $\mathcal{F}$ to $\mathcal{T}$.

At any time, and for every vertex t, $d(t) \geq \mathcal{U}_s(t)$: indeed $(s, \ldots p(t), (p(t), t), t)$ is a path from s de t, and its length is $d(t)$.

We are going to recursively show that at any time $\forall t \in \mathcal{A} \quad d(t) = \mathcal{U}_s(t)$. Notice that this property holds after initialisation.

Let us consider the instant when a vertices of $\mathcal{T}$ of minimal distance is chosen. let us denote by $t_0 \in \mathcal{T} \cup \mathcal{F}$ a vertex minimising $\mathcal{U}_s(t)$. We have

$$\mathcal{U}_s(t_0) = \min_{\substack{v \in \mathcal{N}(t_0) \\ \mathcal{U}_s(v) < \mathcal{U}_s(t_0)}} \mathcal{U}_s(v) + w(v, t_0)$$

Furthermore, if $v \notin \mathcal{A}$, then $\mathcal{U}_s(v) \geq \mathcal{U}_s(t_0)$. Therefore,

$$\mathcal{U}_s(t_0) \geq \min_{v \in \mathcal{N}(t) \cap \mathcal{A}} \mathcal{U}_s(v) + w(v, t_0)$$

Let us consider all the update operation that occurred to t_0 until now. For all neighbors v of t_0 in $\mathcal{A}$, the operation $d(t_0) \leftarrow \min\{d(t_0), d(v) + w((v, t_0))\}$ took place when v was transferred in $\mathcal{A}$.

We thus have

$$d(t_0) = \min_{v \in \mathcal{N}(t) \cap \mathcal{A}} d(v) + w((v, t_0)) = \min_{v \in \mathcal{A} \cup \mathcal{N}(t)} \mathcal{U}_s(v) + w((v, t_0)).$$

$d(t_0) = \mathcal{U}_s(t_0)$ thus holds. Furthermore, for all v in $\mathcal{T}$, $d(t_0) = \mathcal{U}_s(t_0) \leq \mathcal{U}_s(v) \leq d(v)$.

In particular, $d(t_0) \leq d(v)$. The inequality is strict, unless if $\mathcal{U}_s(t_0) = \mathcal{U}_s(v)$. We thus can assert that the set of vertices of $\mathcal{T}$ of minimal evaluated distance coincides with the set of vertices of $\mathcal{T}$ of minimal actual distance.

The chosen vertex t is therefore a vertex with minimal distance, and we have $d(t) = \mathcal{U}_s(t)$, which concludes the proof.

Complexity Every iteration is of $\mathrm{O}(p)$ complexity. If an unstructured set is used to implement $\mathcal{T}$, the selection of the minimal element in $\mathcal{T}$ runs in linear time with respect to the size of $\mathcal{T}$. n iterations being necessary, the overall time complexity of Dijkstra algorithm is thus de $\mathrm{O}(n(n+p)) \subset \mathrm{O}(n^2)$. Many implementations were proposed to decrease this complexity. In particular, it is interesting to consider $\mathcal{T}$ as a priority queue. It can thus be implemented as a *heap*[217, 62]. A heap is an ordered data structure in which insertion and update of an element runs in logarithmic time, while access to the smallest element runs in constant time. The complexity of the algorithm therefore becomes $\mathrm{O}(n(log(n) + p))$.

Improving the running time Other improvements can be made to the running time of the algorithm.

- If one is only interested in finding the shortest path between two vertices s and t, it is possible to stop the algorithm as soon as t is transferred to $\mathcal{A}$ – which can bring a substantial gain of time by avoiding the exploration of a large part of the graph.
- If a prior for distance map is available, it is possible to use meta-heuristics such as A^* algorithm, which allows to guide the exploration of the graph in a supposedly "good" direction [150].
- When no precise prior is available, if one is only interested in quickly finding an approximation of shortest paths, it is possible to use *Best First Search*-like algorithms, which guide the exploration in some direction heuristically – and stops as soon as the target point t is reached [150].

II.2 From discrete to continuous – a first attempt

Let us consider the problem described in I.3.1.3 for $\Omega = [-1,1]\times[-1,1] \subset \mathbb{R}^2$ with a potential P. In this section, we present a first attempt to solve a discretized version of this problem.
Let us discretize Ω with a regular grid with step $h = 1/N : \left\{\frac{i}{N}, \frac{j}{N} \mid -N \leq i,j \leq N\right\}$.
We build a graph which vertices correspond to the grid points – and which edges link all points in a 4-neighborhood (figure 2.2).

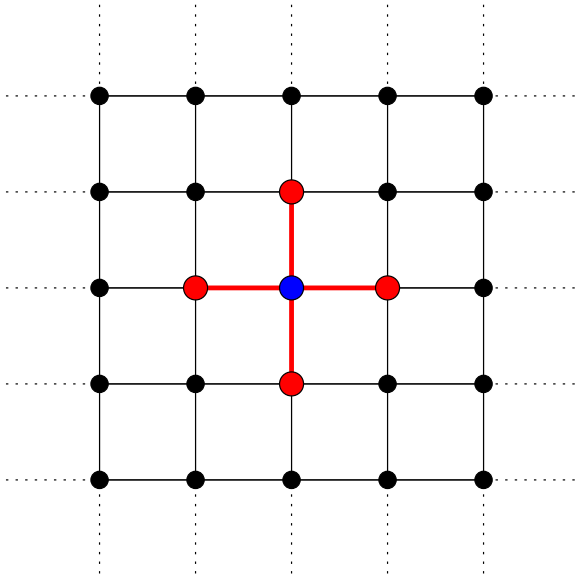


FIG. 2.2 – Local neighborhood system with 4 neighbors : the 4 red vertices are neighbors of the blue vertex.

Then we discretize the Eikonal Equation

$$\|\nabla U\| = P \tag{II.2.1}$$

along the edges.
Let us consider two adjacent points x and y. We then have :

$$\left|\frac{U(x) - U(y)}{h}\right| \approx P(x) \tag{II.2.2}$$

and

$$U(x) \approx U(y) + hP(x) \text{ or } U(x) \approx U(y) - hP(x) \qquad \text{(II.2.3)}$$

In view of updating $U(x)$ from $U(y)$, we have the constraint $U(x) > U(y)$, and with thus select the first equation.
Let us define the weight of an edge (x, y) as $w(x, y) \stackrel{\text{def.}}{=} hP(x)$. The obtained graph is then "compatible" with the continuous problem, in the sense that the length of a path in the graph is equal to the length of the corresponding geometric path in the plane.
Therefore, we can apply Dijkstra algorithm to this graph in order to compute distance maps and shortest paths (a similar formalism is proposed in [98, 101]).
When the potential is uniform, many shortest paths exist between two different points, and they can be distant from the actual straight line shortest path (figure 2.3).

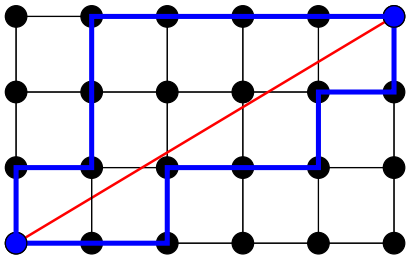


FIG. 2.3 – Shortest paths obtained by Dijkstra algorithm with a 4-neighbors system, for a uniform potential. The obtained shortest paths (in blue) between the two blue points are distant from the actual shortest path (red).

Figure 2.4 shows the result obtained by this method for a uniform potential over a bigger grid, with $s = (0, 0)$.

Proposition II.2.0.3
After the execution of Dijkstra algorithm for a discretization step N, , the approximation d_N of $\mathcal{U}_s$ is equal to $d_N\left(\frac{i}{N}, \frac{j}{N}\right) = \frac{|i|}{N} + \frac{|j|}{N}$.

In particular, as the discretization is refined, if we denote for all $(x, y) \in [-1, 1]^2$ $d_N(x, y) \stackrel{\text{def.}}{=} d_N\left(\frac{\lfloor Nx \rfloor}{N}, \frac{\lfloor Ny \rfloor}{N}\right)$, we have $\lim_{N \to +\infty} d_N(x, y) = |x| + |y| =$

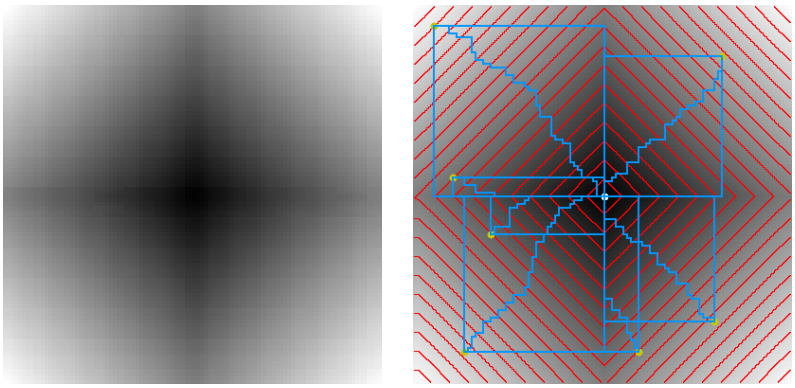


FIG. 2.4 – Results obtained by Dijkstra algorithm with a 4-neighbors system for a uniform potential over a 100×100 grid, starting point s being at the centre. *Left :* distance map ($\mathcal{U}_s$). *Right :* distance map ($\mathcal{U}_s$) along with level sets (red), and some shortest paths from different points to s (cyan)

$\|(x, y)\|_1$.
The trajectories are constrained to the axis directions – and thus this methods outputs an approximation of Manhattan distance from s ($t \to \|s - t\|_1$) instead of the correct $\mathcal{U}_s = t \to \|s - t\|_2$ distance.
It is possible to consider bigger neighborhood-systems, so that trajectories should follow more precise directions. For example, one can consider 8 or 16 neighbors for a generic point (fig 2.5).

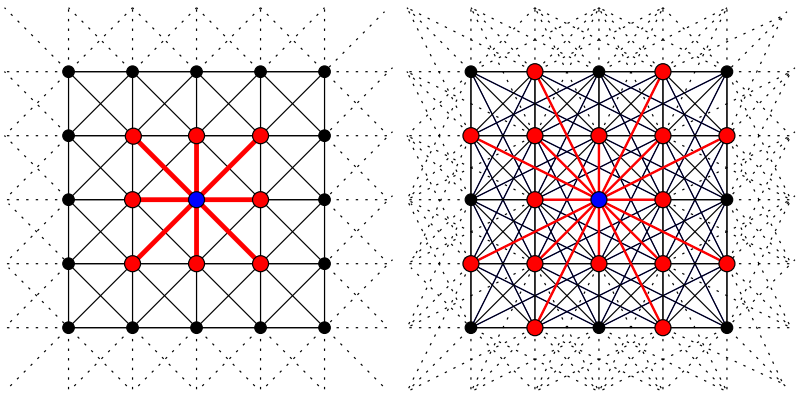


FIG. 2.5 – Local neighborhood-systems with 8 (left) and 16 (right) neighbors. For both figures, red vertices are neighbors of the blue one.

At the sake of an increase of running time, ont can thus improve the quality

of computed solutions (fig 2.6) – in particular when the neighborhood-system increases, level-sets get closer to circles, and the distance map gets closer from its theoretical value. Figure 2.7 also shows some results for a plane separated in two half-planes with constant potentials (1 and 4). The precision of the results increase while the neighborhood-system become more important, but the paths coming from the lower half-plane are still converging near the interface, which is contradictory with the Snell-Descartes law.

Figures 2.14 (top) and 2.15 (bottom) show errors obtained by the algorithm with different neighboring-systems for a uniform potential. Quality of the results improves as expected. Still, for all the considered systems, the trajectories are still constrained to follow a discrete set of directions, and the algorithm remains unable to evaluate distances correctly on other directions, even in refining the discretization.

Furthermore, if one which to increase the number of possible direction, one need to consider from each point neighbors further away. If the potential if varying quickly, this will result in a loss of precision – the neighborhood-system establishes links between spatially far away points, potentially losing precise value of potential between these two points.

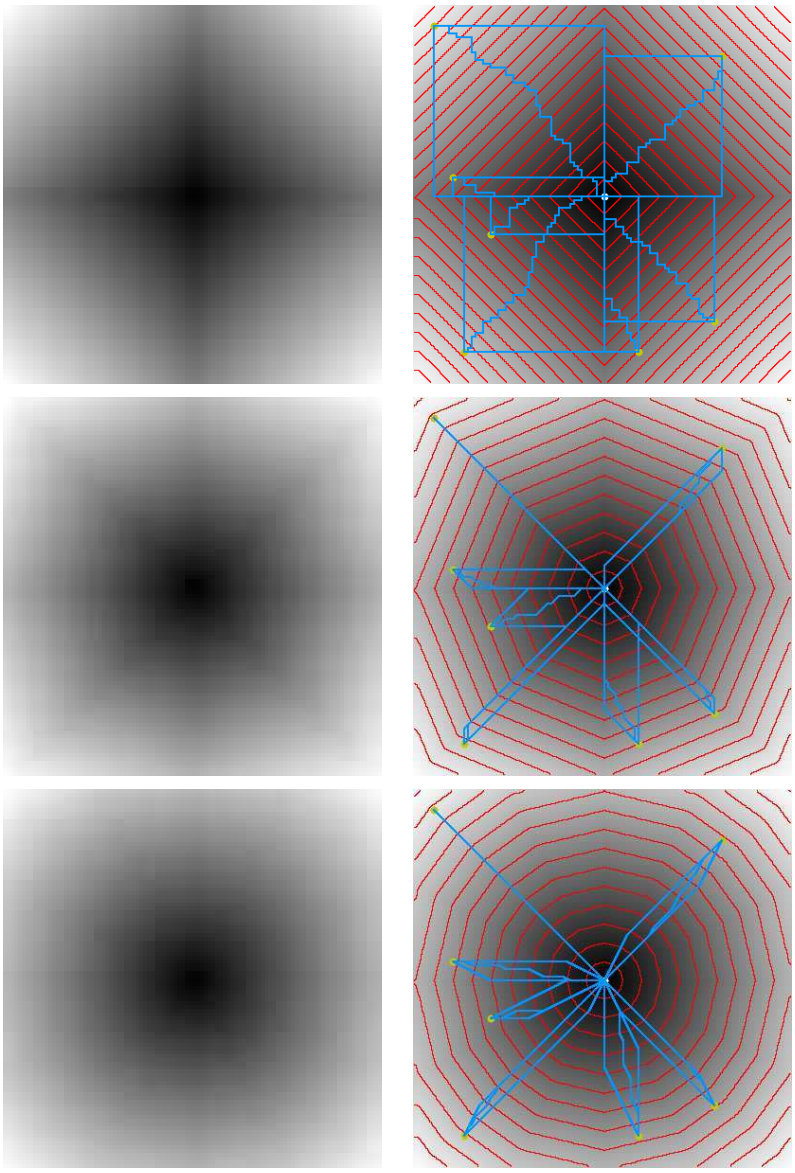


FIG. 2.6 – Distance maps, level sets and shortest paths for a uniform potential. *Top* : 4 neighbors. *Middle* : 8 neighbors. *Bottom* : 16 neighbors.

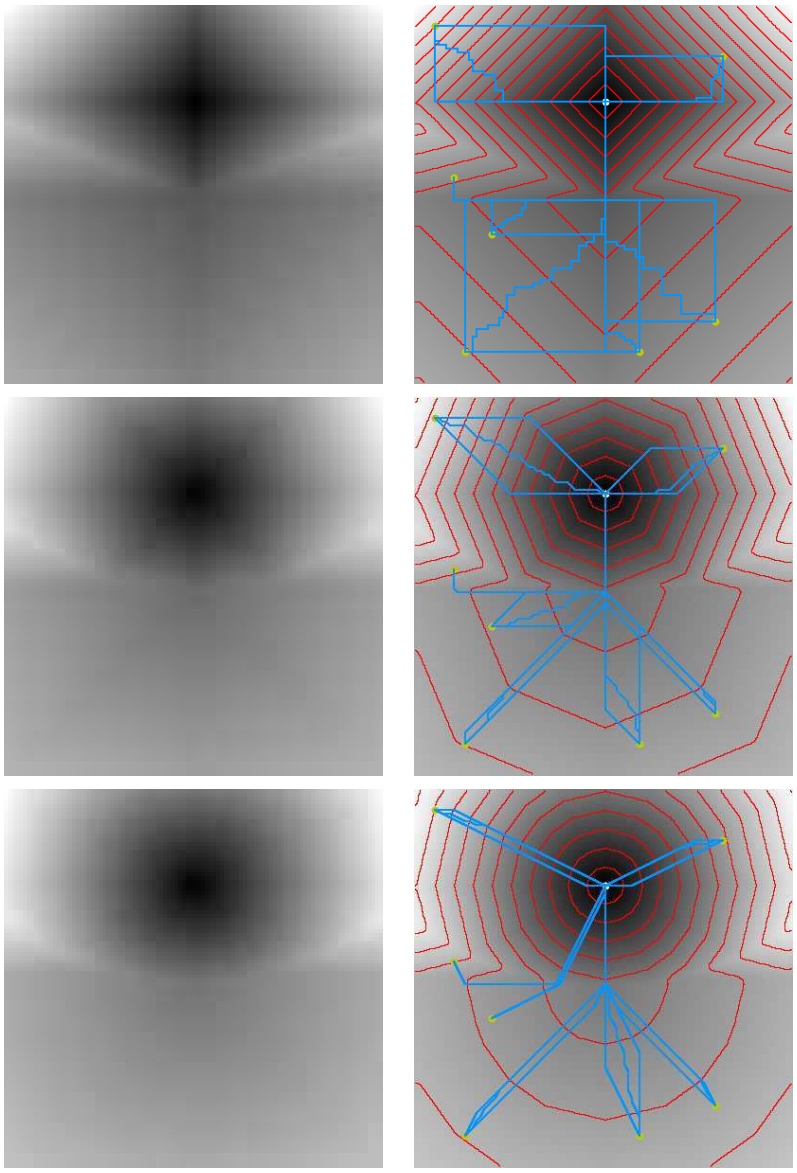


FIG. 2.7 – Distance maps, level sets and shortest paths for a plane separated in two half-planes with constant potentials (1 and 4). *Top* : 4 neighbors. *Middle* : 8 neighbors. *Bottom* Notice that even for the 16 neighbors experiment, trajectories in the bottom half-plane are still far away from the true solution.

II.3 Fast-Marching on a regular grid

The solution proposed in the last section is not fully satisfactory. Another method, based on the acceleration of numerical schemes – through ideas inspired from Dijkstra algorithm – was proposed in [187]. This approach – called *Fast-Marching* – is a continuous version of the algorithms proposed in the previous sections. It was initially written for the 2D case. The general road map for Fast-Marching algorithm is basically the same than for Dijkstra algorithm. A more precise update step allows to relax the constraint of propagation in a finite number of directions. In this section, we will present the algorithm on a regular grid in 2D. The numerical scheme is equivalent to the one proposed in [187] – despite a different shape – which will allow a direct generalisation to more complex cases. We will propose a convergence proof for our scheme – which will be easily extensible to nD and to anisotropic potentials.

II.3.1 Update step

The whole idea behind Fast-Marching update step is to bypass the constraint of propagation along the edges (figure 2.8.)

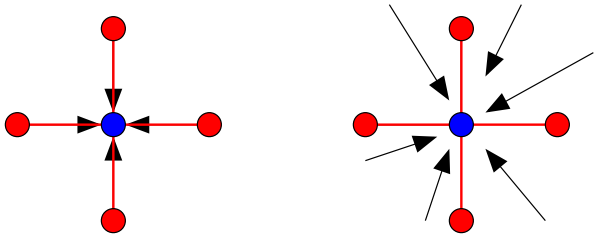


FIG. 2.8 – From Dijkstra to Fast-Marching

Let us consider $E = [-1, 1]^2$, discretized with a regular square grid with step h, and a point (i, j) on the discretization. Its 4 neighbors are $(i + h, j)$, $(i - h, j)$, $(i, j + h)$ and $(i, j - h)$. These five points define four triangles $\{S_i^{(2)}\}_{i\in[1..4]}$ and four edges $\{S_i^{(1)}\}_{i\in[1..4]}$ (figure 2.9).
It is possible to discretize the Eikonal equation on each of the triangles.
As an example, on $S_1^{(2)}$, we obtain

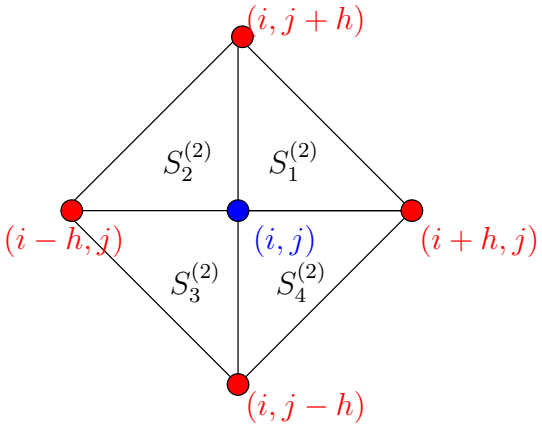


FIG. 2.9 – Neighborhood system induced by 4 neighbors in 2D.

$$\nabla U \approx \left(\frac{U(i+h,j) - U(i,j)}{h}, \frac{U(i,j+h) - U(i,j)}{h} \right) \tag{II.3.1}$$

and

$$(U(i+h,j) - U(i,j))^2 + (U(i,j+h) - U(i,j))^2 = h^2 P(i,j)^2 \tag{II.3.2}$$

denoting $u \overset{\text{def.}}{=} U(i,j)$, we have

$$2u^2 - 2u(U(i+h,j) + U(i,j+h)) + U(i+h,j)^2 + U(i,j+h)^2 - h^2 P(i,j)^2 = 0 \tag{II.3.3}$$

The quadratic equation has 0, 1 or 2 solutions depending on the sign of $\Delta' \overset{\text{def.}}{=} 2h^2 P(i,j)^2 - (U(i+h,j) - U(i,j+h))^2$.
Furthermore, we wish to have $u \geq U(i+h,j)$ and $u \geq U(i,j+h)$. The sum of the roots of the equation being $U(i+h,j) + U(i,j+h)$, only the biggest root u_2 can satisfy this condition. We have

$$u_2 = \frac{U(i+h,j) + U(i,j+h) + \sqrt{\Delta'}}{2} \tag{II.3.4}$$

A simple calculation shows that a sufficient condition so that $u_2 \geq \max\{U(i+h,j), U(i,j+h)\}$ is

$$(U(i+h,j) - U(i,j+h))^2 \leq h^2 P(i,j)^2. \tag{II.3.5}$$

Notice that this condition is stronger than the positivity condition of Δ'.
To sum up, we define :

$$\theta_P^{(2)} : (x,y) \in \mathbb{R}^2 \mapsto \begin{cases} \frac{x+y+\sqrt{2P^2-(x-y)^2}}{2} & \text{if } (x-y)^2 \leq P^2 \\ +\infty & \text{otherwise} \end{cases} \tag{II.3.6}$$

$$s_1^{(2)} \stackrel{\text{def.}}{=} \theta^{(2)}_{hP(i,j)}(U(i+h,j), U(i,j+h)) \tag{II.3.7}$$

In other words, $s_1^{(2)}$ is $+\infty$ or a value :
- which makes the Eikonal equation true in the triangle,
- which is superior to the values of other vertices of the triangles.

Similarly, let us define $s_2^{(2)}, s_3^{(2)}$ and $s_4^{(2)}$ the solutions in the triangles $S_2^{(2)}$, $S_3^{(2)}$ and $S_4^{(2)}$:

$$\begin{aligned} s_2^{(2)} &\stackrel{\text{def.}}{=} \theta^{(2)}_{hP(i,j)}(U(i-h,j), U(i,j+h)) \\ s_3^{(2)} &\stackrel{\text{def.}}{=} \theta^{(2)}_{hP(i,j)}(U(i-h,j), U(i,j-h)) \\ s_4^{(2)} &\stackrel{\text{def.}}{=} \theta^{(2)}_{hP(i,j)}(U(i+h,j), U(i,j-h)) \end{aligned} \tag{II.3.8}$$

Let us also define $\{s_i^{(1)}\}_{i\in[1..4]}$ as the update values obtained by discretizing the Eikonal equation along the edges $\{S_i^{(1)}\}_{i\in[1..4]}$:

$$\theta_P^{(1)} : x \in \mathbb{R} \mapsto x + P \tag{II.3.9}$$

$$\begin{aligned} s_1^{(1)} &\stackrel{\text{def.}}{=} \theta^{(1)}_{hP(i,j)}(U(i+h,j)) \\ s_2^{(1)} &\stackrel{\text{def.}}{=} \theta^{(1)}_{hP(i,j)}(U(i,j+h)) \\ s_3^{(1)} &\stackrel{\text{def.}}{=} \theta^{(1)}_{hP(i,j)}(U(i-h,j)) \\ s_4^{(1)} &\stackrel{\text{def.}}{=} \theta^{(1)}_{hP(i,j)}(U(i,j-h)) \end{aligned} \tag{II.3.10}$$

Let us define

$$\begin{aligned} &\theta_P : (a,b,c,d) \in \mathbb{R}^4 \rightarrow \\ &\min\{\theta_P^{(2)}(a,b), \theta_P^{(2)}(b,c), \theta_P^{(2)}(c,d), \theta_P^{(2)}(d,a), \theta_P^{(1)}(a), \theta_P^{(1)}(b), \theta_P^{(1)}(c), \theta_P^{(1)}(d)\} \end{aligned} \tag{II.3.11}$$

The update scheme of Fast-Marching algorithm consists in computing the solutions of Eikonal equation in all triangles and edges, and to select the minimal value among them.

$$U(i,j) \leftarrow \min_{\substack{i=1..2 \\ j=1..4}} \{s_j^{(i)}\} = \theta_{hP_{(i,j)}}(U(i+h,j), U(i,j+h), U(i-h,j), U(i,j-h)) \tag{II.3.12}$$

Notice that the update scheme in Dijkstra algorithm is

$$U(i,j) \leftarrow \min_{j=1..4} \{s_j^{(1)}\} \tag{II.3.13}$$

Let us recall that in Dijkstra algorithm, the update step could be refined by considering only neighbors with a value smaller than the current point (property II.1.1.2).
The same reasoning holds in the current situation : indeed, for all $j \in [1..4]$, if we denote by $S_j^{(1)}$ the edge $((i,j),A)$, $s_j^{(1)} > U(A)$, and if we denote by $S_j^{(2)}$ the triangle $((i,j),A,B)$, $s_j^{(1)} > \max\{U(A),U(B)\}$. An edge or a triangle cannot be taken into account in the update if the value of one of its vertices is strictly superior to the current value of (i,j).
let us denote by

$$\mathcal{S}_-(i,j)$$

the set of edges or triangles which are adjacent to (i,j) and such that all its vertices distinct from (i,j) have a value inferior to $U(i,j)$.
Then, the update step is equivalent to :

$$U(i,j) \leftarrow \min_{S_j^{(i)} \in \mathcal{S}_-(i,j)} \{s_j^{(i)}\} \tag{II.3.14}$$

This formulation, in addition to allowing to save up computations, will also be useful in the convergence proof.
This update state can be performed iteratively for all the discretization points. However – as for Dijkstra algorithm – one can use causality in order to choose a more clever order.
Fast-Marching algorithm is synthesised on figure 5.
As in Dijkstra algorithm case, in order to avoid unnecessary operations, the update step can be performed by taking into account only triangles/edges

Algorithm 5 Fast-Marching

INPUT: $\left\{\frac{i}{N}, \frac{j}{N} \mid -N \leq i, j \leq N\right\}$ as a discretization of $[0,1]^2$. $s \in S$.
OUTPUT: $\forall t \in V \;\; d(t) = \mathcal{U}_s(t)$

Initialization:
Set $d(s) = 0$ and $d(t) = +\infty$ for all $t \neq s$. Set $A = \emptyset$, $T = \{s\}$ and $F = V \backslash \{s\}$.
while there exists $t \in T$ **do**
  Select $t \in T$ such that $d(t)$ is minimal.
  $T \leftarrow T \backslash \{t\}$, $A \leftarrow A \cup \{t\}$
  for all $v \in \mathcal{N}(t) \backslash A$ **do**
    if $v \in F$ **then**
      $F \leftarrow F \backslash \{v\}$, $T \leftarrow T \cup \{v\}$
    end if
    update v using equation (II.3.14).
  end for
end while

that contains the current point – and in the case of triangles, such that the remaining point belongs to $\mathcal{A}$.

Complexity. The analysis performed for Dijkstra algorithm holds. It shows that the complexity of this algorithm is $O(Nlog(N))$, where N is the number of points explored by the algorithm.

II.3.2 Convergence proof

In this section, we will prove the convergence of this numerical scheme, *i.e.* prove that when the discretization step h tends toward 0, the solution computed by the algorithm tends toward the viscosity solution of Eikonal equation. Notice that another convergence proof is given in [174] – the authors are using a different but equivalent formulation for the update step. The benefit of our framework resides in the ease of generalisation of both the scheme and the convergence proof to more complex cases.
Let us define

$$S(h,(x,y),t,U) \overset{\text{def.}}{=} \frac{\min\limits_{\substack{i=1..2\\ j=1..4}} \{s_j^{(i)}\} - t}{h}$$
$$= \frac{\theta_{hP_{(x,y)}}(U(x+h,y),U(x,y+h),U(x-h,y),U(x,y-h)) - t}{h} \quad \text{(II.3.15)}$$

The update scheme (II.3.12) can thus be rewritten :

$$S(h,(i,j),U(i,j),U) = 0 \quad \text{(II.3.16)}$$

The complete convergence proof lies on two steps.

- Proving that any fix-point of the discrete problem II.3.12 tend toward the viscosity solution of (II.2.1) – which is mainly a specific case of a general proof made in [9] and taken up by [174].
  The proof is based on three characteristics of the scheme ([9]) :
  - *monotony* of the update scheme II.3.12, which can be enunciated as follows :
    (II.3.16) is monotonous if and only if

$$U \leq V \Rightarrow S(h,(i,j),t,U) \leq S(h,(i,j),t,V)$$

  - *stability* of II.3.12 : the scheme is stable if the solution to the discrete problem exists, and is bounded with a bound independent from the discretization step.
  - *consistency* of II.3.12 – which denote the fact that II.3.12 *is* a discretization of Eikonal equation or an equivalent equation. In our case, this can be written

$$\lim_{\substack{h\to 0\\ (x',y')\to(x,y)\\ \xi\to 0}} S(h,(x',y'),\varphi(x',y')+\xi,\varphi+\xi) = H(\nabla\varphi,(x,y))$$

    for every function $\varphi \in \mathcal{C}^\infty$ bounded over E, and where

$$H(\nabla\varphi,(x,y)) = 0$$

    is equivalent to Eikonal equation (H is called *Hamiltonian* associated with the Eikonal equation).

- Prove that the ordering of updates allows to compute such a fix-point. The reasoning is somewhat similar to the one made during Dijkstra algorithm proof. Notice that the first demonstration of this fact was given in [187], yet with a different proof.

Some results. Some preliminary results will be needed during the convergence proof.

Let us define $\Omega_P = \{(x,y) \in \mathbb{R}^2 \mid (x-y)^2 \leq P^2\}$, $\Omega_P^+ = \{(x,y) \in \mathbb{R}^{+2} \mid x \geq y \text{ et } (x-y)^2 \leq P^2\}$ and $\Omega_P^- = \{(x,y) \in \mathbb{R}^{+2} \mid x \leq y \text{ et } (x-y)^2 \leq P^2\}$.

Lemma II.3.2.1 *Properties of* $\theta_P^{(2)}$
Let $(x,y) \in \Omega_P$.
- $\theta_P^{(2)}$ *is continuous over* Ω_P.
- *if* $(x,y) \in \Omega_P$, $a \mapsto \theta_P^{(2)}(x+a, y+a)$ *is non-decreasing over* $\mathbb{R}^+$.
- *if* $(x,y) \in \Omega_P$, $a \mapsto \theta_P^{(2)}(x+a, y)$ *is non-decreasing over* $\mathbb{R}^+$.
- *if* $(x,y) \in \Omega_P$, $a \mapsto \theta_P^{(2)}(x, y+a)$ *is non-decreasing over* $\mathbb{R}^+$.
- *if* $(x-y)^2 = P^2$, $\theta_P^{(2)}(x,y) = \min\{x,y\} + P$.

These properties are illustrated on figure 2.10.

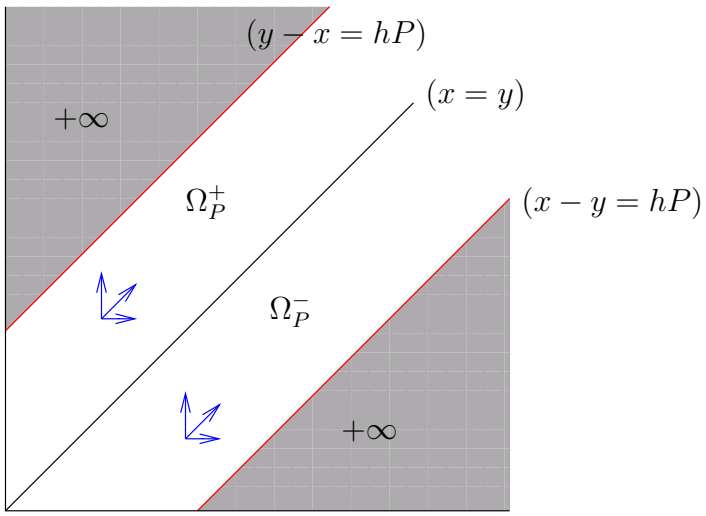


FIG. 2.10 – Illustration of some properties of $\theta^{(2)}$: the function is non-decreasing in all the directions indicated by blue arrows.

Lemma II.3.2.2 *Properties of* $\theta_P^{(1)}$
- $\theta_P^{(1)}$ *is continuous over* $\mathbb{R}$.
- $\theta_P^{(1)}$ *is non-decreasing over* $\mathbb{R}$.

We can therefore deduce the following properties for θ_P

Lemma II.3.2.3 *Properties of* θ_P
- θ_P *is continuous over* $\mathbb{R}^4$.
- θ_P *is a non-decreasing function of each of its variables.*

Proof :
- Using the continuity of $\theta_P^{(1)}$ and $\theta_P^{(2)}$, θ_P is continuous everywhere, except possibly in points such that $(a-b)^2 = P^2$, $(b-c)^2 = P^2$, $(c-d)^2 = P^2$ or $(d-a)^2 = P^2$. As an example, let us assume that $(a-b)^2 = P^2$ and $a > b$. Then we have $a = b + P$
  Then $\theta_P^{(2)}(a,b) = \frac{a+b+\sqrt{P^2}}{2} = \frac{2a+2P}{2} = a + P = \theta_P^{(1)}(a)$.
  Similarly, if $a < b$, we get $\theta_P^{(2)}(a,b) = \theta_P^{(1)}(b)$.
  The $\theta^{(1)}$ mappings "stick" continuously on the border of the set where $\theta^{(2)} < +\infty$, which shows that θ_P is continuous at those points.
- The growing of θ results from the growing of $\theta_P^{(1)}$ and $\theta_P^{(2)}$.

□

Continuity can be geometrically interpreted in the following way : let us assume that for current values of $U(i \pm h, j \pm h)$, the update is done from the the $((i,j),(i+h,j),(i,j+h))$ triangle – II.3.5 being true in this triangle. Let us also assume that $U(i+h,j)$ increases until equality is reached in II.3.5. Then, the solution is equal to $U(i+h,j)$, *i.e.* the gradient of U est collinear with $((i,j),(i,j+h))$. The update value for the triangle is then equal to the update value for the $((i,j),(i,j+h))$ edge (figure 2.11).
Updates from edges are therefore *continuously* taking over from the updates from triangles when the laters become impossible.
We now present the convergence proof of the algorithm :
Proof :

monotony :

Follows immediately from proposition II.3.2.3.

stability :

The existence of a solution of discrete problem can be demonstrated by borrowing an argument from [174].

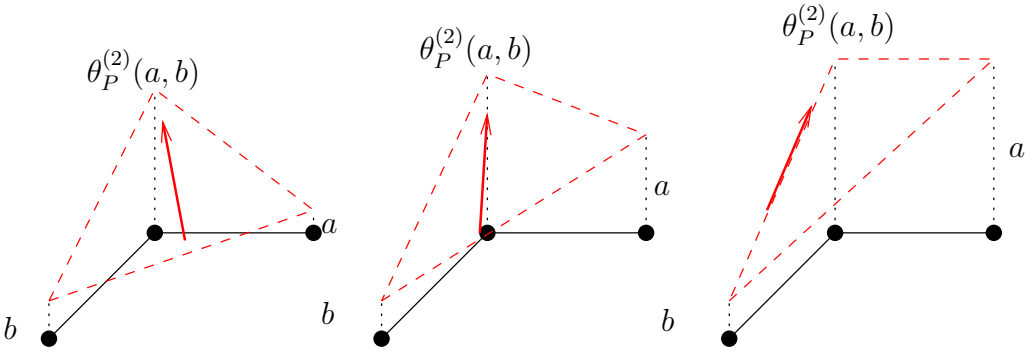


FIG. 2.11 – Continuity of θ_p : the update step from the $\theta_P^{(1)}(a)$ edge continuously taking over from the updates from $\theta_P^{(2)}(a,b)$ triangle when the gradient becomes collinear with the edge – after an increase of b.

Let us consider an algorithm which apply the update state to all the points of the discretization, in the same order as Fast-Marching algorithm, but an infinite number of times (which correspond to the iterative version of Dijkstra algorithm proposed in II.1.1).

For every point (i,j), $d(i,j)$ is thus non-increasing, and inferiorly bounded by 0. Therefore it tends toward some limit denoted $d_{it}(i,j)$.

After an update step over (i,j), we have $S(h,(i,j),U(i,j),U)=0$. $S(h,(i,j),t,U)$ being continuous in t and in U, $S(h,(i,j),t,U)$ tends toward

$S(h,(i,j),d_{it}(i,j),d_{it})$ along the iterations. Furthermore, after an infinite number of iterations, we have $S(h,(i,j),U(i,j),U)=0$. This implies $S(h,(i,j),d_{it}(i,j),d_{it})=0$, and d_{it} is therefore a solution of the discrete scheme.

d_{it} is inferiorly bounded by 0. It is possible to show the existence of an upper bound which depends on the diameter of E, on the minimal potential over E – which is not zero by compacity of E an continuity of the potential.

consistency :

let us consider a function φ $\mathcal{C}^\infty$ bounded over E, $(x,y)\in E$, and $\xi\in\mathbb{R}^{+*}$.

- Firstly, let us note that

$$s_1^{(1)}=\varphi(x'+h,y)+\xi+hP_{x'y'} \tag{II.3.17}$$

and

$$\frac{s_1^{(1)}-\varphi(x',y')-\xi}{h}=\frac{D^{+x}\varphi(x',y')}{h}+P_{x'y'} \tag{II.3.18}$$

tends toward

$$d_x\varphi(x,y)+P_{xy}=\theta_{P_{xy}}^{(1)}(d_x\varphi(x,y)) \tag{II.3.19}$$

when $(x', y') \to (x, y)$, $h \to 0$ and $\xi \to 0$.
Similarly, for the other edges, we find limits $\theta^{(1)}_{P_{xy}}(-d_x\varphi(x,y))$, $\theta^{(1)}_{P_{xy}}(d_y\varphi(x,y))$ and $\theta^{(1)}_{P_{xy}}(-d_y\varphi(x,y))$.

- Let us now consider $S^{(2)}_1$ triangle, and let us assume that $(d_x\varphi(x,y)-d_y\varphi(x,y))^2 < P^2_{xy}$.
  P is continuous, and $\varphi\ \mathcal{C}^\infty$. For any h close enough to 0 and (x', y') close enough to (x, y), we thus have $(\varphi(x'+h, y') + \xi - \varphi(x', y'+h) - \xi)^2 = (\varphi(x'+h, y') - \varphi(x', y') + \varphi(x', y') + \varphi(x', y'+h))^2 \leq h^2 P^2_{x'y'}$.
  Then,

$$s^{(2)}_1 = \frac{2\xi+\varphi(x'+h,y')+\varphi(x',y+h')+\sqrt{2h^2P^2_{x'y'}-(\varphi(x'+h,y')-\varphi(x',y'+h))^2}}{2}$$

and

$$\frac{s^{(2)}_1 - \varphi(x', y') - \xi}{h} = \frac{D^{+x}\varphi(x',y')}{h} + \frac{D^{+y}\varphi(x',y')}{h} + \sqrt{2P^2_{x'y'} - \left(\frac{D^{+x}\varphi(x',y')}{h} - \frac{D^{-y}\varphi(x',y')}{h}\right)^2}. \quad \text{(II.3.20)}$$

When h and (x', y') converge toward their limits, this expression tends to

$$d_x\varphi(x,y) + d_y\varphi(x,y) + \sqrt{2P^2_{xy} - (d_x\varphi(x,y) - d_y\varphi(x,y))^2} = \theta^{(2)}(d_x\varphi(x,y), d_y\varphi(x,y)) \quad \text{(II.3.21)}$$

Similar results can be obtained for the three remaining triangles :
- if $(d_x\varphi(x,y) + d_y\varphi(x,y))^2 < P^2_{xy}$,
  $\frac{s^{(2)}_2 - \varphi(x',y') - \xi}{h}$ tends toward $\theta^{(2)}(d_x\varphi(x,y), -d_y\varphi(x,y))$.
- if $(-d_x\varphi(x,y) + d_y\varphi(x,y))^2 < P^2_{xy}$,
  $\frac{s^{(2)}_3 - \varphi(x',y') - \xi}{h}$ tends toward $\theta^{(2)}(-d_x\varphi(x,y), -d_y\varphi(x,y))$.
- if $(-d_x\varphi(x,y) - d_y\varphi(x,y))^2 < P^2_{xy}$,
  $\frac{s^{(2)}_4 - \varphi(x',y') - \xi}{h}$ tends toward $\theta^{(2)}(-d_x\varphi(x,y), d_y\varphi(x,y))$.

Furthermore, these results remain true in the cases when $(\pm d_x\varphi(x,y) \pm d_y\varphi(x,y))^2 > P^2_{xy}$ – when the limit is $+\infty$.

- Thus – outside the limit cases where $(\pm d_x\varphi(x,y) \pm d_y\varphi(x,y))^2 = P^2_{xy}$, we have

$$S(h, (x', y'), \Phi(x', y') + \xi, \Phi + \xi) = \frac{\min\{s^{(i)}_j - \varphi(x,y)\}}{h} \underset{\substack{h\to 0 \\ (x',y')\to(x,y) \\ \xi\to 0}}{\to} \theta_{P_{xy}}(d_x\varphi(x,y), d_y\varphi(x,y), -d_x\varphi(x,y), -d_y\varphi(x,y)) \quad \text{(II.3.22)}$$

It is possible to show that this equation still holds in the limit cases : for example, let us assume that $(d_x\varphi(x,y) - d_y\varphi(x,y))^2 = P_{xy}^2$ and $d_x\varphi(x,y) > d_y\varphi(x,y)$. Let us consider $((x_k,y_k),h_k,\xi_k) \to ((x,y),0,0)$.
let us denote by $((x_{\psi(k)},y_{\psi(k)}),h_{\psi(k)},\xi_{\psi(k)})$ the subsequence made from terms such that $(\varphi(x_{\psi(k)}+h,y_{\psi(k)}) - \varphi(x_{\psi(k)},y_{\psi(k)}+h))^2 \leq h^2 P^2_{x_{\psi(k)}y_{\psi(k)}}$, and $(x_{\psi'(k)},y_{\psi'(k)},h_{\psi'(k)},\xi_{\psi'(k)})$ the complementary subsequence.
Let us consider $s_k = \frac{s_1^2 - \varphi(x_k,y_k)}{h_k}$. Like above, we have
$\lim_{k\to+\infty} s_{\psi(k)} = \theta^{(2)}(d_x\varphi(x,y), d_y\varphi(x,y))$.
Furthermore, we still have $\lim_{k\to+\infty} \frac{s_1^{(1)} - \varphi(x_k,y_k)}{h_k} = \theta^{(1)}(d_x\varphi(x,y))$
As observed during the continuity proof of θ, these two quantities are equal. We deduce that

$$\frac{\min\{s_1^2, s_1^1\} - \varphi(x_k,y_k)}{h_k} \to \theta^{(1)}(d_x\varphi(x,y)).$$

The other cases can be tackled in the same way. Thus II.3.22 is valid for any value of φ.

The scheme is thus consistent with the following Hamiltonian

$$H(\nabla\varphi,(x,y)) \stackrel{\text{def.}}{=} \theta_{P_{xy}}(d_x\varphi(x,y), d_y\varphi(x,y), -d_x\varphi(x,y), -d_y\varphi(x,y)) \quad \text{(II.3.23)}$$

Furthermore, $\theta^{(1)}$ and $\theta^{(2)}$ being non-decreasing, we have

$$\begin{aligned} &H(\nabla\varphi,(x,y)) = \\ &\quad \min\{\theta^{(2)}(-|d_x\varphi(x,y)|, -|d_y\varphi(x,y)|), \theta^{(1)}(-|d_y\varphi(x,y)|), \theta^{(1)}(-|d_y\varphi(x,y)|)\}. \end{aligned} \quad \text{(II.3.24)}$$

One easily sees that $H(\nabla\varphi,(x,y)) = 0$ if and only if $|\nabla\varphi(x,y)|^2 - P_{xy}^2 = 0$.

The scheme is thus consistent.

ordering :

the proof of Dijkstra algorithm can be exactly transposed here. For any point t of the discretization, let us denote by $d_{it}(t)$ the distance obtained by the algorithm described in the stability proof.. d_{it} is thus a fix-point of the update scheme. Furthermore, we have$d \geq d_{it}$ (the first iteration correspond exactly to the one of Fast-Marching algorithm, and the following iterations can only decrease the values of each point.)

Let us consider the *Fast-Marching* algorithm.

We are going to prove inductively that, at any step of the algorithm, $\forall t \in \mathcal{A}\ \ d(t) = d_{it}(t)$. Notice that this property holds after the initialisation.

Let us consider the time when a point with minimal distance is chosen in $\mathcal{T}$. Let us consider $t_0 \in \mathcal{T} \cup \mathcal{F}$ minimising $d_{it}(t)$.

let us denote by

- $\mathcal{S}_-(t_0)$ the set of triangles/edges adjacent to t_0 and such that all the points of the triangle/edges different from t have a value smaller than $d_{it}(t_0)$,
- $\mathcal{S}_\mathcal{A}(t_0)$ the set of triangles/edges adjacent to t_0 and such that all the points of the triangle/edges different from t are in $\mathcal{A}$.

d_{it} verifies II.3.14. Thus, we have

$d_{it}(t_0) = \min_{S_j^i \in \mathcal{S}_-(t_0)} s_j^i(d_{it})$

Furthermore, if $v \notin \mathcal{A}$, we have $d_{it}(v) \geq d_{it}(t_0)$.

We deduce $d_{it}(t_0) \geq \min_{S_j^i \in \mathcal{S}_\mathcal{A}(t_0)} s_{it}{}_j^i(d_{it})$

Let us consider all the update operation that occurred to t_0 until now. The update from a triangle or an edge in $\mathcal{S}_\mathcal{A}(t_0)$ occurred when the last vertex but one of this triangle or this edge was transferred in $\mathcal{A}$.

We then have $d(t_0) = \min_{S_j^i \in \mathcal{S}_\mathcal{A}(t_0)} s_j^i = \min_{S_j^i \in \mathcal{S}_\mathcal{A}(t_0)} s_j^i(d_{it})$

by hypothesis.

Then $d(t_0) \leq d_{it}(t_0)$. Furthermore, for all v in $\mathcal{T}$, $d(t_0) \leq d_{it}(t_0) \leq d_{it}(v) \leq d(v)$.

In particular, $d(t_0) \leq d(v)$. This inequality is strict, unless if $d_{it}(t_0) = d_{it}(v)$. We thus can assert that the set of vertices of $\mathcal{T}$ of minimal evaluated distance coincides with the set of vertices of $\mathcal{T}$ of minimal actual distance.

The chosen vertex t is therefore a vertex with minimal distance, and we have $d(t) = d_{it}(t)$, which concludes the proof.

□

The monotonicity condition will be the main obstacle to the generalisation of Fast-Marching algorithm to more general Riemannian manifolds.

II.3.2.1 Improving the running time

The calculation (II.3.12) request up to four resolutions of second degree equations. It is possible to reduce this amount of operations.
Let us define $A_x = (i-1, j)$ and $A'_x = (i+1, j)$ if $\mathcal{U}(i-1, j) \leq \mathcal{U}(i+1, j)$, $A_x = (i+1, j)$, and $A'_x = (i-1, j)$ otherwise. Similarly, let us define A_y and A'_y. Up to a switch of two coordinates, we can assume that $A_y \leq A_x$. Figure 2.12 illustrates two possible configurations of $s_1^{(2)}$, $s_2^{(2)}$, $s_3^{(2)}$ and $s_4^{(2)}$. Let us

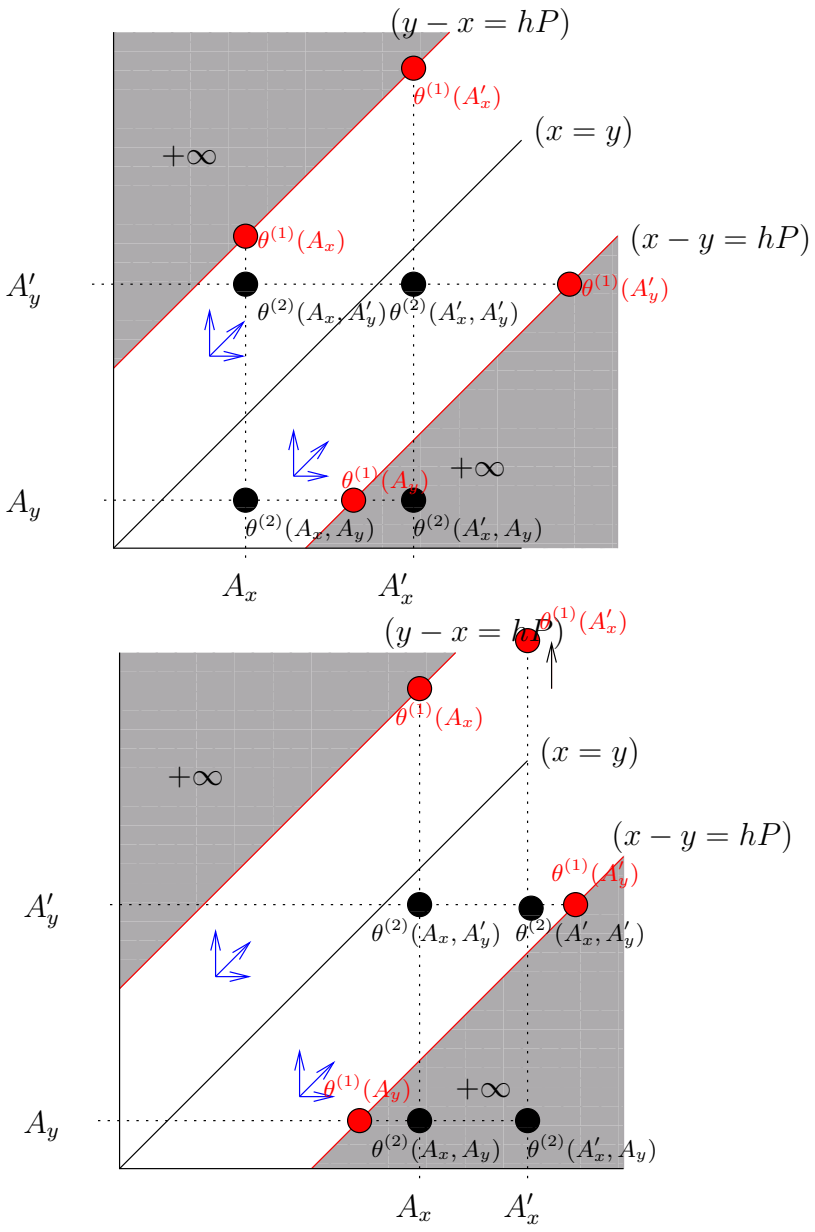


FIG. 2.12 – Some possible configurations for $s_j^{(i)}$.

note that we can also draw points corresponding to values of $s_1^{(1)}$, $s_2^{(1)}$, $s_3^{(1)}$ and $s_4^{(1)}$.

A case-study of the different configurations gives the following results :

Proposition II.3.2.4
$\min\{s_j^{(i)}\}$ *is*

- *either reached for the* $((i,j),A_x,A_y)$ *triangle if the corresponding value is finite,*
- *or reached for the* $((i,j),A_y)$ *edge.*

To summarise, we define

$$U(i,j) \leftarrow \begin{cases} s_*^{(2)} = \frac{U(A_x)+U(A_y)+\sqrt{2h^2P(i,j)^2-(U(A_x)-U(A_y))^2}}{2} \\ \text{if } (U(A_x)-U(A_y))^2 \leq h^2P(i,j)^2 \\ s_*^{(1)} = \min\{U(A_x),U(A_y)\}+hP(i,j) \\ \text{otherwise} \end{cases}$$

which reduces the number of operations to perform with respect to (II.3.12) to at most one resolution of a quadratic equation.

Despite the different formulation, this scheme is equivalent to the one proposed in [188, 174] :

$$(\max\{U_{i,j}-U_{i-1,j},U_{i,j}-U_{i+1,j},0\})^2+(\max\{U_{i,j}-U_{i,j-1},U_{i,j}-U_{i,j+1},0\})^2 = P_{(i,j)}^2 \tag{II.3.25}$$

Other improvements of the running time have been proposed, most of them inspired by variations of Dijkstra algorithm.

- When one is willing to compute a shortest path between two points, it is possible to stop the front propagation when the second point is reached. Another approach consists in propagating fronts simultaneously from both points, and to stop when the two fronts intersect. A gradient descent from the intersection in each front will then give an approximation of the shortest path between the two considered points [50].
- In the same article, a freezing strategy is proposed – which allows to stop front propagation in high-potential areas.
- Inspired by *Best First Search* algorithm, [156] proposes to use heuristics to drive the propagation of the front in the correct direction.

- At the sake of a slight lost of precision, [219] has shown the possibility of implementing the algorithm with a $O(n)$ time complexity, using a *untidy priority queue* data structure instead of a heap to implement $\mathcal{T}$.

Notice that all these strategies can also be applied to the more general versions of Fast-Marching algorithm we will describe in the sequel.

II.3.3 Increasing the neighborhood system

Even if convergence of this algorithm is proved when the discretion steps converges toward 0, it is not an exact algorithm. Figures 2.14 and 2.15 show errors obtained by the algorithm. Unlike the results obtained by Dijkstra algorithm, we can observe that the relative error vanishes as we move away from the origin. This means dually that – for a constant potential – the evaluated distance map converge toward its theoretical value when the the discretization step tends toward zero.

The numerical error of the algorithm is more important in the neighborhood of s, in directions where no edges are present in the neighborhood system. Not unlike the case of Dijkstra algorithm, it is possible to improve the precision of the algorithm by considering a more important neighborhood system (figure 2.13) – as proposed in [46]. The presented system consists in 8 triangles and 8 edges.

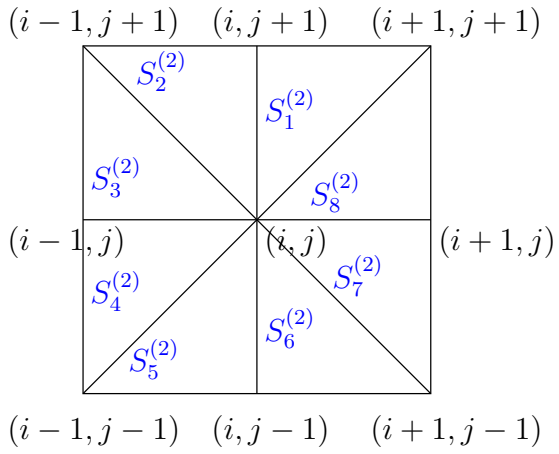


FIG. 2.13 – 8-neighbors system for 2D Fast-Marching

The Fast-Marching algorithm remains the same. The update scheme consists

in selecting the triangle or the edge which produces the minimal value. However, the are two differences with the previously exposed algorithm :

- We need to compute update values for triangles with a different shape – and therefore to find an equivalent of II.3.1 for those triangles. It is a specific case of a more general equation we will introduce in section II.4.
- It is not possible to reduce the amount of necessary calculations as much as in the previous case.

II.3.4 Numerical results

We compare numerical results obtained by the methods exposed in the previous sections.

Figures 2.14 and 2.15 shows results obtained by the increase of the neighborhood system. Figure 2.16 presents similar results for a space consisting of two half-planes with uniform potentials 1 and 4 – for which it is possible to compute the distance map with arbitrary precision.

Increasing the neighborhood system results in an improvement of the results obtained by Dijkstra algorithm. However, as shown in 2.15, the error does not vanish as we move away from the origin – or dually when the the discretization step tends toward zero. On the opposite, it is the case for approximation computed by Fast-Marching algorithm.

Figure 2.17 shows some shortest paths computed from the distances maps.

II.3.5 Generalisation to nD

It is straightforward to generalise the presented algorithm to arbitrary dimension.

Let us consider a n-dimensional space, discretized with a regular grid, and a neighborhood system consisting of $2n$ neighbors. Such a system defines $K_n \stackrel{\text{def.}}{=} 2^n$ simplices of dimension n, $K_{n-1} \stackrel{\text{def.}}{=} \binom{n}{1}2^{n-1} = n2^{n-1}$ simplices of dimension $n-1$ $\ldots K_1 \stackrel{\text{def.}}{=} 2\binom{n}{n-1} = 2n$ simplices de dimension 1 – i.e. $3^n - 1$ simplices (figure 2.18 shows some of these simplices in the case of dimension 4). These simplices are a generalisation of triangles and edges in dimension 2.

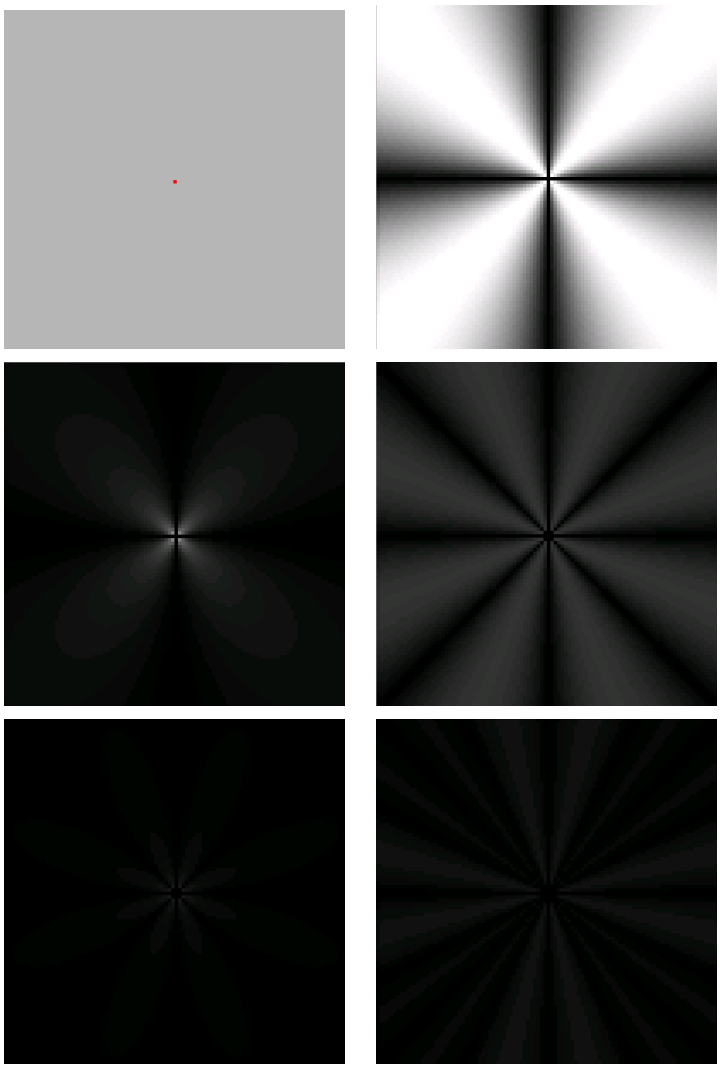


FIG. 2.14 – Relative errors obtained by the different algorithms for a uniform potential over a regular 100×100 grid. *Left column, top to bottom* : the potential, Fast-Marching with 4 neighbors, Fast-Marching with 8 neighbors. *Right column, top to bottom* : Dijkstra with 4 neighbors, Dijkstra with 8 neighbors, Dijkstra with 16 neighbors. All the images are represented with the same gray level scale : black : 0%, white : $\geq 40\%$

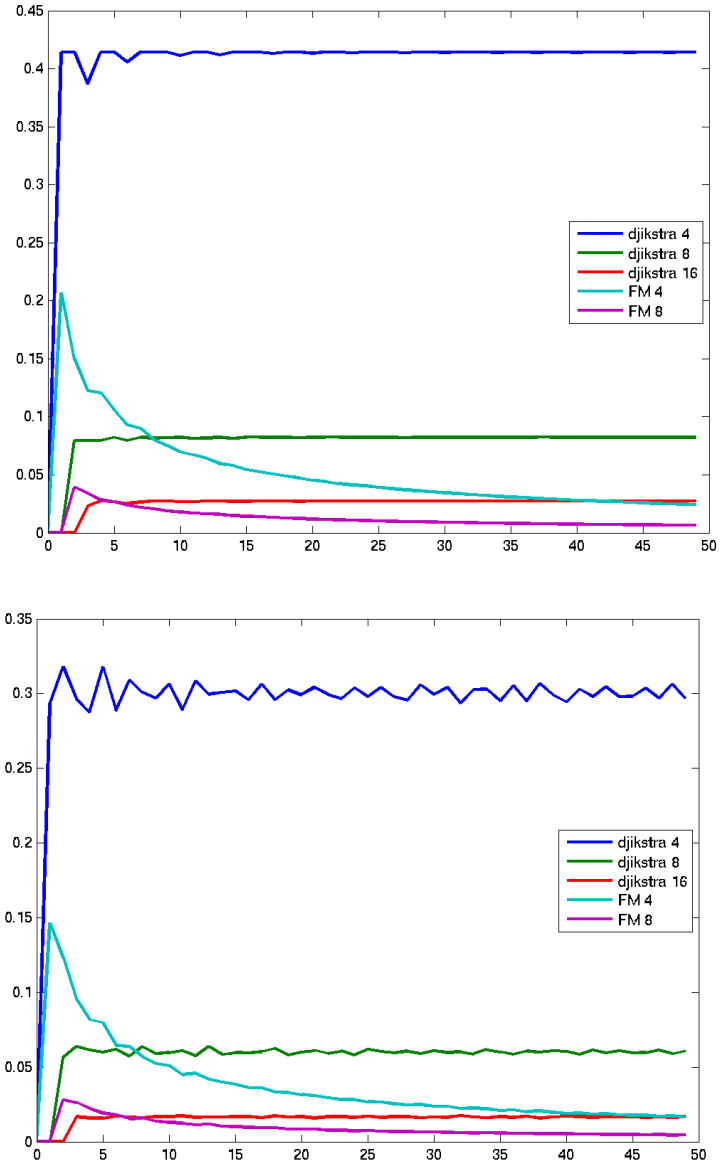


FIG. 2.15 – Relative errors obtained by the different algorithms for a uniform potential over a regular 100×100 grid. *Top :* maximum of relative error for a fixed distance to origin. *Bottom :* $\mathcal{L}_2$ norm of relative error for a fixed distance to origin.

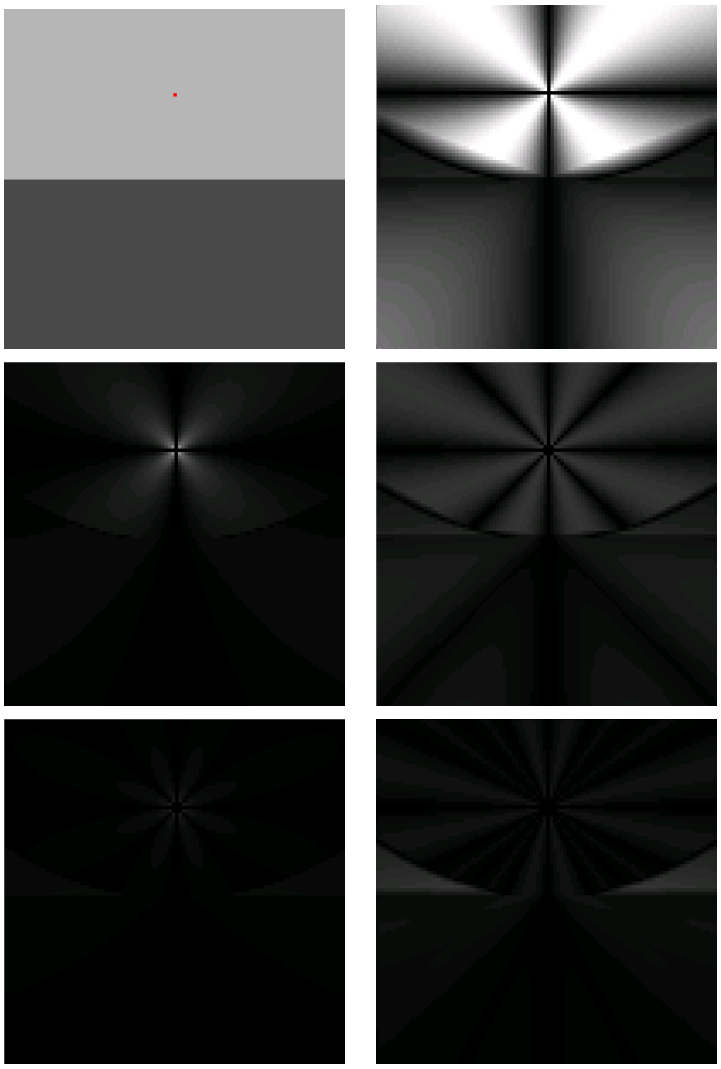


FIG. 2.16 – Relative errors obtained by the different algorithms for a piecewise-constant potential over a regular 100×100 grid. *Left column, top to bottom* : the potential, Fast-Marching with 4 neighbors, Fast-Marching with 8 neighbors. *Right column, top to bottom* : Dijkstra with 4 neighbors, Dijkstra with 8 neighbors, Dijkstra with 16 neighbors. All the images are represented with the same gray level scale : black : 0%, white : $\geq 40\%$

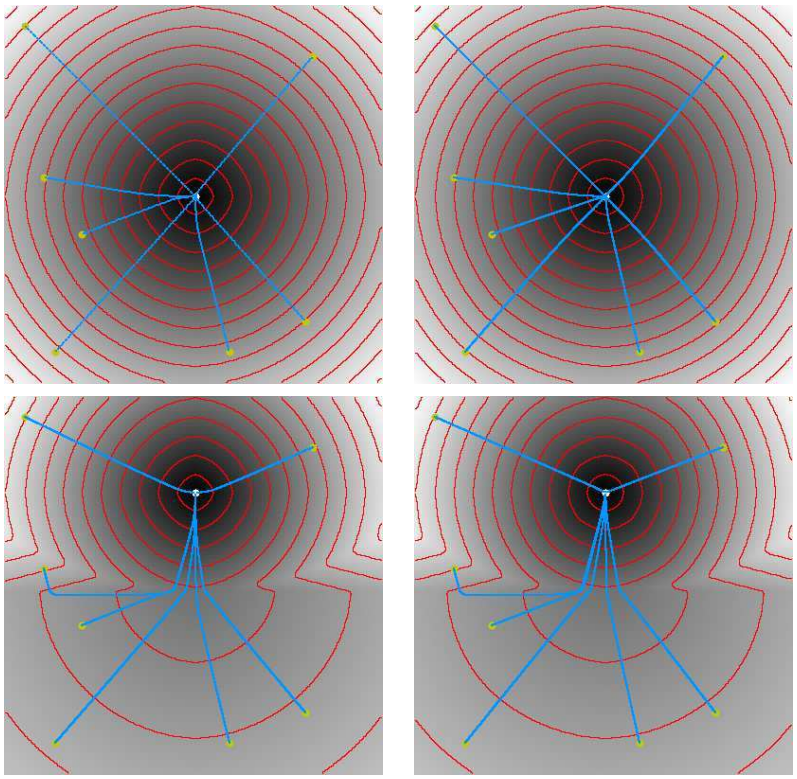


FIG. 2.17 – Shortest paths obtained for a constant potential (*top*), and for a piecewise constant potential (*bottom*), for the Fast-Marching algorithm with 4 (*left*) and 8 (*right*) neighbors.

It is possible to discretize the Eikonal equation on each simplex, and thus to obtain a generalisation of (II.3.1) and (II.3.3). Let us consider a simplex $S_l^{(k)}$ of dimension k, and let us denote by $v_1 \dots v_k$ the values on its vertices. Let us define $u = U(i_1, i_2 \dots i_n)$. We have

$$\nabla U \approx \left(\frac{v_1 - u}{h}, \dots \frac{v_k - u}{h} \right) \tag{II.3.26}$$

and

$$k u^2 - 2u \sum_{i=1}^{k} v_i + \sum_{i=1}^{k} v_i^2 - h^2 P(i,j)^2 \tag{II.3.27}$$

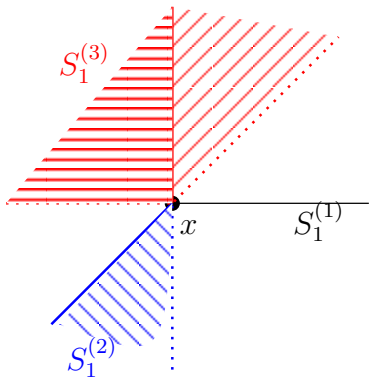


FIG. 2.18 – Some simplices adjacent to x in dimension 3.

Properties shown in 2D can easily be generalised.
In the case when

$$\Delta' \stackrel{\text{def.}}{=} (\sum_{i=1}^{k} v_i)^2 - k(\sum_{i=1}^{k} v_i^2) + kh^2P(i,j)^2$$
$$= kh^2P(i,j)^2 - \frac{1}{2}\left(\sum_{i=1}^{k}\sum_{j=1}^{k}(v_i - v_j)^2\right) \geq 0 \quad \text{(II.3.28)}$$

the bigger solution of this equation is

$$u_2 \stackrel{\text{def.}}{=} \frac{\left(\sum_{i=1}^{k} v_i\right) + \sqrt{\Delta'}}{n} \quad \text{(II.3.29)}$$

In the case when $\Delta' \geq 0$, we thus have

$$u_2 \geq v_l$$
$$\Leftrightarrow$$
$$\sum_{i=1}^{k}(v_i - v_l) \geq 0 \text{ or } \left(\sum_{i=1}^{k}(v_i - v_l)\right)^2 \leq h^2\Delta' \qquad (\mathbf{C_l})$$

If the set $\mathbf{C}$ of all conditions $(\mathbf{C_l})$ is verified, we have $\Delta' \geq 0$ have $\forall l \;\; u_2 \geq v_l$.

We thus define

$$\theta_P^{(k)} : (x_i) \in \mathbb{R}^{2k} \mapsto \begin{cases} \frac{\sum_{i=1}^{k} x_i + \sqrt{kP^2 - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}(x_i - x_j)^2}}{2} & \text{if } \mathbf{C}^1 \\ +\infty & \text{otherwise} \end{cases} \quad \text{(II.3.30)}$$

$$s_l^{(k)} \stackrel{\text{def.}}{=} \theta^{(2)}_{hP(i,j)}(U(i_1 + h \dots i_k \dots i_n) \dots U(i_1 \dots i_k + h \dots i_n)) \quad \text{(II.3.31)}$$

and

$$\theta_P(a_1^+, \dots a_n^+, a_1^-, \dots a_n^-) \in R^{+2n} \stackrel{\text{def.}}{=} \min_{\substack{k=1..n \\ a^{\pm}_{i_1} \dots a^{\pm}_{i_k}}} \{\theta_P^{(k)}(a^{\pm}_{i_1} \dots a^{\pm}_{i_k})\} \quad \text{(II.3.32)}$$

Therefore, we use the following update scheme :

$$U(i_1, \dots i_n) \leftarrow \min_{\substack{i=1..k \\ j=1..K_i}} \{s_j^{(i)}\} = \theta_{hP_{(i,j)}}(U(i_1 + h, \dots, i_n), \dots, U(i_1, \dots, i_n + h), U(i_1 - h, \dots, i_n), \dots, U(i_1, \dots, i_n - h)) \quad \text{(II.3.33)}$$

As in the case of 2D, it is possible to restrict the calculations to simplices such that their vertices have values smaller than current value $U(i, j)$.
The algorithm is then the same as in dimension 2.

Complexity. The update state requires 3^n total computations for each vertex. The complexity of the algorithm is thus *a priori* $O(N(log(N)+n3^n))$, where N is the number of vertices explored by the algorithm.

Correctness. The 2D proof can be exactly transposed. It mainly relies on the following lemma :

Lemma II.3.5.1
- θ_P *is continuous.*
- θ_P *is non-decreasing in each of its variables.*

[1] For practical purposes, in order to check this condition, one just need to compute the maximal solution of the equation. If this solution exists and is bigger than all the v_i, then **C** holds.

Proof :

- As in the 2D case, continuity is derived from the study of connections at points belonging to the border of **C** conditions. More precisely, we can show that if equality holds in ($\mathbf{C_1}$), the solution obtained on the current simplex is equal to the solution obtained on the sub-simplex obtained by removing the l^{th} vertex. This generalises the property illustrated by figure 2.11 :
  Assume that we are on the border of the domain defined by **C**. Then there exists l such that $\left(\sum_{i=1}^{k}(v_i - v_l)\right)^2 = h^2\Delta'$, $\sum_{i=1}^{k}(v_i - v_l) \leq 0$ and $u_2 = v_l$. After some calculations, we can write
  $$v_l = \frac{\sum_{i\neq l} v_i + \sqrt{(k-1)h^2P_{xy}^2 - \frac{1}{2}\sum_{i\neq l}\sum_{j\neq l}(v_i - v_j)^2}}{k-1}$$
  hence
  $$u_2 = v_l = \frac{\sum_{i\neq l} v_i + \sqrt{(k-1)h^2P_{xy}^2 - \frac{1}{2}\sum_{i\neq l}\sum_{j\neq l}(v_i - v_j)^2}}{k-1},$$
  which corresponds with the solution on the sub-simplex obtained by removing the l^{th} vertex (this solution being clearly bigger than v_i for all $i \neq l$).
- Growing is derived from growing of the $\theta^{(k)}$ functions with respect to each of their variables in the domain where they are finite.

□

The convergence proof is now exactly parallel to the one in 2D : stability and ordering are proved in the same way. Monotony of the scheme comes from monotony of θ_P. In order to prove consistency, we can demonstrate as in dimension 2 that if we define

$$S(h,(x_1,\ldots,x_n),t,U) \stackrel{\text{def.}}{=} \frac{\theta_{hP_{(x,y)}}(U(x_1+h,\ldots x_n),\ldots) - t}{h} \quad \text{(II.3.34)}$$

then we have

$$\lim_{\substack{h\to 0\\(x_1',\ldots,x_n')\to(x_1,\ldots,x_n)\\\xi\to 0}} S(h,(x',y'),\varphi(x',y')+\xi,\varphi+\xi) = \theta_{hP_{(x,y)}}(d_1\varphi(x,y),\ldots d_n\varphi(x,y),-d_1\varphi(x,y),\cdots - d_n\varphi(x,y),). \quad \text{(II.3.35)}$$

This quantity vanishes if and only if $\nabla\varphi$ satisfies the Eikonal equation.

II.3.5.1 Improving the running time

The computation of (II.3.33) requires up to the resolution of 3^n quadratic equations. As in the 2D case, it is possible to reduce this number of operations.
Let us define $A_k = (i_1 \dots, i_k+h, \dots i_n)$ if $U(i_1 \dots, i_k+h, \dots i_n) \leq U(i_1 \dots, i_k-h, \dots i_n)$, and $U(i_1 \dots, i_k - h, \dots i_n)$ otherwise. Up to a permutation of coordinates, we can assume that $U(A_1) \leq \dots \leq U(A_n)$.
We then define

$$\begin{aligned} S_*^{(n)} &= ((i_1 \dots i_n), A_1, \dots A_n) \\ S_*^{(n-1)} &= ((i_1 \dots i_n), A_1, \dots A_{n-1}) \\ &\dots \\ S_*^{(1)} &= ((i_1 \dots i_n), A_1) \end{aligned} \tag{II.3.36}$$

and $s_*^n \dots s_*^1$ the corresponding values.
We thus have the following result – which generalise the result obtained in II.3.2.1 :

Proposition II.3.5.2

- *For all $k \in [1..n]$, for all simplex $S_l^{(k)}$ of dimension k, if $s_*^{(k)} \neq \infty$ then $s_*^k \leq s_l^{(k)}$.*
- *For all $k \in [2..n]$ if $s_*^{(k)} \neq +\infty$, then $s_*^{(k-1)} \neq +\infty$ and $s^{(k)} \leq s_*^{(k-1)}$.*

Proof :
Comes immediately from monotony properties of $\theta_P^{(k)}$ in Ω_P^k.

□

We can then use the following algorithm to compute the update step – which was proposed in the appendix of [99] :
which reduces the number of quadratic equations to solve to $n-1$ instead of 3^n.

II.3.6 A step toward anisotropy

The update scheme for an anisotropic is theoretically more complex than the schemes we studied in the previous sections. It will be studied in full generality in section II.4.

Algorithm 6 update(t)

INPUT: A vertex $t = (i_1 \dots i_n)$.
  for k varying from n to 1 **do**
    Compute $s_*^{(k)}$.
    If $s_*^{(k)} \neq \infty$, $U(i_1 \dots i_n) \leftarrow s_*^{(k)}$ and quit
  end for

However, in this section, we will study a useful specific case of anisotropic Fast-Marching algorithm on a regular grid – for which the principal components of the potentials are collinear with the grid axis. For this problem, the previously exposed method works directly. Notice that [186] rapidly mentions a method to solve the equivalent problem of finding distance maps for isotropic potential on orthogonal irregular grid, without explaining precisely how to solve the discretized equation.

Let us consider a n dimensional space, discretized with a regular grid. Let us assume that for any point, the potential has the following expression : $g(x)(\mathbf{v}) = \lambda_{x1} v_1^2 + \dots + \lambda_{xn} v_n^2$ – *i.e.* the tensor g has its principal components aligned with the axis of the grid.

Let us consider a simplex $S_l^{(n)}$ of dimension n, and denote by $v_1 \dots v_n$ the values on the vertices. Let us define $u = U(i_1, i_2, \dots i_n)$. Injecting the discrete form of the gradient in I.3.15, we have

$$\sum_{i=1}^{n} \lambda_{xi} (u - v_i)^2 = 1 \tag{II.3.37}$$

hence

$$\sum_{i=1}^{n} \lambda_{xi} \left(\frac{(u - v_i)}{h} \right)^2 = 1 \tag{II.3.38}$$

$$u^2 \left(\sum_{i=1}^{n} \lambda_{xi} \right) - 2u \left(\sum_{i=1} v_i \lambda_{xi} \right) + \sum_{i=1}^{n} \lambda_{xi} v_i^2 - h^2 = 1 \tag{II.3.39}$$

When

$$\Delta' \stackrel{\text{def.}}{=} (\sum_{i=1}^{n} \lambda_{xi} v_i)^2 - \sum_{i=1}^{n} \lambda_{xi} (\sum_{i=1}^{n} v_i^2) + \sum_{i=1}^{k} \lambda_{xi} h^2 P(i,j)^2$$
$$= \sum_{i=1}^{n} \lambda_{xi} h^2 P(i,j)^2 - \frac{1}{2} \left(\sum_{i=1}^{k} \sum_{j=1}^{k} \lambda_{xi} \lambda_{xj} (v_i - v_j)^2 \right) \geq 0 \quad \text{(II.3.40)}$$

the bigger solution of this equation is

$$u_2 \stackrel{\text{def.}}{=} \frac{\left(\sum_{i=1}^{k} \lambda_{xi} v_i \right) + \sqrt{\Delta'}}{\sum_{i=1}^{n} \lambda_i} \quad \text{(II.3.41)}$$

The update step still consists in computing the update values for each simplex, and to select the minimal one. Similar calculations as in isotropic case show the monotony on each simplex of the previous expression – hence we can deduce monotony and continuity of the update scheme, and then its convergence.

II.3.6.1 Improving the running time

Improving the running time is tougher in this situation.
However, we can notice that if we define $A = (i_1 \pm h, \dots i_k + h, \dots i_1 \pm n)$, $B = (i_1 \pm h, \dots i_k - h, \dots i_1 \pm n)$, and if we assert for example that $U(A) \leq U(B)$, computing solutions on simplices containing A is useless. Indeed, such a solution is bigger than the one in the symmetric simplex obtained by replacing A with B.
We thus define $A_k = (i_1 \dots, i_k + h, \dots i_n)$ if $\mathcal{U}(i_1 \dots, i_k + h, \dots i_n) \leq \mathcal{U}(i_1 \dots, i_k - h, \dots i_n)$, and $\mathcal{U}(i_1 \dots, i_k - h, \dots i_n)$ otherwise. Up to a permutation of coordinates, we can assume that $A_1 \leq \dots \leq A_n$.
We can use the following algorithm to compute the update value :
The total number of quadratic equation to solve is thus 2^n.
Figure 2.19 shows shortest paths and distance maps computed with this method.

Algorithm 7 update(t)

INPUT: A vertex $t = (i_1 \dots i_n)$.

Initialization: Set $s_* = +\infty$.

for k varying from n to 1 **do**

For all k-uplet of points $(A_{i_1} \dots A_{i_k})$, compute the solution s on the simplex $(t, A_{i_1} \dots A_{i_k})$.

$s_* \leftarrow \min\{s, s_*\}$

end for

$U(i_1 \dots i_n) \leftarrow s_*$.

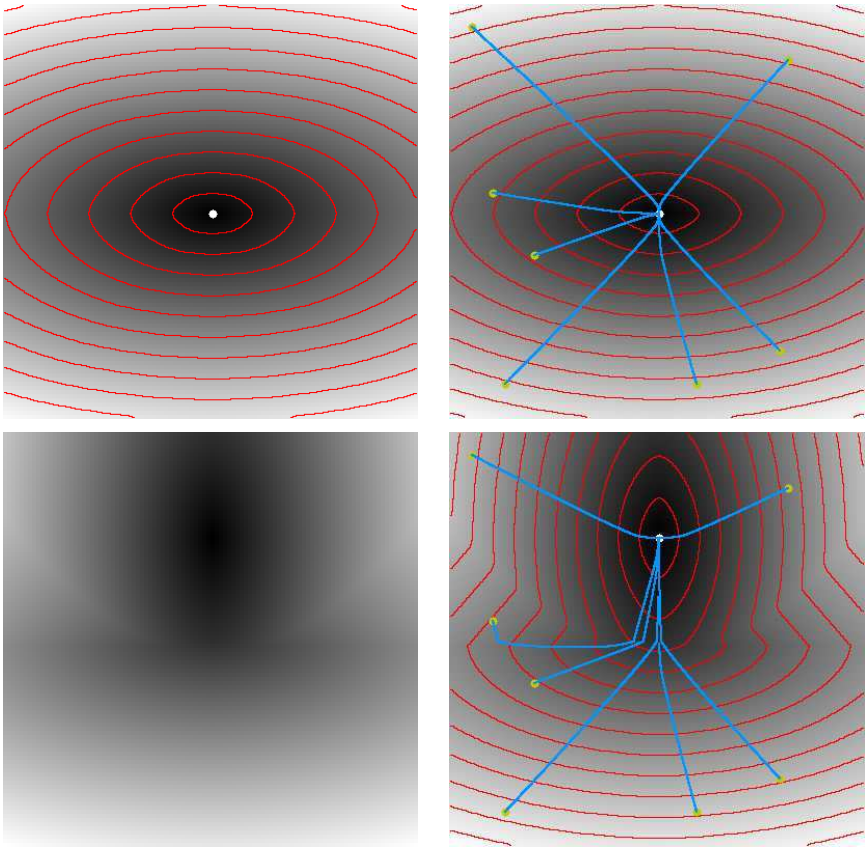


FIG. 2.19 – Distance maps and shortest paths in anisotropic spaces. *Top :* horizontal speed is twice the vertical speed. *Bottom :* in upper half-plane, horizontal speed is twice the vertical speed. The opposite holds in bottom half-plane.

II.4 Anisotropic Fast-Marching, general case

In this section, we propose a scheme for the computation of distance maps and shortest paths in Riemannian manifolds. We generalise results proposed in [97] et [30]. This scheme is directly derived for the ones of previous sections. It also relies on the computation of solutions for each simplices adjacent to the current point. The smallest solution verifying conditions which generalise II.3.5 will be selected as update value.

We will also expose a convergence proof for a large class of cases. In the case of isotropic potential, the condition for convergence is that for any point of the discretization and any adjacent simplex, the angles of the simplex at this point are acute. This is a generalisation of known results in dimension 2 [97, 30].

Notice that in the case of a regular grid in dimension 3, our scheme is equivalent with the one proposed in [163].

II.4.1 Solution computation in a simplex

Generalising the algorithms of previous section on Riemannian manifolds is straightforward. Only the computation of the $s_j^{(i)}$ changes. By the way, as we will find out, cases appear in which convergence of the method is lost.

Inasmuch as introducing anisotropic potentials does not result in extra difficulty, we will directly describe the more general case.

The framework of this section in the one described in I.3.1.6. Let V be a Riemannian manifold of dimension n, discretized with a set of points. We consider a neighborhood system around this point, which consists of several simplices (2.13 and 2.18 are some examples in dimension 2 and 3).

V is locally diffeomorphic to an open subset of $\mathbb{R}^n$, and we will work on such a space to derive the equations in the sequel.

Let x be a point of the discretization of V. Let us consider a simplex $S^{(k)}$ of dimension k, adjacent to x. Up to a translation, we assume that $x = 0$.

The equation we want to discretize is as follow :

$$\|\nabla U\|_x = 1 \tag{II.4.1}$$

which we can rewrite

$$\|M\nabla U\| = 1 \tag{II.4.2}$$

where M is the $n \times n$ symmetric positive definite matrix associated with the potential.
We denote by $x^1 \dots x^k$ the other vertices of the simplex, $v = (v^1 \dots v^k)^T$ the corresponding values, and

$$X \stackrel{\text{def.}}{=} \begin{pmatrix} x_1^1 & \dots & x_n^1 \\ \vdots & & \vdots \\ x_1^k & \dots & x_n^k \end{pmatrix} \tag{II.4.3}$$

We want to estimate $u = U(x) = U(0)$ such that (II.4.2) holds.
Asserting U is affine on the simplex defined by $0, x^1, \dots x^n$. ∇U is therefore constant on the simplex.
For all $i \in [1..k]$ let us consider the function $u_i(\lambda) \stackrel{\text{def.}}{=} U(\lambda x_i)$.
Differentiating this expression, we get : $u_i'(\lambda) =< \nabla U, x^i >$ – which is constant.
Furthermore, we have $u_i(0) = u$ and $u_i(1) = v^i$.
We deduce

$$< \nabla U, x^i >= v^i - u \tag{II.4.4}$$

hence

$$\sum_j U_j x_j^i = v^i - u \tag{II.4.5}$$

and rewriting this in term of matrices,

$$X\nabla U = \mathbf{v} - u\mathbf{1} \tag{II.4.6}$$

If we denote by $X^+ \stackrel{\text{def.}}{=} (X^t X)^{-1} X^t$ the pseudo-inverse of X, we have

$$\nabla U = X^+(\mathbf{v} - u\mathbf{1}) \tag{II.4.7}$$

Notice that X^+, only depends on the geometry of the neighborhood, and can thus be pre-computed.

We also have the constraint (II.4.1), which can be rewritten

$$\nabla U^t M M \nabla U = 1. \tag{II.4.8}$$

$$(X^+(\mathbf{v} - u\mathbf{1}))^t M M X^+(\mathbf{v} - u\mathbf{1}) = 1 \tag{II.4.9}$$

If we define $b \stackrel{\text{def.}}{=} X^{+t} M M X^+$, we thus have

$$(\mathbf{v} - u\mathbf{1})^t b(\mathbf{v} - u\mathbf{1}) = 1 \tag{II.4.10}$$

Hence

$$1 = (\mathbf{v} - t\mathbf{1})^t b(\mathbf{v} - u\mathbf{1}) \tag{II.4.11}$$

$$= u^2(\mathbf{1}^t b\mathbf{1}) - 2u(\mathbf{v}^t b\mathbf{1}) + \mathbf{v}^t b\mathbf{v} \tag{II.4.12}$$

which is a quadratic equation in u.
b is a symmetric positive definite matrix. We denote by $< ., . >_b$ the associated inner product.
The equation becomes

$$u^2||\mathbf{1}||_b^2 - 2u < \mathbf{v}, \mathbf{1} >_b + ||\mathbf{v}||_b^2 - 1 = 0 \tag{II.4.13}$$

When the grid is regular and the potential is isotropic with value P, we have $b = PI_k$, and we find the equation (II.3.27). The case tackled in II.3.6 corresponds corresponds to a diagonal matrix b.
This equation has roots if an only if

$$\Delta' = ||\mathbf{1}||_b^2 + < \mathbf{v}, \mathbf{1} >_b^2 - ||\mathbf{v}||_b^2 ||\mathbf{1}||_b^2 \geq 0 \tag{II.4.14}$$

This has the following geometric interpretation : in the $\mathbb{R}^n$ space equipped with the metric induced by b, the inequality is equivalent to the distance from $\mathbf{v}$ to vect($\mathbf{1}$) being less than 1.
The bigger root is then

$$u_2 = \frac{< \mathbf{v}, \mathbf{1} >_b + \sqrt{\Delta'}}{\|\mathbf{1}\|_b^2} \tag{II.4.15}$$

II.4.1.1 Link between anisotropy and geometry

In the anisotropic framework, if we define

$$X' \stackrel{\text{def.}}{=} XM^{-1} \text{ et } U' = M^{-1}U \tag{II.4.16}$$

we get

$$b = X'^{+t}X'^{+}$$

and

$$\nabla U' = X'^{+}(\mathbf{v} - \mathbf{1}u)$$

Calculations in anisotropic case are then equivalent to calculations in isotropic case where simplices were deformed by the metric of space.
Therefore, from a theoretical, there is no major difference between update steps in isotropic an anisotropic cases.

II.4.1.2 Conditions for convergence

As in the regular grid case, the convergence proof relies on

- the fact that the solution on each simplex is a non-decreasing function of its variables.
- the fact that the solution on each simplex is bigger that the values on other vertices of the simplex.

This second condition can be written

$$(u\mathbf{1} - \mathbf{v}) \geq 0 \quad (\mathbf{C_u}) \tag{II.4.17}$$

In the isotropic case, this has a simple geometric interpretation : it just asserts that the gradient ∇U of the found solution must be in the opposite direction with respect to all edges of the simplex (figure 2.20 (left)).
Let us look for the monotony condition on the simplex.
We start from equation (II.4.15), and differentiate it with respect to v^i. We get

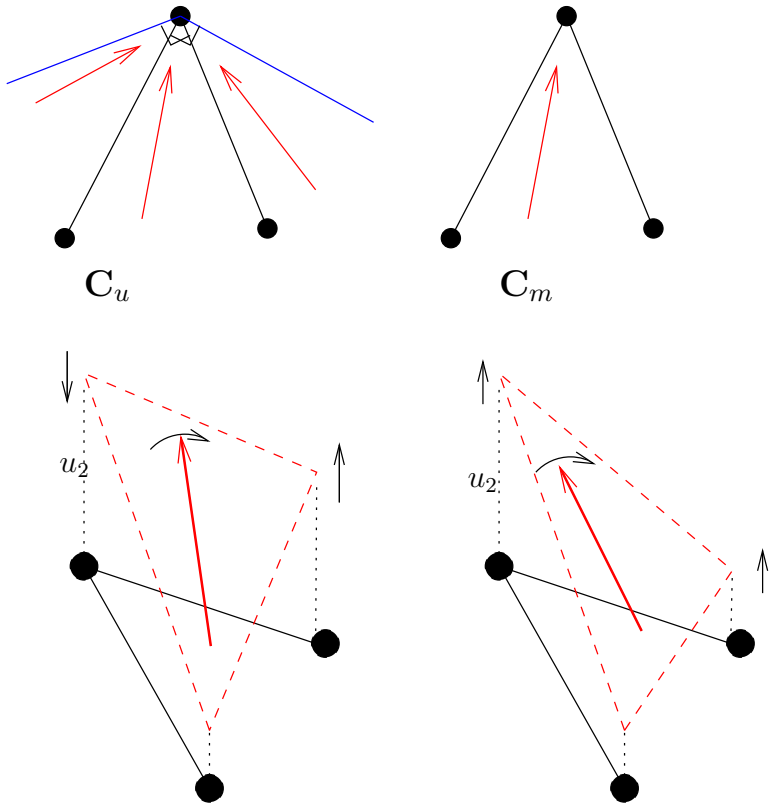


FIG. 2.20 – Geometrical interpretation of conditions $\mathbf{C}_u$ and $\mathbf{C}_m$ for a bi-dimensional simplex (isotropic case). *Top :* $\mathbf{C}_u$ corresponds to the gradient being in the opposite direction with respect to the edges of the simplex. $\mathbf{C}_m$ corresponds to the gradient coming from inside the simplex. *Bottom :* on the left, a solution which satisfies $\mathbf{C}_u$ but not $\mathbf{C}_m$. Increasing the value of the right vertex while keeping the norm of the gradient constant results in a decrease of the solution (black arrows). On the right, a solution satisfying $\mathbf{C}_u$ and $\mathbf{C}_m$.

$$\frac{\partial u_2}{\partial v^i} = \frac{< \frac{\partial \mathbf{v}}{\partial v^i}, \mathbf{1} > \sqrt{\Delta'} + < \frac{\partial \mathbf{v}}{\partial v^i}, \mathbf{1} >< \mathbf{v}, \mathbf{1} > - < \frac{\partial \mathbf{v}}{\partial v^i}, \mathbf{v} >< \mathbf{1}, \mathbf{1} >}{\|\mathbf{1}\|_b^2 \sqrt{\Delta'}} \tag{II.4.18}$$

or

$$\frac{\partial \mathbf{v}}{\partial v^i} = (0, 0, \dots \ 1 \dots, 0, 0) \tag{II.4.19}$$

hence

$$\nabla u_2 = \frac{b\mathbf{1}\sqrt{\Delta'} + b\mathbf{1} < \mathbf{v}, \mathbf{1} > -b\mathbf{v} < \mathbf{1}, \mathbf{1} >}{\|\mathbf{1}\|_b^2\sqrt{\Delta'}} = \frac{b(\mathbf{1}u - \mathbf{v})}{\sqrt{\Delta'}} \tag{II.4.20}$$

i.e. monotony on the simplex holds if and only if

$$b(u_2\mathbf{1} - \mathbf{v}) \geq 0 \quad (\mathbf{C_m}) \tag{II.4.21}$$

In the isotropic case, this condition can be rewritten $X^{+t}\nabla U \leq 0$. From a geometric point of view, it is equivalent to the fact that the gradient comes from inside the considered simplex (figure 2.20 (right)).
In the case of a regular grid with isotropic potential, we have $b = I_k$, and the two conditions $\mathbf{C_u}$ and $\mathbf{C_m}$ coincides – which is coherent with the geometric interpretation – but it is no longer the case in the more general framework. However, we have the following property :

Proposition II.4.1.1
Up to local deformation of the simplex using II.4.16, let us assume that the potential in x is isotropic.
If the angles of the simplex adjacent to x are acute, then
$\mathbf{C_m} \Rightarrow \mathbf{C_u}$.

Proof :
Indeed, if the designated angles are acute, then

$$XX^t \geq 0.$$

Furthermore, let us notice that $b^{-1} = (X^{+t}X^+)^+ = XX^t$.
Let us assume that

$$b(u_2\mathbf{1} - \mathbf{v}) \geq 0.$$

Multiplying by b^{-1}, we have $(u_2\mathbf{1} - \mathbf{v}) \geq 0$.

□

Notice that if the current point x is entirely surrounded by simplices (in the sense that the union of simplices adjacent to x contains a topological neighborhood of x), there will be a simplex containing the gradient, and the $(\mathbf{C_m})$ will be verified. In this case, the previous property asserts that $(\mathbf{C_u})$ will be verified – which will be necessary to the convergence of the scheme.

Finally we denote by $s^{(k)}$ the solution on the simplex which verifying the monotony condition (if it exists). In particular, this solution is superior to the values of other points of the simplex. If such a solution does not exist, we set $s^{(k)} = +\infty$.

Furthermore, we define $\theta_b^{(k)}$ as the function which maps the values on the vertices of the simplex v^i to $s^{(k)}$.

II.4.2 Update scheme

The update scheme simply consists in selecting the smallest value produces by a simplex adjacent to x.

$$U(x) \leftarrow \min_s s^{(i)} \tag{II.4.22}$$

As in the previous sections, the points are explored in a non-increasing ordering. When a point x is transferred to $\mathcal{A}$, the update step is applied to its neighbors. It is also possible to compute updates only from simplices containing x and other points in $\mathcal{A}$.

We denote by $\theta = \min \theta_b^k$ the function that maps the set of values of neighbors of x to the selected update value.

II.4.3 Convergence proof

Here again, the proof relies on the fact that on the border of $\mathbf{C_m}$ conditions, the computed solution is equal to the solution computed on a sub-simplex – which will imply the continuity of θ.

We will need the following lemma :

Lemma II.4.3.1 *If $a \in \mathcal{M}_n(\mathbb{R})$, let us $a_{[i]}$ the matrix from which the i^{th} line and i^{th} column were deleted. If $v \in \mathcal{M}_{n,1}(\mathbb{R})$, let us denote by $v_{[i]}$ the vector from which the i^{th} element was suppressed.*
Let $b \in \mathcal{M}_n(\mathbb{R})$ be a symmetric positive definite matrix.
Then

$$v \text{ solution de } v^t b v = 1 \text{ with } b_i v = 0$$

$$\Rightarrow$$

$$v_{[i]} \text{ solution de } v_{[i]}^t ((b^{-1})_{[i]})^{-1} v_{[i]} = 1$$

Proof :
We define $w = bv$, hence $v = b^{-1}w$.
By hypothesis we have $v^t b v = 1$ and $b_i v = 0$.
Therefore $w^t b^{-1} w = 1$ and $w_i = 0$, and $w_{[i]}^t (b^{-1})_{[i]} w_{[i]} = 1$.
By the way, we have $v_{[i]} = (b^{-1})_{[i]} w_{[i]}$, so that $w_{[i]} = ((b^{-1})_{[i]})^{-1} v_{[i]}$, and $v_{[i]}^t ((b^{-1})_{[i]})^{-1} v_{[i]} = 1$.

□

Let us consider again the discretization of Eikonal equation over the $S^{(k)}$ simplex. (II.4.10) : $(\mathbf{v} - u\mathbf{1})^t b (\mathbf{v} - u\mathbf{1}) = 1$. Let us consider a solution of this equation such that it is at the border of $\mathbf{C_m}$ conditions [2], i.e. $b_l(u\mathbf{1} - \mathbf{v}) = 0$. From the preceding lemma, we have

$$(u\mathbf{1}_{[l]} - v_{[l]})^t ((b^{-1})_{[l]})^{-1} (u\mathbf{1}_{[l]} - v_{[l]}) = 1$$

[2] There is a technical difficulty here – related to positivity condition of Δ' : indeed, $\Delta' \geq 0$ is a necessary condition to the existence of a solution verifying $\mathbf{C_m}$. Therefore, it seems necessary to analyse the behaviour of the solution in the limit case $\Delta' = 0$. However, as in the case of dimension 2 on a regular grid, we can show that $\mathbf{C_m}$ is "stronger" than $\Delta' \geq 0$, i.e. one never has $\Delta' = 0$ and $\mathbf{C_m}$. Indeed, if we assume $\Delta' = 0$, $\mathbf{C_m}$ is rewritten $b\mathbf{1} < v, 1 >_b - bv\|1\|_b^2 \geq 0$. Multiplying with $\mathbf{v}^t$, we have $< v, 1 >_b^2 - \|v\|_b^2 \|1\|_b^2 \geq 0$, which is contradictory with $\Delta' = \|\mathbf{1}\|_b^2 + < v, 1 >_b^2 - \|v\|_b^2 \|1\|_b^2 = 0$.

Yet, $b = X^{+t}MMX^{+}$, hence $b^{-1} = XM^{-1}M^{-1}X^{t}$, and
$(b^{-1})_{[l]} = (X_{[l]})M^{-1}M^{-1}(X_{[l]})^{t}$.
u is then solution of the discretized Eikonal equation on the sub-simplex obtained from $S^{(k)}$ by deleting the l^{th} vertex – this solution being clearly satisfying $\mathbf{C}_m$.)
Continuity of θ follows.

We know give the sketch of convergence proof :

monotony Follows immediately from monotony on each simplex, and from the continuity of θ.

stability The argument is the same as in dimension 2.

consistence In the sequel, we will assume that the simplices of the neighborhood-system homothetically tends toward a single point. It will be the case if the considered space is discretized by regular simplices of side h – or if it is consists of simplices build always in the same way on a regular grid of side h. It is possible to extend the results presented here : for example, they remain true for simplices which volume tends toward 0 and such that the corresponding normalised simplices tend toward a limit simplex. We restricts ourselves to the latter case, in order not to make the notations too heavy.

Let us consider a function $\varphi \in \mathcal{C}^{\infty}$ and $x = (x_1 \dots x_n)$.

Let us consider a point $x' = (x'_1 \dots x'_n)$, and a simplex $(x', x^{(1)} \dots x^{(k)})$ – its associated matrix being hX.

Let us denote $b_{hx'} \stackrel{\text{def.}}{=} (hX)^{+t}M_{x'}M_{x'}(hX)^{+}$ and $u_h = \theta_{b_{hx'}}$ the solution computed on this simplex (if it exists) when the values on other vertices are $\mathbf{v}_h = (\varphi(x^{(1)}) \dots \varphi(x^{(k)}))$. Let us consider $\xi \in \mathbb{R}$

$$u_h + \xi - \varphi(x') - \xi = \frac{<\mathbf{v}_h - \varphi(x'), \mathbf{1}>_{b_{hx'}} + \sqrt{\|\mathbf{1}\|^2_{b_{hx'}} + <\mathbf{v}_h - \varphi(x'), \mathbf{1}>^2_{b_{hx'}} - ||\mathbf{v}_h - \varphi(x')||^2_{b_{hx'}} \|\mathbf{1}\|^2_{b_{hx'}}}}{\|\mathbf{1}\|^2_{b_{hx'}}} \quad \text{(II.4.23)}$$

Furthermore, when $h \to 0$ and $x' \to x$, $\mathbf{v}_h - \varphi(x') \approx hX\nabla\varphi$. Therefore, denoting $b \stackrel{\text{def.}}{=} X^{+t}M_xM_xX^{+}$, we have

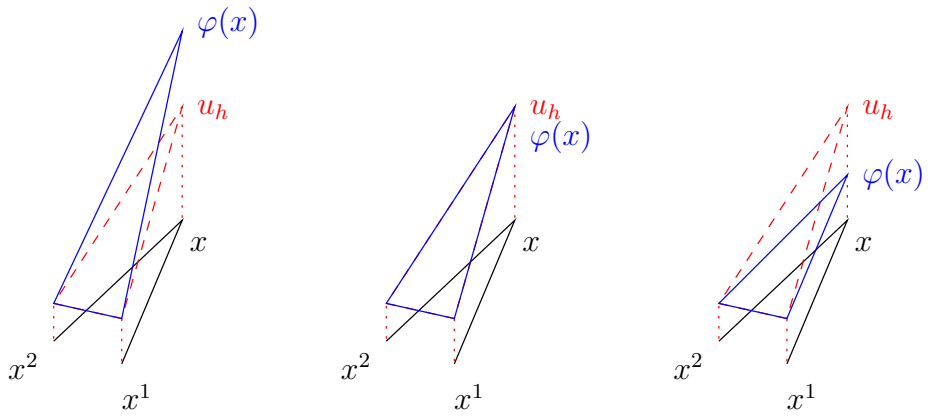


FIG. 2.21 – Consistency : different cases

$$\lim_{\substack{h\to 0\\ x'\to x\\ \xi\to 0}} \frac{u_h - \varphi(x')}{h} =$$

$$\frac{< X\nabla\varphi, \mathbf{1} >_b + \sqrt{\|\mathbf{1}\|_b^2 + < X\nabla\varphi, \mathbf{1} >_b^2 - ||X\nabla\varphi||_b^2 \|\mathbf{1}\|_b^2}}{\|\mathbf{1}\|_b^2} =$$

$$\theta_b^{(k)}(X\nabla\varphi) \quad \text{(II.4.24)}$$

A similar analysis as the one conducted in 2D and relying on the continuity of $\theta = \min \theta_b^{(k)}$ shows that

$\lim_{\substack{h\to 0\\ x'\to x\\ \xi\to 0}} (S(h, x', \varphi(x') + \xi, \varphi + \xi) \overset{\text{def.}}{=} \frac{\theta(.) - \varphi(x')}{h})$ has as a continuous limit. The equivalence between the vanishing of this limit and φ satisfying Eikonal equation remains to be checked.

The underlying intuition is illustrated figure 2.21. We will prove that, in the limit, solutions to the discretized equation exist in a simplex which contains $\nabla\varphi$ (after deformation in the anisotropic case). Depending on how $\|M_x \nabla\varphi\|$ compares to 1, this solution will be strictly inferior, equal, or strictly superior to $\varphi(x)$.

Considering equation (II.4.24), we observe that for each simplex, if $\theta_b^{(k)}(X\nabla\varphi) \neq +\infty$ then

$$\theta_b^{(k)}(X\nabla\varphi) = 0$$
$$\Leftrightarrow < X\nabla\varphi, 1 >_b \leq 0 \text{ et } \|X\nabla\varphi\|_b^2 = \|M_x X^+ X\nabla\varphi\|^2 = 1 \quad \text{(II.4.25)}$$

$$\theta_b^{(k)}(X\nabla\varphi) > 0$$
$$\Leftarrow \|X\nabla\varphi\|_b^2 = \|M_x X^+ X\nabla\varphi\|^2 < 1 \quad \text{(II.4.26)}$$

$$\theta_b^{(k)}(X\nabla\varphi) < 0$$
$$\Leftarrow < X\nabla\varphi, 1 >_b \leq 0 \text{ et } \|X\nabla\varphi\|_b^2 = \|M_x X^+ X\nabla\varphi\|^2 > 1 \quad \text{(II.4.27)}$$

By the way, $X^+X\nabla\varphi$ is the projection of $\nabla\varphi$ onto the linear span of the simplex.

Several cases have to be considered :

- If $\|M_x\nabla\varphi\| < 1$, then for any simplex, $\|M_x X^+ X\nabla\varphi\| < 1$. From (II.4.26) we have $\theta_b^{(k)}(X\nabla\varphi) = +\infty$ or $\theta_b^{(k)}(X\nabla\varphi) > 0$. Therefore $\lim S > 0$.
- If $\|M_x\nabla\varphi\| = 1$, let us consider the n-dimensional simplex containing $\nabla\varphi$ after deformation – such that $\theta_b^{(k)}(X\nabla\varphi) \neq +\infty$. Then we have $\theta_b^{(k)}(X\nabla\varphi) = 0$ from (II.4.25). For other simplices, as in the previous point, we have $\theta_b^{(k)}(X\nabla\varphi) = +\infty$ or $\theta_b^{(k)}(X\nabla\varphi) > 0$ and then $\lim S = 0$.
- If $\|M_x\nabla\varphi\| > 1$, we wish to show that there exists a simplex such that $\theta_b^{(k)}(X\nabla\varphi) \neq +\infty$, $< X\nabla\varphi, 1 >_b \leq 0$ and $\|X\nabla\varphi\|_b^2 = \|M_x X^+ X\nabla\varphi\|^2 > 1$. From (II.4.27), this would entail $\lim S < 0$.
  $\theta_b^{(k)}(X\nabla\varphi) \neq +\infty$ is equivalent to the existence of a solution u of discretized Eikonal equation in the simplex associated with X, with values $X\nabla\varphi$ on the vertices – $\mathbf{C}_m$ being satisfied.
  Let us consider a simplex $S^{(n)}$ which contains $\nabla\varphi$ after deformation, and denote by X_n its associated matrix. We define $u = \varphi(x)$ – it is clear that u is not solution of the discretized equation on $S^{(n)}$. Let us progressively decrease u. Two scenarios can occur.

– Either we get a u value verifying the discretized Eikonal equation, at a stage when $\mathbf{C}_m$ still holds. The problem is then solved.
– Either one of the $\mathbf{C}_m$ conditions is violated before, which means that $b_n(u\mathbf{1} - X_n\nabla\varphi) < 0$. In the limit, ∇U belongs to a sub-simplex $S^{(n-1)}$.

Iterating this process, we travel along a family of decreasing simplices

$$(S^{(n)}, S^{(n-1)}, S^{(n-2)} \dots)$$

. We will denote by X_i and b_i the corresponding matrices. Notice that $< X_i\nabla\varphi, 1 >_{b_i} \leq 0$ holds for all these simplices
If a solution v of the discretized Eikonal equation is found a simplex $S^{(i)}$ such that $\|M_x X_i^+ X_i \nabla\varphi\|^2 \leq 1$. Then v verifies $v \geq \varphi(x)$, which is absurd : indeed the value of u when entering the simplex was strictly smaller than $\varphi(x)$.
This processed necessarily lead to a solution – in the worst case in the $S^{(1)}$ simplex, which concludes the proof.

ordering The argument is the same as in dimension 2.

We thus have the following theorem :

Theorem II.4.3.2
The distance map computed from the algorithm proposed in II.4.2 converges toward the viscosity solution of I.3.15 when the size of the simplices converges to 0.

If the angles of the simplices have a maximal value θ, no obtuse angle can appear under a deformation by a tensor with anisotropy ratio less than $(tan(\theta/2))^{-1}$. This lead to the following sample results.

- In dimension 2 :

> **Theorem II.4.3.3**
> *In the following cases, the distance map computed from the algorithm proposed in II.4.2 converges toward the viscosity solution of I.3.15 when the size of the simplices converges to 0 :*
> – *Regular grid (figure 2.9), and isotropic potentials (section II.3) – or anisotropic potentials with principal components collinear with the grid axis (section II.3.6.)*
> – *8 neighbors system (figure 2.13), potentials with anisotropy ratio less than* $(tan(\pi/8))^{-1} = \frac{1}{\sqrt{2}-1} \approx 2.4$.
> – *Neighborhood system consisting of equilateral triangles, potentials with anisotropy ratio less than* $(tan(\pi/6))^{-1} = \sqrt{3} \approx 1.7$.

- In dimension 3 :

> **Theorem II.4.3.4**
> *In the following cases, the distance map computed from the algorithm proposed in II.4.2 converges toward the viscosity solution of I.3.15 when the size of the simplices converges to 0 :*
> – *Regular grid and isotropic potentials (section II.3.5) – or anisotropic potentials with principal components collinear with the grid axis (section II.3.6.)*
> – $\mathcal{S}_{48}a$ *neighborhood system (cf. section II.5.2), potentials with anisotropy ratio less than* ≈ 1.9.
> – $\mathcal{S}_{48}b$ *neighborhood system (cf. section II.5.2), potentials with anisotropy ratio less than* $(tan(\pi/6))^{-1} = \sqrt{3} \approx 1.7$.
> – *Neighborhood system consisting of regular tetrahedron, potentials with anisotropy ratio less than* $(tan(\pi/6))^{-1} = \sqrt{3} \approx 1.7$.

II.5 Numerical Results

II.5.1 Dimension 2

This section presents some results obtained by the algorithm in 2D. If need be, we applied the algorithm to simplices with obtuse angles – in this case we selected the smallest solution satisfying both $\mathbf{C}_m$ and $\mathbf{C}_u$.

Figure 2.22 shows results obtained with a 4 neighbors system (figure 2.9), with anisotropic potentials non-collinear with the axis. Such a potential creates obtuse angles in the deformed simplices, and the distance map does not seem to converge toward their theoretical value.
Figure 2.23 (top and middle) shows results in the same space, obtained with a 8 neighbors system (figure 2.13). In this case, the algorithm converges. In fact as long as the maximal anisotropy ratio is less than $(tan(\pi/8))^{-1} \approx 2.4$, the deformed angles remain acute – whatever the direction of the tensor is. On the opposite, if anisotropy keeps on increasing (bottom), obtuse angles appear, and convergence is lost.

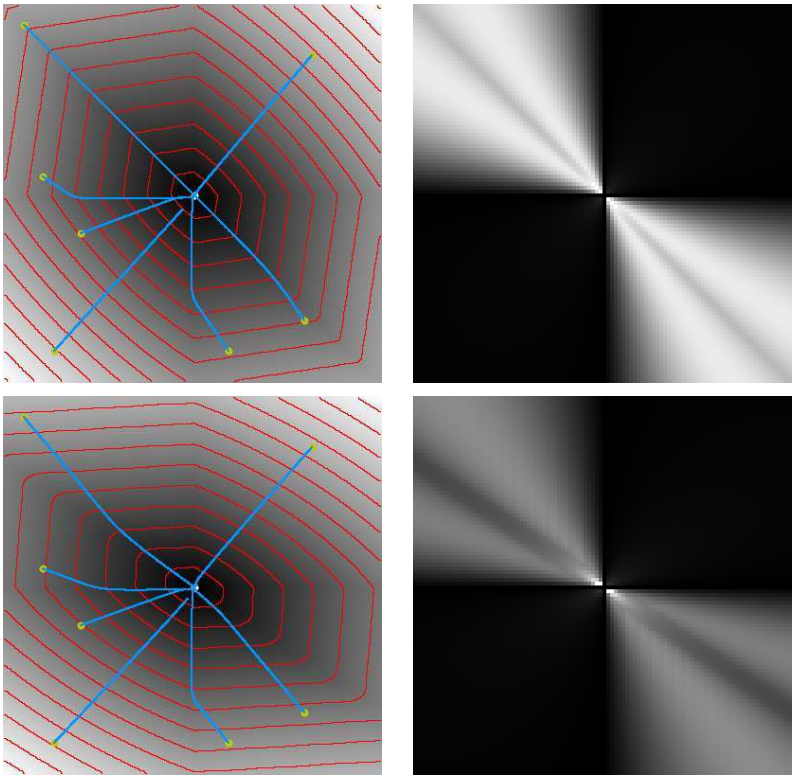


FIG. 2.22 – Distance maps, level sets and shortest paths for a uniform anisotropic potential, obtained with a 4 neighbors system. Anisotropy ratio of the tensor is 2. *Top :* principal direction is collinear with $e_{3\pi/4}$. *Bottom :* principal direction is collinear with $e_{5\pi/6}$.

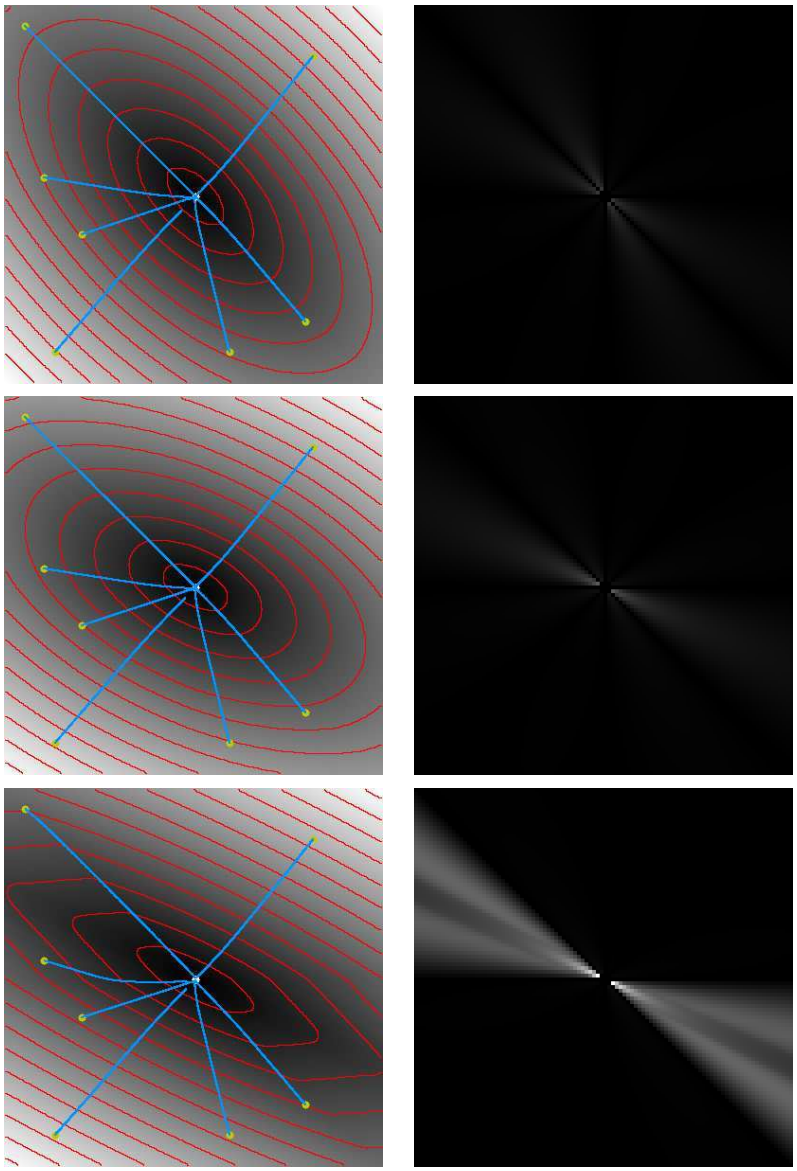
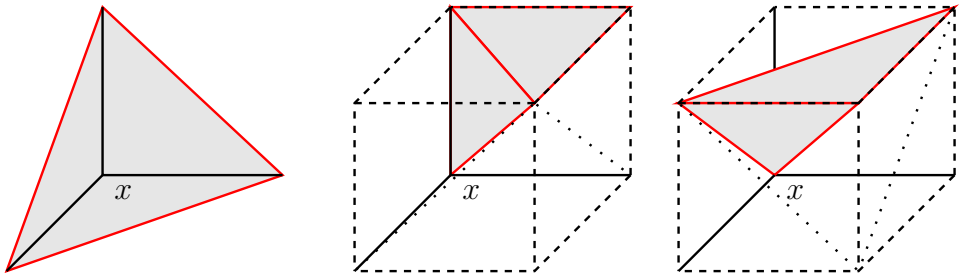


FIG. 2.23 – Distance maps, level sets and shortest paths for a uniform anisotropic potential, obtained with a 4 neighbors system. *Top :* Anisotropy ratio of the tensor is 2, principal direction is collinear with $e_{3\pi/4}$. *Middle :* Anisotropy ratio of the tensor is 2, principal direction is collinear with $e_{5\pi/6}$. *Bottom :* Anisotropy ratio of the tensor is 4, principal direction is collinear with $e_{5\pi/6}$.

II.5.2 Dimension 3

In this section, we present some results obtained by the algorithm in 3D for different uniform potentials, and different neighborhood systems.(figure 2.24). The first system ($\mathcal{S}_8$) consists in 8 simplices. The second ($\mathcal{S}_{48}a$) and third one ($\mathcal{S}_{48}b$) are bigger, and consists in 48 simplices.



8 simplexes 48 simplexes - a 48 simplexes - b

FIG. 2.24 – Different neighborhood systems in dimension 3

The starting set is reduced to a single point. The algorithm is illustrated for three potentials.

- an isotropic potential (figure 2.25). In the $\mathcal{S}_8$ case, the scheme corresponds to the specific case detailed in section II.3.5.
- an anisotropic potential, collinear with the axis – speeds in the different directions being 1,2 and 3 (figure 2.26). In the $\mathcal{S}_8$ case, the scheme corresponds to the specific case detailed in section II.3.6.

  In these cases, for all the neighborhood systems, convergence is proved. However, the choice of a bigger neighborhood system increases the precision. $\mathcal{S}_{48}a$ or $\mathcal{S}_{48}b$ give qualitatively equivalent results.
- the same anisotropic potential, but non-collinear with the axis(figure 2.27). Convergence is lost for $\mathcal{S}_8$. The last figure shows result obtained for $\mathcal{S}_{48}a \cup \mathcal{S}_{48}b$ – which is not significantly better than the ones obtained for the two systems independently.

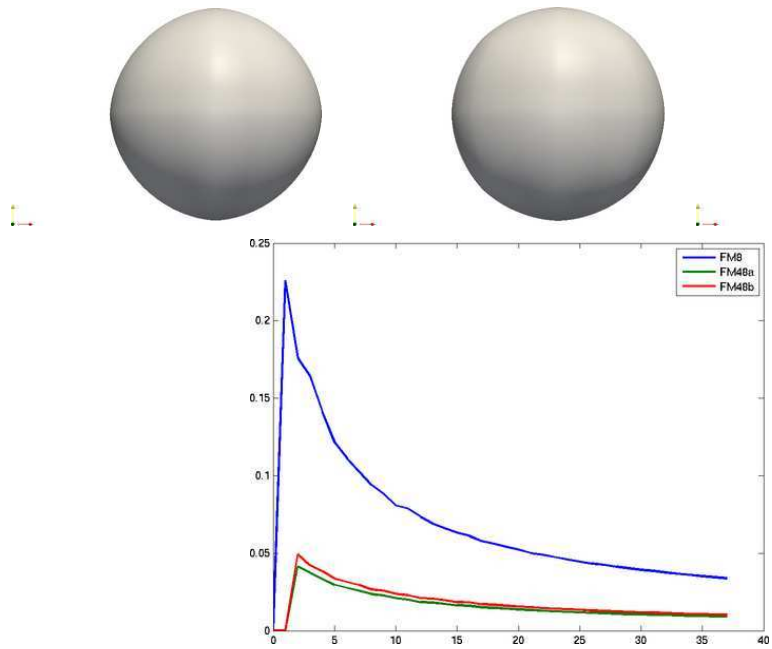


FIG. 2.25 – Results for a uniform isotropic potential. *Top :* level sets for $\mathcal{S}_8$, $\mathcal{S}_{48}a$ and $\mathcal{S}_{48}b$. *Bottom :* mean relative error for the three systems, as a function of distance from starting point

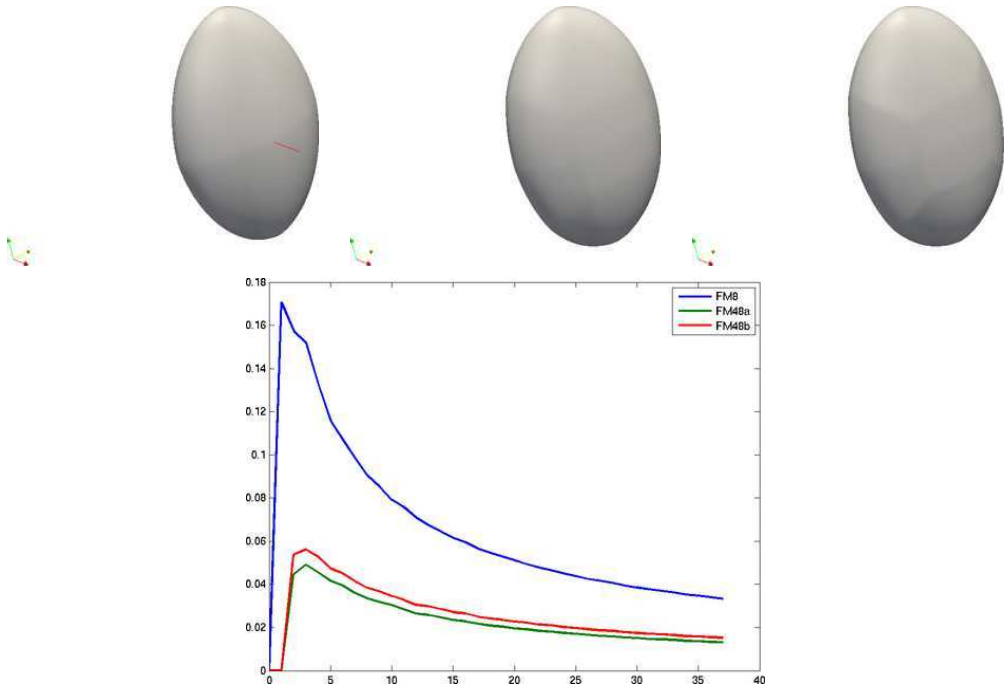


FIG. 2.26 – Results for a uniform anisotropic potential, collinear with the axis. *Top :* level sets for $\mathcal{S}_8$, $\mathcal{S}_{48}a$ and $\mathcal{S}_{48}b$. *Bottom :* mean relative error for the three systems, as a function of distance from starting point.

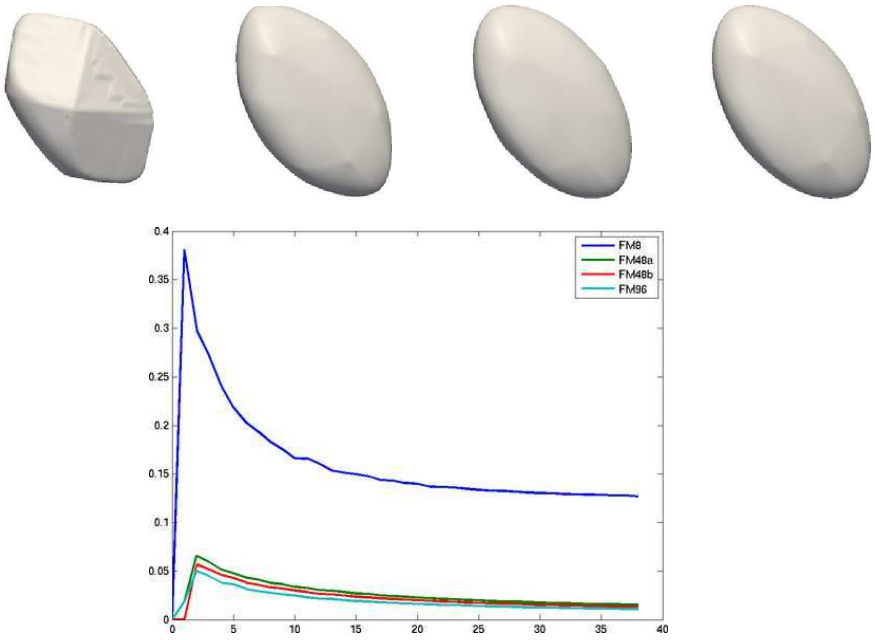


FIG. 2.27 – Results for a uniform anisotropic potential, non-collinear with the axis. *Top :* level sets for $\mathcal{S}_8$, $\mathcal{S}_{48}a$, $\mathcal{S}_{48}b$ and $\mathcal{S}_{48}a \cup \mathcal{S}_{48}b$. *Bottom :* mean relative error for the three systems, as a function of distance from starting point.

II.6 Other algorithms for shortest paths computation

In the case of dimension 2 with a constant potential (section I.3.1.1), numerous methods have been developed to compute distance maps. A recent review of main algorithms can be found in [57]. Notice that in this case, exact algorithms exists, with quasi-linear complexity in the size of space.

It is also possible to use the idea of approximating continuous shortest paths by discrete ones in order to compute geodesics on manifolds represented by random point clouds[201, 136].

In dimension 2, [209] proposed an algorithm similar to Fast-Marching almost simultaneously. This algorithm was inspired by control theory, and can be generalised in dimension 3 [87]. It is equivalent with Fast-Marching in the cas of isotropic potentials, but does not converge to the theoretical solution in more general cases.

Several variants of Fast-Marching have been proposed in order to obtain convergent scheme in presence of obtuse angles – or dually when anisotropy is important.

In dimension 2, [97] proposed a method to suppress obtuse angles based on extending the neighborhood. However this extension increase running time, and its implementation seems to be tricky in bigger dimension. In [190], the authors propose a more general method, based on an extended front – the amount of extension depending on the anisotropy ratio. In the case of parametric manifolds, [197] proposes a fast method for extending the neighborhood.

In dimension 3 – the deformed space being sampled by a regular grid – [31] proposed a generic splitting algorithm based on integer programming, which extends the method proposed in [197].

It is also possible to keep the Fast-Marching general sketch, but to allow updates for points already in $\mathcal{A}$. When the value of such a point is modified, a recursive correction of its neighboring points is performed [104]. For practical purposes, the increase of running time again depends on anisotropy ratio. While the convergence is not guaranteed, the algorithm seems to behave well

from strongly anisotropic metrics.
Finally, a method was proposed for anisotropic potentials in dimension 3 on regular grids [161]. It appears to be equivalent with our formulation in this case.

II.7 Conclusion, discussion

We proposed a new presentation of Fast-Marching algorithm. We emphasised on the connexions with Dijkstra algorithm. Our formulation is easily extended to bigger dimensions, to anisotropic potentials, and to manifolds, and admits a unified proof.
It would be of high interest to compare and synthesise the algorithms for shortest paths computations, in the cases when our scheme is not convergent. To our knowledge, such a work has not been done yet.

Chapitre III

Tubular structures segmentation using shortest paths

Introduction

In this chapter, we propose an application of shortest paths to the segmentation of tubular structures – mainly vessels in bi-dimensional medical images. After an introduction (section III.1), we propose to recast the 2D segmentation problem as a geodesic computation over a 4-dimensional space in section III.2. An additional scale dimension gives access to the local width of the vessels, and allows the direct extraction of the centerline of the vessel. A rotational dimension reduces erroneous detection when two vessels are overlapping.
In section III.3, we then propose an application of this framework to a flow-based vessel segmentation algorithm for optical cortical imaging.
Finally, in section III.4 we show how to apply this framework to the extraction of networks of roads or vessels.

Contents

Publications related to this work

This chapter is based on the work published in the following articles :

- *Extraction of Tubular Structures over an orientation domain.* [167]
- *SIFT-based Sequence Registration and Flow-based Cortical Vessel Segmentation applied to High Resolution Optical Imaging Data.* [168]
- an article currently under review in MIA .

III.1 Tubular structures segmentation

In the sequel, we will denote by *tubular structures* either roads in satellite images, or blood vessels in medical images (figure 3.1). As we will see, both share common characteristics, which allow their segmentation in one unified framework. While our method is originally designed for medical imaging applications, we will also show some of its results on high-resolution satellite imaging.

FIG. 3.1 – *Left :* roads in a satellite image. *Right :* vessels in a medical image (cortical imaging)

Blood Vessels Extracting tubular structures is a central problem in medical imaging. Detection of vessels and vessels networks in bi-dimensional medical images is of primary interest to help medical diagnostic. The extraction of an accurate network allows one to compute meaningful information such as the local width of the vessels and the connectivity of the networks from a single planar observation. These problems are critical in retinal imaging[143] for example, where they allow to diagnose pathologies such as *Diabetic Retinopathy* [47].
Several problems arise to correctly perform the segmentation task. Many acquisition modalities produce highly noisy images. Furthermore, vessels usually exhibit complex tree-like structures that require a careful processing. Another specific difficulty in 2D imaging is the overlapping of vessels : two distinct vessels in real anatomy can give rise to a crossing in the plane of the image (figure 3.2).

Roads

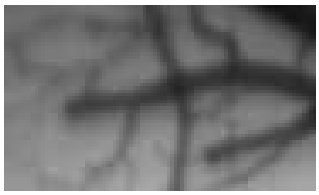


FIG. 3.2 – A crossing of two blood vessels in a cortical image.

Road segmentation is of primary interest for the automatic analysis of satellite images. Automatic or semi-automatic cartography mainly aims at updating geographic information systems [194] – with applications to road traffic management or automated navigation systems.

III.1.1 State of the art

The problem of tubular structure extraction has received considerable attention in the computer vision and medical imaging communities. The recent reviews [100] and [44] give extremely good topical outlines of the domain. A survey on several retinal imaging specific methods can also be found in [126]. Several classes of methods have been proposed to segment tubular structures. They generally rely on the use of a local detector, post-processed by a method that links locally detected structures.
Local detectors allow to detect points belonging to tubular structures or portions of tubular structures depending on the modality of the image. Local detectors include various methods : thresholding of images intensities, ridge or crest detection [8, 164, 34], wavelets [89, 195, 41, 105], line detector for low resolution satellite imaging [61, 135], gabor filters [170], differential operators [153, 112], vesselness measures [63, 53, 114] or matching filters [71, 36, 82, 18, 35, 115, 159] – recently combined with learning processes [72].

Many methods allow to link or post-process the locally detected points.
Among classical methods (inspired by ideas which early arose in computer vision community for edge detection [33]), thresholding [200], fusion processes [93, 76, 130, 218], region growing algorithms [58, 180, 220, 80], front

propagation [128, 188, 50], or pixel classification [41, 171, 193] techniques were proposed. Active contours [134, 151, 172], deformable models [142, 133], and more recently geometric flow based methods [212, 53] can also be used to fit models of tubular structures or boundaries to the data.

Geodesic based methods are another class of methods allowing the linking of local features – usually pixels intensities : the notion of shortest path proved to be efficient for the extraction of salient curves in 2D or 3D images, see for instance [37]. Geodesic curves can also be used to extract tubular structures centerlines in 3D medical images, as proposed by [49] and by [177]. In [120], the authors proposed to extend the shortest path computation to a higher dimensional domain. They include local radius of the tubular structures as an additional scale dimension in order to stabilize the computations and to select the centerline without any post-processing.

Another way of linking local features is the class of tracking methods which start from a point belonging to a vessel (either user-defined, or detected using a *ad-hoc* method with respect to the modality), and iteratively track the vessel by analyzing the neighborhood of the current point in the direction of the tubular structures (look-ahead) [152, 111, 179, 67, 121, 224, 206, 45, 199, 24]. Kalman filtering is also used to robustify the tracking process [214, 223]. While some of these methods can handle junctions, they usually fail to deal robustly with crossings in the case of bi-dimensional medical images.

III.1.2 Shortest paths methods for road/vessels segmentation

Our work was mainly inspired by shortest paths methods such as [37] and [120]. Starting from an image $I : [0,1]^2 \to \mathbb{R}$, the basic idea is to compute road/vessels as shortest paths in the plane of the image. A potential must be designed such that computed shortest paths correspond to actual road/vessels in the images. Since in most medical images, vessels appear to be darker than the background, a natural idea is to design the potential as a non-decreasing function of the gray level – doing so, shortest paths are likely to follow dark areas of the images, *i.e.* vessels. This is illustrated in fig 3.3. The opposite holds for

satellite images, in which roads are usually lighter than the background. Therefore in this case, the potential will be designed as a non-increasing function of the Gray level. These methods can also be extended to 3D images, and can thus be used to segment anatomic structures *e.g.* in endoscopy.

However, as illustrated in figure 3.4, these methods usually fail to find the centerline of the targeted vessel if the centerline does not correspond to a minima of gray level along the section of the tubular structure. They are, as well, unable to directly recover its radius, which evaluation may have significance, *e.g.* in retinal imaging. Several attempts have been made to address this problem. One of them is to apply a Gaussian blurring to the image as a pre-processing step, hoping that after this operation, the potential will be lower at the centerline of vessels. It is however unclear how the intensity of the blurring should be chosen, and how it affects the obtained segmentation. It is also possible to refine a first coarse segmentation using skeletization-like methods as a post-processing [50, 208, 196, 77]. Notice also that there exist an important litterature concerning computation of medial axis (e.g. [192, 28]), but such methods can usually only be applied to binary images. As we will show, our method will be able to compute centerlines in a more intrinsic way.

An attempt to intrinsically compute centerlines and radii is proposed in [120]. The authors propose to lift the 2D image to a 3D space taking into account radius of vessels. They design a local detector which allows to evaluate the likelihood of the presence of the centerline of a vessel of radius r at every point of the image. Then they compute shortest paths in this 3D space, the potential being a non-increasing function of the likelihood. However, due to their choice of local detector, their method is extremely sensitive to initialization and parameters. Their idea of using a radius space was also adapted in a Dijkstra-like framework [160].

Notice that another algorithm inspired from this framework was proposed very recently in [16, 17]. After a preprocessing of the image, it uses an anisotropic fast-marching in such a 3D space to accurately segment vessels – while not handling intersections.

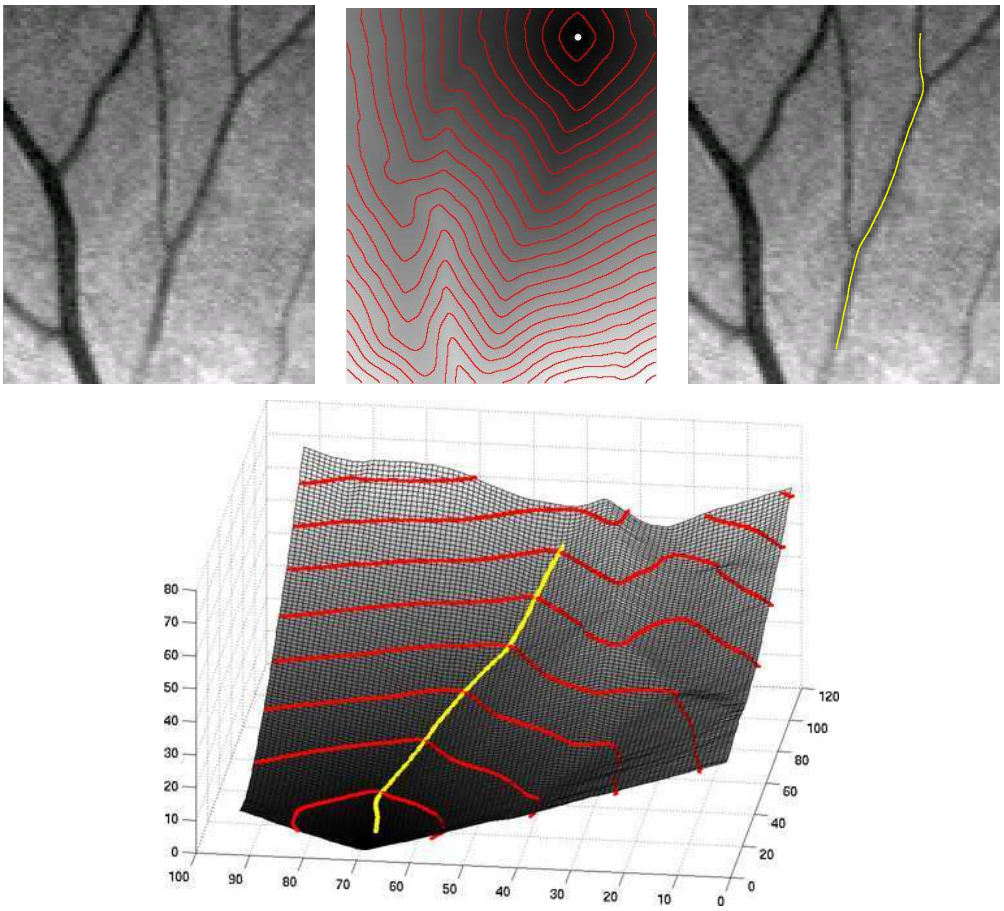


FIG. 3.3 – Vessel segmentation using shortest paths. *Top left :* original retinal image. *Top middle :* distance map computed from the white point (gray level was used as potential) and isodistance lines (red). Notice that the front propagates faster along the vessel. *Top right :* shortest paths computed from another point of the vessel. *Bottom :* synthesis on the distance function elevation map

III.1.3 Overview of our method

Our method goes one step further with respect to the method of [120]. It lifts the 2D image in a 4D radius and orientation space using local detectors of vessels at different orientations and scales. The use of 4D orientation space disambiguates crossing configurations [91], and also allows to perform more

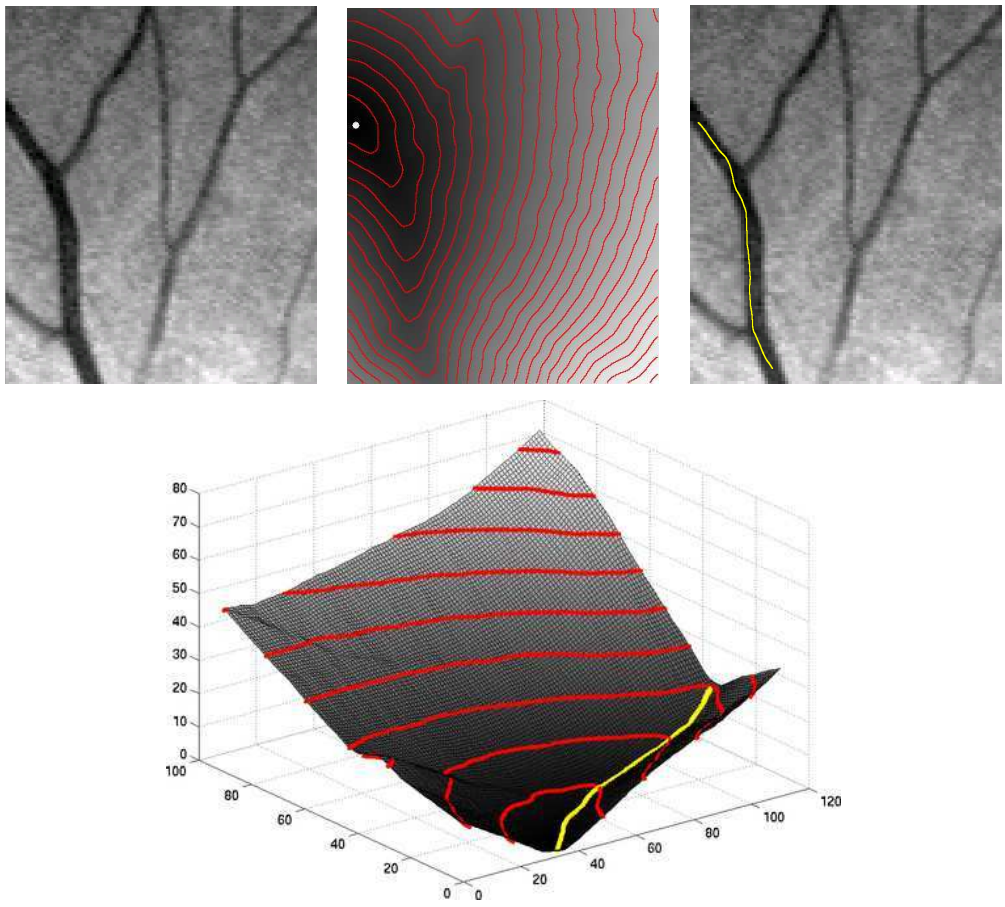


FIG. 3.4 – Vessel segmentation using shortest paths – while the path is included in the vessel, the centerline is not correctly evaluated. *Top left :* original retinal image. *Top middle :* distance map computed from the white point (gray level was used as potential) and isodistance lines (red). *Top right :* shortest paths computed from another point of the vessel. *Bottom :* synthesis on the distance function elevation map

stable and accurate segmentation.

Our method is independent from the local detector used, which can be tuned precisely to the targeted application. It then uses a geodesic based formalism to compute optimal paths in this 4D space, leading to a robust global segmentation of vessels as detailed in section III.2. Unlike methods which rely on a post-processing skeletization to compute the centerlines of the vessels, our method directly and naturally computes both centerlines and radii of

vessels.
We propose an application of this framework to the segmentation of vessels in cortical imaging movies, using the flow information to perform the detection of vessels (section III.3).
Finally, in section III.4, we propose to use this segmentation framework to design an algorithm for network extraction. Based on a tracking framework on extended neighborhoods, our algorithm handles difficult crossing configurations.

III.2 A framework for tubular structure segmentation

In this section, we present our new framework for the segmentation of tubular structures in a 4D radius and orientation space.

III.2.1 Local Vessel Model

An image will be treated as a 2D function $I : [0,1]^2 \to \mathbb{R}$. The local geometry of a vessel is captured with a vessel model $M(x) \in \mathbf{R}$ for $x = (x_1, x_2) \in \Lambda = [-\Lambda_1, \Lambda_1] \times [-\Lambda_2, \Lambda_2]$. This model is a 2D pattern that incorporates our prior knowledge about both the cross section of the vessels and the regularity of vessel.
The prior on the cross section of the vessel is included by considering models $M(x_1, x_2) = m(x_2)$ that only depends on a 1D profile m (figure 3.5). The prior on the regularity of the vessels corresponds to the ratio Λ_1/Λ_2 of the horizontal and vertical dimensions of the model.

Model cross-section for vessels. A 1D profile adapted to both cortical and optical imaging and retinal imaging is defined as

$$m(x_2) = \begin{cases} 1 & \text{for } |x_2| > \Lambda_2/2, \\ \exp(-\alpha\sqrt{(1/2)^2 - (x_2/\Lambda_2)^2}) & \text{otherwise.} \end{cases} \qquad \text{(III.2.1)}$$

This model encompasses medical knowledge about the light reflexion around blood vessels in cortical imaging. The image intensity inside a vessel is as-

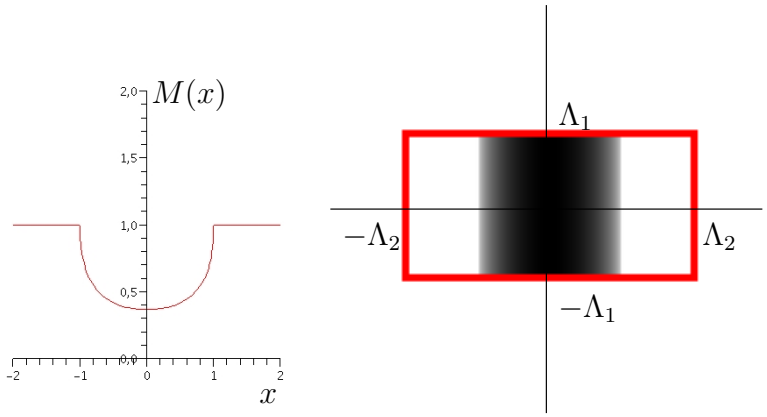


FIG. 3.5 – *Left :* intensity profile along a section. *Right* : a vessel model

sumed to result from a light absorption (with coefficient α) proportional to the vessel width at this point. It is also widely used in the retina image community [36].
The value $\alpha \approx 0.05$ was evaluated from a set of typical cortical images. However, section III.2.8.2 shows that our vessel extraction method is robust with respect to approximate choices of this absorption parameter.

Model for road extraction. A typical road in satellite imaging has a slow variation of intensity along a section. It is efficiently captured by a binary model defined as

$$m(x_2) \stackrel{\text{def.}}{=} \begin{cases} 0 & \text{for } |x_2| > \Lambda_2/2, \\ 1 & \text{otherwise.} \end{cases} \tag{III.2.2}$$

Regularity selection. The ratio Λ_1/Λ_2 of the model dimensions acts as a prior on the regularity of typical vessels. The more typical vessels are curved, the smaller Λ_1/Λ_2 should be. Also, robustness to noisy images forces to use a model with a large enough area $\Lambda_1 \times \Lambda_2$. The value of $(\Lambda_1, \Lambda_2) = (1, 2)$ is used in our numerical experiments. This choice is further discussed in the numerical experiments section.

To overcome the inherent difficulties of the 2D detection problem, additional scale and orientation dimensions are introduced to increase the detectability

of vessels.

III.2.2 Rotated and Scaled Models

The normalized pattern $M(x)$ is rotated and scaled to match the varying orientation and width of vessels. Beside the choice of the pattern cross section m and the dimension $\Lambda_1 \times \Lambda_2$ of the model $M(x)$, the scaling of $\Lambda(r)$ with r is another avenue to introduce some prior about vessels in the image. Small scales cortical and retinal vessels are less regular than large scale vessels. We thus chose to scale the dimensions of the model $\Lambda(r) = r\Lambda$ linearly with the radius r. This causes thin vessels to be detected using a finer correlation analysis.

The warped model $M_{r,\theta}(x)$ for $x \in \Lambda(r,\theta) = R_\theta \Lambda(r)$ is defined as

$$\forall x \in \Lambda(r,\theta), \quad M_{r,\theta}(x) \stackrel{\text{def.}}{=} M(R_{-\theta}(x/r)) \tag{III.2.3}$$

where R_θ is the planar rotation of angle θ.

Figure 3.6 shows examples of models defined with (III.2.1) and (III.2.2) that are rotated and scaled according to (III.2.3).

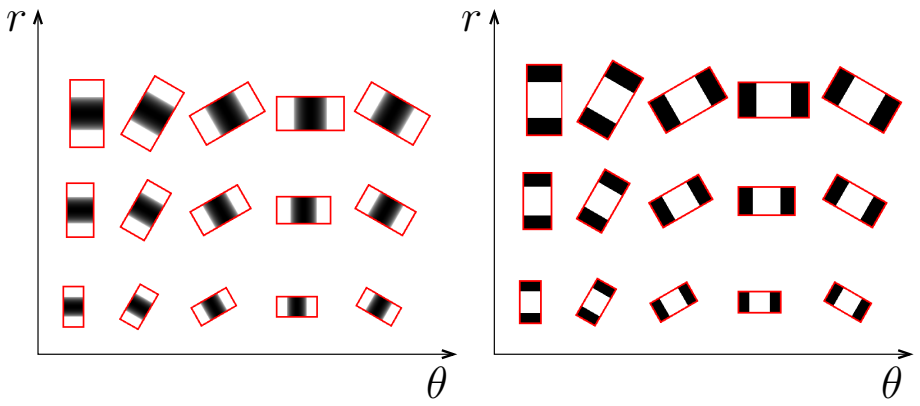


FIG. 3.6 – Vessel models (left) and roads models (right) for different orientations and scales. Here, $\Lambda_1/\Lambda_2 = 1/2$ and $m(.)$ is given by (III.2.1).

III.2.3 Scale/Orientation Lifting

The image I is lifted in a 4D space by adding a scale and an orientation dimension. Let Ω be defined by

$$\Omega \stackrel{\text{def.}}{=} [0,1]^2 \times [r_{\min}, r_{\max}] \times [0, \pi) \tag{III.2.4}$$

the last dimension being periodic. Ω is thus a 4-dimensional manifold.
We call *lifting* the function F computed as the normalized cross-correlation [73] between the image and the local model (III.2.3)

$$\forall \omega = (x, r, \theta) \in \Omega,\ F(\omega) \stackrel{\text{def.}}{=} \mathrm{NCC}_{\Lambda(r,\theta)}(M_{r,\theta}(\cdot), I(x + \cdot)) \tag{III.2.5}$$

where $I(x + \cdot)$ is the image translated by x, $\mathrm{NCC}_A(f, g)$ is the normalized cross-correlation between f and g over the domain A, defined by :

$$\mathrm{NCC}_A(f, g) \stackrel{\text{def.}}{=} \frac{\int_A (f - \bar{f})(g - \bar{g})}{\sqrt{\int_A (f - \bar{f})^2} \sqrt{\int_A (g - \bar{g})^2}} \tag{III.2.6}$$

where $\bar{h} = (\int_A h)/|A|$, $|A|$ being the area of A.
This lifting separates real 3D vessels that overlaps when projected at the same location by the imaging system but have different orientations.
r_{min} and r_{max} are respectively set as the minimum and maximum values of the vessels radius one wishes to detect in the image.
The value $F(x, r, \theta)$ ranges from -1 to 1 and measures the likelihood of observing a vessel at a given location x with a width r and an orientation θ. The normalization of the detector makes it invariant under to intensity variations that occurs in medical images due to the elevation variation of the vessels and the imperfection of the imaging system. Adding a scale dimension yields a robust and regularized estimation of the radius and the center of vessels.
Figure 3.7 shows an example of a cortical image where orientation lifting is crucial to distinguish locally between orientations.

Numerical computations. A medical image is acquired on a on discrete grid of $n \times n$ pixels. The 4D lifting is computed for n_r radii evenly spaced in $[r_{\min}, r_{\max}]$ and n_θ orientations evenly spaced in $[0, \pi)$, with $n_r = 12$ and

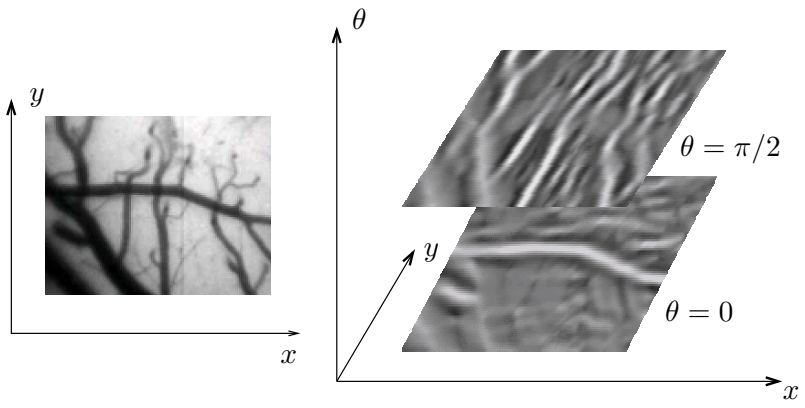


FIG. 3.7 – *left :* An original 2D image. *right :* Its 4D lifting (fixed radius), ranging from -1 (black) to 1 (white). White values indicate likely positions and orientations of vessels.

$n_\theta = 12$ in the experiments. This requires $O((r_{\max} n)^2 n^2 n_r n_\theta)$ operations with $r_{\max} \ll 1$ and $n_r, n_\theta \ll n$.

III.2.4 Lifted Potential

The 4D lifting (III.2.5) defines an isotropic potential ρ over the 4D domain Ω

$$\forall \omega \in \Omega, \quad \rho(\omega) \stackrel{\text{def.}}{=} \max(1 - F(\omega), \epsilon). \tag{III.2.7}$$

The parameter ϵ prevents the potential to vanish and is set to $\epsilon = 10^{-3}$ in the numerical tests.
This potential encodes local information about the presence of a vessel at a given position, scale and orientation.
Notice that this choice is somewhat arbitrary. Any non-increasing function of F could be considered.

III.2.5 Distance Map and Geodesic Computation

The length of a lifted curve $\gamma : [0,1] \to \Omega$ over the lifted domain is defined as

$$\mathcal{L}_F(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \rho(\gamma(t)) \|\gamma'(t)\| \mathrm{d}t. \tag{III.2.8}$$

where the length of the speed vector $v = \gamma'(t) = (v_x, v_r, v_\theta)$ is

$$\|v\|^2 \overset{\text{def.}}{=} v_x^2 + \lambda v_r^2 + \mu v_\theta^2, \tag{III.2.9}$$

(λ, μ) being normalizing constants that controls the penalty on scale and orientation variations along the vessels in the images. In practice, as we will demonstrate in the numerical experiments section, we observed strong robustness with respect to the choice of (λ, μ).

Given a set $\mathcal{A} \subset \Omega$ of seeds points and a set $\mathcal{B} \subset \Omega$ of ending points, a shortest lifted curve $\gamma^*(t) \subset \Omega$ joining $\mathcal{A}$ to $\mathcal{B}$ is defined as a shortest path for the metric $\mathcal{L}_F$

$$\gamma^*(\mathcal{A}, \mathcal{B}) \overset{\text{def.}}{=} \underset{\gamma \in \mathcal{C}(\mathcal{A}, \mathcal{B})}{\operatorname{argmin}} \mathcal{L}_F(\gamma), \tag{III.2.10}$$

where $\mathcal{C}(\mathcal{A}, \mathcal{B})$ is the set of curves γ such that $\gamma(0) \in \mathcal{B}$ and $\gamma(1) \in \mathcal{A}$. The corresponding geodesic distance is $d_F(\mathcal{A}, \mathcal{B}) = \mathcal{L}_F(\gamma^*)$. This definition can be specialized to a single starting point $\mathcal{A} = \{\omega_0\}$ and/or to single ending point $\mathcal{B} = \{\omega_1\}$ to define the geodesic distance between points and/or sets, *e.g.* $d_F(\omega_0, \omega_1) \overset{\text{def.}}{=} d_F(\{\omega_0\}, \{\omega_1\})$.

Therefore, we are in exactly in the framework of shortest paths on a Riemannian manifold introduced in section I.3.1.6.

The tensor associated with the potential is proportional to

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \mu \end{pmatrix}$$

i.e. its principal components are aligned with the canonical basis of the space. Furthermore, up to the periodicity of the θ dimension, we can assume that Ω is a cuboid of $\mathbb{R}^4$.

Ω is discretized as a grid of $N \overset{\text{def.}}{=} n^2 n_r n_\theta$, where extra links are set between points $(i, j, r, 0)$ and $(i, j, r, (n_\theta - 1)\frac{\pi}{n_\theta})$ and therefore we can apply the framework developed in section II.3.6 to compute shortest paths.

A first order Euler-scheme was used to perform the gradient descent – therefore computing γ^* with sub-pixel accuracy.

III.2.6 Shortest Paths and 4D curves

A 4D curve $c(x, x')$ between two points $x, x' \in [0,1]^2$ is computed as a 4D geodesic in Ω between the 4D lifted sets $\mathcal{A}(x)$ and $\mathcal{A}(x')$ defined as

$$\mathcal{A}(x^{(\prime)}) \stackrel{\text{def.}}{=} \left\{ (x^{(\prime)}, r, \theta) \setminus r \in [r_{\min}, r_{\max}], \theta \in [0; \pi) \right\}. \tag{III.2.11}$$

The 4D curve is then defined as

$$c_{x,x'} \stackrel{\text{def.}}{=} \gamma^*(\mathcal{A}(x), \mathcal{A}(x^{'})). \tag{III.2.12}$$

This 4D curve contains three components $c_{x,x'}(t) = (\tilde{x}(t), r(t), \theta(t))$. The path $\tilde{x}(t) \subset [0,1]^2$ is the actual centerline over the image plane, whereas $r(t)$ and $\theta(t)$ give the local width and orientation of the vessel, see Figure 3.8.

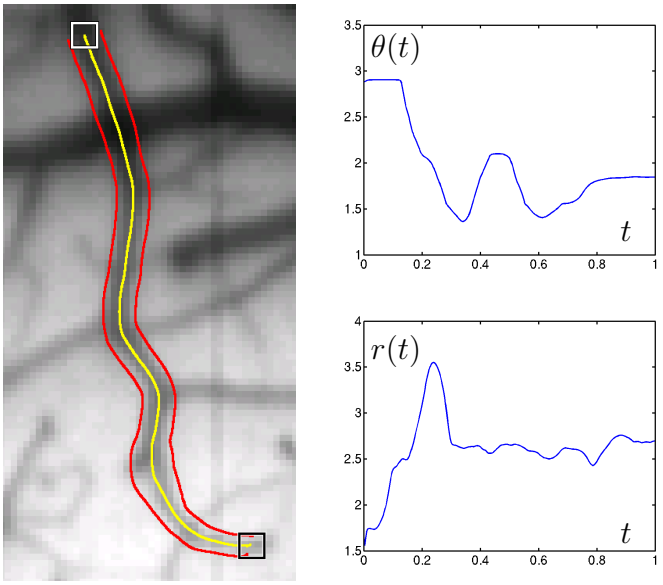


FIG. 3.8 – *Left :* centerline extraction of a vessel in a cortical image. Starting point : white square. Ending point : black square. *Right :* corresponding orientation $\theta(t)$ and radius $r(t)$.

III.2.7 Another interpretation

Due to the structure of the targeted images, we observed that the direction of a shortest path projected in the image plane is approximately equal to the

current angular position θ of a curve. Up to a renormalization, v_θ is then the curvature of the projection of γ in the image plane.
The length of a lifted curve is written

$$\mathcal{L}_F(\gamma) = \int_0^1 \rho(\gamma(t))\sqrt{v_x^2 + \lambda v_r^2 + \mu v_\theta^2}\mathrm{d}t \qquad \text{(III.2.13)}$$

and then its minimization leads to curves with both small length and curvature – which is somewhat similar to an optimization in a Sobolev space.

III.2.8 Evaluation of the Geodesic Centerlines

In this section, we present some results obtained by our method.

III.2.8.1 Accuracy and robustness to noise and parameters choice

The accuracy of the centerline extraction is compared on synthetic data to the two other methods mentioned in the introduction :

- the method of [39, 38], in which a 2D metric is computed from an image intensity blurred with a Gaussian filter. The filtering helps to re-center the geodesic since the smoothed image exhibits a local maxima around the center of the vessels, at the cost of a loss of spatial resolution,
- the method of [120], in which a 3D (space+scale) metric is computed. All the parameters of this 3D model are optimized to give the best results.

In the experiments, our method was used with an absorption parameter $\alpha = 0.1$, which is not optimized to fit the α of all benchmark images.

The precisions of the three algorithms are tested on several phantoms images. This phantoms images are build from five centerlines and radii analytical forms – thus with sub-pixelic accuracy. The cross section corresponds to the model (III.2.1) with parameter $\alpha = 0.01$, $\alpha = 0.1$ and $\alpha = 1$. An additive Gaussian white noise with various amplitudes are added to the phantoms. Ten phantoms are generated for each condition, and each noise level. This leads to a total database of about 3000 images. Figure 3.9 shows some of the obtained phantoms.
For each experiment, the true starting and ending points of each phantom are used, as well as the true starting and ending radii for the [120] method.

Default parameters $\Lambda_1/\Lambda_2 = 0.5$, $n_\theta = 12$, $n_r = 12$, $\mu = 1\frac{\pi}{n_{theta}}$ and $\lambda = 1\frac{r_{max}-r_{min}}{n_r}$ were used for our algorithm. Its sensibility with respect to this choice will be discussed.

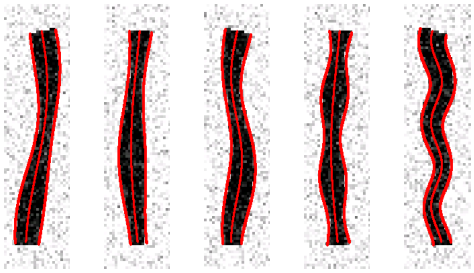


FIG. 3.9 – Some of the phantoms used in our benchmark (basic intensities range from 0 to 1), shown here with a spatially independent Gaussian noise of variance 0.15.

The extracted 4D curve $c(t) = (\tilde{x}(t), r(t), \theta(t))$ is compared to the ground trust c^* using the following errors :

$$\begin{cases} \text{Error}_C(c)^2 = \int_0^1 \|\tilde{x}(t) - \tilde{x}^*(t^*)\|^2 \mathrm{d}t \\ \text{Error}_R(c)^2 = \int_0^1 |r(t) - r^*(t^*)|^2 \, \mathrm{d}t \end{cases} \tag{III.2.14}$$

where t^* is such that $\tilde{x}^*(t)$ is the ground truth centerline point closest to $\tilde{x}(t)$, and where r is the radius computed by the method (proposed method and [120] only), and r^* the ground truth radius.
Figures 3.10,3.11,3.12,3.13 and 3.14 shows $\text{Error}_C(c)$ and $\text{Error}_R(c)$ curves for several synthetic images as a function of the noise level.
Using the 3D space+scale lifting [120] produces results of varying quality, and requires a careful tuning of the parameters to achieve the optimal error rate. [39, 38] with an optimal smoothing generally provides a precise evaluation of the centerline locations, but without any evaluation of the local radius. Notice also that the smoothing parameter achieving the best result varies from one phantom to another. Our method provides both positions and radii with more robustness and accuracy – and outperforms other existing methods, even when the model is not precisely tuned.
Furthemore, figure 3.15 shows an experiment where start and end points have been shifted two pixels to the right. Due to the slow variation of intensity

along the section of the benchmark, the 2D method is sensitive to this shift, while the 4D method recenters the paths, and does not suffer harshly from the shift.

Robustness to $\frac{\Lambda_1}{\Lambda_2}$ Figure 3.16 shows the influence of the choice of $\frac{\Lambda_1}{\Lambda_2}$ on the results. $\frac{\Lambda_1}{\Lambda_2} = 0$ corresponds to a model reduced to a segment. The choice of low values of $\frac{\Lambda_1}{\Lambda_2}$ allows to evaluate radius with a good accuracy when noise level is low, but leads to some instability. On the opposite, a too important value choice leads to precision lost. $\frac{\Lambda_1}{\Lambda_2} = 0.5$ is a good compromise.

Robustness to discretization Figure 3.17 demonstrates the effect of the choice n_θ – *i.e.* the number of angles used in the discretization. It appears that for low n_θ, the quality of the segmentation depends on whether or not the angle of the vessel is approximately present in the discretization of $[0, \pi)$. We thus performed quality tests for 31 rotations of the benchmarks (of angles $\{i/10\}_{i=1..31}$). Low values of n_θ lead to a higher variability in the segmentation, depending on whether or not the vessel direction is aligned with one of the discretized θ value. Overall the algorithm is quite robust to the choice of this quantity.
A similar experiment was run for n_r (figure 3.18). Although the centerline detection is robust to choices of small n_r, the radius evaluation is extremely sensitive to it.

Robustness to speed parameters We performed experiments to asses the dependence of the algorithm with respect to the choice of the angular speed μ (figure 3.19) and radius speed λ (figure 3.20). Centerline segmentation shows little sensitivity with respect to the choice of speed parameter in angular direction – radius estimation is slightly affected is the speed is to low. The choice of speed parameter in radius direction seems to be more important : a too important value will lead to good results for radius estimation when noise level is low, but will show a more unstable behavior when the noise increases. $\lambda = 0.5\frac{r_{max}-r_{min}}{n_r}$ seems to be a good choice.

In all the subsequent sections, the results were obtained with parameters $\Lambda_1/\Lambda_2 = 0.5$, $n_\theta = 12$, $n_r = 12$ ($r_{min} = 1$ and $r_{max} = 6.5$), $\lambda = 0.5\frac{r_{max}-r_{min}}{n_r}$

and $\mu = 0.1\frac{\pi}{n_\theta}$. The choice of a low μ is motivated by the necessity of having a space Ω "wide enough" in angular direction in order to disambiguate crossing configurations.

III.2.8.2 Evaluation on Synthetic Crossings

The 4D lifting (III.2.5) is challenged by testing the extraction of a curved vessel with a self crossing. Figure 3.21 shows that the vessel curve is not correctly extracted with a metric that does not take into account the local orientation. A 2D purely spatial metric or a 3D space+scale metric extracts a curve that does not capture the correct topology of the vessel. Our 4D centerline position+scale+orientation favors the extraction of a longer curve that is both well centered and geometrically faithful to the true vessel.

III.2.8.3 Evaluation on Medical/Satellite Images

Figure 3.22, left, shows vessels extraction for a complex optical imaging of the cortex with several branches and intersections. The centerlines computed from different ending points are overlapping.
Figure 3.22, right, shows vessels extraction on a retinal image from the *DRIVE* database [198, 143]. The starting point is shown with a white square and several end points are shown with black squares. The crossings in this retinal image show the interest of the 4D lifting, that allows to correctly detect the geometry of the vessels.
Figure 3.23 shows a similar experiment in a satellite image.
In figure 3.24, two initial seeds were provided, on the roads going down and to the right from the crossing. Shortest paths were then computed from the two others segments of roads. The crossing is handled correctly, which can not be done by the other methods which do not use an orientation lifting.
Figure 3.8 shows the estimated radius $r(t)$ and orientation $\theta(t)$ for a vessel extracted in a cortical image. Both the centerline position, the radius and the orientation are computed with sub-pixel accuracy.
The precision of our 4D lifting method is evaluated on the DRIVE database [198, 143]. Approximate ground truth centerlines positions and radii are computed from the binary masks available with the database. Figure

3.25, shows the binary segmented vessels together with the ground trust centerline position and boundaries (top and middle), as well as the result of our segmentation algorithm (bottom).

The three geodesic extraction algorithms are applied to these three images between the indicated starting and ending points. Table III.1 report the centerlines position and radii errors $\text{Error}_C(\gamma)$ and $\text{Error}_R(\gamma)$ for each method. For the centerline extraction, due to the lack of precision of the ground truth, there is no significant difference between the proposed 4D lifting method and the space only geodesic extraction with smoothing of the metric. The 3D space+scale lifting [120] method showed unstable behavior with respect to its initialization and parameters, which had to be chosen carefully - for the second image, we did not manage to find parameters giving a correct result. Our 4D lifting method is also more precise for the radii estimation than the 3D lifting.

	DRIVE 1		DRIVE 2		DRIVE 3	
	Error_C	Error_R	Error_C	Error_R	Error_C	Error_R
2D metric	0.40	-	0.38	-	0.30	-
3D metric	1.33	1.67	3.13	3.31	0.53	1.90
4D metric	0.31	0.43	0.35	0.44	0.40	0.47

TAB. III.1 – Centerlines positions Error_C and radii Error_R estimation errors on retina images for the three different methods.

III.2.9 Conclusion and Discussion

We proposed a reliable algorithm to segment tubular structure in bi-dimensional images, between user provided points. Experiments on real and synthetic data show the accuracy and the robustness of the proposed methods.

Furthermore, as it is widely independant of the local detector we used, it is virtually applicable to other modalities. As an example, it could be interesting to consider less naive roads detectors than the simple one that was used in our work.

Questions remains about the choice of the metric and of the potential function. Although we did not notice high sensibility with respect to the choice of any reasonnable potential in our experiments, it would be interesting to understant how design it in order to reach an optimal segmentation for the targeted application. Metric choice could also be used for example to favor faster rotations for smaller vessels. Tuning it would require a careful statistical analysis of a database of manually segmented tubular structures.
Theoretically, this framework could be extended to tri-dimensional images, but the lifting would lead to a 6 or 8 dimensional space – depending if the section of a vessel is modelised by a circle (1 parameter) or an ellipse (3 parameters) – which is likely to be computationally untractable. Moreover, as there is no need of orientation disambiguation in tri-dimensional images, it is not clear that this algorithm would lead to improvments with respect to existing methods. However, as explained in IV.2, several tracks may be followed in order to reduce computational complexity in that case, for example computations on a partial volume or approximation of the update step.

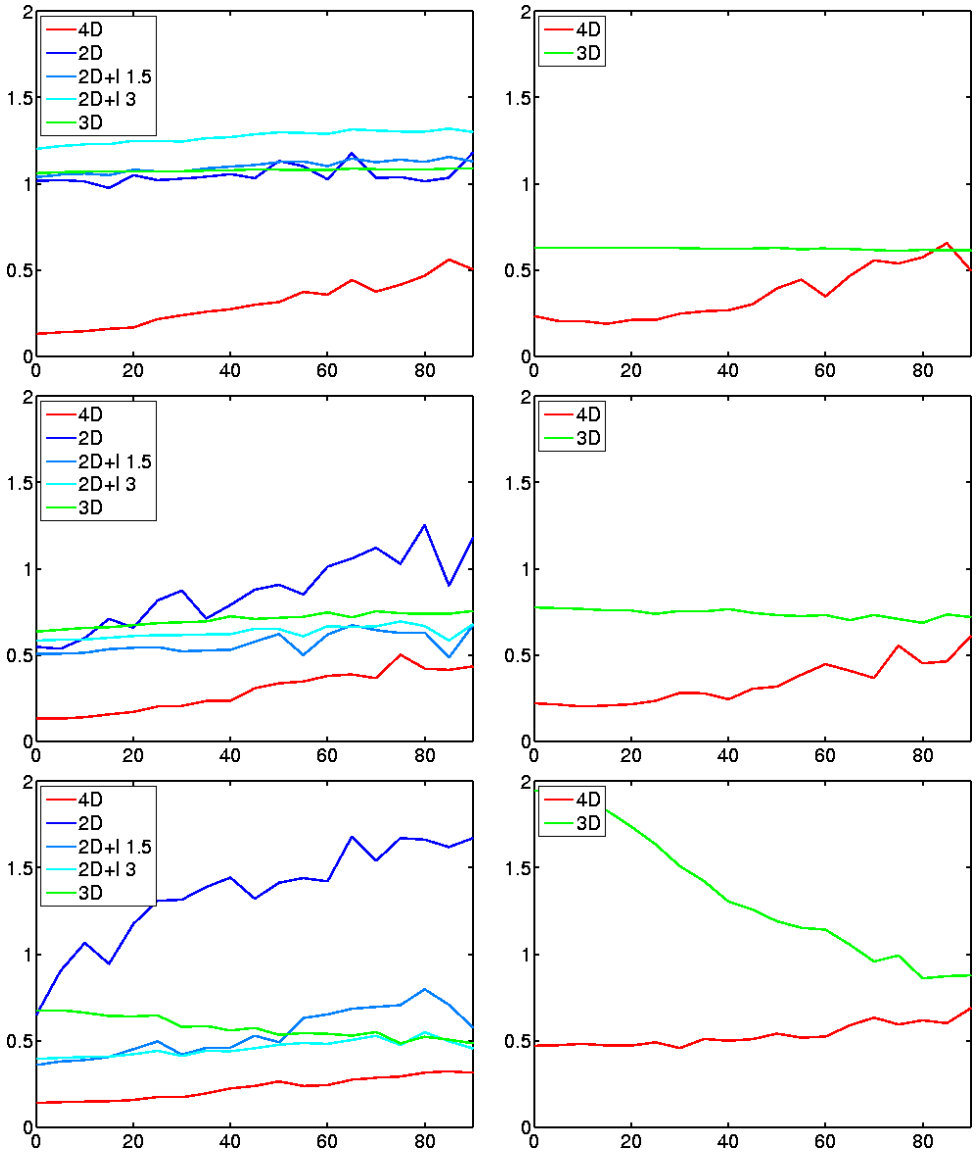


FIG. 3.10 – Centerlines positions and radii estimation errors for the phantom 1 of Figure 3.9.

Top, middle and bottom row respectively show results for phantoms generated with parameters $\alpha = 0.01$, $\alpha = 0.1$ and $\alpha = 1$.

Left column : error $\text{Error}_C(\gamma)$ (in pixel) for the three methods, as a function of the noise level (100σ where σ is the independent Gaussian noise variance). (*red* : 4D metric (space+scale+orientation), *green* : 3D metric (space+scale) [120], *blue* : 2D metric with different pre-smoothing [39, 38])

Right Column : Radii error $\text{Error}_R(\gamma)$ (in pixel) for the 3D and 4D methods, as a function of the noise level.

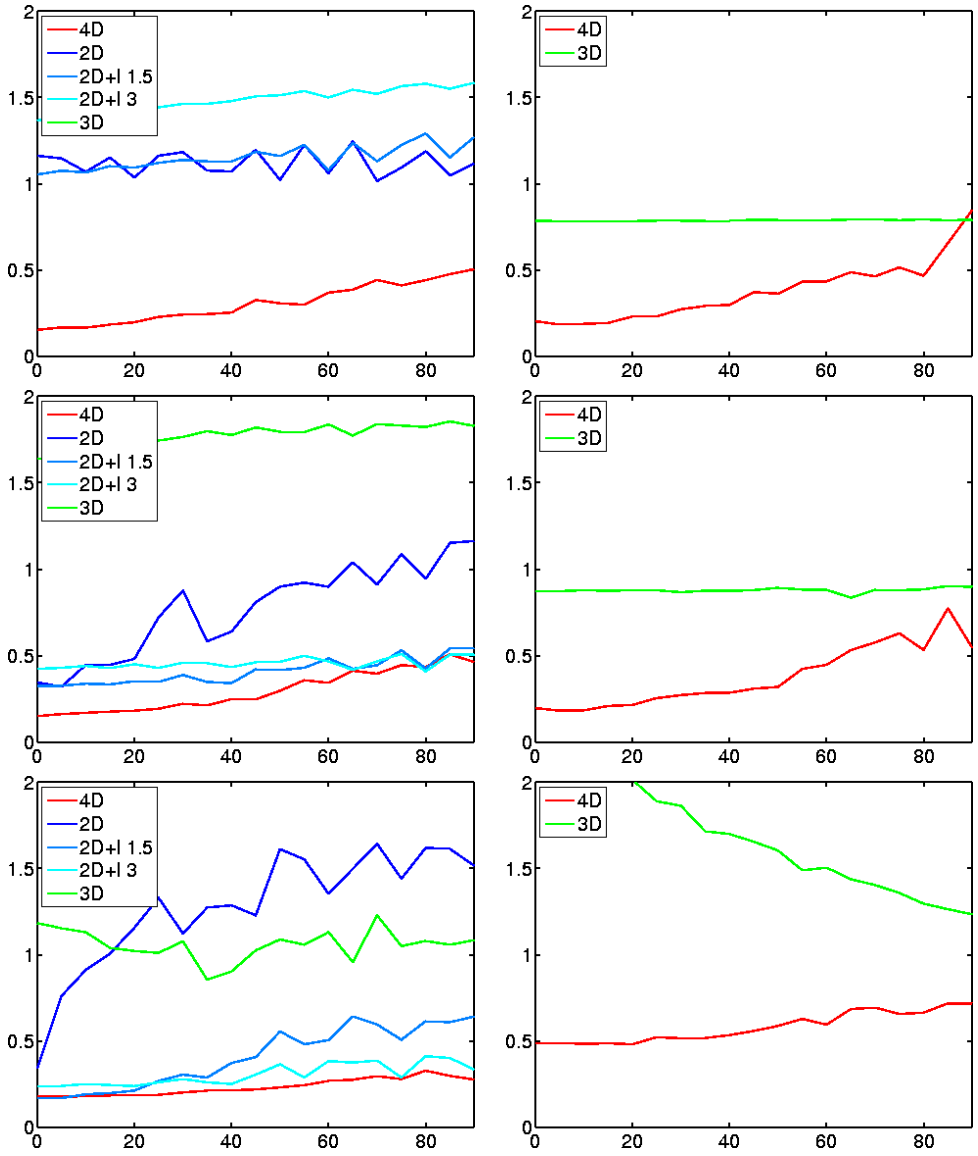


FIG. 3.11 – Centerlines positions and radii estimation errors for the phantom 2 of Figure 3.9. See 3.10 for legend.

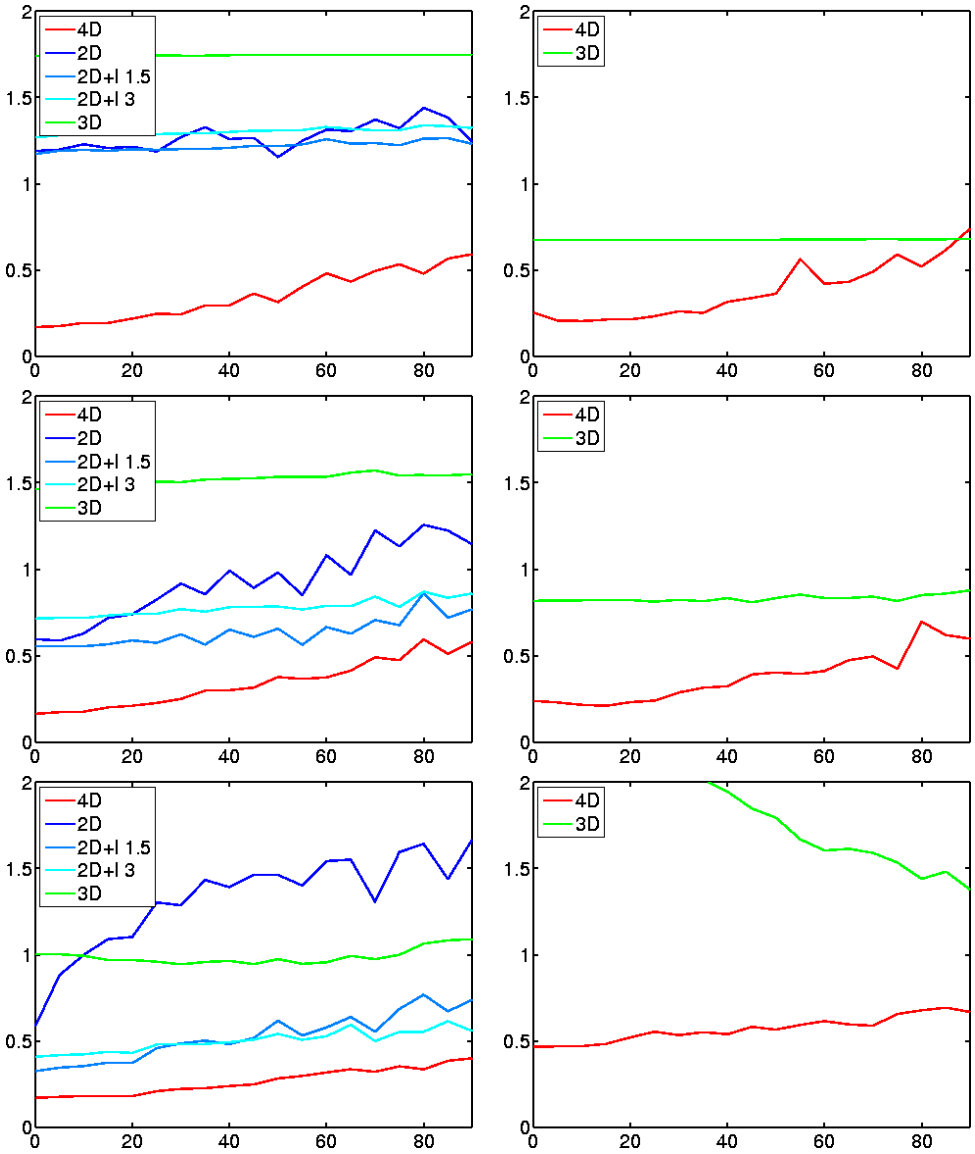


FIG. 3.12 – Centerlines positions and radii estimation errors for the phantom 3 of Figure 3.9.See 3.10 for legend.

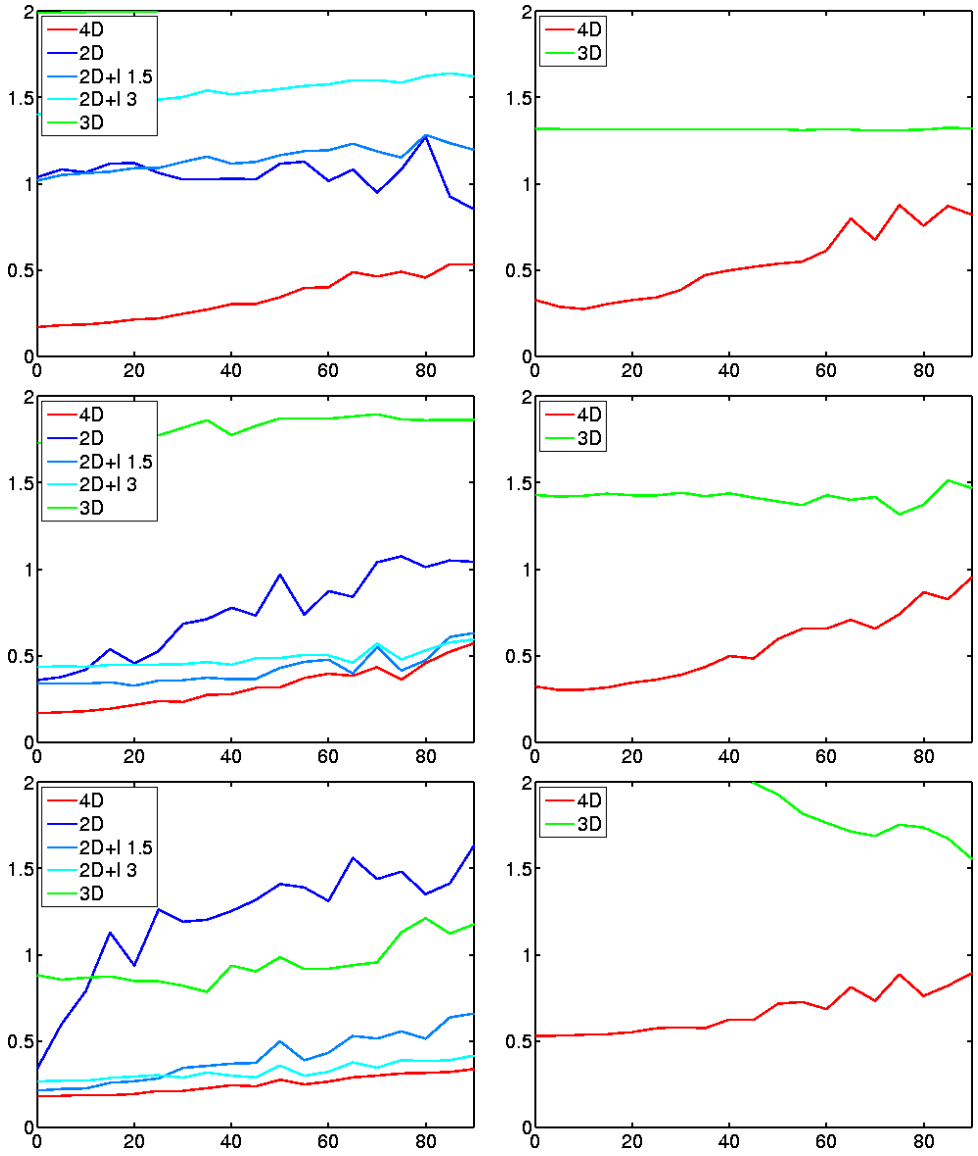


FIG. 3.13 – Centerlines positions and radii estimation errors for the phantom 4 of Figure 3.9.See 3.10 for legend.

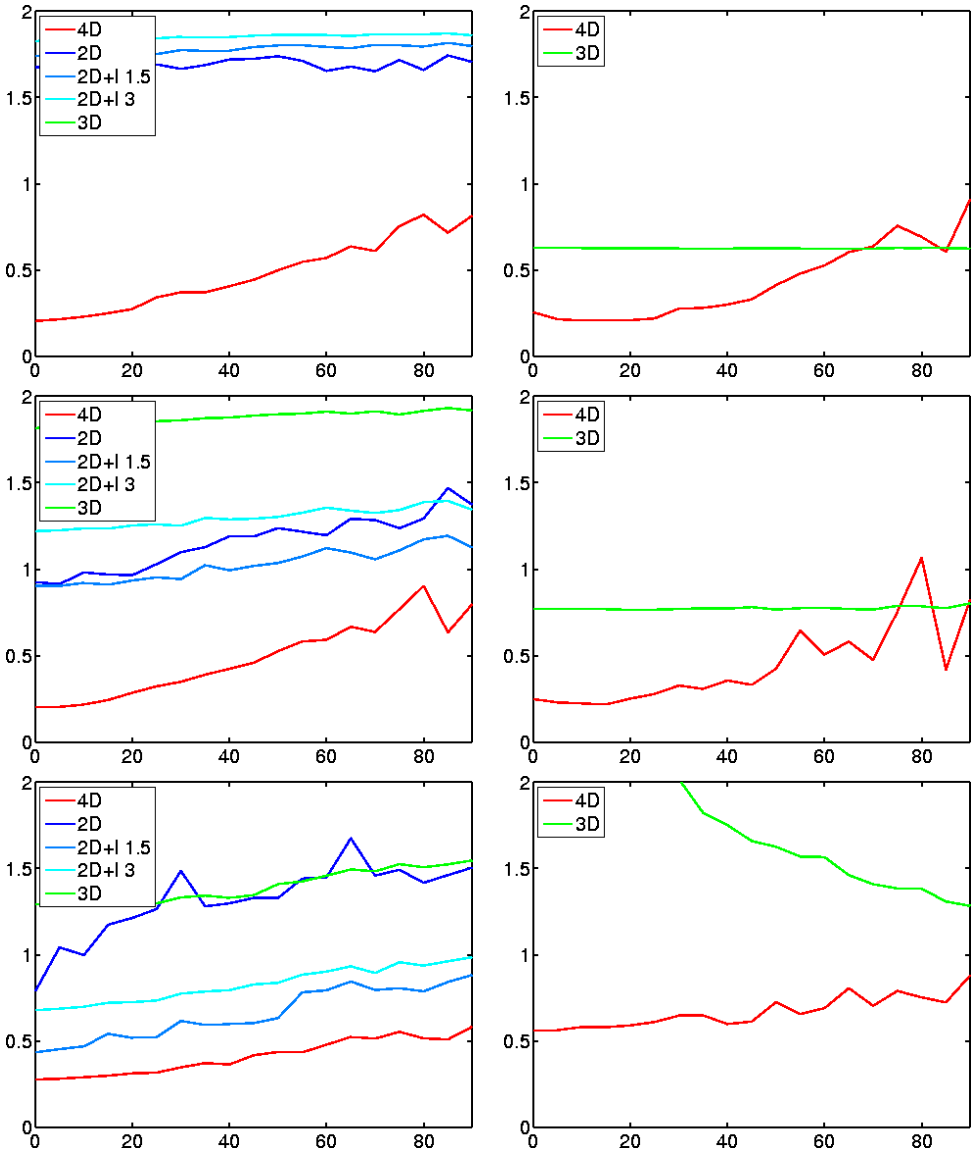


FIG. 3.14 – Centerlines positions and radii estimation errors for the phantom 5 of Figure 3.9.See 3.10 for legend.

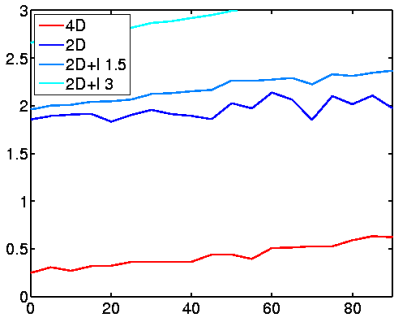


FIG. 3.15 – Centerlines positions and radii estimation errors for the phantom 2 of Figure 3.9 with $\alpha = 0.01$, with start and end points shifted two pixels on the right. Compare results with 3.11 (top left)

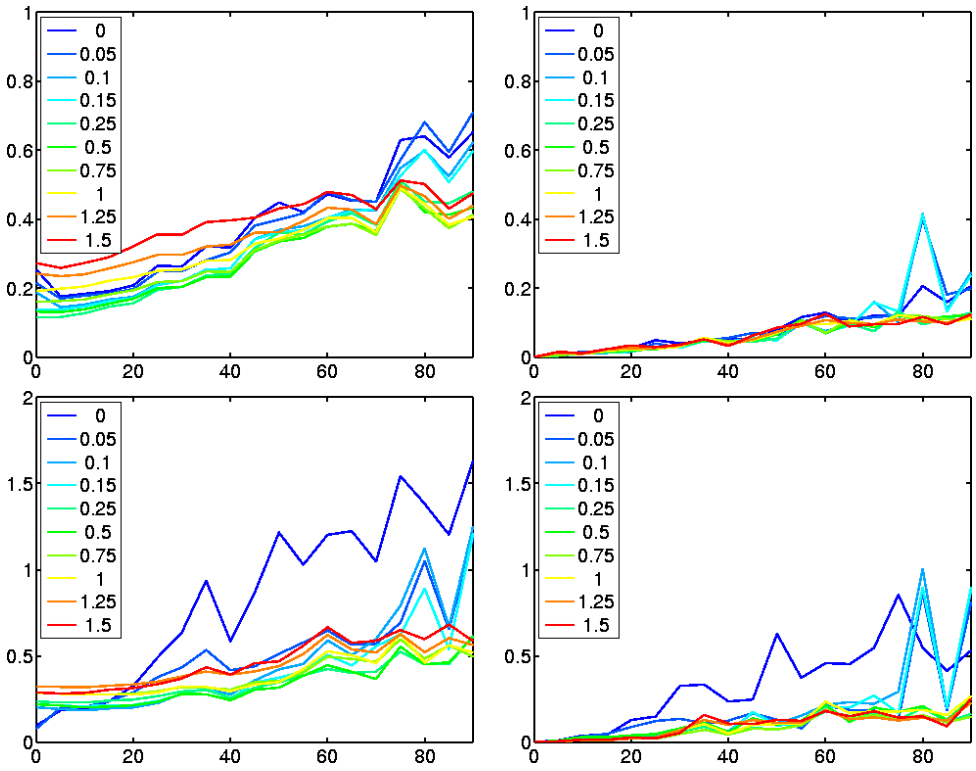


FIG. 3.16 – Influence of the choice of $\frac{\Lambda_1}{\Lambda_2}$: experiments for benchmark 1. Different curves are for several values of $\frac{\Lambda_1}{\Lambda_2}$ (dark blue → cyan → red). *Top Left* : mean centerline errors. *Bottom Left* : standard deviation of centerline errors. *Top Right* : mean radius errors. *Bottom Right* : standard deviation of radius errors.

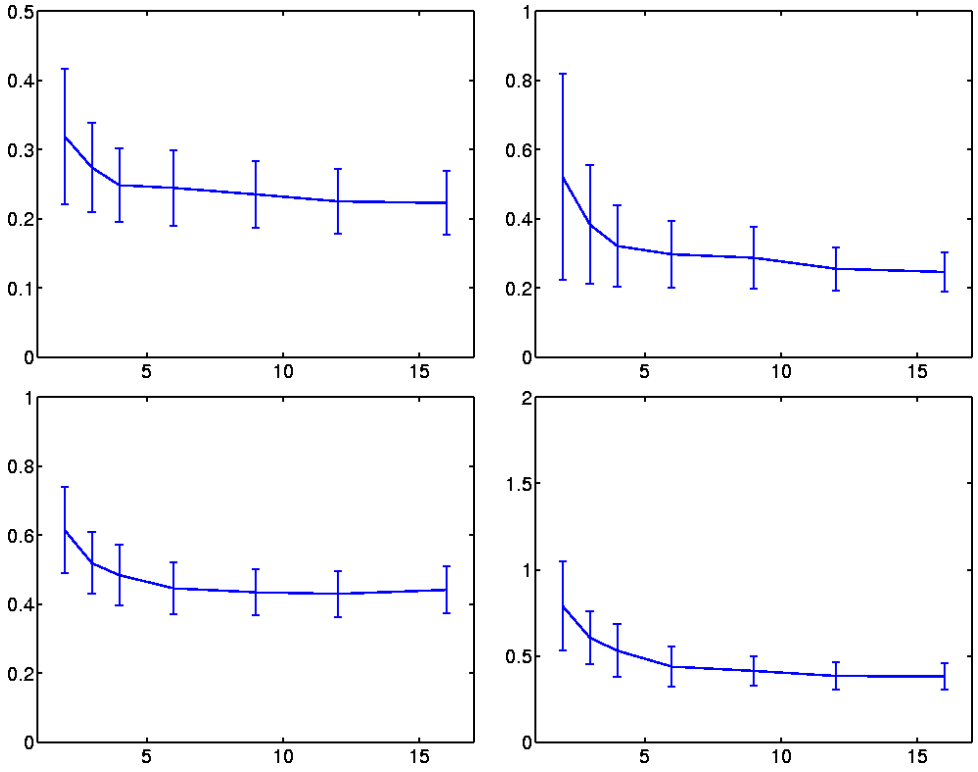


FIG. 3.17 – Influence of the choice of discretization step in angular direction. Results for phantoms 1 (Top row) and 5 (Bottom row) with noise 0.4. *x-axis :* number of angles of the discretization. *Left* : centerline errors, bars represent standard deviation. *Right* : radius errors, bars represent standard deviation.

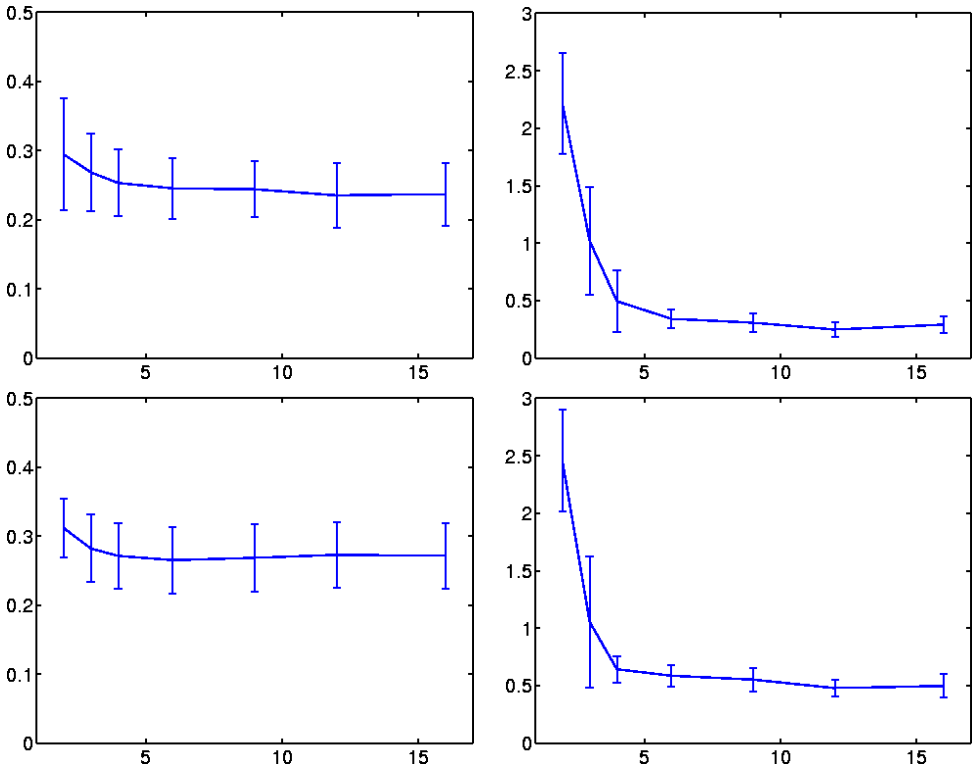


FIG. 3.18 – Influence of the choice of discretization step in radius direction. Results for benchmark 1 (Top row) and 4 (Bottom row) with noise 0.4. *x-axis :* number of radius of the discretization. *Left* : centerline errors, bars represent standard deviation. *Right* : radius errors, bars represent standard deviation.

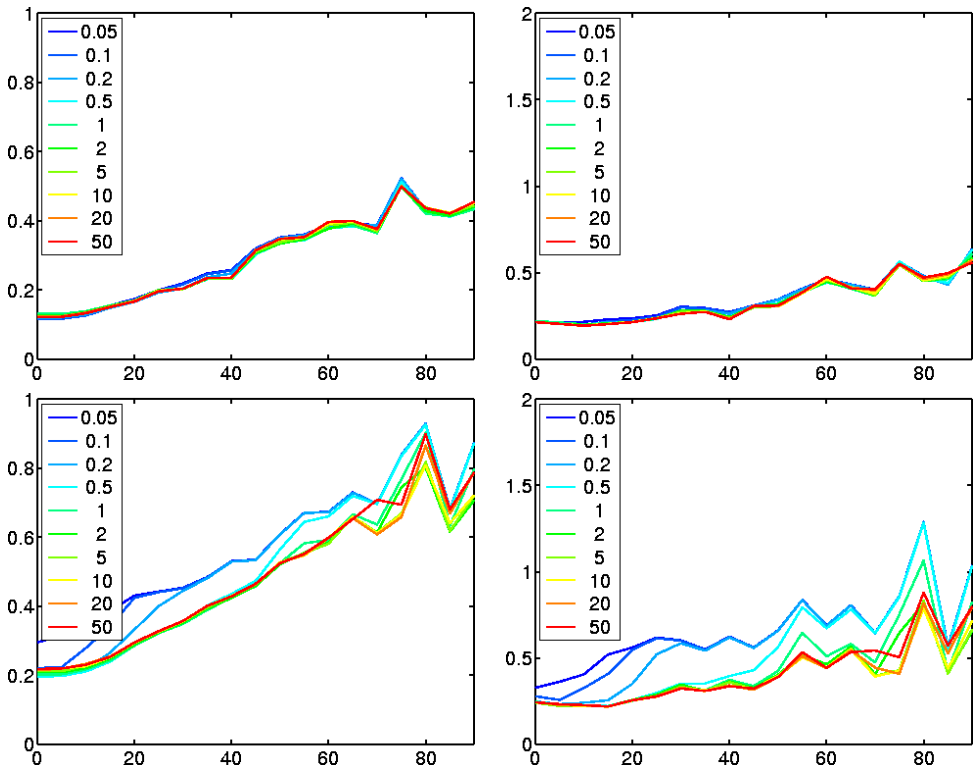


FIG. 3.19 – Influence of the choice of speed in angular direction. Results for benchmark 1 (Top row) and 5 (Bottom row) as a function of noise level. *Left* : centerline errors. *Right* : radius errors.

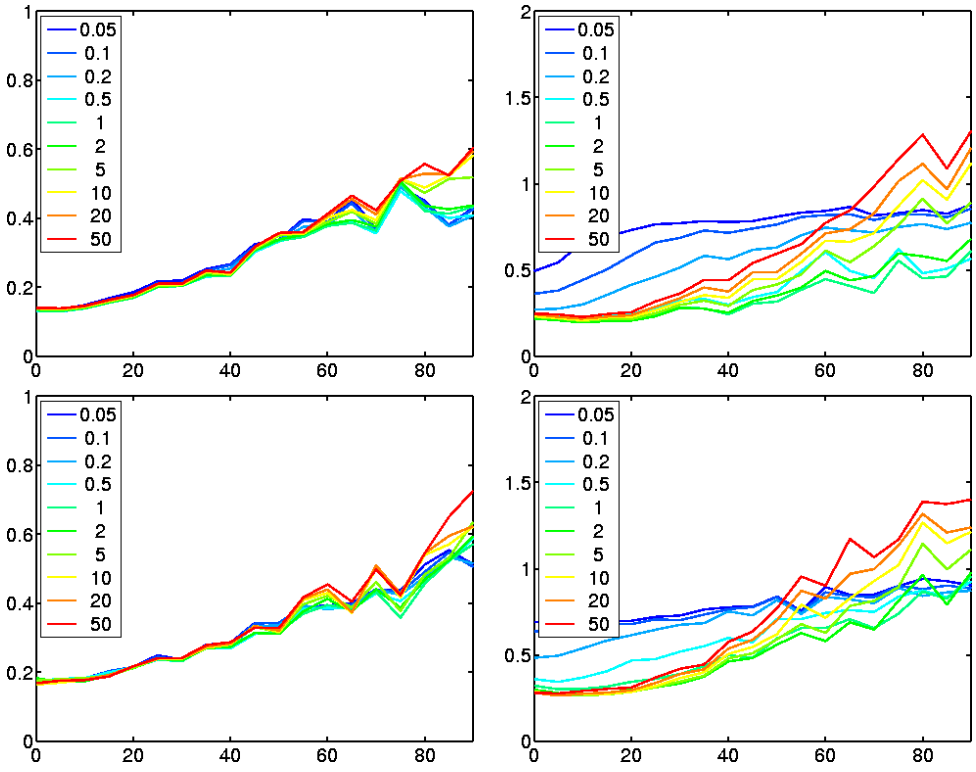
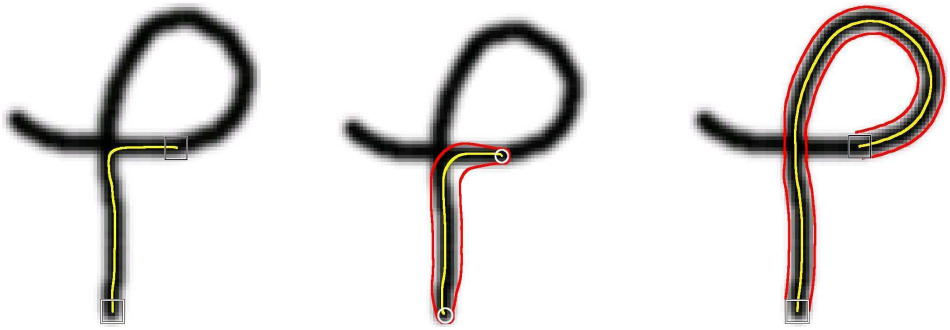


FIG. 3.20 – Influence of the choice of speed in radius direction. Results for benchmark 1 (Top row) and 4 (Bottom row) as a function if noise level. *Left* : centerline errors. *Right* : radius errors.



2D metric [39, 38] (position)

3D metric [120] (position+scale)

our method : 4D metric (position+scale+orientation)

FIG. 3.21 – Comparison of the 2D [39, 38], 3D [120] and 4D lifting (our method) when encountering a self-crossing.

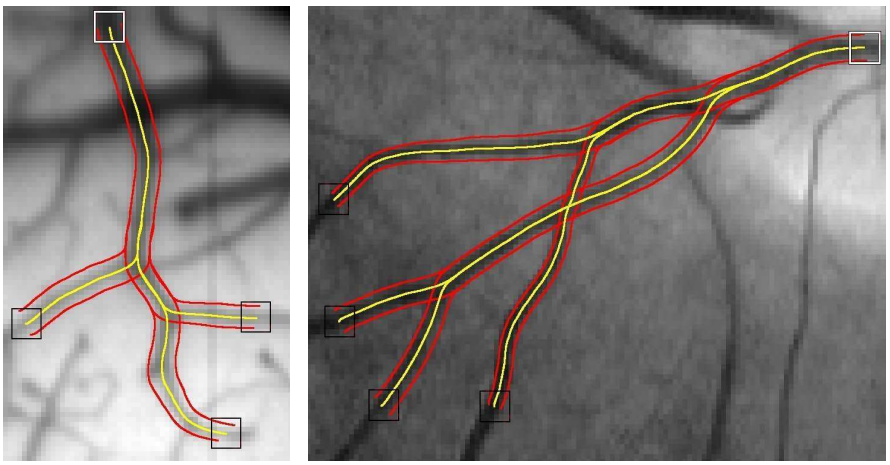


FIG. 3.22 – Centerlines positions and radii extraction of vessels in a cortical image (left), and in a retinal image (right).

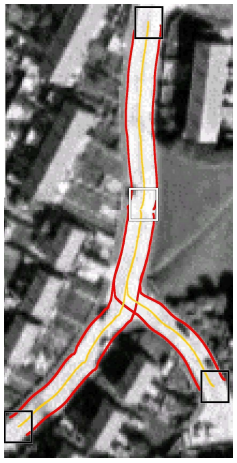


FIG. 3.23 – Centerlines positions and radii extraction of roads in a satellite image.

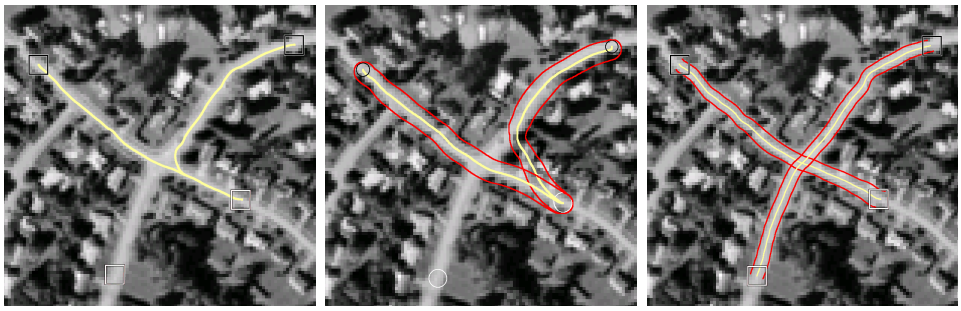


FIG. 3.24 – Centerlines and radii extraction of roads for the three tested methods. Two starting points (white squares/circles) and two ending points (black squares/circles) were provided for each method. From top to bottom : 2D, 3D, 4D methods.

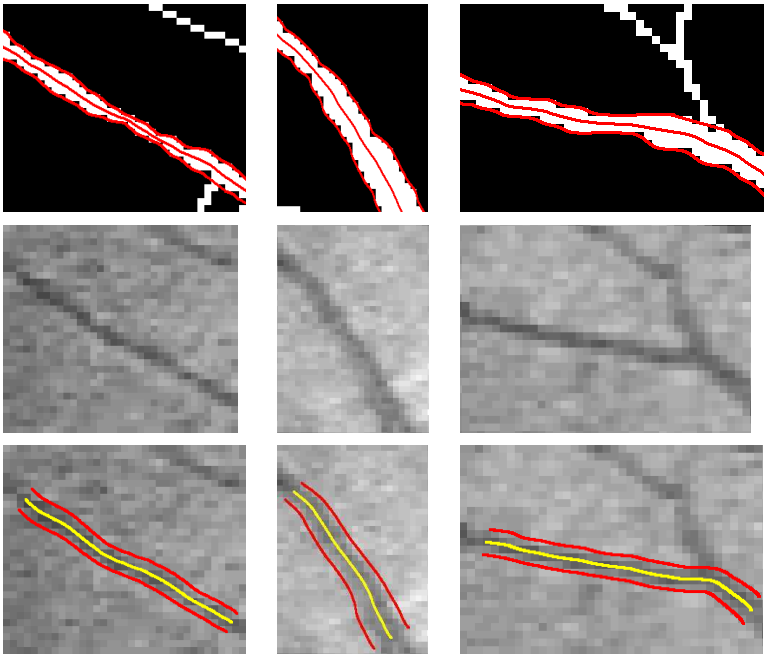


FIG. 3.25 – Binary segmented images from the DRIVE database, together with the extracted ground trust. *middle* : corresponding images with ground-truth centerline and boundary and initial and ending points. *bottom* : centerlines positions and boundaries computed with our method.

III.3 Application to flow-based extraction

In this section, we present an application of our shortest paths vessels segmentation algorithm to the analysis of *cortical optical imaging*.

III.3.1 Introduction

Magnetic Resonance Imaging (MRI) is a widely used medical imaging modality, discovered in the 1970's [11, 129]. It allows the tri-dimensional imaging of several tissues with good contrast and high spatial definition, while being non-invasive. Its basic principle is to put a subject in a high intensity magnetic field, therefore aligning the protons in water molecules with the field. A second field is then applied briefly, changing the alignment of protons. When relaxing to the alignment induced by the first magnetic field, the protons emit a signal in radio frequency, which can be detected. The use of non-constant magnetic fields allows one to locate the spatial position from which the signal was emitted. Furthermore, the intensity of this signal is related to properties of the tissues from which it originates. This leads to a tri-dimensional image of an organ, usually discretized in voxels whose resolution can be under $1mm^3$ (figure 3.26).

FIG. 3.26 – A sagittal slice of my head, acquired with MRI.

Functional Magnetic Resonance Imaging (fMRI) [109, 146] is a variant of MRI for the imaging of neural activity. When neurons in an area of brain activate, one can observe a subsequent increase in blood flow in the area, aiming at providing glucose and oxygen to the neurons. This phenomenon is called *Hemodynamic Response*. fMRI is able to detect this response, through

the detection of changes in oxyhemoglobin concentration, and can thus issue an activation map of brain. Due to its high spatial precision, fMRI has become in a few years one of the most widely used technique for functional imaging of the brain.
However, fMRI does not directly measure the neural electric activity, but the hemodynamic response. There is therefore a strong need for relating hemodynamic response to neural activity [32, 48].
Moving red blood cells (RBCs) can be directly "seen" by optical imaging of the cortex at adequate wavelengths [26], allowing to quantify blood flow in vascular networks [75]. However, to achieve a robust, fast and reliable determination of the small, eventually activity-evoked changes in cerebral blood flow (CBF), some obstacles still have to be overcome [211].
In particular, vessels segmentation is a highly time-consuming task if relying on user input, but is a challenge for standard automatic methods due to the weakness of contrast of small vessels and ambiguities posed by crossing and branching points.
Here, we present a new algorithmic approach based on our shortest path algorithm, allowing to segment vessels by using flow information rather than anatomical information.

III.3.2 Pre-processing

III.3.2.1 Sequence Registration

Images were acquired at 200Hz with a CCD, upon illumination at 570nm, from the primary visual cortex of an awake macaque who had a 1cm^2 transparent cranial window chronically implanted above the area of interest. Even though, during the experiment, the monkey's head is thoroughly stabilized, the curvature of the cortical surface, its position with respect to the camera and the exact morphology of the vasculature change slightly under the effects of the heart-beat and the monkey's body movements. These movements can be as large as a few pixels, and can be relatively fast (until hundred of Hz). An inter-frame spatial matching step is therefore required to be able to further process each image-sequence. [225] offers a recent survey of several image such image registration methods.

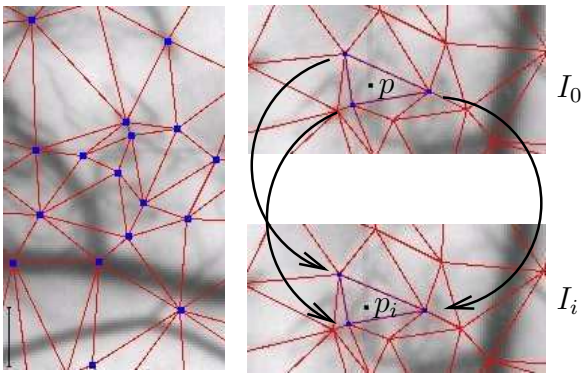


FIG. 3.27 – *Left : Blue :* SIFT points on a part of the first frame of the sequence. *Red :* corresponding Delaunay triangulation. *Right :* Registration between two frames : a point belonging to a triangle in the first image is registered to the point with same barycentric coordinates in the corresponding triangle of the second image.Scale-bar is $500\mu m$ in all figures

We used a ad-hoc features-based method for registering a complete sequence, based on Scale-Invariant Feature Transform (SIFT) descriptors [122]. SIFT is a state-of-the-art fast and robust algorithm for extracting and characterizing salient features from an image which can deal with several computer vision problems. For each image, the SIFT algorithm yields a number (controlled by a threshold) of 2D points p with sub-pixel precision, along with a descriptor vector v_p in $\mathbb{R}^{128}$ for each point p, which represents the image around the detected point. The main feature of the SIFT detector is that the points and descriptors obtained are invariant with respect to scale, rotation, and illumination changes.
Our method can be described by the following steps :

1. *Features Detection :* the SIFT algorithm is applied to each image of the sequence, to detect characteristic points along with their descriptors (figure 3.27) after images have been smoothed with a narrow Gaussian filter ($\sim$ 2 pixels) to remove high spatial frequency components.
2. *Features Matching :* we use one frame (usually the first, 0) as a reference and match its SIFT keypoints to those of other frames. Using some threshold δ, we keep from the set of these keypoints only those p_0

which match with one and only one keypoint p_i ($||v_{p_0}, v_{p_i}||_2 < \delta$) in every other frame i. Notice that no spatial information is used during this step : only the points' descriptors are used during the matching process, not their positions. This potentially allows for large movements between frames.

3. *Full Image Matching :* the third step is intended to extend the matching of the characteristic points to the whole space. For this purpose, we first apply Delaunay triangulation [25] to form a mesh M_0 which vertices are the SIFT points of the reference frame (figure 3.27). Then each triangle (p_{0a}, p_{0b}, p_{0c}) of this mesh is matched to its counterpart (p_{ia}, p_{ib}, p_{ic}) in each other image i using an affine transformation (figure 3.27).

However naive, this method seems to be fast and to be well suited for the registration of almost unchanging (up to a rigid transform) images.

III.3.2.2 Beer-Lambert correction

The Beer-Lambert law predicts the measured signals as a function of the absorption of the illumination light by the tissues. If we separate the absorption by the RBCs from the one from vessels or other cortical tissues, we get $I \approx I_0 e^{-\alpha 2d} e^{-\beta 2d'}$, where I is the reflected light intensity, I_0 the incident light, α the absorption coefficient of vessel, d the width of the vessel at the considered point, β the absorption coefficient of the RBCs and d' its width. Thus the signal of interest - *e.g.* the presence of RBCs can be extracted by applying the following filter to each point of the sequence : $d' \backsim -log(\frac{I}{I_{base}})$ where $I_{base} = I_0 e^{-\alpha 2d}$
For each point I_{base} is evaluated as a robust minimal intensity throughout the registered sequence. Such a normalization using the minimal intensity instead of average intensity [211] enhances the signal from RBCs motion in the vessels without increasing the noise outside.

III.3.3 Flow-based vessels extraction

Blood-flow based image segmentation To adapt the shortest path formalism to a flow-based extraction of vessels, we replace the light-intensity

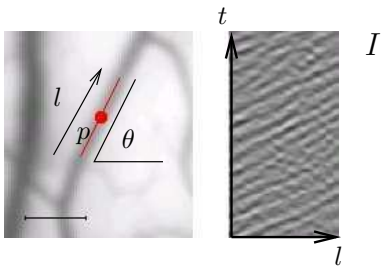


FIG. 3.28 – Extraction of the $space-time$ image. *Left :* neighborhood of p in the direction θ. *Right :* corresponding $space-time$ image.

(gray level) information by a value depending on the presence – or absence – of blood-flow. For a point p and an orientation θ, we determine whether flow following the direction θ is present at p throughout the sequence. To achieve this, we first extract a 2-dimensional $space-time$ image from a small neighborhood of p in our sequence of frames in direction θ (figure 3.28), yielding an image $I(l,t)$. Using the same structure tensor formalism as in [211], we compute the following tensor :

$$T(x,\theta,t) = \left\langle \left(\frac{\partial I}{\partial t}, \frac{\partial I}{\partial l}\right) \left(\frac{\partial I}{\partial t}, \frac{\partial I}{\partial l}\right)^T \right\rangle$$

As noted in [211], this orientation of this tensor can be used to locally evaluate the inclination of stripes in the image, and therefore the speed of flow. Furthermore, the ratio between its two eigenvalues (*i.e.* its anisotropy) gives an indicator of the presence of flow in that direction – the more the tensor is anisotropic, the more likely there is significant flow.
Let $\bar{T}(x,\theta)$ be the mean of this tensor over the time sequence. We propose to use the ratio $\rho(x,\theta)$ between the two eigenvalues of $\bar{T}(x,\theta)$ as an indication of the presence of flow at point x in direction θ.
Let Ω be defined by

$$\Omega \stackrel{\text{def.}}{=} [0,1]^2 \times [r_{\min}, r_{\max}] \times [0,\pi) \tag{III.3.1}$$

We propose the following approach inspired by the work presented in section III.2.3 to segment vessels based on flow information as shortest paths.

Assuming that the anisotropy of the tensors is constant across the section of a vessel, we propose the following model for anisotropy, for a vessel of width Λ_2 :

$$m(x_2) \overset{\text{def.}}{=} \begin{cases} 1 & \text{for } |x_2| > \Lambda_2/2, \\ 0 & \text{otherwise.} \end{cases} \tag{III.3.2}$$

This assumes that the anisotropy is constant inside and outside a vessel, and that it is more important inside. Notice that this rough assumption should be further investigated. We then define a set of scaled and rotated models as defined in (III.2.3).
Then, we denote by F the following quantity :

$$\forall \omega = (x, r, \theta) \in \Omega,\ F(\omega) \overset{\text{def.}}{=} \mathrm{NCC}_{\Lambda(r,\theta)}(M_{r,\theta}(\cdot), \bar{T}(x + \cdot, \theta))$$

A potential is thus defined over the 4D domain as

$$\forall \omega \in \Omega, \quad \rho(\omega) \overset{\text{def.}}{=} \max(1 - F(\omega), \epsilon). \tag{III.3.3}$$

Vessels as then extracted as shortest paths for this potential, as explained in III.2.5.

Figure 3.29 shows some results of vessels extracted by this method, superimposed on the first image of the sequence. Note how the smoothness in orientation imposed by our method allows the extraction of the vessel, even when crossings are cluttering the image. Also, in the left image, the segmented vessels has a very bad contrast with respect to the background, but is still segmented, which shows the interest of using a flow based segmentation for this modality.

III.3.4 Results

III.3.4.1 Frame registrations : rigid vs. non-rigid

Figure 3.30 compares the performance of our SIFT-based registration method with a classical rigid registration algorithm. Notice our method correctly registers the borders of the vessels.

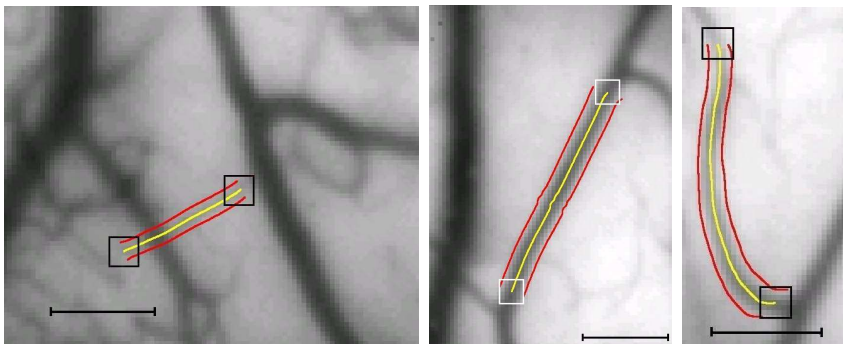


FIG. 3.29 – Automatically extracted vessels. Initial and final points are shown with squares. Notice that only flow information (vs anatomic information) was used to perform these segmentations.

III.3.4.2 Average flow in the vasculature

Figure 3.31 shows RBCs' speeds in three automatically segmented vessels. RBCs were found to cross any given section of the vessel one-by-one. Also, linear RBC density along the vessels'axis was found to be essentially equal for all three vessels ($D_1 \sim D_2 \sim D_3 \sim 6.7 \pm 1.18\ mm^{-1}$). The RBC current conservation equation $V_1 D_1 + V_2 D_2 + V_3 D_3 = 0$ is therefore satisfied within the variability of the data (where V_i are the RBCs' speeds in the vessels, and D_i the density of RBCs).

III.3.4.3 Variations of the flow in time

Estimation of velocity changes of the RBCs flow inside the vessels is much sensible to the accuracy of frame registration and vessel extraction. We performed such estimations on a trial of our monkey experiment presenting a high level of vibrations. Figure 3.32 shows that in the rigid registration case, the data remains too much polluted by signals originating from outside the vessel and no flow estimation is possible ; whereas the SIFT registration allows to deal with these vibrations most of the time (except when they are faster than frame acquisition rate, resulting in blurred raw images).

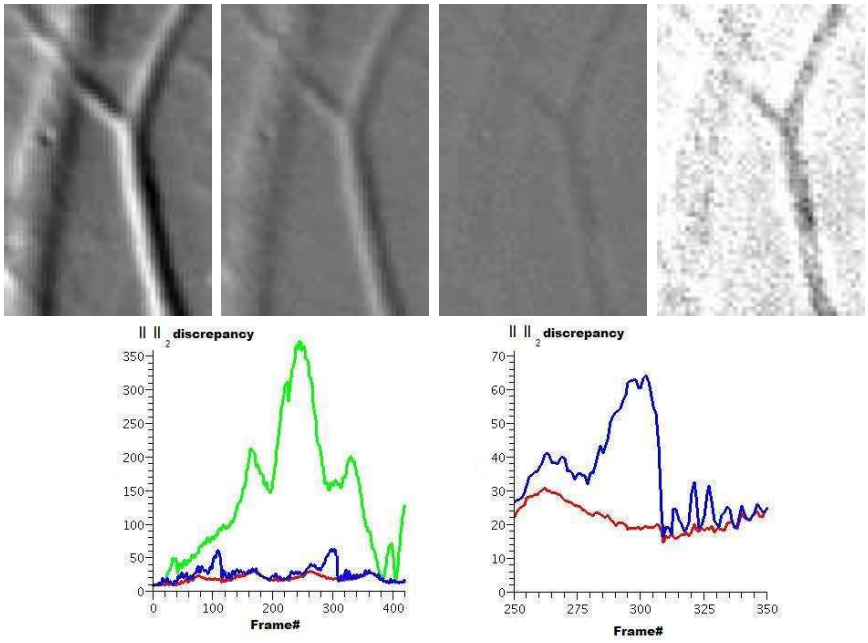


FIG. 3.30 – *Top :* ratio between frame 0 and frame 300 of a representative sequence, on an area of interest. From left to right : raw (no registration), rigid registration, SIFT-based registration (clipping range - *i.e.* gray-level intensity scale - is the same for the 3 images), SIFT-based registration with a clipping range ten times smaller. *Bottom :* $||\ ||_2$ comparison of each frame in the whole sequence to frame 0 (for the area of interest). Raw, rigid registration and SIFT-based registration are respectively represented in green, blue and red. *Left :* whole sequence. *Right :* zoom on frames 250 to 350

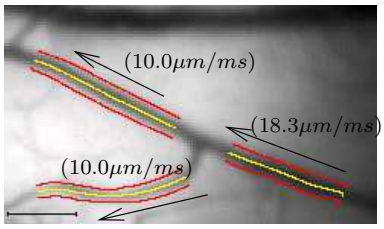


FIG. 3.31 – RBCs' speeds in three automatically segmented vessels

III.3.5 Conclusion and Discussion

Using the non-rigid image registration described here, we were able to achieve far better spatial matching between the vasculature in different frames. As a result, the blood flow signal could be recovered in vessels that did not yield any signal upon rigid registration. The obtained RBC flow could also be validated for conservation in vascular branching points, the total number of RBCs flowing in and out being found to match. The described data processing will hopefully allow increasing the accuracy and the sensitivity of optical imaging-based blood flow measurements, in particular with respect to reliably mapping over large vascular networks the small activity dependent blood flow changes elicited by neuronal activation. However, in view of the great difficulty of cortical imaging acquisition, we had only one sequence at our disposal in order to assess the quality of our methods. Validation on other data set should thus be needed.

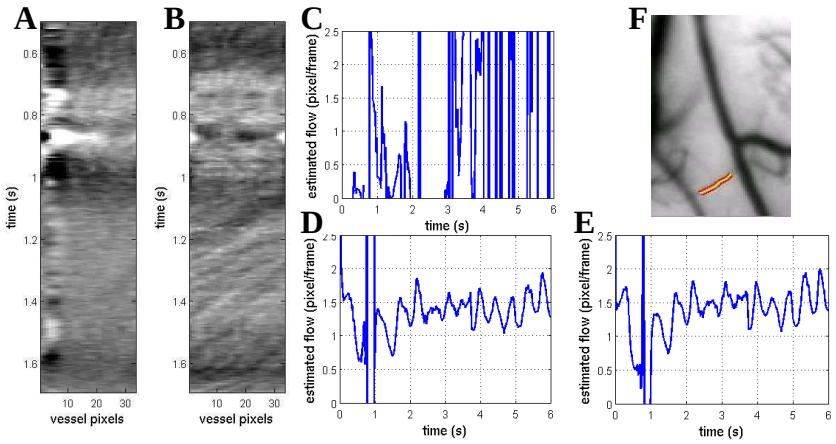


FIG. 3.32 – Comparison of the estimations of RBCs velocity changes after rigid vs non-rigid sequence registration. (F) Vessel considered, extracted using flow-based segmentation. (A,B) Space-time data extracted along this vessel after rigid and non-rigid registrations respectively. (C,D) Corresponding estimates of RBCs velocity, using the tensor structure information : only in the non-rigid case it is possible to estimate the velocity and then detect heart-pulsation changes. (E) Estimation in the non-rigid case, when averaging the structure tensor over the whole section of the vessel : only little information is added compared to using only the middle line of the vessel (D)

III.4 Application to Network of Curves Extraction

III.4.1 Introduction

In section III.2, we presented a framework for the extraction of vessels or roads between two user-defined points. It is however interesting for many medical applications to automatically extract full networks of vessels. In this section, we propose an extension of this work to extract full networks of vessels.

The proposed algorithm consists in an iterative growing of the network. At each step of the algorithm, a set of key points and junction points is added to seed new geodesic branches that are connected to the current network. The length τ of these branches is fixed and defines the granularity of the network. Notice that [15] recently proposed a similar growing-of-minimal-paths framework, but it is specialized in the segmentation of closed curves in 2D and meshing of surfaces in 3D.

III.4.2 Extension Domain

Given a network $\mathcal{A} = \mathcal{A}^{(i)}$ obtained after i steps of the algorithm, the growing process computes an extended network $\mathcal{A}^{(i+1)}$ by adding new geodesics that have an Euclidean length $\tau > 0$. This ensures that the branches of $\mathcal{A}^{(i+1)}$ have equal length so that its distribution is uniform, avoiding clusters of geodesic curves.

As in section A.2.1, the Euclidean geodesic distance $\mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega_0)$ from ω_0 to $\mathcal{A}$ is the Euclidean length of the geodesic $\gamma^* = \gamma(\mathcal{A}, \omega_0)$ joining ω_0 to $\mathcal{A}$

$$\mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega_0) = \int_0^1 \|(\gamma^*)'(t)\| dt.$$

The extension domain is defined as

$$\mathcal{E}(\mathcal{A}, \tau) = \partial\mathcal{B}^{\text{Euc}}(\mathcal{A}, \tau) \cap \mathcal{B}(\mathcal{A}, \tau\sigma) \tag{III.4.1}$$

with notations :

$$\begin{aligned}\mathcal{B}^{\text{Euc}}(\mathcal{A},r) &= \left\{\omega \in \Omega \setminus \mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega) \leq r\right\}\\ \partial\mathcal{B}^{\text{Euc}}(\mathcal{A},r) &= \left\{\omega \in \Omega \setminus \mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega) = r\right\}\\ \mathcal{B}(\mathcal{A},r) &= \left\{\omega \in \Omega \setminus \mathcal{U}_{\mathcal{A}}(\omega) \leq r\right\}\\ \partial\mathcal{B}(\mathcal{A},r) &= \left\{\omega \in \Omega \setminus \mathcal{U}_{\mathcal{A}}(\omega) = r\right\}\end{aligned} \tag{III.4.2}$$

It is composed of points $\omega \in \Omega$ that can be reached from $\mathcal{A}$ by geodesics of Euclidean length τ. We also want these points to be meaningful according to their geodesic distance to $\mathcal{A}$, which requires that $\mathcal{U}_{\mathcal{A}}(\omega) \leq \tau\sigma$. This imposes that for any point on $\mathcal{E}(\mathcal{A},\tau)$, $\mathcal{U}_{\mathcal{A}}(\omega)/\mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega) \leq \sigma$, *e.g.* the average value of ρ along the geodesic curve joining ω to $\mathcal{A}$ is better than σ. The threshold σ thus guarantees that the extension domain does not extend in areas where no vessel is present. σ must be selected according to the average response ρ of the vessel detector for the targeted application.
Figure 3.33 shows a typical extension domain around a single vessel.

Numerical computation. The computation of $\mathcal{U}_{\mathcal{A}}^{\text{Euc}}$ is described in appendix A.
In order to save time, the propagation for the computation of both $\mathcal{U}_{\mathcal{A}}^{\text{Euc}}$ and $\mathcal{U}_{\mathcal{A}}$ is performed only on the grid points ω that satisfy $\mathcal{U}_{\mathcal{A}}(\omega) \leq \tau\sigma$.
Figure 3.33 shows the level sets of the geodesic distance, computed inside the region where $\mathcal{U}_{\mathcal{A}}(\omega) \leq \tau\sigma$. $\mathcal{E}(\mathcal{A},\tau)$ is the intersection of the Euclidean ball border (light) and the geodesic ball (gray).

III.4.3 Network Extension

A set $\mathcal{K}(\mathcal{A})$ of locally optimal key points are seeded on the extension domain $\mathcal{E}(\mathcal{A},\tau)$. Theses points are the extremities of the new geodesic branches added to the current network $\mathcal{A}$.

Key points selection. A key point $\omega \in \mathcal{K}(\mathcal{A})$ is a local minimum of the geodesic distance, as measured using a neighborhood of size δ in the spatial domain. A point $\omega = (x, r, \theta) \in \mathcal{E}(\mathcal{A},\tau)$ is a local minimum of the geodesic distance if

$$\forall\tilde{\omega} = (\tilde{x},\tilde{r},\tilde{\theta}) \in \mathcal{E}, \quad |x-\tilde{x}| \leq \delta \quad \Longrightarrow \quad \mathcal{U}_{\mathcal{A}}(x) \leq \mathcal{U}_{\mathcal{A}}(\tilde{x}). \tag{III.4.3}$$

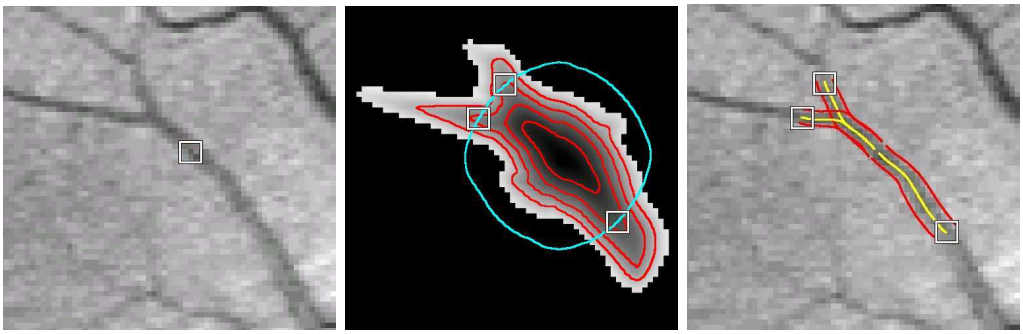


FIG. 3.33 – *left :* retinal image. Projection of the starting set $\mathcal{A}$ is indicated by a white square. *middle :* 2D schematic representation of the distance from $\mathcal{A}$, restricted to $\mathcal{B}(\mathcal{A}, \tau\sigma)$ (for each 2D point x, $min_{r,\theta}\mathcal{U}_{\mathcal{A}}(x, r, \theta)$ is represented). Corresponding level-sets are shown. A 2D representation of $\partial\mathcal{B}^{\text{Euc}}(\mathcal{A}, \tau)$ is superimposed, and key points are indicated by white squares. *right :* key points connexions to $\mathcal{A}$.

The set of key points is

$$\mathcal{K}(\mathcal{A}) = \underset{\omega\in\mathcal{E}(\mathcal{A},\tau)}{\text{loc.argmin}}\, \mathcal{U}_{\mathcal{A}}(\omega), \tag{III.4.4}$$

Extraction of local minimum of geodesic saliency is sensitive to noisy data sets, in particular in flat areas where no vessel is present. The size δ of the spatial neighborhood should be adapted to the noise level of the image. δ is set to 4 pixels in the numerical experiments.
Figure 3.34 shows a key point detected on the boundary of the extension domain in a synthetic example. Figure 3.33 shows that several key points are detected on a medical image near a branching of vessels.

Key points connexion. An augmented network is obtained by linking each $\omega \in \mathcal{K}(\mathcal{A})$ to the current network $\mathcal{A}$. The geodesic $\gamma^*(\omega, \mathcal{A})$ linking ω to $\mathcal{A}$ is computed and is added to the existing network. These paths are likely to be vessels segments starting from the initial set $\mathcal{A}$. This requires no additional computation since $\mathcal{U}_{\mathcal{A}}(\omega)$ is readily available inside $\mathcal{B}(\mathcal{A}, \tau\sigma)$, and $\gamma^*(\omega, \mathcal{A}) \subset \mathcal{B}(\mathcal{A}, \tau\sigma)$.
We denote the union of these paths by :

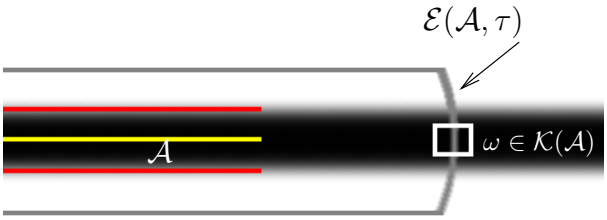


FIG. 3.34 – Schematic display of the extension domain extension domain $\mathcal{E}(\mathcal{A},\tau)$ where a single key point $\omega \in \mathcal{K}(\mathcal{A})$ is detected.

$$\mathbf{K}(\mathcal{A}) = \bigcup_{\omega\in\mathcal{K}(\mathcal{A})} \gamma^*(\omega,\mathcal{A}) \tag{III.4.5}$$

III.4.4 Network Junctions

In the case when $\mathcal{A}$ consists of several starting points, the procedure described in the previous section may not be sufficient to extract a complex network. Linking different part of the network is required in some cases (figure 3.35). Also, noisy images generate a network whose topology might progressively diverge from the true network, and correcting this requires to join several parts of it. This is achieved by computing a set of junction points $\mathcal{J}(\mathcal{A}) \subset \mathcal{B}(\mathcal{A},\tau\sigma)\cap\mathcal{B}^{\text{Euc}}(\mathcal{A},\tau)$ and linking these points to the network with geodesics.

Junction points selection. The geodesic distance $\mathcal{U}_\mathcal{A}$ is singular at a points ω that are connected by two geodesics to two different networks points $\omega_1,\omega_2 \in \mathcal{A}$. These two points are necessarily at equal geodesic distances $d_F(\omega,\omega_1) = d_F(\omega,\omega_2) = \mathcal{U}_\mathcal{A}(\omega)$. To ensure that these two points belong to different parts of the network that should be joined, we impose that they are far away according to the topology of $\mathcal{A}$, as measured by their distance $D_\mathcal{A}(\omega_1,\omega_2)$ defined as the Euclidean length of the shortest path from ω_1 to ω_2 in $\mathcal{A}$. Also, like in the case of extension domain, we require that the points are meaningful from the point of view of underlying vessels, *e.g.* that $\mathcal{U}_\mathcal{A}(\omega_1)/\mathcal{U}^{\text{Euc}}_\mathcal{A}(\omega_1)$ and $\mathcal{U}_\mathcal{A}(\omega_2)/\mathcal{U}^{\text{Euc}}_\mathcal{A}(\omega_2)$ are less than σ.

We denote as $\omega \in \mathcal{J}_0(\mathcal{A})$ the set of singular points whose closest network points (ω_1, ω_2) satisfy $D_{\mathcal{A}}(\omega_1, \omega_2) > \eta, \mathcal{U}_{\mathcal{A}}(\omega_1)/\mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega_1) \leq \sigma$ and $\mathcal{U}_{\mathcal{A}}(\omega_2)/\mathcal{U}_{\mathcal{A}}^{\text{Euc}}(\omega_2) \leq \sigma$. η is set to 10 pixels in the numerical experiments.
Similarly to key points (III.4.4), junctions points are local minimum of the geodesic distance, but are restricted to be singular points

$$\mathcal{J}(\mathcal{A}) = \underset{\omega \in \mathcal{J}_0(\mathcal{A})}{\text{loc.argmin}}\, \mathcal{S}_{\mathcal{A}}(\omega) \tag{III.4.6}$$

where (III.4.3) defines a local minimum. Figure 3.35 shows an example of junction points where two parts of $\mathcal{A}$ are aligned along the same vessel.

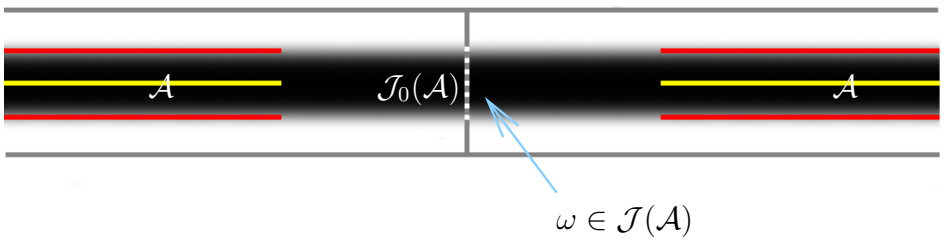


FIG. 3.35 – Junction point $\omega \in \mathcal{J}(\mathcal{A})$.

Junction connexion. Each junction point $\omega \in \mathcal{J}(\mathcal{A})$ is linked to the network $\mathcal{A}$ by extracting the two geodesics γ_1^* and γ_2^* linking ω to its two closest points $\omega_1, \omega_2 \in \mathcal{A}$. Numerically, the set $\mathcal{J}_0$ is determined during the Fast Marching propagation as points where the fronts emanating from different base points in $\mathcal{A}$ are collapsing. A careful initialization of the gradient descent around the point ω is needed because the distance function $\mathcal{U}_{\mathcal{A}}$ is singular at this location. In order to compute the geodesic to ω_1, we perform a gradient descent by using a numerical approximation of the gradient that only depends on values at points belonging to the front emanating from ω_1. The same holds for ω_2 .Two proper gradients are therefore computed that define the two descent directions for γ_1^* and γ_2^*.
We denote the union of these paths by :

$$\mathbf{J}(\mathcal{A}) = \bigcup_{\omega \in \mathcal{J}(\mathcal{A})} \gamma_1^*(\omega, \mathcal{A}) \cup \gamma_2^*(\omega, \mathcal{A}) \tag{III.4.7}$$

III.4.5 Vessels Cropping

Using geodesics that emanate from both key points and junctions points, a refined network $\bar{\mathcal{A}}^{(i+1)}$ is obtained that extend the initial network $\mathcal{A} = \mathcal{A}^{(i)}$

$$\bar{\mathcal{A}}^{(i+1)} = \mathcal{A}^{(i)} \cup \mathbf{K}(\mathcal{A}^{(i)}) \cup \mathbf{J}(\mathcal{A}^{(i)}) \tag{III.4.8}$$

Since the extremities of this refined network lie at a fixed distance τ from $\mathcal{A}$, the network might extend slightly beyond the boundaries of vessels. The final extended network $\mathcal{A}^{(i+1)}$ is obtained from $\bar{\mathcal{A}}^{(i+1)}$ using a cropping process that remove part of the network that are unlikely to belong to vessels.
For each curve $\{\gamma^*(t)\}_{t=0}^{1}$ emanating from a key point in the refined network $\bar{\mathcal{A}}^{(i+1)}$, a cropped curve is computed as $\{\gamma^*(t)\}_{t=0}^{t_c}$, where t_c is the minimum t satisfying $F(\gamma^*(t)) \leq \sigma$. The final network $\mathcal{A}^{(i+1)}$ is obtained from $\bar{\mathcal{A}}^{(i+1)}$ by cropping all the geodesic curves.

III.4.6 Overview of the Algorithm

Starting from an initial set $\mathcal{A}^{(0)}$ of (either isolated or not) seed points, the network is progressively grown by inserting new key points and junction points. In practice, a set $\{x_1, \ldots, x_K\}$ of spatial locations are provided either by the user or in an automatic way depending on the modality, and $\mathcal{A}^{(0)} = \{\mathcal{A}(x_k)\}_{k=1}^{K}$. This leads to the following steps :

1. *Initialization :* the initial points are $\mathcal{A}^{(0)}$, set $i \leftarrow 0$.
2. *Computing the extension domain :* compute $\mathcal{E}(\mathcal{A}^{(i)}, \tau)$ as explained in Section III.4.2.
3. *Seeding key points :* compute the set of key points $\mathcal{K}(\mathcal{A}^{(i)})$ as explained in Section III.4.3.
4. *Seeding junction points :* compute the set of key points $\mathcal{J}(\mathcal{A}^{(i)})$ as explained in Section III.4.4.
5. *Network extension :* compute the extended network $\bar{\mathcal{A}}^{(i+1)}$ defined in (III.4.8) by connecting seeded point.

6. *Network pruning :* compute $\mathcal{A}^{(i+1)}$ from $\bar{\mathcal{A}}^{(i+1)}$ as explained in Section III.4.5.

7. *Stop :* if $\mathcal{A}^{(i+1)} \neq \mathcal{A}^{(i)}$, set $i \leftarrow i+1$ and go back to 2.

Multi-pass refinement. The algorithm presented in the previous section uses a fixed τ, and might thus fail to detect vessel extremities. Indeed, if the vessel extremity is located far from $\mathcal{E}(\mathcal{A}^{(i)}, \tau)$, it might not be part of a geodesic starting from the key points $\mathcal{K}(\mathcal{A}^{(i)})$. To address this issue, a refinement pass is added if $\mathcal{A}^{(i+1)} = \mathcal{A}^{(i)}$, which lower the value of τ. In the numerical experiments, we have used a set of values $\tau = \{\tau_{\max}, \tau_{\max}/2, \tau_{\max}/4\}$. Reducing the value of τ does not require to re-compute $\mathcal{U}_{\mathcal{A}}$ and $\mathcal{U}_{\mathcal{A}}^{\text{Euc}}$.

III.4.7 Numerical Experiments

Experiments were carried out on both synthetic and medical images. For all the presented results, we used discretization $n_r = 12$ and $n_\theta = 12$ for radius and orientation dimensions. The speed on the orientation dimension what set to $\mu = 0.1\frac{\pi}{n_\theta}$, and the speed on the radius dimension what set to $\lambda = 0.5\frac{r_{max}-r_{min}}{n_r}$. Otherwise indicated, the values $\sigma = 0.25$ and $\tau_{max} = 48$ where used for synthetic examples, and $\sigma = 0.33$ and $\tau_{max} = 36$ for medical examples where the quality of vessel is less good on average.

Phantom experiment of figure 3.36 (top) shows the behavior of our method in case of junctions. All the junctions are handled correctly by the algorithm. Phantom experiment of figure 3.36 (bottom) shows the behavior of our method in case of a (self-)crossing. The correct segmentation of the vessel is retrieved by the algorithm. All the examples where computed from a single user-provided seed point.

Figures 3.37, 3.38 and 3.39 show results on medical images. For figures 3.37, 3.38, the only required human-interaction is to set the initial points, *e.g.* to point out the relevant structure to segment in the image. For figure 3.38, initial points were computed automatically as local minima of ρ. Notice that depending on the image modality, the initial point could be computed automatically (*e.g.* detection of optical papilla on retinal images).

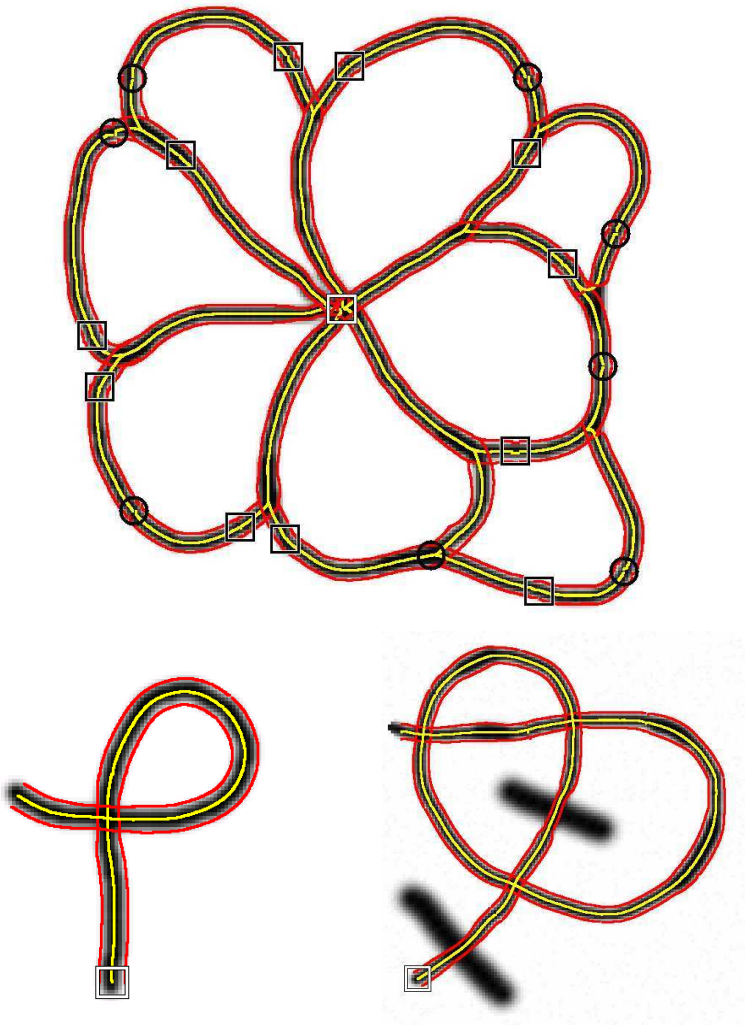


FIG. 3.36 – *top :* Multiple junctions example on synthetic data, after full running of the proposed method ($\tau_{max} = 96$) *White square :* user provided initial seed. *Black squares :* key points. *Black circles :* junction points. *bottom :* Crossing examples on synthetic data, after full running of the proposed method. *White square :* user provided initial seed.

III.4.8 Conclusion and discussion

The networks of curves extraction algorithm proposes a framework which naturally extend the geodesic method by defining the network extension notion. This method was tested on several synthetic and medical examples, and for

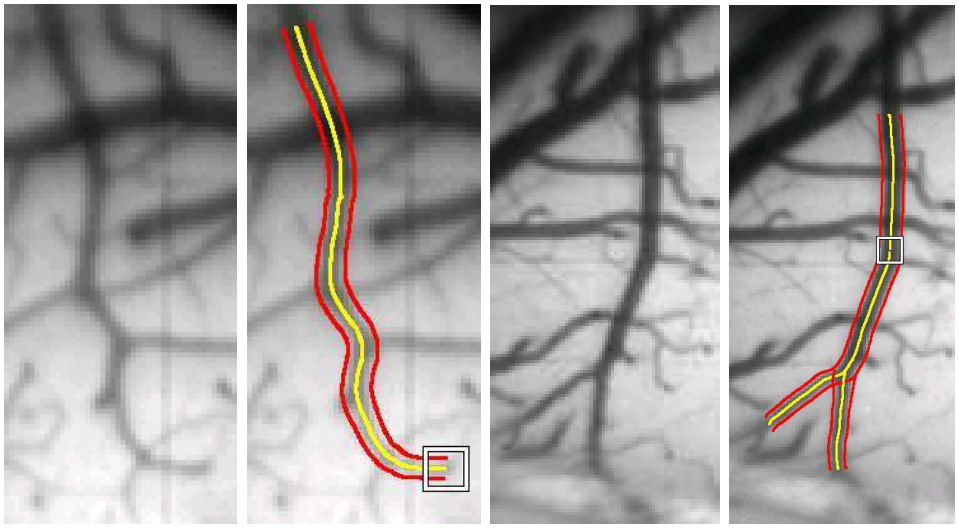


FIG. 3.37 – Experiments of the network extraction algorithm on cortical images. *White square :* user-defined initial seed. Notice the correct handling of intersections and forks.

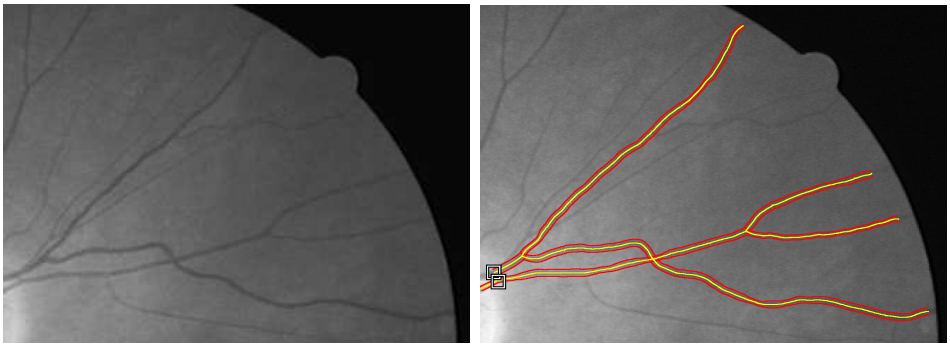


FIG. 3.38 – Experiments of the network extraction algorithm on retinal images. Two initial points were provided (*white squares*). Intersections and forks are correctly handled.

different kinds of initial conditions.

Some problems remain, and their precise analysis could lead to improvments in our algorithm.

- The overall speed of the algorithm could be improved. One could consider implementing speed-up versions of Fast-Marching. Furthermore, as Fast-

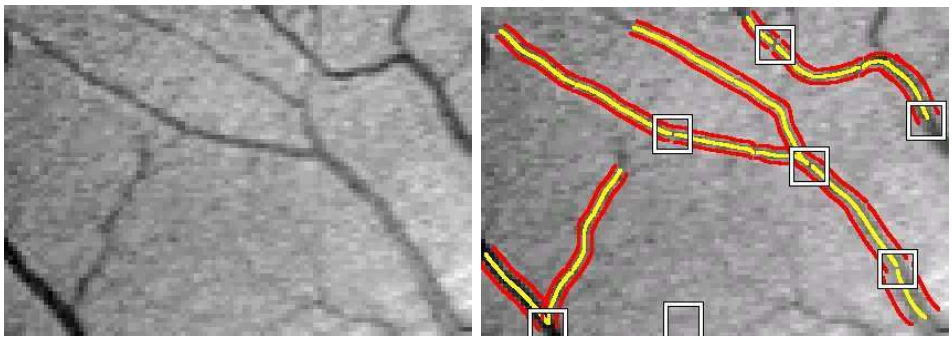


FIG. 3.39 – Experiments of the network extraction algorithm on retinal images. Initial seeds (*white squares*) were computed automatically as local minima of ρ over an extended 15 pixels neighborhood. Junction between different parts of the network are correctly handled. Notice that an incorrect seed (bottom), did not give birth to any local network.

Marching starts from the full current network at each step, many computations are performed several times – one could thus consider freezing like strategies, or partially reuse already computed distances map in some way.

- The algorithm does not make an actual difference between crossings and junctions – which are just disriminated by the angle of incidence of the vessel(s). The speed parameter on angular direction acts as a selection parameter for an admissible junction angle. Two vessels crossing with a small angle will lead to a false segmentation. It seems difficult to overcome this limitation without the help of a post-processing step.
- In our implementation, the parameters τ and σ were set globally by hand, and will decide wether or not an intersection is crossed or not. It would be interesting to be able to learn those parameters, or to make them be (locally) adaptable to characteristics of the image.
- Extremities of vessels are sometimes miss-handled (cf figure 3.39). This is due to the fact that parameter τ cannot be decreased too much without having incertainty in the speed along the shortest path. Again, a post-processing could handle this case.

Chapitre IV

HARDI-tracking using shortest paths

Introduction

In this chapter, we propose an application of shortest paths formalism to the problem of fiber tracking in High Angular Resolution Diffusion Imaging.

Diffusion Magnetic Resonnance Imaging (DMRI) [11] is derived from MRI (cf. section III.3.1), but allows to evaluate the probability distribution of water molecules in any direction at any point of a tissue. Its main application is to produce an image of white water fiber bundles in the human brain : due to organization and physico-chimical properties of the neurons axones, water molecules tend to diffuse faster along such bundles. Using DMRI imaging, one can then assess the presence of a white matter fiber at a given point of the brain, in a given direction. White matter fibers bundles are known to convey neural information between different part of the brain, and studying their anatomy helps to improve the knowledge of neuroscientists with respect to the connexion of different parts of the brain, and to its way of operating.

Many new diffusion models and fiber tracking algorithms have recently appeared in the literature always seeking better brain connectivity assessment, in particular regarding complex fiber configuration such as crossing, branching or kissing fibers. Clinical applications are also asking for robust tractography methods, as they are the unique *in vivo* tool to study the integrity

of brain connectivity.
The most commonly used model is the diffusion tensor (DTI) [12], in which diffusion is measured in the three principal directions (figure 4.1, *left*). This modality is only able to characterize one fiber compartment per voxel, and is not adapted to areas of fibers crossings.
Several alternatives have been proposed to overcome this limitation of DTI, mainly using high angular resolution diffusion imaging (HARDI) [210]. Several competing HARDI reconstruction technique exist in the literature, which all have their advantages and disadvantages. Nonetheless, the community seems to now agree that a sharp orientation distribution function (ODF), often called fiber ODF or fiber orientation density function (fODF) [86, 207, 90, 51], able to discriminate low angle crossing fibers needs to be used for fiber tractography (figure 4.1, *right*).

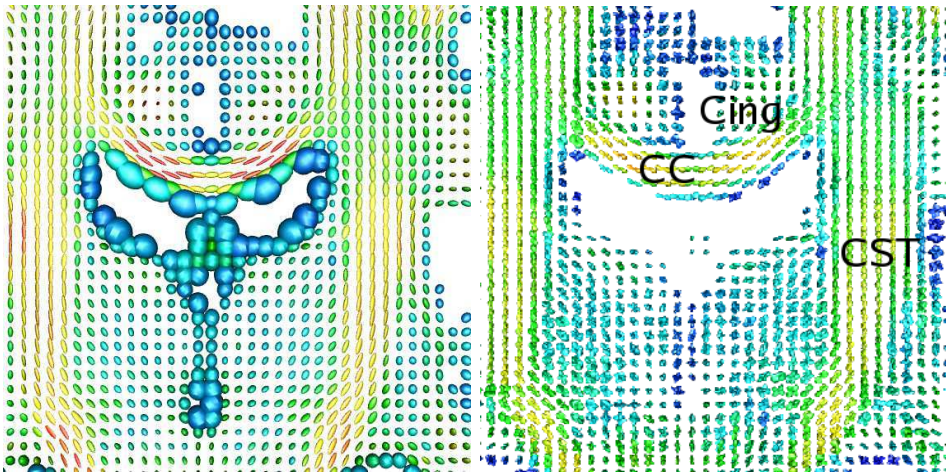


FIG. 4.1 – DTI of a human brain (*left*) and fODF (*right*) on the same coronal slice. Fibers of the *Corpus Callosum* (CC) and of the *Corticospinal Tract* can be seen in the plane of the image, as well as a section of *Cingulum* (Cing).

Three classes of algorithms exist to compute fibers or evaluate connectivities between different part of the brain from the volumic data : deterministic, probabilistic and geodesic. A large number of tractography algorithms have been developed for DTI, which are limited in regions of fiber crossings. While HARDI-based extensions of streamline deterministic [210, 107, 20, 215, 51] and probabilistic [147, 154, 191, 14, 178, 221, 51] tracking algorithms have

flourished in the last few years, there has not been, to our knowledge, any proposition to generalize DTI geodesic tracking [163, 88] for HARDI measurements.

In this chapter, we develop an algorithm for brain connectivity assessment using geodesics in HARDI. We propose to recast the problem of finding connectivity maps in the white matter to the calculation of shortest paths on a Riemannian manifold. This Riemannian manifold is a cross-product between white matter volume and a unit sphere representing the possible direction of fibers. The potential will be defined from fiber ODFs computed from HARDI measurements.

Anisotropy will be used in order to constraint the paths to follow a direction in the white matter which is coherent with the position on path on the unit sphere. Notice that in chapter III, this was unnecessary, due to the structure of the vessels : with our model, it is very unlikely for example to find shortest paths perpendicular to vessels.

Contents

Publication related to this work

This chapter is based on the work published in [165].

IV.1 Method

Firstly, let us recall some basics definitions about Riemannian manifolds – these definitions were already introduced in section I.3.

Let (V, g) be a Riemannian manifold *i.e.*

- V is a k-dimensional manifold
- for all $x \in V$, $g(x)$ is a bilinear symmetric positive definite application on T_xV, inducing a *metric* $||y||_x \stackrel{\text{def.}}{=} \sqrt{g(x)(y,y)}$ over that manifold.

The *length* of a smooth curve $\gamma : [0,1] \to V$ is then defined as

$$\mathcal{L}(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \|\gamma'(t)\|_{\gamma(t)} \mathrm{d}t \stackrel{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T g(\gamma(t))\gamma'(t)} dt. \tag{IV.1.1}$$

Given a set $\mathcal{A} \subset V$ of seeds points and a set $\mathcal{B} \subset V$ of ending points, a *geodesic* $\gamma^*(t) \subset V$ joining $\mathcal{A}$ to $\mathcal{B}$ is defined as a curve with minimal length between $\mathcal{A}$ and $\mathcal{B}$:

$$\gamma^*(\mathcal{A}, \mathcal{B}) \stackrel{\text{def.}}{=} \underset{\gamma \in \mathcal{C}(\mathcal{A},\mathcal{B})}{\operatorname{argmin}} \mathcal{L}(\gamma), \tag{IV.1.2}$$

where $\mathcal{C}(\mathcal{A}, \mathcal{B})$ is the set of curves γ such that $\gamma(0) \in \mathcal{A}$ and $\gamma(1) \in \mathcal{B}$. The corresponding *geodesic distance* is $d(\mathcal{A}, \mathcal{B}) \stackrel{\text{def.}}{=} \mathcal{L}(\gamma^*(\mathcal{A}, \mathcal{B}))$.
Following A, let us also define the *Euclidean length* of the curve γ

$$\mathcal{L}_{euc}(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \|\gamma'(t)\| \mathrm{d}t. \tag{IV.1.3}$$

and

$$\mathcal{L}_{sq}(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \|\gamma'(t)\|^2_{\gamma(t)} \mathrm{d}t. \tag{IV.1.4}$$

If we interpret the metric induced by g as as the inverse of a "speed" tensor over V, for any smooth curve γ, $\mathcal{L}(\gamma)/\mathcal{L}_{euc}(\gamma)$ can be thought of as the average of inverse speed along the curve, while
$\sqrt{\mathcal{L}_{sq}(\gamma)/\mathcal{L}_{euc}(\gamma) - (\mathcal{L}(\gamma)/\mathcal{L}_{euc}(\gamma))^2}$ represents the standard deviation of this quantity.

Connectivity measures. Considering $\mathcal{A}$ and $\mathcal{B}$ two subset of V we then define

$$\begin{aligned}
\mathcal{C}(\mathcal{A}, \mathcal{B}) &\stackrel{\text{def.}}{=} \frac{\mathcal{L}(\gamma^*(\mathcal{A}, \mathcal{B}))}{\mathcal{L}_{euc}(\gamma^*(\mathcal{A}, \mathcal{B}))}, \quad \mathcal{C}_{max}(\mathcal{A}, \mathcal{B}) \stackrel{\text{def.}}{=} \max_{t \in [0..1]} \|(\gamma^*(\mathcal{A}, \mathcal{B}))'(t)\|_{\gamma(t)} \\
\mathcal{C}_\sigma(\mathcal{A}, \mathcal{B}) &\stackrel{\text{def.}}{=} \sqrt{\frac{\mathcal{L}_{sq}(\gamma^*(\mathcal{A}, \mathcal{B}))}{\mathcal{L}_{euc}(\gamma^*(\mathcal{A}, \mathcal{B}))} - \left(\frac{\mathcal{L}(\gamma^*(\mathcal{A}, \mathcal{B}))}{\mathcal{L}_{euc}(\gamma^*(\mathcal{A}, \mathcal{B}))}\right)^2}
\end{aligned} \tag{IV.1.5}$$

$\gamma^*(\mathcal{A},\mathcal{B})$ being a geodesic between $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{C}(\mathcal{A},\mathcal{B})$, $\mathcal{C}_\sigma(\mathcal{A},\mathcal{B})$ and $\mathcal{C}_{max}(\mathcal{A},\mathcal{B})$ are respectively measures of average inverse speed, inverse speed standard deviation, and worst inverse speed to reach $\mathcal{B}$ from $\mathcal{A}$. They can therefore be interpreted as three different connectivity measures between $\mathcal{A}$ and $\mathcal{B}$ (see A).

IV.1.1 HARDI Riemannian manifold

We now explain how we recast the fibers bundles tracking problem from HARDI data to the calculation of connectivity maps on a Riemannian manifold.

let us denote by $E \subset \mathbb{R}^3$ the white matter volume, $\mathcal{S}$ the unit sphere and $V \overset{\text{def.}}{=} E \times \mathcal{S}$. Using such a 5-dimensional space can disambiguate crossing configurations since in such a space $(x,y,z,e_{\theta,\varphi})$ and $(x,y,z,e_{\theta',\varphi'})$ are completely different points. The idea was introduced [91], but the authors proposed to segment rather than track bundles using level-sets.

At every point $(x,y,z) \in E$, we can compute the fODF $f_{xyz} : e_{\theta,\varphi} \in \mathcal{S} \rightarrow f_{xyz}(e_{\theta,\varphi}) \in \mathbb{R}^+$.The full data can thus be naturally modelled as a mapping f from V to $\mathbb{R}^+ : f : (x,y,z,e_{\theta,\varphi}) \in V \mapsto f_{xyz\theta\varphi} \overset{\text{def.}}{=} f_{xyz}(e_{\theta,\varphi}) \in \mathbb{R}^+$.

Let us define the metric g at any point $(x,y,z,e_{\theta,\varphi})$ of V as

$$g^{-1}_{xyz\theta\varphi} \overset{\text{def.}}{=} \left(\begin{array}{ccc:cc} \overbrace{\rho(f_{xyz\theta\varphi}) & 0 & 0}^{E} & \overbrace{0 & 0}^{\mathcal{S}} \\ 0 & \rho(f_{xyz\theta\varphi}) & 0 & 0 & 0 \\ 0 & 0 & \rho(f_{xyz\theta\varphi}) & 0 & 0 \\ \hdashline 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & \alpha \end{array}\right) = \left(\begin{array}{cc} \rho(f_{xyz\theta\varphi})I_3 & 0 \\ 0 & \alpha I_2 \end{array}\right)$$

where ρ is an increasing function from $\mathbb{R}^+$ to $\mathbb{R}^{+*}$ and α is a parameter controlling the speed on the angular space $\mathcal{S}$ with respect to the speed on the E volume. Such a metric "favors" paths going through areas of high diffusion (figure 4.2).

Recasting the problem in the white matter volume, let us consider two points (x_1,y_1,z_1) and $(x_2,y_2,z_2) \in E$ between which we wish to estimate the connectivity. Let us denote $\mathcal{A} = \{x_1,y_1,z_1,e_{\theta,\varphi} \mid e_{\theta,\varphi} \in \mathcal{S}\}$ and $\mathcal{B} = \{x_2,y_2,z_2,e_{\theta,\varphi} \mid e_{\theta,\varphi} \in \mathcal{S}\} \subset E \times \mathcal{S}$.

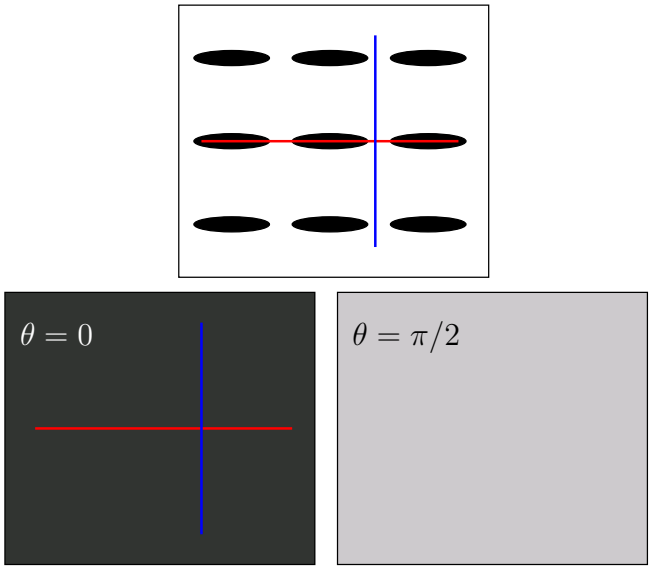


FIG. 4.2 – Illustration of the proposed potential in 2D. Starting from a $2D \rightarrow (\theta \rightarrow \mathbb{R}^+)$ dataset (*top*), we interpret it as a $(2D \times \theta) \rightarrow \mathbb{R}^+$ mapping. Slices for $\theta = 0$ (*bottom left*) and $\theta = \pi/2$ (*bottom right*) are represented. Potential is lower on the $\theta = 0$ slice. Paths (in blue and red) are computed on this $(2D \times \theta)$ space, and then reprojected in 2D. Notice however that the blue path is not consistant as it was computed in the $\theta = 0$ slice while having a $\theta = \pi/2$ direction.

$\mathcal{C}(\mathcal{A}, \mathcal{B})$, $\mathcal{C}_\sigma(\mathcal{A}, \mathcal{B})$ and $\mathcal{C}_{max}(\mathcal{A}, \mathcal{B})$ are then natural measures of connectivity between (x_1, y_1, z_1) and (x_2, y_2, z_2). Furthermore, let us denote by $\pi : E \times \mathcal{S} \rightarrow E$ the projection such that $\pi(x, y, z, e_{\theta,\varphi}) = (x, y, z)$. To the geodesic $\gamma^*(\mathcal{A}, \mathcal{B})$ in $E \times \mathcal{S}$ then corresponds a projected path $\pi(\gamma^*(\mathcal{A}, \mathcal{B}))$ in $E \subset \mathbb{R}^3$. Since $\gamma^*(\mathcal{A}, \mathcal{B})$ follows a high diffusion trajectory, $\pi(\gamma^*(\mathcal{A}, \mathcal{B}))$ is likely to follow an actual fiber bundle in the volume. With this point of view, α can be seen as a smoothing parameter of the angular variations of the fibers.
However, among the paths $\gamma : [0, 1] \rightarrow V$, we would like to favor the ones such that at every point $\pi(\gamma)$ follows the corresponding $e_{\theta,\varphi}$ direction : if $\gamma(t_0) = (x_0, y_0, z_0, e_{\theta_0,\varphi_0})$, we would like to have

$$\begin{aligned}(\pi(\gamma)_x(t_0), \pi(\gamma)_y(t_0), \pi(\gamma)_z(t_0)) \approx \\ \pm e_{\theta_0,\varphi_0} ||(\pi(\gamma)_x(t_0), \pi(\gamma)_y(t_0), \pi(\gamma)_z(t_0))||\end{aligned} \tag{IV.1.6}$$

The blue curve in figure 4.2 shows a path which is not satisfying this constraint, but as the same length as the red path.
In order to encourage these paths and thus to penalize paths which are transversal to fibers, we propose the following approach : let us consider a point $(x, y, z, e_{\theta,\varphi})$. Instead of using an isotropic metric $\rho(f_{xyz\theta\varphi})I_3$ in the first three directions, one would like to favor propagation along the $e_{\theta,\varphi}$ direction. In order to do so, $\rho(f_{xyz\theta\varphi})I_3$ is replaced by the following matrix :

$$(R_{\theta,\varphi})^T \begin{pmatrix} \rho(f_{xyz\theta\varphi}) & 0 & 0 \\ 0 & min(\epsilon, \rho(f_{xyz\theta\varphi})) & 0 \\ 0 & 0 & min(\epsilon, \rho(f_{xyz\theta\varphi})) \end{pmatrix} R_{\theta,\varphi}$$

where $R_{\theta,\varphi}$ is a rotation which maps the first axis to the $e_{\theta,\varphi}$ direction, and ϵ is some constant. As long as $\rho(f_{xyz\theta\varphi}) > \epsilon$, this tensor favors propagation in the $e_{\theta,\varphi}$ direction. However if $\rho(f_{xyz\theta\varphi}) \leq \epsilon$ (*i.e.* if the diffusion is small at this point), this does not make sense, and we keep the isotropic tensor defined by $\rho(f_{xyz\theta\varphi})I_3$. Figure 4.3 illustrate this : in the $\theta = 0$ slice, where potential is low, we encourage propagation in the $\theta = 0$ direction. The red curve will then be shorter than the blue one.

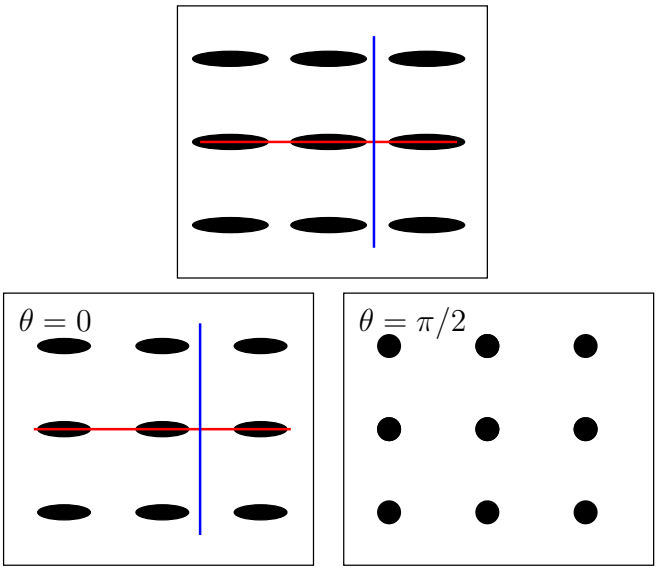


FIG. 4.3 – Illustration of the proposed corrected potential in 2D.

The choice of this metric is a natural way of handling the 5-dimensional

HARDI data and to obtain connectivity maps and fibers. It ensures that (i) the full HARDI angular information is used, (ii) geodesics go through areas of high diffusion, (iii) geodesics travel in those areas in the correct directions and (iv) crossing configurations are disambiguated.

Notice also that the analysis conducted in III.2.7 apply to this framework : the choice of the metric described above favors curves with low curvature.

IV.2 Implementation

For our problem, E was discretized as a subset of a 3-dimensional grid, at the HARDI measurement spatial definition – notice that due to the non-rectangular shape of E, we use the method desribed in appendix A.1 to prevent the front to be computed outside E, e.g. to propagate outside the white matter volume. $\mathcal{S}$ was meshed in such a way that every vertex of the mesh corresponds to a direction of HARDI measurements – leading to a 6 neighbors system. Furthermore, in order to achieve good precision, we chose to use a 26-neighborhood in the discretization of E.
However, computing distance map using Fast-Marching algorithm is this framework is unreallistic. Recall that the update state is of exponential complexity in the dimension of the space. In the proposed framework, every point of the discretization has 156 neighbors, and is surrounded by thousands of simplices.
Since we are mainly interested in precision in the high diffusion directions, we propose to compute $d(\mathcal{A}, \{x\})$ at each point by using Dijkstra local update step for the 156 neighbors. The Fast-Marching local update step is then only applied for the simplices S_d of $\mathcal{S}_{48}a$ (see section II.5.2) in the 3 first dimensions which contain $\pm e_{\theta,\varphi}$ direction, and their sub-simplices (see figure 4.4, *second scheme*). Furthermore, we perform this computation only if the diffusion is important enough (*i.e.* $\rho(f_{xyz\theta\varphi}) > \epsilon$) at current point. We also chose to update from a simplice only if the computed values satisfies monoticity and upwinding conditions described in II.4. The update step is thus the following.

$$U(x) \leftarrow \min\{\min_{S_j^{(1)}}\{s_j^{(1)}\}, s_d\} \tag{IV.2.1}$$

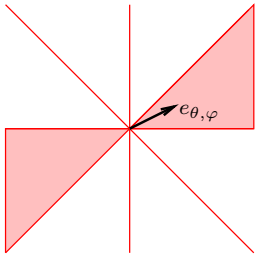


FIG. 4.4 – Illustration of the simplices used in the update step, in 2D, with a 8-neighbors system. All the 1-dimensional simplices are used, while for simplices of bigger dimension, only the ones containing $\pm e_{\theta,\varphi}$ direction are used.

Figure 4.5 shows an application of this strategy to a uniform anisotropic potential in dimension 2.

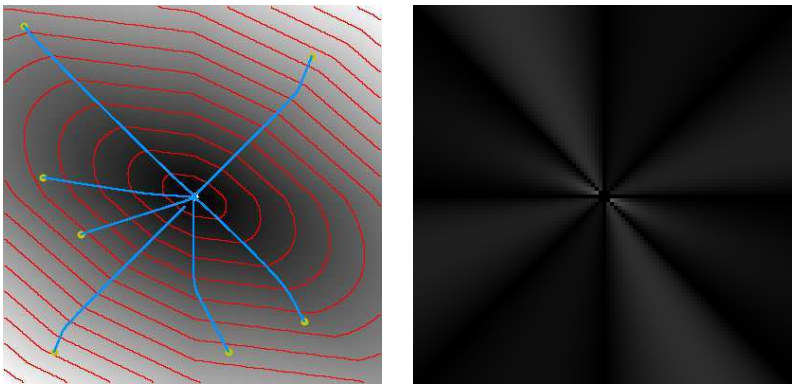


FIG. 4.5 – Results of the mixed Dijkstra-FastMarching algorithm for a uniform anisotropic potential in dimension 2, using the neighborhood system described in 2.9.

This leads to tracktable computations, while the precision in the fibers direction is preserved. This choice will be further discussed in the experimental results section.

IV.3 Experimental results

IV.3.1 Real HARDI data

The HARDI dataset was acquired on a whole-body 3 Tesla Magnetom Trio scanner (Siemens, Erlangen) equipped with an 8-channel head array coil [4]. The spin-echo EPI sequence, TE = 100 ms, TR = 12 s, 128 x 128 image matrix, FOV = 220 x 220 mm^2, consists of 60 evenly distributed diffusion encoding gradients with a b-value of 1000 s/mm^2 and 7 images without any diffusion weightings. The measurement of 72 slices with 1.7mm thickness (no gap), which covered the whole brain, was repeated three times, resulting in an acquisition time of about 45 minutes. The SNR in the white matter of this S_0 image was estimated to be approximately 37. Additionally, fat saturation was employed, 6/8 partial Fourier imaging, Hanning window filtering and parallel GRAPPA imaging with a reduction factor of 2.
From these HARDI measurements, the fiber ODF was reconstructed. As mentioned in the introduction, several fiber ODF reconstruction algorithm exist [86, 207, 90, 51]. Here, we used the analytical spherical deconvolution transform of the q-ball ODF using spherical harmonics [51]. We used an order 4 estimation with symmetric deconvolution fiber kernel estimated from the real data, resulting in a profile with FA = 0.7 and $[355, 355, 1390] \times 10^{-6} mm^2/s$.
The geodesic tracking is performed within a white matter mask was obtained from a minimum fractional anisotropy (FA) value of 0.1 and a maximum ADC value of 0.0015. These values were optimized to produce agreement with the white matter mask from the T1 anatomy. The mask was morphologically checked for holes in regions of low anisotropy due to crossing fibers.

IV.3.2 Geodesic connectivity results

For each bundle except the Superior Longitudinal Fasciculus (SLF), experiments were carried out with $\rho(f) = ln(f)/ln(2)$, $\epsilon = 1$ and $\alpha = 2$ after thresholding values of the fODF under 1 to avoid negative values – the choice of a logarithmic function for ρ was driven by both the need to compact the highly variable values of the fODF (many other methods perform a linear

voxelwise rescale – which is not suitable for our purpose), and the need to avoid strong anisotropy that will lead to violations of the upwinding conditions (II.4.17). Our method however demonstrates robustness with respect to the exact choice of these parameters.

Since SLF has high curvature, we set angular speed $\alpha = 8$ in order to favor tracking of actual SLF rather than projections on the occipital cortex. Runtime was about 75min for each bundle. It can be further reduced by computing only some of the connectivity maps, or by computing them only on a subset of white matter. While results presented below show connectivity maps on the full maps, experiments show that the bundles can be retrieved by stopping the algorithm when 20% of the mask has been visited. The runtime is then reduced to about 12min.

Figure 4.6 shows connectivity measures and some geodesics obtained from different seeds manually placed into major fibers bundles, which agree with our knowledge of the white matter anatomy. Notice the correctness of the maps on Corticospinal Tract (CST) , which does not spread into the Corpus Callosum (CC). Also, the Cingulum (Cg), which is a thin structure close to CC is correctly handled by our method. This clearly shows the advantage of using a 5D space : since fibers in Cg and CC are perpendicular, these two bundles are very distant in our 5D space, while they are extremely close in 3D. Other fibers bundles are also correctly retrieved, such as the Inferior Fronto-Occipital Fasciculus (IFO) and the Anterior Thalamic Radiations (ATR). Furthermore, coherent results are obtained by the three proposed connectivity measures.

On figure 4.7 isosurfaces of the connectivity maps are shown for all the previous fibers bundles, and for the corresponding fibers in the right hemisphere. Notice that the lower part of SLF is missed in the right hemisphere.

Figure 4.8 shows some geodesics in the left hemisphere.

Figure 4.9 shows results on Corpus Callosum (CC). Several experiments were ran from manually provided seeds. Notice that CC is not segmented by our method. Rather, fibers – include spenium on the posterior part of CC – are tracked from each given seed. Cingulum is also represented. While this anatomical structure is very close to CC in 3D space, it is not in our 5D segmentation space, and thus it is correctly not retrieved by our method.

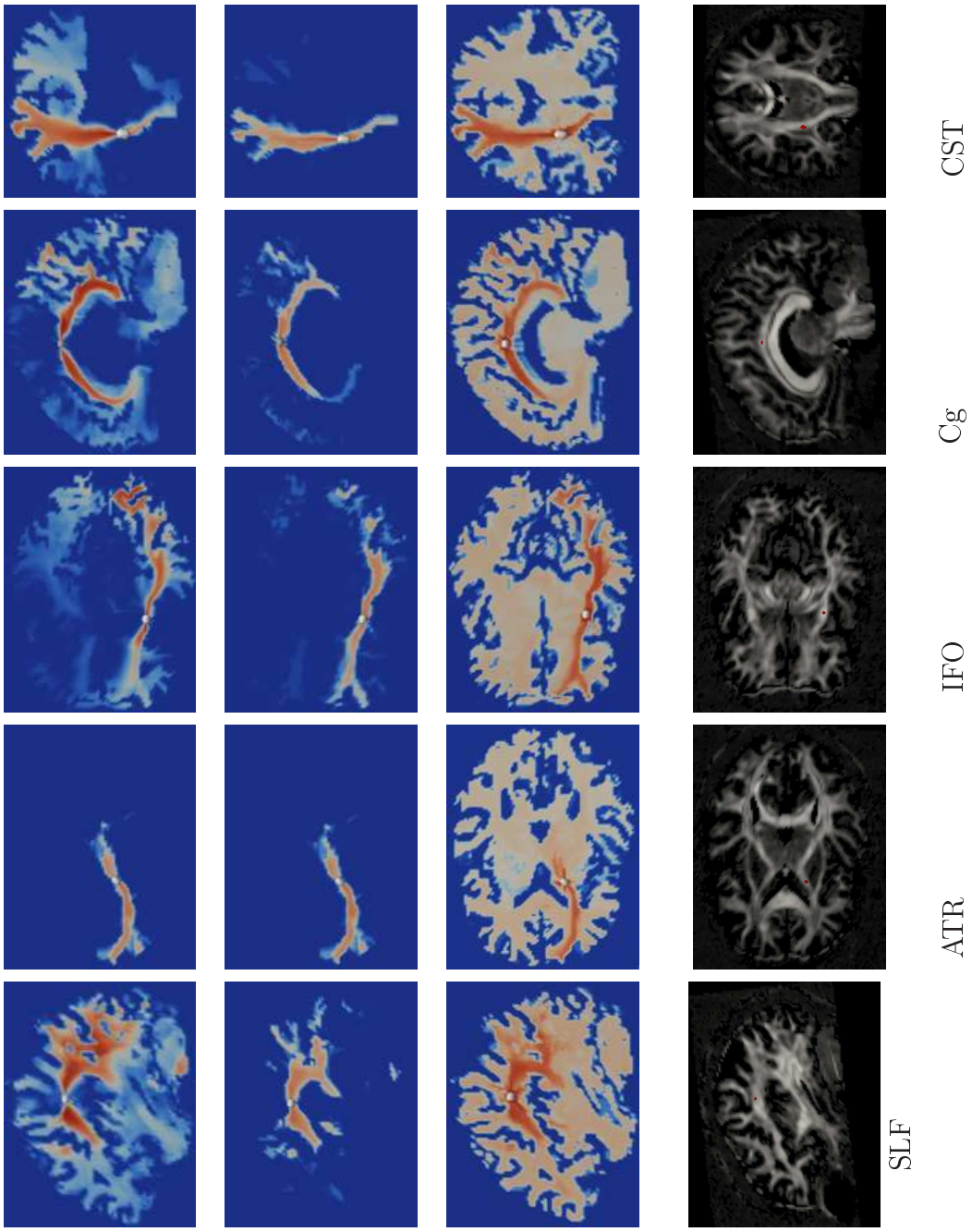


FIG. 4.6 – Geodesic tracking results on five major fibers bundles in left hemisphere. From left to right, $\mathcal{C}$, $\mathcal{C}_{max}$, $\mathcal{C}_{sigma}$ and FA.

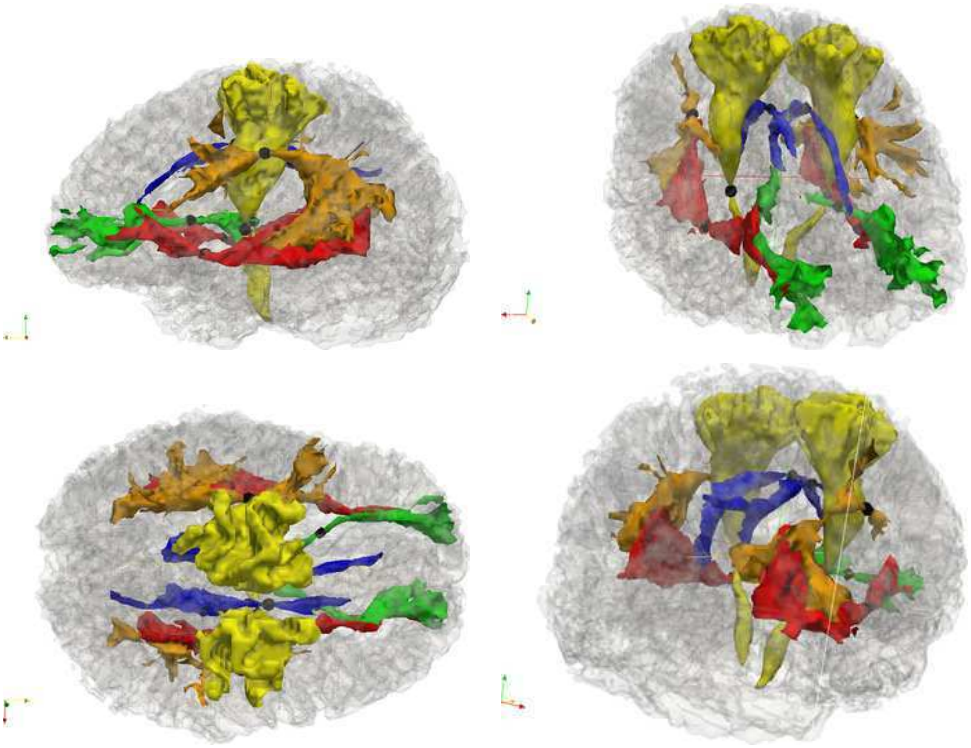


FIG. 4.7 – Geodesic tracking results on major fibers bundles – left and right hemispheres. Isosurfaces of the connectivity measures are shown. Each bundle in a different color. In yellow, the CST ($\mathcal{C}$) ; in blue, the Cg ($\mathcal{C}$) ; in red, the IFO ($\mathcal{C}_{max}$) ; in orange, the SLF ($\mathcal{C}_{max}$) ; in green, the ATR ($\mathcal{C}$). *Bottom Row :* some corresponding geodesics.

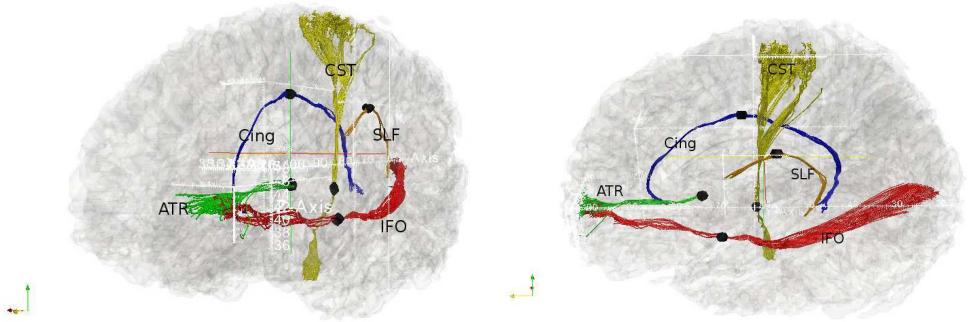


FIG. 4.8 – Geodesics corresponding to major fibers bundles in left hemisphere. In yellow, the CST ; in blue, the Cg ; in red, the IFO ; in orange, the SLF ; in green, the ATR.

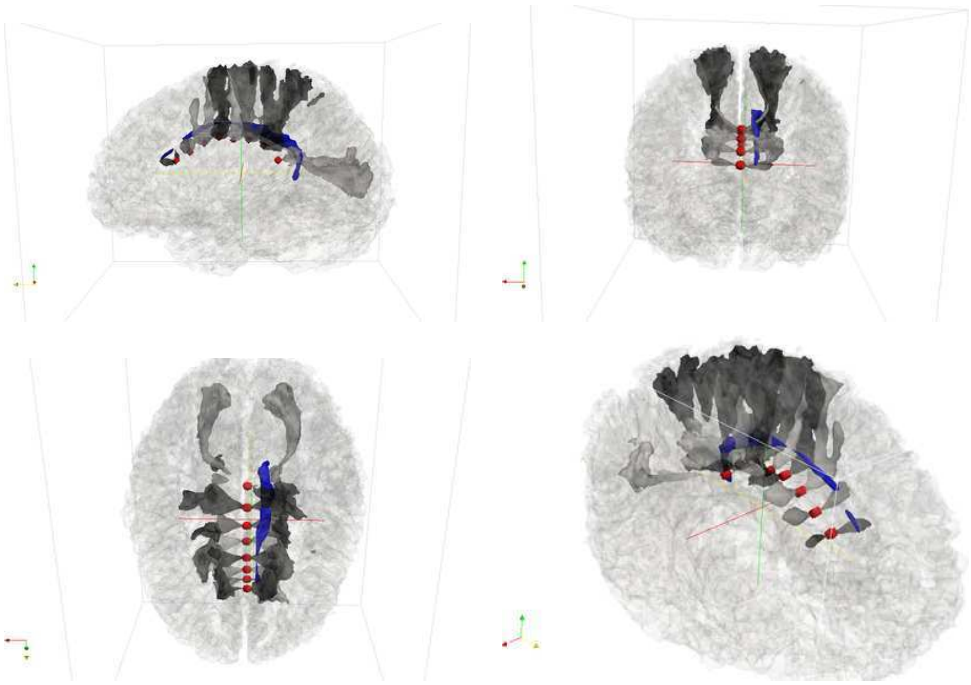


FIG. 4.9 – Geodesic tracking results on Corpus Callosum (CC). Seeds are indicated in red. Cingulum in left hemisphere is also represented (Red)

IV.3.3 Comparison with existing methods

In this section, we compare our results with results obtained by other methods on the same data :

- The GCM algorithm of [163, 118] (figures 4.10 and 4.11). Tensors were evaluated from the raw data using the framework developped in [117]. Connectivity measures corresponding to our $\mathcal{C}$ and $\mathcal{C}_\sigma$ were computed as indicated in [163, 118]. We furthermore computed the $\mathcal{C}_{max}$ connectivity measure.
- The deterministic HARDI tracking algorithm described in [52] (figure4.12).

While GCM are faster than our method, the obtained C and C_{max} results are less focussed on the bundles of fibers, and are subject to "leaks" in other bundles (Cg, SLF, and link to the opposite hemisphere through CC for CST). Moreover, shallow bundles not aligned with the grid seem to be missed by the method (e.g. Cg, lower part of CST)

On all experiments, C_{sigma} is sensitive to the grid orientation, and gives results of varying quality (figure 4.10).
The deterministic tracking approach (figure 4.12) gives generally satisfying results, but is also subjet to leaks (leak in opposite hemisphere for ATR, leak in CC from Cg). Due to its high curvature and its ambiguity, SLF is also not tracked correctly.
Overall, while these two methods are faster, our method seems to perform in a comparable or better way on the selected tracks.

IV.3.4 Approximation quality

In this section, we discuss the choice of (IV.2.1) as an approximation of more complete Fast-Marching update steps. We computed connectivity maps using 4 different update schemes : (1) pure Dijkstra algorithm, (2) (IV.2.1) scheme, (3) : (2) + Fast-Marching update state applied to the neighboring simplices in the three first directions (4) Fast-Marching update state applied to all simplexes in the three first directions. Figure 4.13 synthesizes those schemes. Isosurfaces of connectivity maps are shown figure 4.14, for the 4 schemes, and the same connectivity value. While pure Dijkstra algorithm produces different results, the other methods provided qualitatively equivalent results. This plaid for the use of scheme (2), which is the fastest among those three.

IV.4 Conclusion and Discussion

We presented a geodesic based tracking algorithm on HARDI data. Our method rapidly estimates connectivity maps inside a white matter mask from seed points, without the need for an explicit computation of fibers. All the directions of HARDI measurments are used by our method. Our experiments plaid for the use of a 5D space and show that our method is able to recover complex fiber bundles, which are often difficult to track.
However, our experiments are preliminar. A full validation of the method would imply a systematic study on a inter-subject large database, as well as a the study of the dependancy of the method with respect to its parameters, including the choice of ρ.

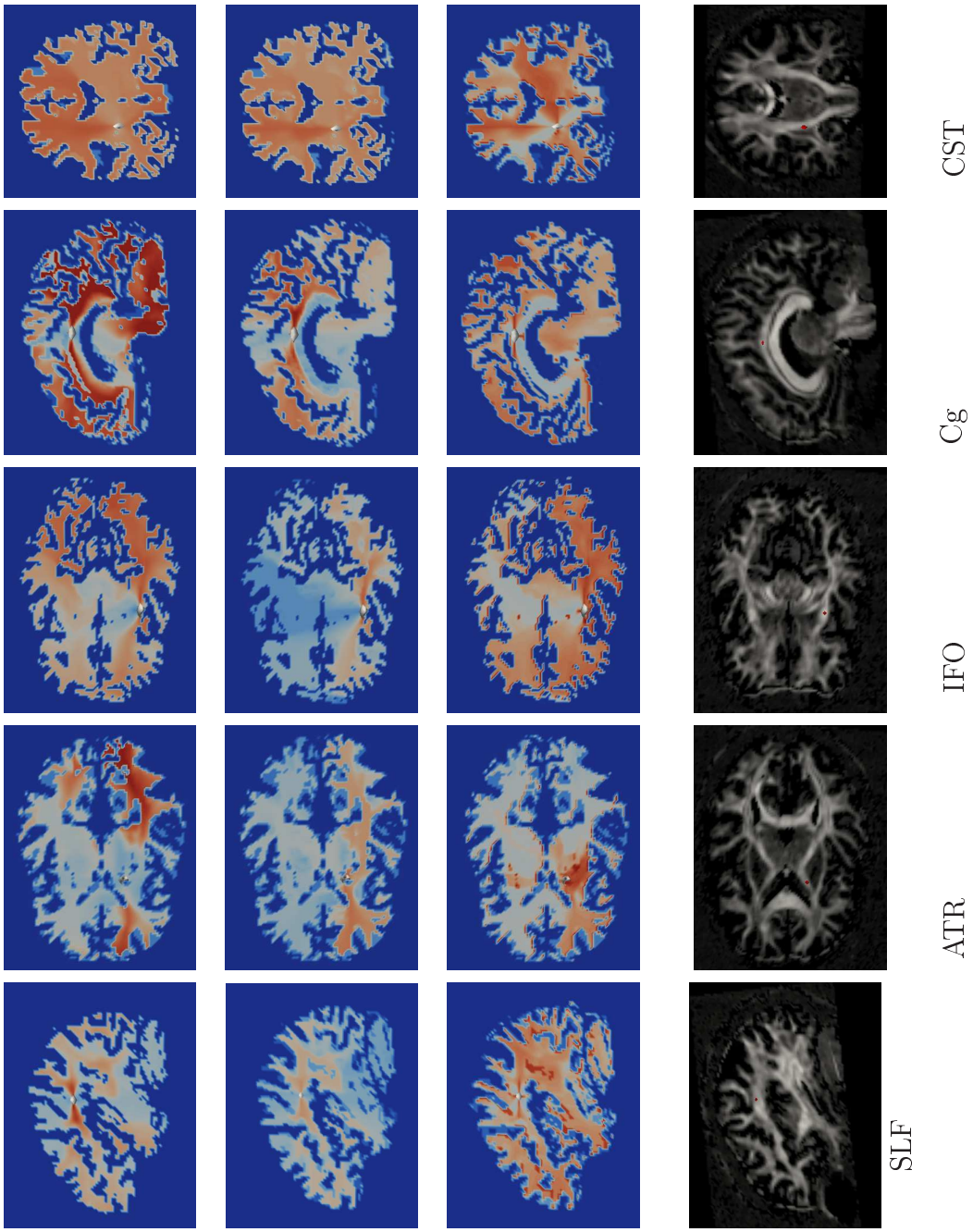


FIG. 4.10 – GCM results on five major fibers bundles in left hemisphere. From left to right, $\mathcal{C}$, $\mathcal{C}_{max}$, $\mathcal{C}_{sigma}$ and FA.

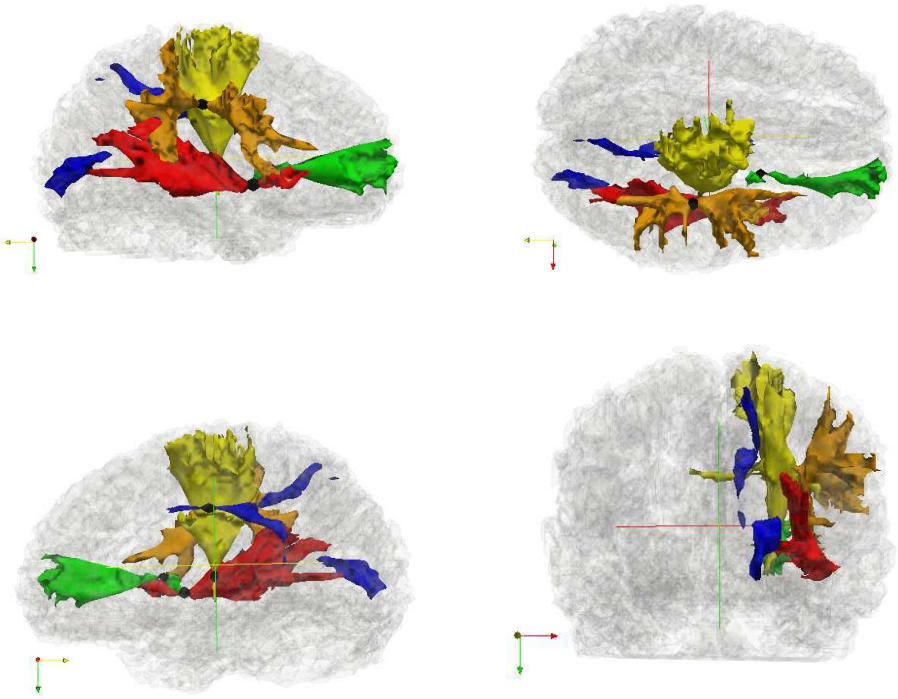


FIG. 4.11 – GCM results on major fibers bundles in the left hemispheres. Isosurfaces of the connectivity measures are shown. Each bundle in a different color. In yellow, the CST ($\mathcal{C}$) ; in blue, the Cg ($\mathcal{C}$) ; in red, the IFO ($\mathcal{C}_{max}$) ; in orange, the SLF ($\mathcal{C}_{max}$) ; in green, the ATR ($\mathcal{C}$).

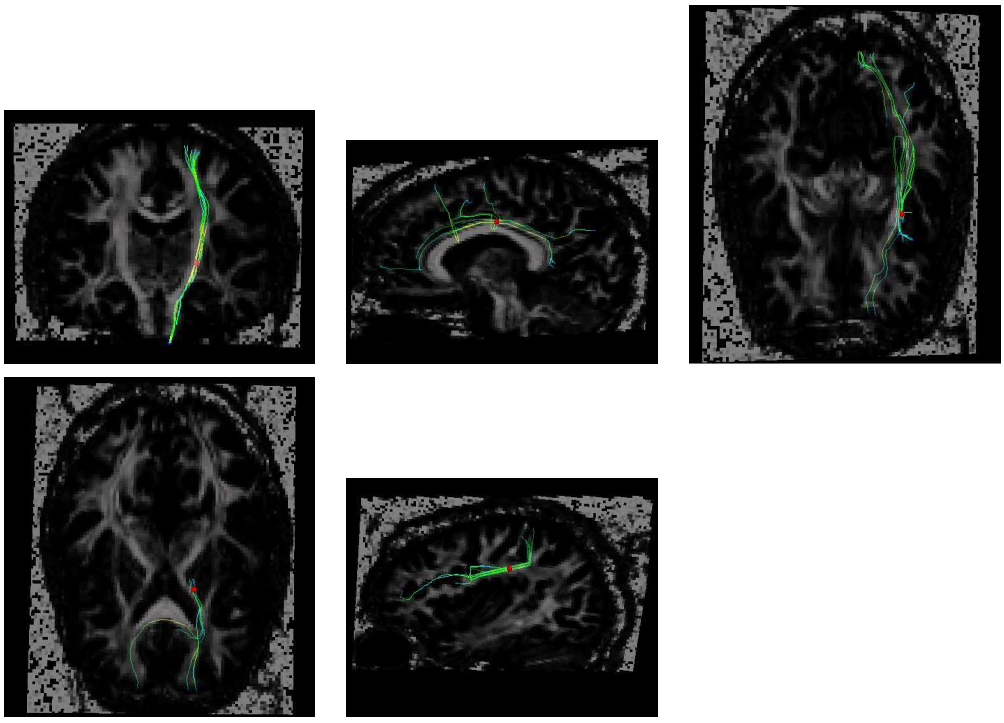


FIG. 4.12 – Deterministic tracking results on five major fibers bundles. From top to bottom and from left to right : ATR, Cg, IFO, ATR, SLF

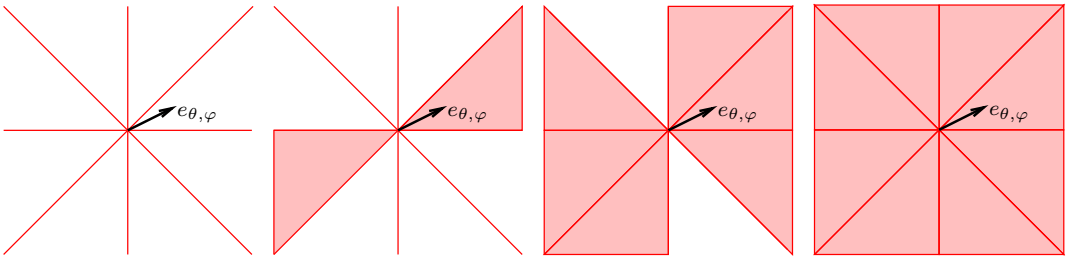


FIG. 4.13 – Illustration of the simplices used in the different update schemes in 2D, with a 8-neighbors system. From left to right : (1), (2), (3) and (4).

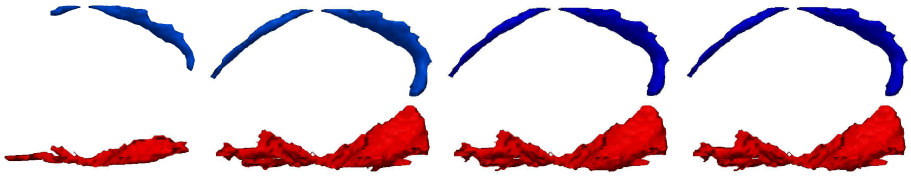


FIG. 4.14 – Isosurfaces of $\mathcal{C}$ for *Cingulum* (*top*), and $\mathcal{C}_{max}$ for *IFO* (*bottom*). From left to right, schemes (1), (2), (3) and (4) were used.

Conclusion générale

Le travail de thèse dont il est rendu compte dans ce manuscrit a porté sur l'application de méthodes de calcul de plus courts chemins à différentes problématiques tirées du domaine de l'imagerie médicale : segmentation de vaisseaux et de réseaux de vaisseaux pour différentes modalités, et calculs de cartes de distances dans la matière blanche à partir de données d'IRM de diffusion à haute résolution angulaire.

D'un point de vue théorique, sa contribution principale est une présentation unifiée de différentes versions des *Fast-Marching* – donnant une vision géométrique de l'algorithme, et permettant d'effectuer une preuve de convergence relativement simple dans le cas le plus général. Le chapitre correspondant se veut également une tentative de clarification par rapport à des références considérées comme classiques, mais qui contiennent néanmoins nombre d'imprécisions. Le cœur de cette thèse porte sur des applications de ces algorithmes.

Du point de vue applicatif, une idée centrale du travail présenté est celle de se placer dans des espaces où l'orientation des structures anatomiques considérées est représentée explicitement. Ceci est évidemment naturel et important dans le cadre de calculs de cartes de connectivités au sein de la matière blanche, les données que nous avons à disposition rendant essentiellement compte de l'orientation des faisceaux de fibres. Mais nous avons également montré l'intérêt d'introduire ce genre de méthodes dans le cadre de segmentation d'images bidimensionnelles, pour lesquelles il n'y a aucune information a priori concernant l'orientation des structures à segmenter.

Dans le cadre d'images bidimensionnelles, nous avons proposé un formalisme permettant de segmenter de façon robuste des structures tubulaires, tout en évaluant leur rayon. Les applications à différentes modalités, et en particu-

lier l'extension proposée à la segmentation à partir de flot optique suggère que notre méthode pourrait trouver d'autres applications dans le cadre de l'imagerie médicale. Le travail concernant la segmentation automatique de réseaux ouvre également des perspectives vers la création de nouveaux systèmes automatiques ou semi-automatiques de traitement d'images médicales. L'optimisation du temps de calcul n'a pas été une préoccupation centrale de notre travail. Son amélioration pourrait permettre l'inclusion de nos méthodes dans des interfaces de type *livewire*.

Concernant les applications au calcul de cartes de connectivité dans la matière blanche, l'algorithme que nous avons proposé – s'il n'a pas encore été testé sur des jeux de données complet – offre des premiers résultats intéressants : en un temps faible comparé à celui de l'acquisition des données, il permet d'obtenir des cartes de connectivité correspondant à nos connaissances anatomique, y compris pour des faisceaux fins et/ou proches d'autres faisceaux, tels le *Cingulum*. Étant donné la dimension de l'espace considéré pour cette méthode, notre parti pris a été de sacrifier la précision – en particulier, notre schéma ne converge pas vers une solution théorique – afin de diminuer le temps de calcul. Nous avons cependant veillé à conserver la précision dans les directions principales des fibres. Une étude plus approfondie de ce que nous perdons par rapport aux *Fast-Marching* complets serait intéressante. Il serait également judicieux de valider cette méthode sur une étude à plusieurs sujets, et de la comparer à d'autres méthodes existantes.

Annexe A

Appendix to shortest paths computation

Let us consider the framework described in II.4.

A.1 Shortest paths computation on a subset of $\mathbb{R}^n$ or V

Fast-Marching can be easily adapted to the computation of shortest paths on a subset Ω of $\mathbb{R}^n$ or of the considered manifold V (section I.3.1.2).

A first solution is to put the points of the discretization outside Ω in $\mathcal{A}$ at the beginning of the algorithm with $+\infty$ value – or to simply remove them from the discretization. Therefore, those points will not be update, nor they will participate in updates of their neighbors.

However, in view of performing a gradient descent to compute geodesics, it is desirable to dispose of the value of U at any point immediately outside Ω – which will allow a unified evaluation of the gradient in Ω. One possible solution is to label such points, and to set their initial value to $+\infty$. These points we behave as Ω points during the execution of the algorithm, with the exception that they will not be used to update their neighbors values. An evaluation of U will thus be available for those points, without perturbing the values obtained for points in Ω.

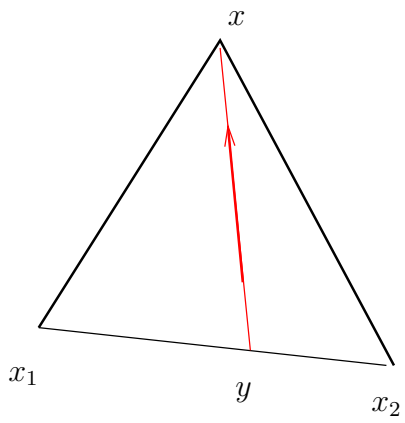


FIG. 1.1 – Approximation of the shortest path in the simplex used during the update step.

A.2 Connectivity measures

A.2.1 Definitions, computations

If we interpret the local metric as the inverse of a speed tensor (section I.3.1.6), we saw that a shortest path can be considered as shortest in a temporal sense. The average potential of the shortest path between two points can then be interpreted as a connectivity measure between these two points in several contexts (chapter II). The standard deviation of speed, as long as its minimal value along the path can also be meaningfull for the connectivity assesment between two points (chapter III.)

In order to estimate those quantities, we propose to use a generalization of the process described in [163]. Let us consider an update step in a simplex of matrix X. Condition $\mathbf{C}_m$ implies that $M^{-2}\nabla U$ comes from inside the simplex (figure 1.1).

Inside the simplex, we can approximate the shortest path to x with the line going through x of direction $M^{-2}\nabla U$. Let us denote by y the intersection of this line and the facet opposed to x in the simplex.

Furthermore, we write $y = \sum_{i=1}^{n} \lambda_i x_i$ in barycentric coordinates [1].

[1] The equation of the opposite facet of the simplex is given by $\sum e_i t_i = 1$, where $\mathbf{e} \overset{\text{def.}}{=} X^{+}\mathbf{1}$. We immediately deduce the coordinates of point y. λ is then expressed as $X^{+t}y$.

For any point x, let us denote by $U_{euc}(x)$ the estimated Euclidean length along the shortest path from x to the origin. We will use the following approximation :

$$U_{euc}(x) \approx \sum_{i=1}^{n} \lambda_i U_{euc}(x_i) + \|x - y\| \tag{A.2.1}$$

Average potential along a geodesic is then given by $\mathcal{C} = U/U_{euc}$.
Similarly, we can estimate the square of the potential, averaged along a geodesic (U_{sq}) and the maximal potential (or the minimal speed) along a geodesic (U_{max}).

$$U_{sq}(x) \approx \sum_{i=1}^{n} \lambda_i U_{sq}(x_i) + \|x - y\| \frac{\|x - y\|_b^2}{\|x - y\|^2} \tag{A.2.2}$$

$$U_{max}(x) \approx \max\{\min_{i=1}^{n}\{\lambda_i U_{max}(x_i)\}, \frac{\|x - y\|_b}{\|x - y\|}\} \tag{A.2.3}$$

$\mathcal{C}_{max} = U_{max}$ can be seen directly as a connectivity measure.
The standard deviation of potential along a geodesic – $\mathcal{C}_\sigma = \sqrt{U_{sq}/U_{euc} - (U/U_{euc})^2}$ – measures the "regularity" of the trajectory between two points.

A.2.2 Numerical results

We will not give any convergence results for the calculation of $\mathcal{C}$, $\mathcal{C}_{max}$ et $\mathcal{C}_\sigma$. We present results obtained for these measures for two different potential maps in dimension 2. A 4 neighbors system is used in all the experiments.

A.2.2.1 "Vessels" Potential

The first tested potential mimics a vessel (cf. section III.1). It is equal to 1 in all the space except in a shallow vessel in which its value is 1/4 – which favors front propagation (figure 1.2, *left*).
Figure 1.3 presents the obtained results. The three connectivity measures are minimal inside the vessel.
Figure 1.4 shows similar results for a noisy potential. Gaussian noise was added in the lower left part, and the vessels was cut in its right part (figure

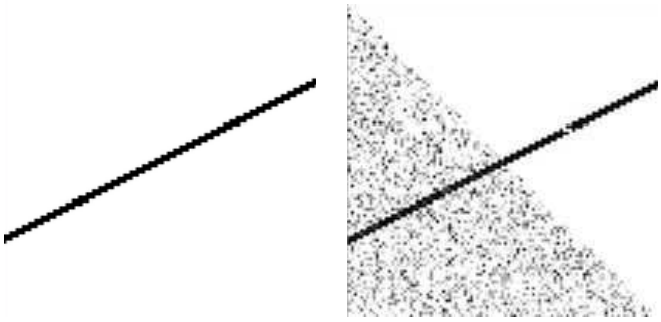


FIG. 1.2 – Potentials used for testing connectivity measures. Noise free (*left*) and noisy (*right*).

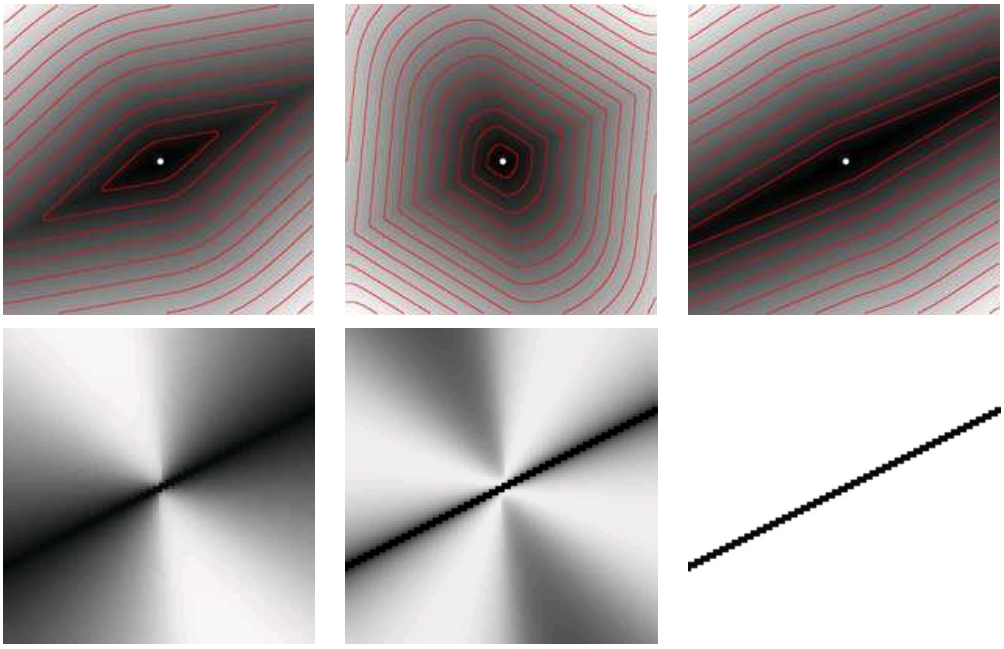


FIG. 1.3 – Connectivity measures for "vessel" potential. *Top, from left to right :* U, U_{euc} and U_{sq}. *Bottom, from left to right :* the extremal intensity values being given, $\mathcal{C}$ (0.2, 1), $\mathcal{C}_\sigma$ (0, 0.4), $\mathcal{C}_{max}$ (0.2, 1).

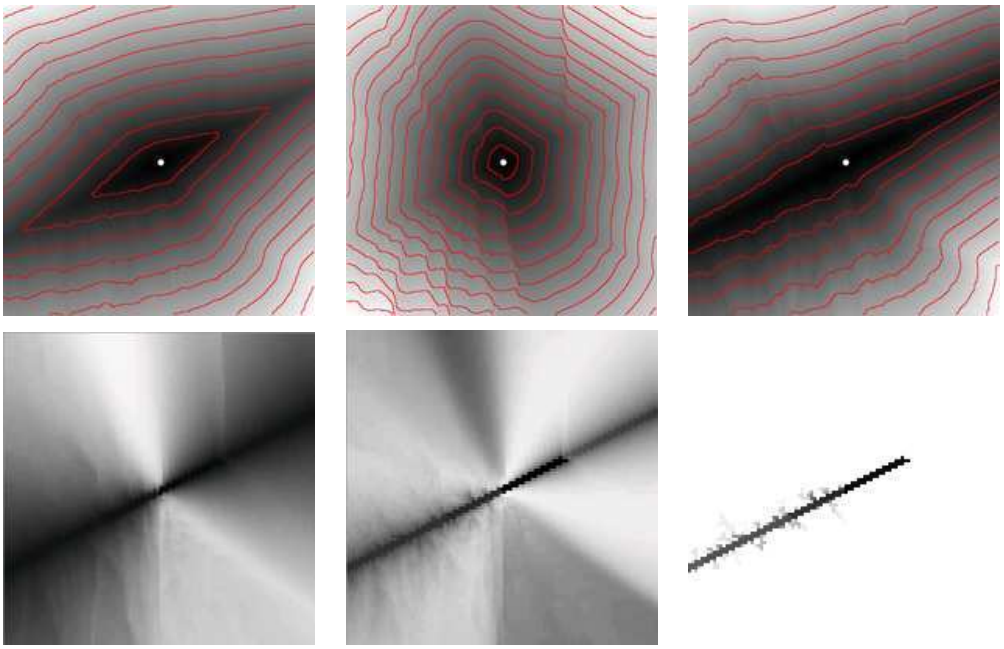


FIG. 1.4 – Connectivity measures for noisy "vessel" potential. *Top, from left to right :* U, U_{euc} and U_{sq}. *Bottom, from left to right :* the extremal intensity values being given, $\mathcal{C}$ (0.2, 1), $\mathcal{C}_\sigma$ (0, 0.4), $\mathcal{C}_{max}$ (0.2, 1).

1.2, *right*). $\mathcal{C}_\sigma$ et $\mathcal{C}_{max}$ seem to be more sensitive to noise than $\mathcal{C}$. These to measures are also more disturbed by the delete piece of vessel.

A.2.2.2 Anisotropic Potential

The second potential is a uniform anisotropic potential. The tensor is alogned with the axis, and horizontal speed is twice as much as horizontal vertical speed. Figure 1.5 shows some results for this potential. $\mathcal{C}$ et $\mathcal{C}_{max}$ exhibit the expected behavior – i.e. they are smaller in the horizontal direction. Since the shortest paths are straight lines in this context, the expected value for $\mathcal{C}_\sigma$ is 0 at any point. In this experiments, we found exact values in this axis directions. In other direction, they range between 0 and 0.2, with maximum around the starting point.

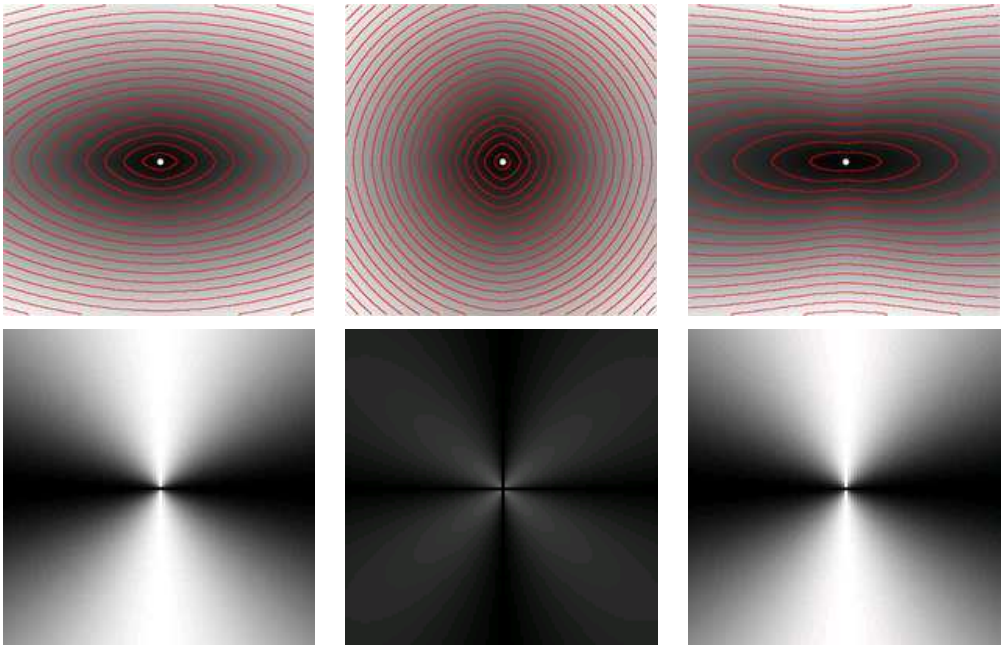


FIG. 1.5 – Connectivity measures for anisotropic potential. *Top, from left to right :* U, U_{euc} and U_{sq}. *Bottom, from left to right :* the extremal intensity values being given, $\mathcal{C}$ (0.5, 1), $\mathcal{C}_\sigma$ (0, 0.5) – maximale value in the plane is lower than 0.2, $\mathcal{C}_{max}$ (0.5, 1).

Annexe B

Electrodes registration in EEG using discrete optimization

This appendix has been adapted from the research report [140] corresponding to a work published in [166]. It is an early work in this thesis, independant from the rest of the presented methods. However, since the algorithm described here is currently used by EEG experimenters, and since developping a full framework for the localization of EEG electrodes from pictures would be of high interest, we believe it is interesting to reproduce this appendix as a reference.

B.1 Introduction

Electroencephalography (EEG) is a widely used method for both clinical and research purposes. Clinically, it is used e.g. to monitor and locate epilepsy, or to characterize neurological disorders such as sleeping or eating disorders and troubles related to multiple sclerosis. Its main advantages are its price compared to magnetoencephalography (MEG), and its very good time resolution compared e.g. to fMRI. Conventionally, EEG readings were directly used to investigate brain activity from the evolution of the topographies on the scalp. Nowadays, it is also possible to reconstruct the brain sources that gave rise to such measurements, solving a so-called inverse problem. To this purpose, it is necessary to find the electrode positions and to relate them to

the head geometry recovered from an anatomic MRI. Current techniques to do so are slow, tedious, error prone (they require to acquire each of the electrodes in a given order with a device providing 3D coordinates[106]) and/or quite expensive (a specialized system of cameras is used to track and label the electrodes[175]). Our goal is to provide a cheap and easy system for electrode localization based on computer vision techniques.
In modern EEG systems, the electrodes (64, 128 or even 256) are organized on a cap that is placed on the head. system, electrodes, obtain such a once to obtain used as a some roots between the and those multiple pictures of the head wearing the cap from various positions. As a preliminary step, electrodes are localized and their 3D positions are computed from the images by self-calibration (a technique that recovers the cameras' positions from the image information [59]) and triangulation. These are standard techniques that can provide 3D point coordinates with a quite good accuracy. Remains the problem of electrode identification which labels each 3D position with the name of the corresponding electrode. Finding a solution to this last problem is the focus of this paper. Note, that a good labeling software can also improve current systems by removing acquisition constraints (such as the recording of the electrodes in a given order) and by providing better user interfaces.
We propose a method that recovers this labeling from just a few (two or three) manually annotated electrodes. The only prior is a reference, subject independent, 3D model of the cap. Our framework is based on combinatorial optimization (namely on an extension of the Loopy Belief Propagation algorithm[148]) and is robust to soft deformations of the cap caused both by sliding effects and by the variability in subjects' head geometry.

B.2 Problem definition

The inputs of our method consist in :

- a template EEG cap model providing labeled electrodes, along with their 3D positions (in fact, as we will explain further, an important feature of our method is that only the distances between close electrodes are used). $\mathcal{L}$ will denote the set of labels (e.g. $\mathcal{L} = \{Fpz, Oz, \cdots\}$), and $C = \{C_l,\ l \in \mathcal{L}\}$ will be their corresponding 3D positions. C_l could be for example the average

position of electrode l among a variety of prior measures. However, in our experiments, it was just estimated on one reference acquisition.

- the measured 3D positions of the electrodes to label, obtained by 3D reconstruction from images. We will denote by $M = \{M_i,\ i \in [1..n]\}$, these n 3D points.

The output will be a labeling of the electrodes, i.e. a mapping φ from $[1..n]$ to $\mathcal{L}$. Note that n could be less than the total number $|\mathcal{L}|$ of electrodes in cases where some electrodes are of the cap are not used.

B.3 Motivation

In this section, we discuss other possible approaches for the electrode labeling problem. As it will be detailed in section B.6, we have tried some of these methods without any success. This will motivate our energy-based combinatorial approach. A simple method could consist in a 3D registration step, followed by a nearest-neighbor labeling. Let T be a transformation that sends M into the spatial referential of C. A straight labeling could be :

$$\varphi(i) = \arg\min_{l \in \mathcal{L}} d(C_l, T(M_i))$$

where $d(A, B)$ denotes the Euclidean distance between points A and B. Actually, we first tested two direct ways of obtaining an affine transformation T :

- *moment-based affine registration* : in this case, we computed first and second order moments of the sets of points M and C and choose T as an affine transformation which superimposes these moments.
- *4 points manual registration* : here, we manually labeled 4 particular electrodes in M and took for T the affine transformation which exactly sends these 4 electrodes to the corresponding positions in C.

As explained in section B.6, we observed that these two approaches give very bad average results. One could argue that this might be caused by the quality of the registration. A solution could be to use more optimal affine registration methods, like Iterative Closest Points[222, 22]. Yet, a close look at what caused bad labeling in our experiments, reveals that this would not improve the results. The main reasons are indeed that (i) the subject whose

EEG has to be labeled does not have the same head measurements than the template, and moreover that (ii) the cap is a soft structure that shifts and twists from one experiment to another.
It is clear that only a non-rigid registration could send M close to C. However, modeling the problem in term of space deformation is not suitable. For instance, a Thin-Plate Spline[27, 79] based algorithm would not be adapted. Actually, a more suitable framework could be a deformable shape matching one. We could see our problem as a shape registration one, based on shape deformation and *intrinsic* shape properties[183], rather than on deforming the ambient space in order to make the shapes match. Because of the topology of the electrodes on the cap, relations between points are also of importance. In that sense, our problem is close to the one investigated by Coughlan et al. [42, 5], which they solve recovering both deformations and soft correspondences between two shapes. Yet, in our case, we see two main differences : (i) labeling, rather than shape matching, is the key issue, and (ii) enforcing relational constraints between points are more important than regularizing deformations. For these reasons, we propose a method based on optimal labeling for which the only (soft) constraints are the distances between nearby points, without modeling any deformation.
In the remaining of the article, we first state our model and the associated energy ; we then discuss our choice for an energy minimization algorithm. Finally, we validate our method giving qualitative and quantitative results on real experiments.

B.4 Proposed framework

The complete pipeline of our system is depicted figure 2.1. As we already explained, we do not consider here the 3D reconstruction step, but only the labeling one. From the measured data M, we construct an undirected graph $G = (V, E)$, where $V = [1..n]$ is the set of vertices and E a certain set of edges which codes the relations between nearby electrodes. As it will become clear in the following, the choice of E will tune the rigidity of the set of points M. Practically, the symmetric k-nearest neighbors or all the neighbors closer than a certain typical distance, are two valid choices. Given an edge $e = (i, j) \in E$

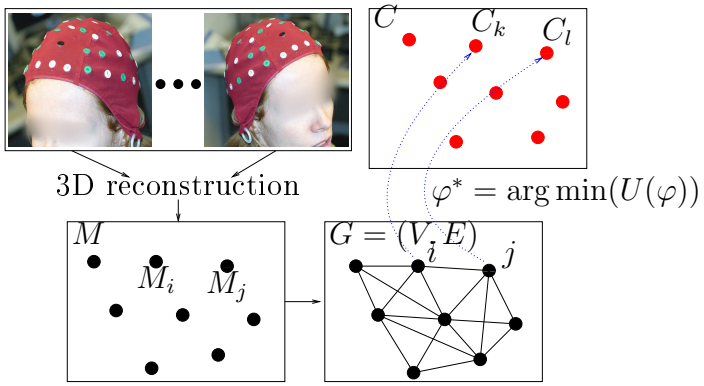


FIG. 2.1 – Complete pipeline : we obtain 3D positions M (bottom left) by reconstruction from several (usually 6) pictures (top left). A graph G then is constructed from these positions (bottom right). Considering a template cap and associated positions C (top right), we label the measured electrodes by estimating $\varphi^* = \arg\min(U(\varphi))$. In this example, $\varphi(i) = k$, $\varphi(j) = l$.

for $i \in V$ and $j \in V$, we denote by $d_{ij} = d(M_i, M_j)$ the distance between points M_i and M_j in the measured data and by $\tilde{d}_{ij} = d(C_{\varphi(i)}, C_{\varphi(j)})$ the reference distance between the electrodes $\varphi(i)$ and $\varphi(j)$. In order to preserve in a soft way the local structure of the cap, we propose to simply minimize the following energy :

$$U(\varphi) = \sum_{(i,j)\in E} \rho(d_{ij}, \tilde{d}_{ij}) \tag{B.4.1}$$

where ρ is a cost-function which penalizes differences between the observed and template distances. Note that, whereas the global one-to-one character of φ is not explicitly enforced by this model, the local rigidity-like constraints enforce it. Graph rigidity theory is a very complex domain (see for example [23] as an introduction), beyond the purpose of this article.
Following the classical framework of Markov Random Fields (MRF) [125, 21, 70], this can be rewritten as maximizing the following function :

$$P(\varphi) = exp(-U(\varphi)) = \prod_{(i,j)\in E} exp(-\rho(d_{ij}, \tilde{d}_{ij})) = \prod_{(i,j)\in E} \Psi_{i,j}(\varphi(i), \varphi(j)) \tag{B.4.2}$$

Normalizing P by dividing by the sum over all the possible mappings φ,

yields a *Gibbs distribution* over a MRF derived from graph G with $\mathcal{L}$ as the set of possible labels each vertex. The problem is thus reduced to the classical case of finding a *Maximum A Posteriori* (MAP) configuration of a Gibbs distribution :

$$p(\varphi) = \frac{1}{K} \prod_{i \in V} \psi_i(\varphi(i)) \prod_{(i,j) \in E} \psi_{i,j}(\varphi(i), \varphi(j)) \tag{B.4.3}$$

where K is a normalizing constant and $\psi_i(\varphi(i)) = 1$ in our case.

B.5 Energy minimization

The problem of finding a MAP configuration of a Gibbs distribution being NP-complete [102], we cannot expect to get an algorithm that optimally solves every instance of the problem. Since the seminal work of Geman & Geman [70], who proposed an algorithm that warrants the probabilistic convergence toward the optimal solution – however with an unreasonable run-time – several methods have been investigated to maximize general distributions like (B.4.3). Among these, minimal-cut based methods (often referred to as *GraphCuts*), introduced in computer vision and image processing by [74], has received many attention (see [85, 29]). These methods can achieve global optimization for a restricted class of energies[84]. For more general energies, approximations were proposed [169]. As we experimented[140], these approximations fail to recover a correct labeling in our problem, which belongs to a class of multilabel problems that are not easily tackled by *GraphCuts*.

As a consequence, we opted for a completely different but widely spread algorithm, namely *Belief Propagation* (BP), and more precisely for its variant adapted to graphs : *Loopy Belief Propagation* (LBP). Please see [60] for a recent reference. Briefly, it consists in propagating information through the edges of the graph : each node i sends *messages* to its neighbors k, measuring the estimated label of k from its own point of view. Messages are passed between nodes iteratively until a convergence criterion is satisfied. This algorithm is neither guaranteed to converge nor to converge to an optimal solution. However, it behaves well in a large variety of early vision problems. Empirical and theoretical convergence of this family of methods were studied

for instance in [141, 216].
Actually, we designed for this work an original and faster version of LBP. It is an improved version of LBP based on the idea of [103].
Let us first explain classical LBP algorithm.

B.5.1 LBP

Loopy Belief Propagation (LBP) algorithm [60] is a widely used method to find approximate solutions to the MAP problem when the sub-modularity condition is not fulfilled. It consists in propagating information through the vertices of the mesh seen as a graph : roughly speaking, each node i sends *messages* to its neighbors k, measuring the estimated label of k from the point of view of i.
The LBP algorithm is derived from an exact algorithm working on trees called *Belief Propagation* (BP) or *Max-Product* algorithm [149]. In the original BP, messages measuring belief in a local labeling propagate from the leafs to the root of the tree. Then a backward pass is computed in which label that maximizes the belief is chosen at each node, knowing the label of the father.
Let us introduce some notations :
r the root of the tree, s the application that maps a node to its sons and f the application that maps a node to its father. L is the set of the leafs of the tree.
$m_{i\to j}$ will denote the message passed by node V_i to node V_j. $m_{i\to j}(l)$ is a measure of how confident node V_i is that node V_j is given the l label, i.e. $\varphi(V_j) = l$.
We denote by $b_i(l_i, l_{f(i)}) = \psi_i(l_i)\psi_{f(i)i}(l_{f(i)}, l_i)\prod_{j\in s(i)} m_{j\to i}(l_i)$ for $l_i \in C$ and $l_{f(i)} \in C$ the joint belief that node $V_{f(i)}$ is assigned label $l_{f(i)}$ and node V_i is assigned label l_i.
The BP algorithm is described in algorithm 8.

When the graph is not a tree, the ordered treatment of BP is impossible to apply. However, disregarding the relation of paternity of the nodes, it is still possible to pass messages from nodes to nodes in the graph. A belief can also be computed the same way as for BP. The idea of LBP is then to apply the message passing simultaneously or sequentially to all the neighboring

Algorithm 8 Belief Propagation

$K = L$
Forward pass
while $V_i \in K$ **do**
  remove V_i from K
  compute $m_{i \to f(i)}(l_{f(i)}) = max_{l_i}(b_i(l_i, l_{f(i)}))$ for all $l_{f(i)} \in C$.
  compute $\delta(l_{f(i)}) = argmax_{l_i}(b_i(l_i, l_{f(i)}))$ for all $l_{f(i)} \in C$.
  if all sons of $V_{f(i)}$ have been treated, add $V_{f(i)}$ to K
end while
Backward pass
$\bar{\varphi}_r = \delta_r$
$K = s(r)$
while $V_i \in K$ **do**
  remove V_i from K
  compute $\bar{\varphi}_i = \delta_i(\bar{\varphi}_{f(i)})$
  $K = K \cup s(V_i)$
end while
return $\bar{\varphi}$

nodes of the graph. A stopping criterion is then to be defined - usually a convergence criterion or a fixed number of iterations.
Let us adapt slightly the notations and denote by $N(i)$ the set of neighbor nodes of V_i.
$m_{i \to j}^t$ is the message passed by node V_i to node V_j at time t. Let $b_i^t(l_i, l_j) = \psi_i(l_i)\psi_{ij}(l_i, l_j) \prod_{k \in N(i), k \neq j} m_{k \to i}^t(l_i)$ for $(l_i, l_j) \in C^2$ be the joint belief for neighbor nodes V_i and V_j. Finally, let $b_i^t(l_i) = \psi_i(l_i) \prod_{k \in N(i)} m_{k \to i}^t(l_i)$ be the *belief vector* at node V_i and time t (taking into account all the neighbors of node V_i).
This leads to algorithm 9.

This algorithm is neither guaranteed to converge nor to converge to an optimal solution. However, it behaves well in a large variety of early vision problems. Empirical and theoretical convergence of this kind of methods were studied for instance in [141] and [216].
Notice that the complexity of one step of this algorithm is basically $|(C)|^2|E|$

Algorithm 9 Loopy Belief Propagation

set $m^0_{p\to q}(l_q) = 1$ for all $(p,q) \in E$.
for $t = 1, t \leq T, t++$ **do**
  for all (i,j) in V **do**
    $m^t_{i\to j}(l_j) = max_{l_i \in C}(b^{t-1}_i(l_i, l_j))$
  end for
end for
return $\bar{\varphi}_i = argmax_{l_i \in C} b^T_i(l_i)$

where $|E|$ is the number of edges of the graph.

B.5.2 Improving belief propagation

Several methods have been proposed to improve both the convergence and the quality of results obtained by LBP algorithm. [213] proposed a slightly different algorithm based on a different theoretical framework with interesting convergence properties. More recently, [103] proposed an interesting modification of LBP based on label pruning according to current belief at each node, and on a choice of a priority order for covering all nodes. But, their method show a greedy behavior, since a label cannot appear again once it has been pruned.

A new intermediate and simpler version of LBP based on label pruning is proposed here. It is based on the idea that if a label is very unlikely for a given vertex, it ought to be useless to use this label for the calculation of the outgoing messages for this vertex. Hence, after each step, the *belief vector* $b^t_i(l_i)$ is computed for each node as well as its maximum and minimum values M^t_i and m^t_i. Then, each label with a belief lower than the geometric mean g^t_i of m^t_i and M^t_i is declared *inactive* for the next iteration only, e.g. it won't be considered as a candidate label in computing outgoing messages toward the neighbors of V_i (notice that the choice of the mean is somewhat arbitrary. It should be adapted to the structure of the belief vector. For our application, we didn't notice effect of the choice of a threshold between 0.5 and 0.8 over speed nor quality of results).

Let us denote by Act^t_i the set of active labels of V_i computed at iteration t.

Our method is described algorithm 10.

Algorithm 10 Fast Loopy Belief Propagation

set $Act_i^0 = C$ for all $V_i \in V$
set $m_{p\to q}^0(l_q) = 1$ for all $(p,q) \in E$.
for $t = 1, t \leq T, t++$ **do**
  for all (i,j) in V **do**
    $m_{i\to j}^t(l_j) = max_{l_i \in Act_i^{t-1}}(b_i^{t-1}(l_i, l_j))$
    set $Act_i^t = \{l_i : l_i \geq g_i^t\}$
  end for
end for
return $\bar{\varphi}_i = argmax_{l_i \in C} b_i^T(l_i)$

The $|C|^2$ factor for each edge in the complexity for one step is then replaced by a $|C||C'|$ where $|C'|$ is the number of *active* labels of the original node.

B.6 Experiments

We used 6 sets of 63 electrodes. Each set consists in 63 estimated three dimensional points, acquired on different subjects with the same EEG cap and manually labeled. To test our algorithm as extensively as possible, we ran the algorithm on each set, taking successively each of the other sets as a reference. We hence simulated 30 different pairs (M, C). At least one electrode in M was manually labeled (see further).

E was chosen the following way : we first estimated a typical neighbor distance by computing the maximum of the nearest neighbor distance for all electrodes in M, and then considered as belonging to E, every pair of distinct electrodes within less than three times this distance. In order to accelerate and enforce convergence, we used the three following technical tricks :

- we used our modified LBP algorithm
- we added a classical momentum term [141]
- denoting by V_f the subset of V of the manually labeled electrodes, we added the set of edges $V_f \times V$ to E, allowing accurate information to propagate quickly in the graph.

Although non indispensable, this led to a mean running time of less than 11s on a standard 3GHz PC and to a smaller number of non converging optimization.

The cost-function ρ was of the form $\rho(x,y) = \frac{x}{y+\epsilon} + \frac{y}{x+\epsilon}$ where ϵ is a small positive constant. We did not notice sensitivity with respect to this choice, as far as the following key conditions are fulfilled : (i) penalizing differences between x and y and (ii) penalizing small values of x or y. This latest condition enforces (yet does not warrant) a one-to-one mapping φ.

Different experiments where carried out. First, the prior consisted in manually labeling electrodes Fpz, Oz, and $T8$. In that case, our method recovers all the electrodes, which was, as expected, not at all the case with an affine registration+nearest neighbor approach (see figure 2.2). Actually, we observed that labeling $(Oz, T8)$ seems sufficient. Yet, without any further data, we do not consider that labeling two electrodes only is reliable. Figure 2.4 shows a result on a case where affine registration does not work and the final 3D reconstruction with our method.

To demonstrate the robustness of our algorithm, we also tested hundreds of other conditions, in which 1, 2 or 3 randomly chosen electrodes were "manually" labeled. Non-convergence was only observed for non reasonable choices of "manually" labeled electrodes : indeed, if they are chosen on the sagittal medium line, there is an undetermination due to the left-right symmetry of the cap. This does not occur when the electrodes are set by a human operator. The classification error rates are low (see figure 2.2 again) but not negligible. This makes us plead for a manual labeling of two or three fixed and easy to identify electrodes, e.g. $(Fpz, Oz, T8)$.

Finally, we also successfully tested cases for which $n < |\mathcal{L}|$, i.e. when some electrodes are missing : if a few electrodes were forgotten in the 3D reconstruction process, our algorithm should still be able to label the detected ones. This should allow us to find which electrodes were forgotten, to compute their approximate 3D position from the template cap model and to use this information to detect them back in the pictures. To carry our experiments, we removed randomly from 1 to 10 electrodes in the data sets to be labeled. Labelisation was performed using the $(Fpz, Oz, T8)$ prior as explained above. Results are synthetized figure 2.3.

	NC	misclassified labels
Affine registration (moment based)	-	48.7%
Affine registration (4 manual points)	-	21.3%
Our method - $(Fpz, Oz, T8)$ manually labeled	0%	0%
Our method - $(Oz, T8)$ manually labeled	0%	0%
Our method - 3 random electrodes labeled	0%	0.03%
Our method - 2 random electrodes labeled	0.3%	0.2%
Our method - 1 random electrode labeled	4.2%	3,7%

FIG. 2.2 – Classification errors. NC gives the percentage of instances of the problem for which our method did not converge. Misclassified labels percentages are estimated only when convergence occurs.

missing electrodes	mislabeled electrodes
1	0%
2	0%
3	0.01%
4	0.02%
5	0.02%
6	0.04%
7	0.04%
8	0.3%
9	1.1%
10	1.1%

FIG. 2.3 – Results with missing electrodes.

B.7 Discussion

Experiments show that our framework leads to fast, accurate and robust labeling on a variety of data sets. We consider providing on the WEB in a near future an complete pipeline including our algorithm - ranging from 3D reconstruction of electrodes to their labeling. Such a system would only require a standard digital camera and would imply minimal user interaction

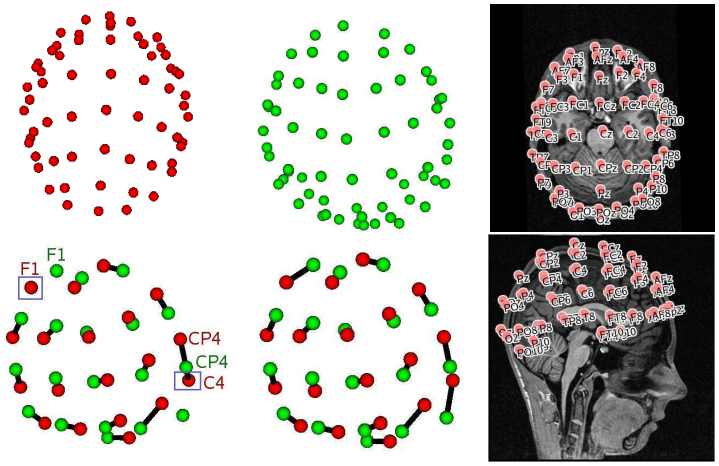


FIG. 2.4 – A sample result. M is in red and C in green. Top left : 63 estimated 3D electrodes positions. Top center : reference. Bottom left : subset of a labeling with the moment based algorithm ; C4 is wrongly labeled CP4, and F1 is labeled F3 (not shown). Bottom center : a subset of correct correspondences retrieved by our algorithm. Top and bottom right : full labeling retrieved by our algorithm, superimposed with anatomical MRI

(manually labeling three electrodes).

Note that the flexibility of our MRF formulation allows different priors. We plan for instance to use the color of electrodes on the images as a further prior for labeling. This could lead to a fully automated system, where no user interaction would be required.

Bibliographie

[1] Shimbel A. Structure in communication nets. In Polytechnic Press of the Polytechnic institute of Brooklyn, editor, *Proceedings of the Symposium on Information Networks*, pages 199–203, 1954. (Page 40.)

[2] Varol Akman. *Unobstructed shortest paths in polyhedral environments*. Springer-Verlag New York, Inc., New York, NY, USA, 1987. (Page 32.)

[3] Schrijver Alexander. *Combinatorial Optimization*. Springer-Verlag Berlin Heidelberg New York, 2003. (Page 17.)

[4] A. Anwander, M. Tittgemeyer, D. Y. von Cramon, A. D. Friederici, and T. R. Knosche. Connectivity-based parcellation of broca's area. *Cerebral Cortex*, 17(4) :816–825, 2007. (Page 164.)

[5] A.Rangarajan, J.M. Coughlan, and A.L. Yuille. A bayesian network framework for relational shape matching. In *9th IEEE ICCV*, pages 671–678, 2003. (Page 186.)

[6] R. Ardon and L. Cohen. Fast constrained surface extraction by minimal paths. In *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 233–244, 2003. (Page 33.)

[7] Roberto Ardon and Laurent D. Cohen. Fast constrained surface extraction by minimal paths. *Int. J. Comput. Vision*, 69(1) :127–136, 2006. (Page 33.)

[8] Stephen Aylward, Stephen Pizer, David Eberly, and Elizabeth Bullitt. Intensity ridge and widths for tubular object segmentation and description. In *Proceedings of the 1996 Workshop of MMBIA '96*, page 131, Washington, DC, USA, 1996. IEEE Computer Society. (Page 104.)

[9] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analalysis*, 4 :271–283, 1991. (Page 58.)

[10] Alberto Bartesaghi and Guillermo Sapiro. A system for the generation of curves on 3d brain images. *Human Brain Mapping*, 14(1), 2001. (Page 33.)

[11] Peter J. Basser, James Mattiello, and Denis Lebihan. Mr diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66 :259–267, 1994. (Pages 135 and 155.)

[12] P.J. Basser, J. Mattiello, and D. LeBihan. Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance B*, 103(3) :247–254, 1994. (Page 156.)

[13] Pierre-Louis Bazin and Dzung L. Pham. Topology correction of segmented medical images using a fast marching algorithm. *Comput. Methods Prog. Biomed.*, 88(2) :182–190, 2007. (Page 26.)

[14] T. E. J. Behrens, H. Johansen-Berg, S. Jbabdi, M. F. S. Rushworth, and M. W. Woolrich. Probabilistic diffusion tractography with multiple fibre orientations. what can we gain ? *NeuroImage*, 34(1) :144–155, 2007. (Page 156.)

[15] F. Benmansour, S. Bonneau, and L. Cohen. Finding a closed boundary by growing minimal paths from a single point on 2d or 3d images. In *IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*, page to appear, 2007. (Page 145.)

[16] Fethallah Benmansour and Laurent D. Cohen. Tubular anisotropy for 3d vessels segmentation. In *SSVM*, pages 14–25, 2009. (Page 106.)

[17] Fethallah Benmansour, Laurent D. Cohen, Max W.K. Law, and Albert C.S. Chung. Tubular anisotropy for 2d vessels segmentation. In *CVPR*, June 2009. (Page 106.)

[18] H. E. Bennink, Hans C. van Assen, Geert J. Streekstra, Renester Wee, J. A. E. Spaan, and Bart M. ter Haar Romeny. A novel 3d multi-scale lineness filter for vessel detection. In Nicholas Ayache, Sébastien Ourselin, and Anthony J. Maeder, editors, *Proceedings of MICCAI (2)*, volume 4792 of *Lecture Notes in Computer Science*, pages 436–443. Springer, 2007. (Page 104.)

[19] C. Berge. *Théorie des graphes et ses applications.* Collection Univesitaire des Mathématiques, Dunod, Paris, 1958. (Page 41.)

[20] Ørjan Bergmann, Gordon Kindlmann, Sharon Peled, and Carl-Fredrik Westin. Two-tensor fiber tractography. In *ISBI*, pages 796–799, Arlington, Virginia, USA, 2007. (Page 156.)

[21] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal Royal Statis. Soc.*, B-148 :192–236, 1974. (Page 187.)

[22] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2) :239–256, 1992. (Page 185.)

[23] B.Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1) :65–84, 1992. (Page 187.)

[24] M. Bicego, S. Dalfini, G. Vernazza, and P. Murino. Automatic road extraction from aerial images by probabilistic contour tracking. In *ICIP03*, pages III : 585–588, 2003. (Page 105.)

[25] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*, chapter Voronoi diagrams : Euclidian metric, Delaunay complexes, pages 435–443. Cambridge University Press, 1998. (Page 138.)

[26] T. Bonhoeffer and A. Grinvald. *Brain Mapping : the Methods (Toga, AW, Mazziotta, JC Eds.)*, chapter Optical imaging based on intrinsic signals : the methodology., pages 55 – 226 97. Academic Press, California, 1996. (Page 136.)

[27] F.L. Bookstein. Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI*, 11(6) :567–585, 1989. (Page 186.)

[28] Sylvain Bouix, Kaleem Siddiqi, and Allen Tannenbaum. Flux driven automatic centerline extraction. *Medical Image Analysis*, 9(3) :209–221, 2005. (Page 106.)

[29] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR '98*, page 648, Washington, DC, USA, 1998. IEEE. (Page 188.)

[30] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 2008. (Pages 25, 39, and 80.)

[31] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Weighted distance maps computation on parametric three-dimensional manifolds. *J. Comput. Phys.*, 225(1) :771–784, 2007. (Page 99.)

[32] RB Buxton, K Uludag, DJ Dubowitz, and TT Liu. Modeling the hemodynamic response to brain activation. *Neuroimage*, 23(Suppl 1) :S220–S233, 2004. (Page 136.)

[33] J. Canny. A computational approach to edge detection. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 8(6) :679–698, November 1986. (Page 104.)

[34] K. V. Chandrinos, M. Pilu, R. B. Fisher, and P. E. Trahanias. Image processing techniques for the quantification of atherosclerotic changes. In *Mediterranian Conf. Medical and Bio. Eng. and Computing.*, 1998. (Page 104.)

[35] Thitiporn Chanwimaluang and Guoliang Fan. An efficient algorithm for extraction of anatomical structures in retinal images. In *ICIP*, volume 1, pages 1093–1096, 2003. (Page 104.)

[36] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging*, 8 :263–269, 1989. (Pages 104 and 110.)

[37] L. Cohen and R. Kimmel. Fast marching the global minimum of active contours, 1996. (Page 105.)

[38] Laurent Cohen and Ron Kimmel. Global minimum for active contour models : A minimal path approach. *International Journal of Computer Vision*, 24 :57–78, 1997. (Pages 28, 116, 117, 122, and 132.)

[39] L.D. Cohen. *Minimal Paths and Fast Marching Methods for Image Analysis*. Nikos Paragios and Yunmei Chen, 2005. (Pages 116, 117, 122, and 132.)

[40] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to algorithms*. MIT Press and McGraw-Hill, 1990. (Page 22.)

[41] D. J. Cornforth, H. F. Jelinek, J. J. G. Leandro, J. V. B Soares, R. M. Cesar-Jr., M. J. Cree, P. Mitchell, and T. R. J. Bossomaier. Development of retinal blood vessel segmentation methodology using wavelet transforms for assessment of diabetic retinopathy. *Complexity International*, 11, 2005. (Pages 104 and 105.)

[42] J.M. Coughlan and S.J. Ferreira. Finding deformable shapes using loopy belief propagation. In *7th ECCV*, pages 453–468, 2002. (Page 186.)

[43] Michael G. Crandall, Hitoshi Ishii, and Pierre L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1) :1–67, 1992. (Page 36.)

[44] Creativision. Human machine intergation for vessel segmentation, 2008. `http ://retina.incubadora.fapesp.br/portal/references/references`. (Page 104.)

[45] M.J. Cree, D.J. Cornforth, and H.F. Jelinek. Vessel segmentation and tracking using a two-dimensional model. In *Proceedings of the Image and Vision Computing Conference, New Zealand 2005, University of Otago*, 2005. (Page 105.)

[46] Per-Erik Danielsson and Qingfen Lin. A modified fast marching method. In *SCIA*, pages 1154–1161, 2003. (Page 67.)

[47] Huang Davidand and Kaiser Peter K.and Lowder Careen Y.and Traboulsi Elias. *Retinal Imaging*. 2005. (Page 103.)

[48] Thomas Deneux. *Hemodynamic Models : Investigation and Application to Brain Imaging Analysis*. PhD thesis, École Polytechnique, 2006. (Page 136.)

[49] T. Deschamps and L.D. Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4) :281–299, December 2001. (Pages 28 and 105.)

[50] Thomas Deschamps and Laurent D. Cohen. Fast extraction of tubular and tree 3d surfaces with front propagation methods. In *In Proc. of 16th ICPR, 1 :731 ̆2013734*, pages 731–734, 2002. (Pages 66, 105, and 106.)

[51] M Descoteaux, R Deriche, T. R. Knösche, and A Anwander. Deterministic and probabilistic tractography based on complex fibre orientation distributions. *IEEE Transactions in Medical Imaging*, 28(2) :269–286, Febuary 2009. (Pages 156 and 164.)

[52] M. Descoteaux, R. Deriche, T.R. Knosche, and A. Anwander. Deterministic and probabilistic tractography based on complex fibre orientation distributions. *MedImg*, 28(2) :269–286, February 2009. (Page 168.)

[53] Maxime Descoteaux, Louis Collins, and Kaleem Siddiqi. Geometric flows for segmenting vasculature in mri : Theory and validation. In *In Medical Imaging Computing and Computer-Assisted Intervention*, pages 500–507, 2004. (Pages 104 and 105.)

[54] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271, December 1959. (Page 41.)

[55] L Euler. *De linea brevissima in superficie quacunque duo qualibe puncta jungente*. 1732. (Page 32.)

[56] L. Evans. *Partial Differential Equations*. American Mathematical Society, 2002. (Page 38.)

[57] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms : A comparative survey. *ACM Comput. Surv.*, 40(1) :1–44, 2008. (Pages 26 and 99.)

[58] B. Fang, W. Hsu, and M.L. Lee. Reconstruction of vascular structures in retinal images. In *ICIP*, volume 2, pages 157–160, 2003. (Page 104.)

[59] O. Faugeras, Q.T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001. (Page 184.)

[60] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision, 2004. (Pages 188 and 189.)

[61] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. pages 741–752, 1987. (Page 104.)

[62] Robert W. Floyd. Algorithm 245 : Treesort. *Communications of the ACM*, 7(12) :701, 1964. (Page 45.)

[63] Ro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. Multiscale vessel enhancement filtering. In *Proceedings of MICCAI*, volume 1496, pages 130–137. Springer-Verlag, 1998. (Page 104.)

[64] Max Frenkel and Ronen Basri. Curve matching using the fast marching method. In A. Rangarajan, editor, *proceedings of EMMCVPR*, 2003. (Page 30.)

[65] P. Fua and Y. G. Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3 :45–56, 1990. (Page 28.)

[66] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian Geometry*. Springer, Berlin, 1993. (Pages 32, 33, and 34.)

[67] X. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. Thom. A method of vessel tracking for vessel diameter measurement on retinal images. In *ICIP01*, volume 2, pages 881–884, 2001. (Page 105.)

[68] S Garrido, L Moreno, M Abderrahim, and F Martin. Path planning for mobile robot navigation using voronoi diagram and fast marching. *Intelligent Robots and Systems*, october 2006. (Page 27.)

[69] S. Garrido, L. Moreno, and D. Blanco. Exploration of a cluttered environment using voronoi transform and fast marching. *Robot. Auton. Syst.*, 56(12) :1069–1081, 2008. (Page 27.)

[70] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6(6) :721–741, Nov. 1984. (Pages 187 and 188.)

[71] Michael Goldbaum, Saied Moezzi, Adam Taylor, Shankar Chatterjee, Jeff Boyd, Edward Hunter, and Ramesh Jain. Automated diagnosis and image understanding with object extraction, object classification, and inferencing in retinal images. In *in*

Image Processing, proceedings of IEEE International Conference on Image Processing, volume 3, pages 695–698, 1996. (Page 104.)

[72] G. Gonzalez, F. Fleuret, and P. Fua. Learning rotational features for filament detection. In *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. (Page 104.)

[73] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992. (Page 112.)

[74] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51 :271–279, 1989. (Page 188.)

[75] A. Grinvald, T. Bonhoeffer, I. Vanzetta, A. Pollack, E. Aloni, R. Ofri, and D. Nelson. High-resolution functional optical imaging : from the neocortex to the eye. *Ophthalmol. Clin. North. Am.*, 17(1) :53–69, 2004. (Page 136.)

[76] D. Guo and P. Richardson. Automatic vessel extraction from angiogram images. In *Computers in Cardiology*, pages 441–444, 1998. (Page 104.)

[77] William E. Hart, Michael Goldbaum, Paul Kube, and Mark R. Nelson. Automated measurement of retinal vascular tortuosity. In *Proceedings of AMIA Fall Conference*, pages 459–463, 1997. (Page 106.)

[78] M.S. Hassouna and A.A. Farag. Robust skeletonization using the fast marching method. In *ICIP05*, pages I : 437–440, 2005. (Page 26.)

[79] H.Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *CVPR*, pages 2044–2051, 2000. (Page 186.)

[80] W. Higgins, W. Sypra, R. Karwoski, and E. Ritman. System for analyzing higresolution three-dimensional coronary angiograms. *IEEE Transactions on Medical Imaging*, 15 :377–385, 1996. (Page 104.)

[81] Jeffrey Ho, Jongwoo Lim, Ming-Hsuan Yang, and David J. Kriegman. Integrating surface normal vectors using fast marching method. In *9th European Conference on Computer Vision (ECCV 2006)*, pages 239–250, May 2006. (Page 30.)

[82] A. Hoover, V. Kouznetsova, , and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. In *Proceedings IEEE Transactions on Medical Imaging*, number 19 in 3, pages 203–210. IEEE Computer Society, 2000. (Page 104.)

[83] H. Hopf and W. Rinow. Ueber den begriff der vollstandigen differentialgeometrischen flachen. *Comm. Math. Helv.*, 3 :209–225, 1931. (Page 34.)

[84] H. Ishikawa. Exact optimization for markov random fields with convex priors, 2003. (Page 188.)

[85] H. Ishikawa and D. Geiger. Mapping image restoration to a graph problem, 1999. (Page 188.)

[86] K. M. Jansons and D. C. Alexander. Persistent angular structure : new insights fom diffusion magnetic resonance imaging data. *Inverse Problems*, 19 :1031–1046, 2003. (Pages 156 and 164.)

[87] S. Jbabdi, P. Bellec, R. Toro, J. Daunizeau, M. Pélégrini-Issac, and H. Benali. Accurate anisotropic fast marching for diffusion-based geodesic tractography. *Journal of Biomedical Imaging*, 2008(1) :1–12, 2008. (Pages 31 and 99.)

[88] S Jbabdi, P Bellec, R Toro, J Daunizeau, M Pelegrini-Issac, and H Benali. Accurate anisotropic fast marching for diffusion-based geodesic tractography. *International Journal of Biomedical Imaging*, 2008 :1–12, 2008. (Page 157.)

[89] H. F. Jelinek and R. M. Cesar-Jr. Segmentation of retinal fundus vasculature in nonmydriatic camera images using wavelets. In J. Suri and T. Laxminarayan, editors, *Angiography and Plaque Imaging : Advanced Segmentation Techniques*, pages 193–224. CRC Press, 2003. (Page 104.)

[90] Bing Jian and Baba C. Vemuri. A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted mri. *IEEE Transactions on Medical Imaging*, 26(11) :1464–1471, 2007. (Pages 156 and 164.)

[91] L. Jonasson, X. Bresson, P. Hagmann, J. Thiran, and V. Wedeen. Representing Diffusion MRI in 5D Simplifies Regularization and Segmentation of White Matter Tracts. *IEEE Transactions on Medical Imaging*, 26 :1547–1554, 11 2007. (Pages 107 and 159.)

[92] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. (Page 40.)

[93] Y. Kawata, N. Niki, and T. Kumazaki. An approach for detecting blood vessel diseases from cone-beam ct image. In *Proceedings of the International Conference on Image Processing*, volume 2, page 2500, Washington, DC, USA, 1995. IEEE Computer Society. (Page 104.)

[94] R. Kimmel and A. M. Bruckstein. Global shape from shading. *Computer Vision and Image Understanding*, 62(3) :360–369, November 1995. (Page 30.)

[95] R. Kimmel and A. M. Bruckstein. Regularized laplacian zero crossings as optimal edge integrators. *Int. J. Comput. Vision*, 53(3) :225–243, 2003. (Page 28.)

[96] R. Kimmel and J.A. Sethian. Fast marching methods for robotic navigation with constraints. Technical Report 669, CPAM,Univ. of California, Berkeley, 1996. (Page 27.)

[97] R. Kimmel and J.A. Sethian. Fast marching methods on triangulated domains. *PNAS*, 95(15) :8431–8435, July 1998. (Pages 80 and 99.)

[98] Ron Kimmel and Nahum Kiryati. Finding shortest paths on surfaces by fast global approximation and precise local refinement. *Journal of Pattern Recognition and Artificial Intelligence*, 10 :643–656, 1996. (Page 48.)

[99] Ron Kimmel and James A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14 :2001, 2001. (Pages 27, 30, 32, and 76.)

[100] C. Kirbas and F. Quek. A review of vessel extraction techniques and algorithms. Technical report, VisLab Wright State University, Dayton, Ohio, Nov 2000. (Page 104.)

[101] N. Kiryati and G. Szekely. Estimating shortest paths and minimal distances on digitized three-dimensional surfaces. *PR*, 26(11) :1623–1637, November 1993. (Page 48.)

[102] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts ? In *ECCV (3)*, pages 65–81, 2002. (Page 188.)

[103] N. Komodakis and G.Tziritas. Image completion using global optimization. In *CVPR '06*, Washington, DC, USA, 2006. IEEE Computer Society. (Pages 189 and 191.)

[104] Ender Konukoglu. *Modeling Glioma Growth and Personalizing Growth Models in Medical Images*. PhD thesis, University of Nice-Sophia Antipolis, 2009. (Page 99.)

[105] S.P. Kozaitis and R.H. Cofer. Lineal feature detection using multiresolution wavelet filters. *PhEngRS*, 71(6) :689–698, June 2005. (Page 104.)

[106] D. Kozinska and K. Nowinski. Automatic alignment of scalp electrode positions with head mrivolume and its evaluation. In *Engineering in Medicine and Biology, BMES/EMBS Conference*, Atlanta, Oct 1999. (Page 184.)

[107] B. W. Kreher, J. F. Schneider, J. Mader, E. Martin, Hennig J, and K.A. Il'yasov. Multitensor approach for analysis and tracking of complex fiber configurations. *Magnetic Resonance in Medicine*, 54 :1216–1225, 2005. (Page 156.)

[108] N Kriegeskorte and R Goebel. An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical mr volumes. *Neuroimage*, 14(2) :329–46, August 2001. (Page 26.)

[109] K K Kwong, J W Belliveau, D A Chesler, I E Goldberg, R M Weisskoff, B P Poncelet, D N Kennedy, B E Hoppel, M S Cohen, and R Turner. Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *PNAS*, 89(12) :5675–5679, June 1992. (Page 135.)

[110] Synge J. L. *Geometrical Optics. An introduction to Hamilton's method*. 37. Cambridge Tracts in Mathematics and Mathematical Physics, Cambridge, University Press, New York, 1937. (Page 28.)

[111] M. Lalonde, L. Gagnon, and M.-C. Boucher. Non-recursive paired tracking for vessel extraction from retinal images. In *Proceedings of the 31st International Symposium on Robotics*, pages 61–68, may 2000. (Page 105.)

[112] B. S. Y. Lam and H. Yan. A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields. *IEEE Transactions on Medical Imaging*, 27(2) :237–246, 2008. (Page 104.)

[113] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Also available at http ://planning.cs.uiuc.edu/. (Pages 22 and 27.)

[114] Max W. Law and Albert C. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *ECCV '08 : Proceedings of the 10th European Conference on Computer Vision*, pages 368–382, Berlin, Heidelberg, 2008. Springer-Verlag. (Page 104.)

[115] Max W.K. Law and Albert C.S. Chung. Vessel and intracranial aneurysm segmentation using multi-range filters and local variances. In Nicholas Ayache, Sébastien Ourselin, and Anthony Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, volume 4791 of *LNCS*, pages 866–874. Springer, 2007. (Page 104.)

[116] John M. Lee. *Riemannian Manifolds*. Springer-Verlag, 1980. (Page 33.)

[117] Lenglet. *Geometric and Variational Methods for DTI Processing*. PhD thesis, INRIA/Université de Nice-Sophia Antipolis, December 2006. (Page 168.)

[118] Christophe Lenglet, Emmanuel Prados, Jean-Philippe Pons, Rachid Deriche, and Olivier Faugeras. Brain Connectivity Mapping using Riemannian Geometry, Control Theory and PDEs. *SIAM Journal on Imaging Sciences*, 2 :285–322, 2009. (Page 168.)

[119] Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz. First annual report. Technical report, Investigation of Model Technique, Case Institute of Technology in Cleveland Ohio, 1957. (Page 41.)

[120] Hua Li and Anthony Yezzi. Vessels as 4d curves : Global minimal 4d paths to extract 3d tubular surfaces. In *CVPRW '06*, page 82, Washington, DC, USA, 2006. IEEE Computer Society. (Pages 105, 106, 107, 116, 117, 120, 122, and 132.)

[121] I. Liu and Y. Sun. Recursive tracking of vascular networks in angiograms based on the detection-deletion scheme. *IEEE Transactions on Medical Imaging*, 12(2) :334–341, 1993. (Page 105.)

[122] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003. (Page 137.)

[123] Ford L.R. Network flow theory. Technical report, RAND-Santa Monica, August 1956. (Page 40.)

[124] L.A. Lyusternik. *Shortest paths, variational problems*, volume 13. Popular lectures in mathematics, PERGAMON, 1964. (Page 32.)

[125] P. Lévy. Chaînes doubles de markov et fonctions aléatoires de deux variables. *C.R.Académie des sciences*, 226 :53–55, 1948. (Page 187.)

[126] Mai S. Mabrouk, Nahed H. Solouma, and Yasser M. Kadah. Survey of retinal image segmentation and registration. *ICGST International Journal on Graphics, Vision and Image Processing*, 6 :1–11, 2006. (Page 104.)

[127] R. Malladi and J. A. Sethian. A real-time algorithm for medical shape recovery. In *ICCV '98 : Proceedings of the Sixth International Conference on Computer Vision*, page 304, Washington, DC, USA, 1998. IEEE Computer Society. (Page 28.)

[128] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation : A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 :158–175, 1995. (Page 105.)

[129] P. Mansfield. Multi-planar image formation using nmr spin echoes. *Journal of Physics C : Solid State Physics*, 10(3) :L55–L58, 1977. (Page 135.)

[130] F. Mao, S. Ruan, A. Bruno, C. Toumoulin, R. Collorec, and P. Haigron. Extraction of structural features in digital subtraction angiography. In *Proceedings of the Biomedical Engineering Days*, pages 166–169, august 1992. (Page 104.)

[131] Alberto Martelli. Application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19(2) :73–83, February 1976. (Page 21.)

[132] Pavel Matula, Jan Huben, and Michal Kozubek. Fast marching 3d reconstruction of interphase chromosomes. In Milan Sonka, Ioannis A. Kakadiaris, and Jan Kybic, editors, *ECCV Workshops CVAMIA and MMBIA*, volume 3117 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2004. (Page 28.)

[133] T. McInerney and D. Terzopoulos. T-snakes : Topology adaptive snakes. *Medical Image Analysis*, 4 :73–91, 2000. (Page 105.)

[134] John Melonakos, Eric Pichon, Sigurd Angenent, and Allen Tannenbaum. Finsler active contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3) :412–423, 2008. (Page 105.)

[135] Nicolas Merlet and Josiane Zerubia. New prospects in line detection by dynamic programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4) :426–431, 1996. (Page 104.)

[136] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4) :647–668, 1987. (Page 99.)

[137] Joseph S. B. Mitchell and Christos H. Papadimitriou. The weighted region problem : finding shortest paths through a weighted planar subdivision. *J. ACM*, 38(1) :18–73, 1991. (Page 27.)

[138] J.S.B. Mitchell. Planning shortest paths (phd thesis). Technical report, Department of Operations Research, Stanford University, August 1986. (Page 27.)

[139] E.F. Moore. The shortest path through a maze. In Harvard University Press, editor, *Proceedings of an International Synposium on the Theory of Switching (Cambridge, Massachusetts)*, pages 285–292, Cambridge, April 1959. (Page 40.)

[140] M.Péchaud, R.Keriven, and T.Papadopoulo. Combinatorial optimization for electrode labeling of EEG caps. Technical Report 07-32, CERTIS, July 2007. (Pages 183 and 188.)

[141] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference : An empirical study. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999. (Pages 189, 190, and 192.)

[142] Delphine Nain, Anthony Yezzi, and Greg Turk. Vessel segmentation using a shape driven flow. In *Medical Image Computing and Comptuer-assisted Intervention - MICCAI*, pages 51–59, 2004. (Page 105.)

[143] M. Niemeijer, J.J. Staal, B. van Ginneken, M. Loog, and M.D. Abramoff. Comparative study of retinal vessel segmentation methods on a new publicly available database. In J. Michael Fitzpatrick and M. Sonka, editors, *SPIE Medical Imaging*, volume 5370, pages 648–656. SPIE, SPIE, 2004. (Pages 103 and 119.)

[144] Bjorn Nilsson and Anders Heyden. Segmentation of dense leukocyte clusters. In *MMBIA '01 : Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'01)*, page 221, Washington, DC, USA, 2001. IEEE Computer Society. (Page 30.)

[145] Lauren O'Donnell, Steven Haker, and Carl-Fredrik Westin. New approaches to estimation of white matter connectivity in diffusion tensor mri : Elliptic pdes and geodesics in a tensor-warped space. In *MICCAI '02 : Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part I*, pages 459–466, London, UK, 2002. Springer-Verlag. (Page 31.)

[146] S. Ogawa, T.M. Lee, A.S. Nayak, and P. Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic Resonance in Medecine*, 14 :68–78, 1990. (Page 135.)

[147] G. J. M. Parker and D. C. Alexander. Probabilistic anatomical connectivity derived from the microscopic persistent angular structure of cerebral tissue. *Philosophical Transactions of the Royal Society, Series B*, 360 :893–902, 2005. (Page 156.)

[148] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988. (Page 184.)

[149] J. Pearl. *Probabilistic Reasonning in Intellignet System :Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988. (Page 189.)

[150] Judea Pearl. *Heuristics : intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984. (Page 46.)

[151] T. Peng, I. H. Jermyn, V. Prinet, and J. Zerubia. An extended phase field higher-order active contour model for networks and its application to road network extraction from vhr satellite images. In *Proc. European Conference on Computer Vision (ECCV)*, Marseille, France, octobre 2008. (Page 105.)

[152] P. Perez, A. Blake, and M. Gangnet. JetStream : probabilistic contour extraction with particles. In *Proceedings of IEEE International Conference on Computer Vision, ICCV'01*, volume 2, pages 524–531, Vancouver, Canada, July 2001. (Page 105.)

[153] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions ob Pattern Analysis and Machine Intelligence*, 12(7) :629–639, 1990. (Page 104.)

[154] M. Perrin, C. Poupon, Y. Cointepas, B. Rieul, N. Golestani, C. Pallier, D. Riviere, A. Constantinesco, D. Le Bihan, and J.-F. Mangin. Fiber tracking in q-ball fields using regularized particle trajectories. In *IPMI*, pages 52–63, 2005. (Page 156.)

[155] Gabriel Peyré and Laurent D. Cohen. Geodesic remeshing using front propagation. *Int. J. Comput. Vision*, 69(1) :145–156, 2006. (Page 32.)

[156] Gabriel Peyre and Laurent D. Cohen. Landmark-based geodesic computation for heuristically driven path planning. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2 :2229–2236, 2006. (Page 66.)

[157] K. Polthier and M. Schmies. Geodesic flow on polyhedral surfaces. In E. Gröller, H. Löffelmann, and W. Ribarsky, editors, *Data Visualization˜'99*, pages 179–188. Springer-Verlag Wien, 1999. (Page 32.)

[158] Konrad Polthier and Markus Schmies. Straightest geodesics on polyhedral surfaces. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Courses*, pages 30–38, New York, NY, USA, 2006. ACM. (Page 32.)

[159] Kelvin Poon, Ghassan Hamarneh, and Rafeef Abugharbieh. Live-vessel : Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In Nicholas Ayache, Sébastien Ourselin, and Anthony Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, volume 4792 of *LNCS*, pages 444–451. Springer, 2007. (Page 104.)

[160] Miranda Poon, Ghassan Hamarneh, and Rafeef Abugharbieh. Live-vessel : Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In *Proceedings of MICCAI (2)*, pages 444–451, 2007. (Page 106.)

[161] Emmanuel Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem.* PhD thesis, University of Nice-Sophia Antipolis, France, October 2004. (Page 100.)

[162] Emmanuel Prados and Olivier Faugeras. Unifying approaches and removing unrealistic assumptions in shape from shading : Mathematics can help. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume 3024 of *Lecture Notes in Computer Science*, pages 141–154. Springer, 2004. (Page 30.)

[163] Emmanuel Prados, Christophe Lenglet, Jean-Philippe Pons, Nicolas Wotawa, Rachid Deriche, Olivier Faugeras, and Stefano Soatto. Control theory and fast marching techniques for brain connectivity mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, June, 2006*, volume 1, pages 1076–1083. IEEE, June 2006. (Pages 80, 157, 168, and 178.)

[164] V. Prinet, O. Monga, C. Ge, S. L. Xie, and S. D. Ma. Thin network extraction in 3d images : Application to medical angiograms. In *ICPR '96 : Proceedings of the International Conference on Pattern Recognition*, volume 3, page 386, Washington, DC, USA, 1996. IEEE Computer Society. (Page 104.)

[165] M. Péchaud, M. Descoteaux, and R. Keriven. Brain connectivity using geodesics in hardi. In *MICCAI*, London, England, September 2009. (Page 157.)

[166] M. Péchaud, R. Keriven, T. Papadopoulo, and J.-M. Badie. Combinatorial optimization for electrode labeling of eeg caps. In *MICCAI, 10th International Conference*, Brisbane, Australia, Oct 2007. (Pages 11 and 183.)

[167] M. Péchaud, G. Peyré, and R. Keriven. Extraction of tubular structures over an orientation domain. In *CVPR '09 : Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2009. IEEE Computer Society. (Pages 11 and 102.)

[168] M. Péchaud, I. Vanzetta, T. Deneux, and R. Keriven. Sift-based sequence registration and flow-based cortical vessel segmentation applied to high resolution optical imaging data. In *Proceedings of ISBI 2008*, Paris, May 2008. (Pages 11 and 102.)

[169] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *ICCV '05*, pages 1048–1054, Washington, DC, USA, 2005. IEEE. (Page 188.)

[170] R. M. Rangayyan, F. J. Ayres, F. Oloumi, F. Oloumi, and P. Eshghzadeh-Zanjani. Detection of blood vessels in the retina with multiscale Gabor filters. *Journal of Electronic Imaging*, 17(2) :023018 :1–7, 2008. (Page 104.)

[171] E. Ricci and R. Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging*, 26(10) :1357–1365, 2007. (Page 105.)

[172] J. Ricny. A-priori information driven model for road segmentation in high resolution images. In *OBIA06*, pages xx–yy, 2006. (Page 105.)

[173] Van Uitert Robert and Bitter Ingmar. Subvoxel precise skeletons of volumetric data based on fast marching methods. *Medical physics*, 34 :627–638, 2007. (Page 26.)

[174] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3) :867–884, 1992. (Pages 30, 57, 58, 60, and 66.)

[175] G. S. Russell, K. J. Eriksen, P. Poolman, P. Luu, and D. M. Tucker. Geodesic photogrammetry for localizing sensor positions in dense-array eeg. *Clinical Neurophysiology*, 116 :1130–1140, 2005 (see also http ://www.egi.com/c_gps.html). (Page 184.)

[176] Lang S. *Introduction to Differentiable Manifolds*. Springer Verlag, New York, second edition, 2002. (Page 31.)

[177] Alberto Santamaría-Pang, C. M. Colbert, P. Saggau, and Ioannis A. Kakadiaris. Automatic centerline extraction of irregular tubular structures using probability

volumes from multiphoton imaging. In *Proceedings of MICCAI (2)*, pages 486–494, 2007. (Page 105.)

[178] Peter Savadjiev, Jennifer S. W. Campbell, Maxime Descoteaux, Rachid Deriche, G. Bruce Pike, and Kaleem Siddiqi. Labeling of ambiguous sub-voxel fibre bundle configurations in high angular resolution diffusion mri. *NeuroImage*, 41(1) :58–68, 2008. (Page 156.)

[179] Michiel Schaap, Rashindra Manniesing, Ihor Smal, Theo van Walsum, Aad van der Lugt, and Wiro J. Niessen. Bayesian tracking of tubular structures and its application to carotid arteries in cta. In *Proceedings of MICCAI (2)*, pages 562–570, 2007. (Page 105.)

[180] H. Schmitt, Michael Grass, Volker Rasche, Oliver Schramm, S. Hähnel, and K. Sartor. An x-ray based method for the determination of the contrast agent propagation in 3d vessel structures. *IEEE Transactions on Medical Imaging*, 21(3) :251–262, 2002. (Page 104.)

[181] Alexander Schrijver. *Combinatorial optimization : polyhedra and efficiency*. volume A, paths, flows, matchings, chapter 1-38. Springer, 2003. Schrijver. (Page 40.)

[182] E.L. Schwartz, A. Shaw, and E. Wolfson. A numerical solution to the generalized mapmaker's problem : Flattening nonconvex polyhedral surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9) :1005–1008, 1989. (Page 32.)

[183] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Alignment-based recognition of shape outlines. In *4th International Workshop on Visual Form*, pages 606–618, 2001. (Page 186.)

[184] M. Sermesant, E. Konukoglu, H. Delingette, Y. Coudiere, P. Chinchaptanam, K.S. Rhode, R. Razavi, and N. Ayache. An anisotropic multi-front fast marching method for real-time simulation of cardiac electrophysiology. In *Proceedings of Functional Imaging and Modeling of the Heart 2007 (FIMH'07)*, volume 4466 of *LNCS*, pages 160–169, 7-9 June 2007. (Page 30.)

[185] Maxime Sermesant, Ender Konukoglu, Hervé Delingette, Yves Coudière, Phani Chinchapatnam, Kawal S. Rhode, Reza Razavi, and Nicholas Ayache. An anisotropic multi-front fast marching method for real-time simulation of cardiac electrophysiology. In Frank B. Sachse and Gunnar Seemann, editors, *FIMH*, volume 4466 of *Lecture Notes in Computer Science*, pages 160–169. Springer, 2007. (Page 31.)

[186] J. Sethian. Level set methods and fast marching methods : Evolving interfaces in computational geometry, 1998. (Pages 36, 39, and 77.)

[187] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4) :1591–1595, 1996. (Pages 53 and 59.)

[188] J. A. Sethian. *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999. (Pages 25, 26, 32, 66, and 105.)

[189] James A. Sethian and Er Mihai Popovici. Three dimensional traveltime computation using the fast marching method, in expanded abstracts from the 67th ann. *Internat. Mtg. of Soc. Expl. Geophys*, 1778 :1778–1781, 1997. (Page 30.)

[190] James A. Sethian and Alexander Vladimirsky. Ordered upwind methods for static hamilton–jacobi equations : Theory and algorithms. *SIAM J. Numer. Anal.*, 41(1) :325–363, 2003. (Page 99.)

[191] K. K. Seunarine, P. A. Cook, K. Embleton, G. J. M. Parker, and D. C. Alexander. A general framework for multiple-fibre pico tractography. In *Medical Image Understanding and Analysis*, 2006. (Page 156.)

[192] Kaleem Siddiqi and Stephen Pizer, editors. *Springer*. Addison-Wesley Longman Publishing Co., Inc., 2008. (Page 106.)

[193] C. Sinthanayothin, J. Boyce, and C. T. Williamson. Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. *British Journal of Ophthalmology*, 83 :902–910, 1999. (Page 105.)

[194] Michael J. De Smith, Michael F. Goodchild, and Paul A. Longley. *Geospatial Analysis : A Comprehensive Guide to Principles, Techniques and Software Tools*. Matador, 2006. (Page 104.)

[195] J. V. B. Soares and R. M. Cesar-Jr. Segmentation of retinal vasculature using wavelets and supervised classification : Theory and implementation. In H. F. Jelinek and M. J. Cree, editors, *Automated Image Detection of Retinal Pathology*. CRC Press, 2007. To appear. (Page 104.)

[196] E. Sorantin, C. Halmai, B. Erdohelyi, K. Palagyi, L. Nyul, K. Olle, B. Geiger, F. Lindbichler, G. Friedrich, and K. Kiesler. Spiral-ct-based assessment of tracheal stenoses using 3-d-skeletonization. *IEEE Transactions on Medical Imgaging*, 21(3) :263–273, 2002. (Page 106.)

[197] Alon Spira and Ron Kimmel. An efficient solution to the eikonal equation on parametric manifolds. In *In INTERPHASE 2003 meeting, Isaac Newton Institute for Mathematical Sciences, 2003 Preprints, Preprint no. NI03045-CPD*, pages 315–327, 2003. (Page 99.)

[198] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4) :501–509, 2004. (Page 119.)

[199] Carsten Steger. An unbiased detector of curvilinear structures. Technical report, FG BV, Informatik IX, Technische Universitat Munchen, 1996. (Page 105.)

[200] L Sukkaew, B Uyyanonvara, S A Barman, and J Jareanjita. Automated vessels detection on infant retinal images. In *Proceedings of International Conference on Control, Automation and Systems (ICCAS2004)*, pages 321–325, August 2004. (Page 104.)

[201] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3) :553–560, 2005. (Page 99.)

[202] F. SÃ©gonne, E. Grimson, and B. Fischl. Topological correction of subcortical segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, Montreal, Nov 2003. (Page 26.)

[203] Xiaodong Tao, Xiao Han, Maryam E. Rettmann, Jerry L. Prince, and Christos Davatzikos. Statistical study on cortical sulci of human brains. In *IPMI '01 : Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, pages 475–487, London, UK, 2001. Springer-Verlag. (Page 33.)

[204] Alexandru Telea and Jarke J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *VISSYM '02 : Proceedings of the symposium on Data Visualisation 2002*, pages 251–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. (Page 26.)

[205] John T.Moy. *OSPF : Anatomy of an Internet Routing Protocol*. Addison Wesley, 200. (Page 21.)

[206] Y. A. Tolias and S. M. Panas. A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *IEEE Transactions on Medical Imaging*, 17 :263–273, April 1998. (Page 105.)

[207] J-Donald Tournier, Fernando Calamante, and Alan Connelly. Robust determination of the fibre orientation distribution in diffusion mri : Non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 35(4) :1459–1472, 2007. (Pages 156 and 164.)

[208] T. Tozaki, Y. Kawata, N. Niki, H. Ohmatsu, , and N. Moriyama. 3-d visualization of blood vessels and tumor using thin slice ct. *IEEE Nuclear Science Symposium and Medical Imaging Conference*, 3 :1470–1474, 1995. (Page 106.)

[209] John N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40 :1528–1538, 1995. (Page 99.)

[210] David S. Tuch. *Diffusion MRI of Complex Tissue Structure*. PhD thesis, Harvard University and Massachusetts Institute of Technology, 2002. (Page 156.)

[211] I. Vanzetta, T. Deneux, G. S. Masson, and O. D. Faugeras. Cerebral blood flow recorded at high sensitivity in two dimensions using high resolution optical imaging. In *ISBI*, pages 1264–1267, 2006. (Pages 136, 138, and 139.)

[212] Er Vasilevskiy and Kaleem Siddiqi. Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :1565–1578, 2002. (Page 105.)

[213] V.Kolmogorov. Convergent tree-reweighted message passing for energy minimization. Technical Report MSR-TR-2005-38, 2005. (Page 191.)

[214] G. Vosselman and J. de Knecht. Road tracing by profile matching and kalman filtering. In *Ascona95*, pages 265–274, 1995. (Page 105.)

[215] V.J. Wedeen, R.P. Wang, J.D. Schmahmann, T. Benner, W.Y.I. Tseng, G. Dai, D.N. Pandya, P. Hagmann, H. D'Arceuil, and A.J. de Crespigny. Diffusion spectrum magnetic resonance imaging (dsi) tractography of crossing fibers. *NeuroImage*, 41(4) :1267–1277, July 2008. (Page 156.)

[216] Y. Weiss and D. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEETIT*, 47, 2001. (Pages 189 and 190.)

[217] J.W. J. Williams. Algorithm 232 : Heapsort. *Communications of the ACM*, 7 :347–348, 1964. (Page 45.)

[218] Yong Yang, Shuying Hung, and Nini Rao. An automatic hybrid method for retinal blood vessel extraction. *International Journal of applied Mathematics and Computer Science*, 18(3) :399–407, 2008. (Page 104.)

[219] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. O(n) implementation of the fast marching algorithm. *J. Comput. Phys.*, 212(2) :393–399, 2006. (Page 67.)

[220] P J Yim and P L Choyke. Gray-scale skeletonization of small vessels in magnetic resonance angiography. *IEEE Transactions on Medical Imaging*, 19 :568–576, june 2000. (Page 104.)

[221] Fan Zhang, Edwin R. Hancock, Casey Goodlett, and Guido Gerig. Probabilistic white matter fiber tracking using particle filtering and von mises-fisher sampling. *Medical Image Analysis*, 13(1) :5–18, Febuary 2008. (Page 156.)

[222] Z. Zhang. iterative point matching for registration of free-form curves. Technical Report RR-1658, INRIA, 1992. (Page 185.)

[223] J. Zhou, W.F. Bischof, and T.M. Caelli. Robust and efficient road tracking in aerial images. In *CMRT05*, pages xx–yy, 2005. (Page 105.)

[224] L. Zhou, M. S. Rzeszotarski, L. J. Singerman, and J. M. Chokreff. The detection and quantification of retinopathy using digital angiograms. *IEEE Transactions on Medical Imaging*, 13(4) :619–626, 1994. (Page 105.)

[225] B. Zitova and J. Flusser. Image registration methods : a survey. *Image and Vision Computing*, 21(11) :977–1000, October 2003. (Page 136.)