Fidelity metrics between curves and surfaces: currents, varifolds and normal cycles

Nicolas Charon[*], Benjamin Charlier[**,†], Joan Glaunès[‡], Pietro Gori[¶] and Pierre Roussillon[§]
* Johns Hopkins University, Center of Imaging Sciences
** Univ. Montpellier, Institut Montpelliérain Alexander Grothendieck
† Institut du Cerveau et de la Moëlle Épinière, ARAMIS
‡ MAP5, Université Paris Descartes, Paris, France
§ ENS Cachan, CNRS, Université Paris-Saclay, CMLA, 94235 Cachan, France
¶ Télécom ParisTech, LTCI, équipe IMAGES, Paris, France

Abstract

This chapter provides an overview of some mathematical and computational models that have been proposed over the past few years for defining data attachment terms on shape spaces of curves or surfaces. In all these models, shapes are seen as elements of a space of generalized distributions, such as currents or varifolds. Then norms are defined through Reproducing Kernel Hilbert Spaces (RKHS) which lead to shape distances that can be conveniently computed in practice. These were originally introduced in conjunction with diffeomorphic methods in Computational Anatomy, and have indeed proved to be very efficient in this field. We provide a basic description of these different models and their practical implementation, then discuss the respective properties and potential advantages or downsides of each of them in diffeomorphic registration problems.

1. Introduction

In Chapter 4, following the classical setting proposed by Grenander, metrics on shape spaces were defined through the action of diffeomorphism groups equipped with right-invariant metrics. In particular, the LDDMM framework inntroduced earlier provides a convenient way to generate diffeomorphic transformations and such right-invariant metrics. In that case, the resulting distance between two given shapes is given through the solution of an exact registration problem obtained by optimizing the deformation cost over all possible deformation fields that match the source shape on the target.

This approach, however, only applies if both shapes belong to the same orbit ; in other words, if there exists a deformation in the group that can exactly deform one shape on the other. Such an assumption is routinely violated in practical scenarios involving shapes extracted from biomedical imaging data. Indeed, those shapes are typically affected by many other variations including noise, potential topological variations or segmentation artifacts all of which are poorly modeled within a pure diffeomorphic setting. From a statistical perspective, it is in fact more reasonable to make the computation of shape distances rather as insensitive as possible to those types of perturbations which are not morphologically relevant.

To that end, a standard approach is to relax the exact matching constraint by instead considering inexact registration. Specifically, the original boundary value problem is replaced by the minimization of a composite functional involving a weighted combination of the previous deformation cost and a fidelity or data attachment term between the deformed source shape and the target. The registration problem is thus *a priori* well-posed even for a target outside the orbit of the source shape. The fidelity term and its weight will typically depend on the nature of the shapes (landmarks, images, curves...) as well as the expected noise model and level on the observations. This inexact formulation for LDDMM (as well as for other deformation models) has been very standard in early applications to landmarks [JM00] or images [BMTY05]. In those works, fidelity terms consist in the simple Euclidean norm between the landmark positions or the sum of squared differences of the pixel values of the images.

Regarding the construction of fidelity terms, shape spaces of curves, surfaces and more gen-

erally submanifolds of $\mathbb{R}^k$ constitute a particularly challenging case if compared for instance to landmarks or images. The main reason is that submanifolds embody a fundamental invariance to reparametrization and thus cannot be *a priori* treated as labelled point sets like in the situation of landmarks. In concrete applications, such shapes are discretized as vertices and meshes. Yet meshes obtained *e.g* from different subjects generally do not contain the same number of vertices and no predefined correspondences exist between them that could provide a straightforward equivalent to Euclidean distances. Many approaches have thus resolved to first extracting and pairing feature points from the two meshes in order to reduce the problem to registration of the obtained landmarks.

This chapter is dedicated to a particular class of methods providing efficient and robust metrics between submanifolds themselves (discrete or continuous) that completely bypass the need for such preprocessing steps. The primary objective is to get adequate fidelity terms to be embedded into inexact registration frameworks for curves and surfaces. Namely, one seeks metrics that are intrinsically independent of a specific shape parametrization or sampling rate, that can be evaluated and differentiated directly in competitive computational time, and that can compare a large class of objects beyond smooth geometries while being reasonably robust to the aforementioned shape artifacts.

The frameworks presented in this chapter all follow the general philosophy and mathematical ideas introduced in the field of Geometric Measure Theory [Fed69, Alm66, All72] in which shapes like submanifolds are basically represented in certain spaces of *generalized measures* such as *currents*, *varifolds* or *normal cycles*. The sections below present and compare several of these approaches and computational models, since the earliest work of [GTY04] in this context, by emphasizing the relevance and applicability to shape analysis and computational anatomy. We also point out to the reader that some recent and related works such as [FCVP17] have exploited alternative ideas from optimal transport theory for similar purpose. We do not present those works here for the sake of general concision.

2. General setting and notations

In the entire chapter, and although the mathematical settings readily extend to general submanifolds, we will restrict the presentation to the cases of curves and surfaces embedded in $\mathbb{R}^2$ or $\mathbb{R}^3$ as these constitute the bulk of the applications considered in this book. The methods we introduce below can equally deal with smooth curves and surfaces but also piecewise smooth and discrete objects, which are all encompassed by the notion of *rectifiable subsets* in geometric measure theory.

Unless stated otherwise, curves and surfaces are meant in the sense of rectifiable curves and rectifiable surfaces. For the sake of concision and clarity, we will however not introduce in detail the precise definition and properties of rectifiable subsets; the interested reader is referred for example to [Mor95]. Instead, we will here adopt the following intuitive setting: we will call $\mathcal{M} \subset \mathbb{R}^k$ a rectifiable curve (resp. surface) embedded in the ambient space $\mathbb{R}^k$ with either $k = 2$ or $k = 3$, if for almost every $x \in \mathcal{M}$, there exists a tangent space $T_x\mathcal{M}$ of dimension one (resp. two) of $\mathcal{M}$ at x. This basically allows for the presence of isolated singularities like in a polygonal curve or polyhedral surface. A very important notion for the rest of the chapter will be the one of orientation. If an orientation of $\mathcal{M}$ is given, *i.e* the subspaces $T_x\mathcal{M}$ are oriented in a "continuous" way on $\mathcal{M}$, then we will equivalently represent $T_x\mathcal{M}$ by a single unit vector in $\mathbb{S}^{k-1}$, which is either the oriented tangent vector $\vec{t}(x)$ in the case of a curve or the oriented unit normal vector $\vec{n}(x)$ in the case of a surface.

3. Currents

The use of currents in computational anatomy has been introduced first to define fidelity metrics between surfaces in [GV06], and between curves in [GQMY08]. These works extend the earlier work on matching distributions [GTY04] which can be seen in turn as a particular case of the currents model, as will be shown later.

3.1. Curves and surfaces as currents

Let $\mathcal{M}$ be an oriented shape, *i.e* a curve in $\mathbb{R}^k$ ($k = 2$ or $k = 3$), or an oriented surface in $\mathbb{R}^3$. As mentioned in Section 2, we only assume regularity at almost every point in $\mathcal{M}$, which allows for shapes with or without boundary, and possibly several disconnected components and branching parts.

For any continuous vector field $\vec{w} : \mathbb{R}^k \to \mathbb{R}^k$ in the ambient space $\mathbb{R}^k$, one may compute its integral along $\mathcal{M}$, which can be denoted $[\mathcal{M}](\vec{w})$, and writes:

- if $\mathcal{M}$ is a curve:

$$[\mathcal{M}](\vec{w}) := \int_{\mathcal{M}} \left\langle \vec{w}(x), \vec{t}(x) \right\rangle d\ell(x). \tag{0.1}$$

- if $\mathcal{M}$ is a surface:

$$[\mathcal{M}](\vec{w}) := \int_{\mathcal{M}} \langle \vec{w}(x), \vec{n}(x) \rangle \, dS(x). \tag{0.2}$$

The application $\vec{w} \to [\mathcal{M}](\vec{w})$ defines a linear mapping between the space of continuous vector fields into $\mathbb{R}$, which characterizes the shape $\mathcal{M}$. This mapping $[\mathcal{M}]$ is called the **current** associated with $\mathcal{M}$. The main advantage of this somewhat abstract setting is to view shapes as elements of a vector space, allowing to consider linear combinations of such elements, and to define norms, which is the basis of the construction.

Let us define more precisely these notions:

Definition 1. • A **current** S is a continuous linear mapping $\vec{w} \mapsto S(\vec{w})$ from $C_0(\mathbb{R}^k, \mathbb{R}^k)$ (the space of continuous vector fields of $\mathbb{R}^k$ that vanish at infinity), into $\mathbb{R}$.

- The **current associated with an oriented shape** $\mathcal{M}$ is the current $[\mathcal{M}]$ defined by formula (0.1) or (0.2) for any $\vec{w} \in C_0(\mathbb{R}^k, \mathbb{R}^k)$.

Let us make a few remarks about this definition. First, the continuity requirement for the linear mapping is equivalent to the inequality

$$|S(\vec{w})| \leq \|\vec{w}\|_\infty,$$

satisfied for all $\vec{w} \in C_0(\mathbb{R}^k, \mathbb{R}^k)$. Second, we assume that vector fields vanish at infinity to get completeness of the space $C_0(\mathbb{R}^k, \mathbb{R}^k)$ with respect to the infinity norm.

3.2. Kernel metrics on currents

The next step is to define a tractable distance between currents which will be used as a data fidelity term. This is done using the framework of Reproducing Kernel Hilbert Spaces (RKHS) [Aro50]. As previously, we first give the main idea before stating precise definitions. The construction starts by the choice of a kernel function $K : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$. The kernel metric between two shapes $\mathcal{M}_1, \mathcal{M}_2$ is then defined as the quantity

$$\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} := \int_{\mathcal{M}_1} \int_{\mathcal{M}_2} K(x, y) \left\langle \vec{t}_1(x), \vec{t}_2(y)) \right\rangle d\ell_1(x) \, d\ell_2(y), \tag{0.3}$$

for curves, or

$$\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} := \int_{\mathcal{M}_1} \int_{\mathcal{M}_2} K(x,y) \, \langle \vec{n}_1(x), \vec{n}_2(y)) \rangle \, dS_1(x) \, dS_2(y). \tag{0.4}$$

The notation $\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'}$ means that the formula defines an inner product between the currents associated with the shapes, as will be detailed in the following definitions. Assuming this point for the moment, one may use the associated norm to define a dissimilarity between two shapes:

$$\| [\mathcal{M}_1] - [\mathcal{M}_2] \|^2_{W'} = \langle [\mathcal{M}_1], [\mathcal{M}_1] \rangle_{W'} - 2 \langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} + \langle [\mathcal{M}_2], [\mathcal{M}_2] \rangle_{W'} \tag{0.5}$$

To go further and define precisely these notions, we need to use the theory of reproducing kernels for spaces of vector-valued functions. The theory of reproducing kernels was first developed by Aronszajn [Aro50] in the scalar case, which is the most commonly used setting, with applications in many fields such as scattered data interpolation [Duc77] and statistical learning [HSS08]. Shortly after Aronszajn, Laurent Schwartz [Sch64] extended the theory to a more general and abstract setting, including de facto the case of vector-valued functions. Several other works have focused on this case [CDVTU10, MP05, DR04, CA04, DVUV13]. The following definitions and theorem rephrase results found in [MG14]. We refer to this work for a detailed study. Here we only consider the case of *scalar* kernels for simplicity (but still for vector-valued functions spaces).

Definition 2. • Let $K : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ and $s \in \mathbb{N}$. We say that K is an **s-admissible kernel** if

1. K is a positive kernel, *i.e* for any $n \geq 1$, any $x_1, \ldots, x_n \in \mathbb{R}^k$, the $n \times n$ matrix with entries $K(x_i, x_j)$ is a positive semi-definite symmetric matrix.
2. For any $x \in \mathbb{R}^k$, $K(\cdot, x) \in C^{2s}_0(\mathbb{R}^k \times \mathbb{R}^k, \mathbb{R})$ (i.e. it is of class C^{2s} and vanishes at infinity).

It is called simply admissible if it is 0-admissible.

- Let $s \in \mathbb{N}$. A Hilbert space W of vector fields $\vec{w} : \mathbb{R}^k \to \mathbb{R}^k$ is called s-admissible if W is continuously embedded in $C^s_0(\mathbb{R}^k, \mathbb{R}^k)$. It is called simply admissible if it is 0-admissible.

Typical examples of such kernels used in applications are the Gaussian kernel $K(x,y) = \exp(\|x-y\|^2/\sigma^2)$, or the Cauchy kernel $K(x,y) = \frac{1}{1+\|x-y\|^2/\sigma^2}$, where x, y are points in $\mathbb{R}^k$ and $\sigma > 0$ is a fixed scale parameter.

Let us mention that for all material presented here, including the use of the derived discrete formulation for LDDMM registration algorithms, the 0-admissibility (i.e. continuity) assumption is sufficient. However a stronger 1-admissibility assumption is usually required for some theoretical results to hold for registration problems in the non-discrete case [Gla05]. This is not an issue at all in practice since usual kernels such as the Gaussian and Cauchy kernels, have in fact C^∞ regularity, which implies that their corresponding Hilbert space will be p-admissible for any p, as ensured by the following theorem:

Theorem 1. *([MG14], th. 2.6 and 2.11) Any p-admissible kernel K is associated with a unique p-admissible Hilbert space W of vector fields in $\mathbb{R}^k$, such that*

$$\forall x, \alpha \in \mathbb{R}^k, \ \forall w \in W, \ \langle K(\cdot, x)\alpha, w \rangle_W = \langle w(x), \alpha \rangle .$$

K is called the **reproducing kernel** *of space W.*

Combining Definitions 1 and 2, we see that by choosing an admissible kernel K, one defines an admissible Hilbert space W of vector fields such that any current S belongs to its dual space W', since

$$\forall \vec{w} \in W, \qquad |S(\vec{w})| \leq \|\vec{w}\|_\infty \leq \|\vec{w}\|_W .$$

Hence one can consider the canonical inner product in W' between any two currents. Now to

derive formula (0.3), let us first rewrite the action of a single current associated with a curve $\mathcal{M}$ on a vector field $\vec{w}$:

$$\begin{aligned}[\mathcal{M}](\vec{w}) &= \int_{\mathcal{M}} \left\langle \vec{w}(x), \vec{t}(x) \right\rangle d\ell(x) \\ &= \int_{\mathcal{M}} \left\langle K(\cdot,x)\vec{t}(x), \vec{w} \right\rangle_W d\ell(x) \\ &= \left\langle \int_{\mathcal{M}} K(\cdot,x)\vec{t}(x) d\ell(x)\ ,\ \vec{w} \right\rangle_W\end{aligned}$$

This shows that $x \mapsto \int_{\mathcal{M}} K(\cdot,x)\vec{t}(x)$ is the unique Riesz representer in W of the current $[\mathcal{M}]$ in W'. This representer is simply a convolution of the tangent bundle of the curve with the kernel. Consequently, considering now two currents $[\mathcal{M}_1]$, $[\mathcal{M}_2]$, one obtains the following expression of their inner product:

$$\begin{aligned}\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} &= [\mathcal{M}_1] \left(x \mapsto \int_{\mathcal{M}_2} K(x,y)\vec{t}_2(y) d\ell_2(y) \right) \\ &= \int_{\mathcal{M}_1}\int_{\mathcal{M}_2} K(x,y) \left\langle \vec{t}_1(x), \vec{t}_2(y) \right\rangle d\ell_1(x)\ d\ell_2(y)\end{aligned}$$

In the case of surfaces, the derivation of equation (0.4) is strictly similar.

3.3. The discrete model

The computation of the kernel metric (0.3) and (0.4) between shapes requires a discretization of the integrals and unit tangent or normal vectors. In practice shapes are given as meshes: unions of line segments in the case of curves, of triangles in the case of surfaces, with connectivity information.

Let us assume that $\mathcal{M}$ is such a mesh, *i.e* $\mathcal{M}$ is a union of line or triangle elements $f_1, \ldots, f_m$. We will denote f_i^k the vertices of each element f_i, with $k = 1, 2$ for curves and $k = 1, 2, 3$ for surfaces. We assume that the orientation is encoded through the ordering of these vertices. We can further define for each element f_i:

- for curves: its center $c_{f_i} = (f_i^1 + f_i^2)/2$, tangent vector $\vec{\tau}_{f_i} = (f_i^2 - f_i^1)$, length $\ell_{f_i} = \|\vec{\tau}_{f_i}\|$, and unit tangent vector $\vec{t}_{f_i} = \vec{\tau}_{f_i}/\ell_{f_i}$,
- for surfaces: its center $c_{f_i} = (f_i^1 + f_i^2 + f_i^3)/3$, normal vector $\vec{\nu}_{f_i} = \frac{1}{2}((f_i^2 - f_i^1) \times (f_i^3 - f_i^1))$, area $S_{f_i} = \|\vec{\nu}_{f_i}\|$, and unit normal vector $\vec{n}_{f_i} = \vec{\nu}_{f_i}/S_{f_i}$

A simple discrete approximation of formula (0.3) is obtained by first writing the double integral as a double sum of double integrals over each pair of triangles and then approximating the values of the kernel $K(x,y)$ by a single value at the centers of the elements. For curves this reads

$$\begin{aligned}\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} &= \int_{\mathcal{M}_1}\int_{\mathcal{M}_2} K(x,y) \left\langle \vec{t}_1(x), \vec{t}_2(y) \right\rangle d\ell_1(x)\ d\ell_2(y) \\ &= \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} \int_{f_{1,i}}\int_{f_{2,j}} K(x,y) \left\langle \vec{t}_{f_{1,i}}, \vec{t}_{f_{2,j}} \right\rangle d\ell_1(x)\ d\ell_2(y) \\ &\approx \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} \int_{f_{1,i}}\int_{f_{2,j}} K(c_{f_{1,i}}, c_{f_{2,j}}) \left\langle \vec{t}_{f_{1,i}}, \vec{t}_{f_{2,j}} \right\rangle d\ell_1(x)\ d\ell_2(y) \\ &\approx \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} K(c_{f_{1,i}}, c_{f_{2,j}}) \left\langle \vec{t}_{f_{1,i}}, \vec{t}_{f_{2,j}} \right\rangle \ell_{f_{1,i}}\ \ell_{f_{1,i}} \\ &= \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} K(c_{f_{1,i}}, c_{f_{2,j}}) \left\langle \vec{\tau}_{f_{1,i}}, \vec{\tau}_{f_{2,j}} \right\rangle\end{aligned}$$

A nice characteristic of this approximation is that it corresponds to the exact expression of

the inner product in W' when replacing the currents $[\mathcal{M}_1]$, $[\mathcal{M}_2]$ by sums of Dirac functionals (located at the faces centers). More precisely, we can define:

$$[\mathcal{M}_1] \approx [\mathcal{M}_1]_d = \sum_{i=1}^{m_1} \delta_{c_{f_{1,i}}}^{\vec{\tau}_{f_{1,i}}} \quad and \quad [\mathcal{M}_2] \approx [\mathcal{M}_2]_d = \sum_{j=1}^{m_2} \delta_{c_{f_{2,j}}}^{\vec{\tau}_{f_{2,j}}} \tag{0.6}$$

where for any $x, \vec{u} \in \mathbb{R}^k$, $\delta_x^{\vec{u}}$ denotes the evaluation functional $\vec{w} \mapsto \langle \vec{w}(x), \vec{u} \rangle$. Using the definition of the reproducing kernel K, one then gets exactly that

$$\langle [\mathcal{M}_1]_d, [\mathcal{M}_2]_d \rangle_{W'} = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} K(c_{f_{1,i}}, c_{f_{2,j}}) \left\langle \vec{\tau}_{f_{1,i}}, \vec{\tau}_{f_{2,j}} \right\rangle . \tag{0.7}$$

The approximation in the case of surfaces is again strictly similar and yields

$$\langle [\mathcal{M}_1]_d, [\mathcal{M}_2]_d \rangle_{W'} = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} K(c_{f_{1,i}}, c_{f_{2,j}}) \left\langle \vec{\nu}_{f_{1,i}}, \vec{\nu}_{f_{2,j}} \right\rangle . \tag{0.8}$$

The approximation error can be easily computed, either directly from the formula, or using the Dirac currents representation. This is postponed to Section 4 since as we will see metrics on currents can be viewed as special cases of metrics on varifolds.

3.4. Examples of registration using currents metrics

We present here some experiments of diffeomorphic registration of surfaces using kernel metrics on currents as data attachment term. Let us briefly precise the problem : given two shapes $\mathcal{M}_1$, $\mathcal{M}_2$, the registration is performed through the minimization of

$$J(\phi) = \gamma \; d(id, \phi)^2 + \| [\phi(\mathcal{M}_1)] - [\mathcal{M}_2] \|_{W'}^2 ,$$

where ϕ denotes a one-to-one mapping belonging to a specific group of diffeomorphisms endowed with a right-invariant metric d (see chapter **??**). In a discrete setting, assuming $\mathcal{M}_1$ and $\mathcal{M}_2$ are meshes, $[\phi(\mathcal{M}_1)]$ and $[\mathcal{M}_2]$ are replaced by their discrete approximations, which reduces the problem to a finite-dimensional LDDMM problem. This is tackled down using a geodesic shooting algorithm, optimizing over initial momentum vectors located at vertices of $\mathcal{M}_1$.

Figure 0.1 shows two examples of registrations of hippocampal cortical surfaces segmented from MRI images as described in [RBT+12]; the first one is acquired with high resolution and the other one with low resolution. Registration was performed using an LDDMM algorithm with kernel metrics on currents as data attachment term. The deformation kernel was chosen to be a sum of four Cauchy kernels with widths $\sigma_V = 10, 5, 2.5, 1.125$ (in mm), while the kernel on currents was chosen to be a Cauchy kernel with width $\sigma_W = 1$. To avoid local minima, two registrations with larger scale ($\sigma_W = 20$ and $\sigma_W = 5$) were performed first and used as initializations. As can be noticed in these examples, registrations using currents metrics for such closed shapes perform very well. Currents are also very efficient when dealing with rough or noisy segmentations, as they locally average orientations through the kernel (see Figure 0.2).

Diffeomorphic registration using currents has been used for several shape analysis applications in medical imaging (see for example [VQGM07, DFP+11b, MKB+11]). It has also been the basis for extensions, mainly for template estimation problems [GJ06, MMY10] and methods for analysing longitudinal datasets [DPT+09].

The need for more sophisticated methods like varifolds and normal cycles comes when dealing with shapes presenting singular features like boundaries, branching points, or high curvature points.

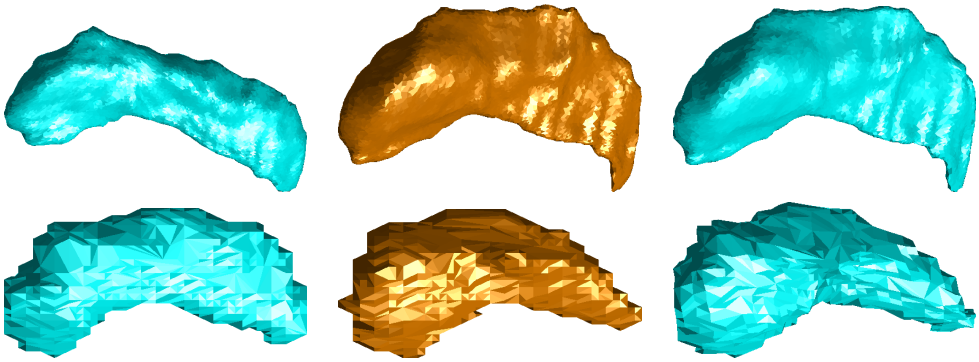
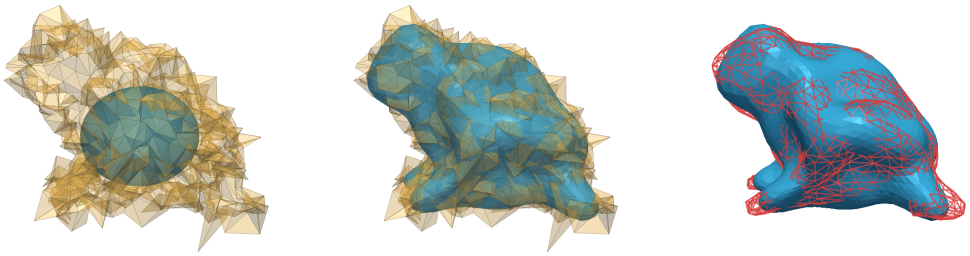


Figure 0.1 Examples of diffeomorphic registrations using kernel metrics on currents. Top row: hippocampal surfaces at high resolution ; bottom row: hippocampal surfaces at low resolution. From left to right: source surface, target surface, deformed source.



Template and target Current matching Result vs Ground truth

Figure 0.2 Registration between a sphere (blue) and a noisy frog (orange). The "true" smooth frog surface (red wireframe) is not used to estimate the registration. It is only used to benchmark with the results.

4. Varifolds

Despite the clear benefits of kernel-based metrics on currents for the problem of shape comparison, there are two important aspects that are worth emphasizing. First, the current representation and metrics are strongly tied to the orientation given to both shapes. While it is often not a difficult issue to find proper and consistent orientation like in the case of closed curves or surfaces for example, some cases are much more involved, if not simply ill-posed. A striking example are fiber bundles with potential crossing or tangle, as we illustrate later in Section 4.4. Another particularity of currents is the linearity of the representation. As we saw, this can be a strength when (properly oriented) data are corrupted by noise since the resulting oscillations tend to cancel out. However, it can also result in artificial destruction or creation of thin structures which may be incorrectly equated to noise in this model, *cf.* [CT13].

Part of these issues can be addressed through the extended framework of varifolds. Varifolds were introduced in geometric measure theory in the works of [Alm66, All72] and first adapted to the context of computational anatomy in [CT13] as a way to eliminate the previous orientation requirement. In the latest work [KCC17], a little more general framework named oriented varifold was proposed, leading to a class of metrics that can, or not, rely on the orientation and allows to recover previous frameworks of currents and varifolds as particular cases. We adopt and summarize this latter approach in what follows. Note that, with a slight abuse of vocabulary, we will call here in short a varifold what corresponds in all rigor to an oriented varifold.

4.1. Representation by varifolds

In the same spirit as Section 3, we will again characterize curves and surfaces through their effect on certain spaces of test functions. With the notations of Section 2:

Definition 3. • A varifold is a distribution on the product $\mathbb{R}^k \times \mathbb{S}^{k-1}$, namely a continuous linear form on a given space W of smooth test functions $\mathbb{R}^k \times \mathbb{S}^{k-1} \to \mathbb{R}$.

- If $\mathcal{M}$ is an oriented curve, we associate the varifold $\{\mathcal{M}\} \in W'$ defined, for any test function $w \in W$, by the line integral:

$$\{\mathcal{M}\}(w) = \int_{\mathcal{M}} w(x, \vec{t}(x)) d\ell(x)\,. \tag{0.9}$$

- If $\mathcal{M}$ is an oriented surface, we associate the varifold $\{\mathcal{M}\} \in W'$ defined, for any test function $w \in W$, by the surface integral:

$$\{\mathcal{M}\}(w) = \int_{\mathcal{M}} w\,(x, \vec{n}(x))\, dS(x)\,. \tag{0.10}$$

The product $\mathbb{R}^k \times \mathbb{S}^{k-1}$ can be interpreted here as the space of position $\times$ tangent space direction and formulas (0.9) and (0.10) amount in representing a curve or a surface as the distribution of its points with unit tangent or normal vector attached. We point out that an important difference with currents is that the dependency of w in $\vec{t}$ (or $\vec{n}$) is not anymore constrained to be linear. However, equations (0.9) and (0.10) still rely, *a priori*, on an orientation of $\mathcal{M}$. We will see that it is not a necessity if the space of test functions W in the previous definitions is carefully chosen in order to recover invariance to the choice of orientation. More fundamentally, the actual specification of W is in fact critical and can lead to a wide range of properties of the representations and metrics, as we discuss in the next section.

4.2. Kernel metrics

As with currents, we will once again focus on the particular class of test functions given by RKHS. In the context of varifolds, an admissible space W is a Hilbert space that is continuously embedded in $C_0(\mathbb{R}^k \times \mathbb{S}^{k-1})$, the space of continuous test functions on $\mathbb{R}^k \times \mathbb{S}^{k-1}$ that decay to 0 at infinity. By the standard theory of RKHS, such a space is equivalently described by a positive and continuous kernel K on $\mathbb{R}^k \times \mathbb{S}^{k-1}$.

In the rest of this section, following the setting of [KCC17], we will restrict to real valued and separable kernels K, namely

$$K((x,u),(y,v)) = K_p(x,y)K_s(u,v) \text{ for all } (x,u),(y,v) \in \mathbb{R}^k \times \mathbb{S}^{k-1},$$

where K_p, K_s are continuous positive kernels on $\mathbb{R}^k$ and $\mathbb{S}^{k-1}$ respectively and such that $K_p(x,y)$ vanish as $\|(x,y)\| \to +\infty$.

With those assumptions, W and its dual W' are Hilbert spaces, and we write $\|\cdot\|_{W'}$ the Hilbert norm on W'. Moreover for any rectifiable curve or surface $\mathcal{M}$, we have that $\{\mathcal{M}\} \in W'$ and the resulting inner product then writes as:

$$\langle \{\mathcal{M}_1\}, \{\mathcal{M}_2\} \rangle_{W'} = \int_{\mathcal{M}_1} \int_{\mathcal{M}_2} K_p(x,y) K_s(\vec{t}_1(x), \vec{t}_2(y)) d\ell(x) d\ell(y) \tag{0.11}$$

for curves or

$$\langle \{\mathcal{M}_1\}, \{\mathcal{M}_2\} \rangle_{W'} = \int_{\mathcal{M}_1} \int_{\mathcal{M}_2} K_p(x,y) K_s(\vec{n}_1(x), \vec{n}_2(y)) dS(x) dS(y) \tag{0.12}$$

for surfaces.

Now, this gives a new class of candidates for fidelity terms which we define by

$$d_W(\{\mathcal{M}_1\}, \{\mathcal{M}_2\})^2 = \|\{\mathcal{M}_1\} - \{\mathcal{M}_2\}\|^2_{W'} = \|\{\mathcal{M}_1\}\|^2_{W'} - 2\langle \{\mathcal{M}_1\}, \{\mathcal{M}_2\} \rangle_{W'} + \|\{\mathcal{M}_2\}\|^2_{W'}. \tag{0.13}$$

In other words, this is the "distance" on curves or surfaces induced by the representation of varifolds and the Hilbert metric on W'. Note that since the representation $\{\mathcal{M}\}$ does not depend on a parametrization of $\mathcal{M}$, the quantity $d_W(\mathcal{M}_1, \mathcal{M}_2)$ is independent of the choice of a

parametrization for $\mathcal{M}_1$ or $\mathcal{M}_2$. From formulas (0.11) and (0.12), we see that, in essence, such metrics are comparing the relative positions of points $\mathcal{M}_1$ and $\mathcal{M}_2$ through the kernel K_p jointly with the relative direction of their tangent spaces as measured by K_s.

As a side note, $\|\cdot\|_{W'}$ gives a metric on the distribution space W' but d_W may still only result in a pseudo-distance as two varifolds $\{\mathcal{M}_1\}$ and $\{\mathcal{M}_2\}$ may coincide in W' even though $\mathcal{M}_1$ and $\mathcal{M}_2$ are distinct. This happens essentially if the space of test functions W is not rich enough. Sufficient conditions on the kernels K_p and K_s to recover a true distance are given in [KCC17], to which we refer for more details. Similarly, specific regularity conditions are necessary to guarantee existence of solutions to registration problems involving those fidelity terms.

Remark 1. In shape analysis, another usual property of metrics that is very often desired is **invariance to rigid motions**, namely for all $(R, a) \in \mathrm{O}_k \rtimes \mathbb{R}^k$ and all $\mathcal{M}_1, \mathcal{M}_2$

$$\langle\{R\mathcal{M}_1 + a\}, \{R\mathcal{M}_2 + a\}\rangle_{W'} = \langle\{\mathcal{M}_1\}, \{\mathcal{M}_2\}\rangle_{W'}.$$

Within the setting presented here, this can be satisfied easily by restricting K_p to the class of **radial** kernels $K_p(x, y) = \rho(\|x - y\|^2)$ with $\rho : \mathbb{R}_+ \to \mathbb{R}$ and K_s to the class of **zonal** kernels on $\mathbb{S}^{k-1}$, *i.e* such that $K_s(\vec{u}, \vec{v}) = \gamma(\langle\vec{u}, \vec{v}\rangle)$ with $\gamma : [-1, 1] \to \mathbb{R}$.

Let us now consider a few specific examples of kernels, which all result in rigid-invariant distances between shapes. For K_p, common choices include Gaussian kernels $K_p(x, y) = e^{-\|x-y\|^2/\sigma^2}$ or Cauchy kernels $K_p(x, y) = \frac{1}{1+\|x-y\|^2/\sigma^2}$, where in both cases σ is a scale parameter that determines the spatial sensitivity of the metric. Sums of kernels with different scales can be also used in order to define multiscale distances. The choice of the spherical kernel K_s has very important impact on the resulting metric, we discuss a few special cases below.

Example 1. If $K_s(\vec{u}, \vec{v}) = 1$ then the metric is essentially insensitive to the tangent or normal vectors' components in the two shapes. Equivalently, it can be interpreted as viewing shapes as standard distributions on $\mathbb{R}^k$ and comparing them through the metrics obtained by the single kernel K_p. This exactly corresponds to the simplest model of *measures* that was introduced in [GTY04] originally to treat point clouds.

Example 2. If $K_s(\vec{u}, \vec{v}) = \langle\vec{u}, \vec{v}\rangle$, *i.e* the restriction of the linear kernel of $\mathbb{R}^k$ to $\mathbb{S}^{k-1}$, then we find, as a particular case, the metrics based on currents of formulas (0.3) and (0.4).

Example 3. Another possible choice is $K_s(\vec{u}, \vec{v}) = e^{-2\langle\vec{u},\vec{v}\rangle/\sigma_s^2}$ which is the restriction of a Gaussian kernel of width $\sigma_s > 0$ on $\mathbb{S}^{k-1}$. Such a kernel induces non-linearity with respect to u and v which, as we will see, leads to important differences with currents.

Example 4. When K_s is chosen to be orientation-invariant, *i.e* if $K_s(\vec{u}, \vec{v}) = K_s(\vec{u}, -\vec{v}) = K_s(-\vec{u}, \vec{v})$, then interestingly the metric defined by formula (0.11) or (0.12) is completely independent of the orientation given to vectors $\vec{t}(x)$ or $\vec{n}(x)$. In that case, orienting $\mathcal{M}_1$ or $\mathcal{M}_2$ is unnecessary and one can basically select any of the two possible unit vector at each point x. This leads to the particular class of metrics on varifolds that were considered in [CT13]. Examples of such symmetric kernels, besides the trivial one of Example 1, are $K_s(u, v) = \langle\vec{u}, \vec{v}\rangle^2$ which is known as the Binet kernel on the sphere or $K_s(\vec{u}, \vec{v}) = e^{-2\langle\vec{u},\vec{v}\rangle^2/\sigma_s^2}$ (the squared scalar product makes the kernel orientation-invariant).

This framework for constructing fidelity terms also enables simple multiscale registration strategies. One approach which is used in some of the examples presented previously and below consist in solving sequentially several registration problems using finer and finer kernel sizes. An-

other possible approach is to directly combine different scales in the fidelity metric by considering sums of kernels for K_p and/or K_s, which is inspired from a similar idea proposed in the context of multiscale deformation models by [RVW+11].

4.3. Discrete model

The discrete model for currents' representation and metrics introduced in Section 3.3 can be easily adapted to the more general situation of varifolds. Adopting the same notations, given two curve or surface meshes, one can approximate the double integrals in the kernel metrics given by equations (0.11) and (0.12) respectively as:

$$\langle\{\mathcal{M}_1\},\{\mathcal{M}_2\}\rangle_{W'} \approx \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} K_p(c_{f_{1,i}},c_{f_{2,j}})K_s(\vec{t}_{f_{1,i}},\vec{t}_{f_{2,j}})\|\vec{\tau}_{f_{1,i}}\|\,\|\vec{\tau}_{f_{2,i}}\| \tag{0.14}$$

and

$$\langle\{\mathcal{M}_1\},\{\mathcal{M}_2\}\rangle_{W'} \approx \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} K_p(c_{f_{1,i}},c_{f_{2,j}})K_s(\vec{n}_{f_{1,i}},\vec{n}_{f_{2,j}})\|\vec{\nu}_{f_{1,i}}\|\,\|\vec{\nu}_{f_{2,i}}\|. \tag{0.15}$$

Note once again that if $K_s(\vec{u},\vec{v}) = \langle\vec{u},\vec{v}\rangle$ then we find the same expressions as in equations (0.7) and (0.8). Recall also that the orientation of vectors $\vec{t}_{f_i}$ or $\vec{n}_{f_i}$ depends on the ordering of vertices in each face, which in general needs to be defined consistently across faces. Yet with the orientation-invariant kernels of Example 4, orienting the mesh is unneeded as any orientation at each face gives the same value in (0.14) and (0.15).

These discrete formulas can be also interpreted as the varifold inner products between the approximations of $\{\mathcal{M}_1\}$ and $\{\mathcal{M}_2\}$ as finite combinations of Diracs:

$$\{\mathcal{M}_1\} \approx \{\mathcal{M}_1\}_d = \sum_{i=1}^{m_1}\|\vec{\nu}_{f_{1,i}}\|\,\delta_{(c_{f_{1,i}},\vec{t}_{f_{1,i}})} \quad\text{and}\quad \{\mathcal{M}_2\} \approx \{\mathcal{M}_2\}_d = \sum_{j=1}^{m_2}\|\vec{\nu}_{f_{2,j}}\|\,\delta_{(c_{f_{2,j}},\vec{t}_{f_{2,j}})}$$

in the case of curves and with similar equivalent expressions for surfaces. In the above equations, a Dirac varifold $\delta_{(x,\vec{u})}$ is the linear functional defined for any $w \in W$ by $\delta_{(x,\vec{u})}(w) = w(x,\vec{u})$. Then, for any polygonal curve or polyhedral surface, one can show (*cf.* [CT13] and [KCC17]) that, with the assumptions on kernels introduced in Section 4.2 and the extra assumptions that both kernels are C^1, there exists a constant $C \geq 0$:

$$\|\{\mathcal{M}\} - \{\mathcal{M}\}_d\|_{W'} \leq C|\mathcal{M}|\max_i \operatorname{diam}(f_i) \tag{0.16}$$

where $|\mathcal{M}|$ is the total length or area of $\mathcal{M}$ and $\operatorname{diam}(f_i)$ is the diameter of the face f_i. In other words, the approximation error for the varifold norm is controlled by the maximum diameter of the mesh faces and will thus be small if the mesh is sufficiently refined. Similar approximation bounds then follow for the discrete metric formulas of (0.14) and (0.15). In addition, the gradient of the metric and distance with respect *e.g.* to the vertices' positions in $\mathcal{M}_1$ is easily obtained by simple chain rule differentiation.

4.4. Examples and applications

We now illustrate several different properties and potential advantages or downsides of the previous metrics through a few examples of diffeomorphic registration on curves and surfaces. As previously, we rely on the LDDMM model for the deformation framework coupled here with general varifold data attachment terms. In all these experiments, spatial kernels K_p are chosen Gaussian (we denote by σ the width parameter) and we focus primarily on the effects of the second kernel K_s for which we compare the choices of Examples 2, 3 and 4.

Figure 0.3 shows a fairly challenging registration experiment on closed curves. With currents (linear kernel), notice the appearance of degenerate structures and the fact that the two humps are

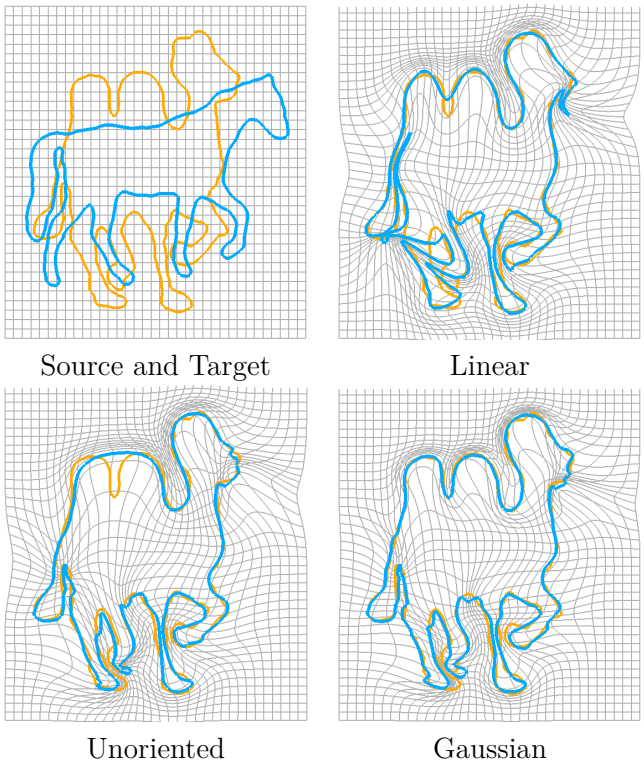


Figure 0.3 Diffeomorphic registration of a template horse (blue) to the target (orange). The estimated deformed template is shown for some varifold metrics using various kernels K_s.

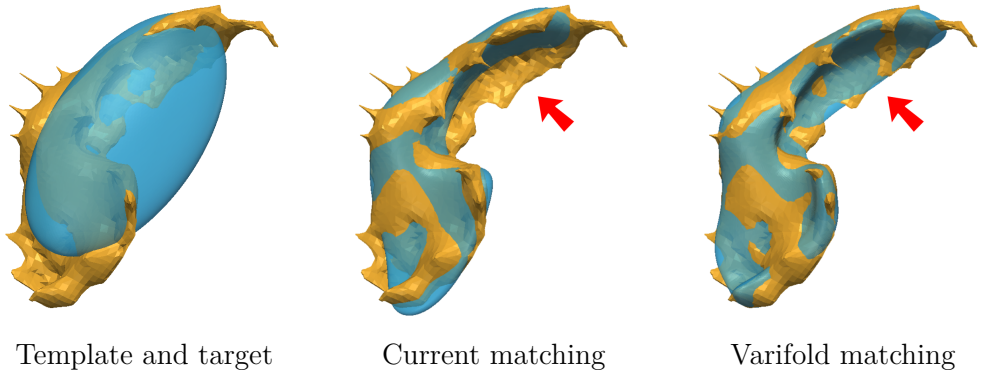


Figure 0.4 Registration between an ellipsoid and an hippocampal surface (left) with the linear kernel (middle) and the Binet kernel (right). The hippocampal surface is segmented from a MRI and contains several segmentation artifacts (spikes).

not well recovered: this is a downside consequence of the cancellation effect that was discussed above. It is again very specific to the linearity of this model as these effects do not occur with the two other metrics. Orientation-invariant kernels like the Binet kernel of Example 4 still display difficulties in recovering the convoluted double hump by instead creating a single 'average' hump. In this particular example where orientation is well-defined and relevant, oriented but nonlinear kernels like the spherical Gaussian of Example 3 achieve the most accurate registration as evidenced in the figure.

The previous observations can clearly have impact on registration of real data as well. We show one example among others in Figure 0.4 where an ellipsoid (blue) is registered on an hippocampal surface segmented from an MRI (orange). We compare the estimated registrations using the linear and Binet kernels for the orientation. There are differences to be noticed between the two

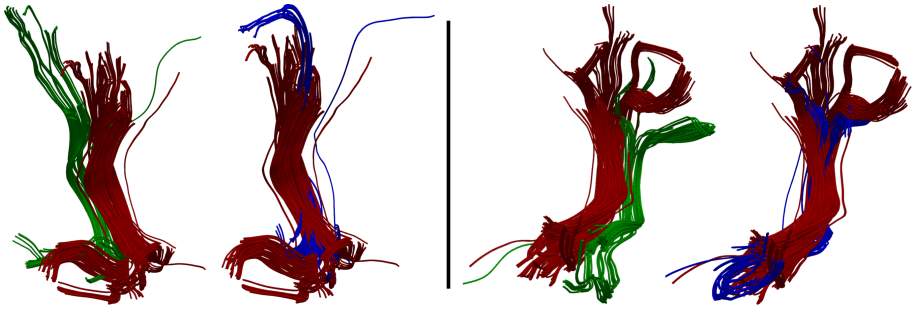


Figure 0.5 Registration of white matter fiber bundles using the Binet kernel for the fidelity term. Two views are shown in which the template is in green, the target in red and the registered template in blue.

results, most notably around the thin part of the hippocampal surface where the overlapping of the registered surface is less accurate in the case of the linear kernel compared to the matching obtained with the Binet kernel (red arrow). This is a clear manifestation of the cancellation of thin shape structures in the representation of currents.

Finally, orientation-invariant fidelity terms can prove very useful in the case of fiber bundle data for which defining an orientation of each fiber that is consistent can be particularly cumbersome and even ill-posed in some situations. We show an example of registration between two white matter fiber tracts in Figure 0.5 using the unoriented Binet kernel.

Besides theses cases, the benefits of oriented or unoriented varifold metrics have also been put to use in several other applications to shape analysis of medical data which include cortical surfaces [RLLS16], complexes of subcortical surfaces [DPC$^+$14] or lung vessels [PCD$^+$16].

5. Normal cycles

Currents and varifolds provide representations for oriented or unoriented shapes which rely on first order information of the shape (*i.e* tangential or normal directions). Depending on the applications, it may be useful to have a second-order model, that takes into account the curvature. For this purpose, we define in this section an alternative representation based on *normal cycles*. This is a summary of the work presented in [RG16].

Rather than the shape itself, the idea of normal cycles is to consider its unit normal bundle, *i.e* the shape attached with its normal vectors at each point. More precisely, the normal cycle is the current associated with the unit normal bundle. It has been introduced first by Zähle [Zä86] as a generalization of curvatures for sets with singularities.

In this section, we propose to briefly present the representation of shapes with normal cycles. We will see that this representation takes into account the boundary of the shapes (*e.g* extremities and branching points for curves), and thus is sensitive to topological change. Moreover, it is sensitive to high-curvature regions of the shape. By introducing kernel metrics on normal cycles (as for currents and varifolds), we are once again able to obtain an explicit form for the metric between discrete shapes represented as normal cycles.

5.1. Differential forms and currents

Similarly to currents and varifolds, normal cycles are defined through their evaluation on test functions. To define adequate spaces of test functions in that case, we need to recall some notions about differential forms and currents in more general dimensions than the setting of Section 3.

Definition 4. . Let $k \in \mathbb{N}^*$.

- A m-differential form, or simply m-form, is a mapping $x \mapsto \omega(x)$ from $\mathbb{R}^k$ to the space of alternating m-linear forms of $\mathbb{R}^k$.
- A m-current in $\mathbb{R}^k$ is a continuous linear mapping from the space of continuous m-forms to $\mathbb{R}$.

The space of test functions that we will consider in the following will be $\Omega_0^{k-1}(\mathbb{R}^k \times \mathbb{S}^{k-1})$, the space of continuous $k-1$-differential forms on $\mathbb{R}^k \times S^{k-1}$, vanishing at infinity.

Remark 2. For curves in $\mathbb{R}^2$ ($k=2$, $m=1$) or surfaces in $\mathbb{R}^3$ ($k=3$, $m=2$), one can canonically identify a m-differential form to a continuous vector field, and we retrieve the definition given in Section 3.

5.2. Unit normal bundle and normal cycle

To keep the discussion as simple as possible, we will only consider curves and surfaces in $\mathbb{R}^3$. However, the notion of normal cycles can be generalized to any m-dimensional surface in $\mathbb{R}^k$.

Definition 5 (Unit normal bundle and normal cycle)**.** Consider $\mathcal{M}$ a compact C^2 curve or surface (possibly with a boundary)

- The *unit normal bundle* of $\mathcal{M}$ is
$$\mathcal{N}_\mathcal{M} = \{(x, n) \mid x \in \mathcal{M}, \vec{n} \in \mathrm{Nor}(\mathcal{M}, x)\}$$
where $\mathrm{Nor}(\mathcal{M}, x)$ is the set of all the unit normal vectors of $\mathcal{M}$ at point x.
- The *normal cycle* of $\mathcal{M}$, denoted $N(\mathcal{M})$ is the current associated with $\mathcal{N}_\mathcal{M}$:
$$N(\mathcal{M})(\omega) := [\mathcal{N}_\mathcal{M}](\omega) = \int_{\mathcal{N}_\mathcal{M}} \langle \omega(x, \vec{n}), \tau_{\mathcal{N}_\mathcal{M}}(x, \vec{n}) \rangle dS(x, \vec{n}).$$
where $\tau_{\mathcal{N}_\mathcal{M}}(x, \vec{n})$ is the unit tangent vector to $\mathcal{N}_\mathcal{M}$ at point $(x, \vec{n})$ and $\omega \in \Omega_0^{k-1}(\mathbb{R}^k \times \mathbb{S}^{k-1})$.

Remark 3. $\mathcal{N}_\mathcal{M}$ is a 2-dimensional surface in $\mathbb{R}^3 \times \mathbb{S}^2$ independently of the dimensionality of $\mathcal{M}$. Indeed, it can be canonically associated with an ε-expansion of the initial shape, for ε small enough. See [RG16] and the references therein for more details. Thus, the integration over the unit normal bundle is a surface area integration. Hence the notation $dS(x, \vec{n})$ in the definition of normal cycles.

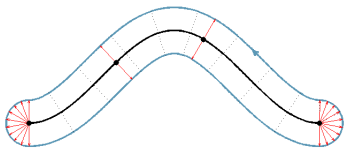


Figure 0.6 Illustration of the unit normal bundle for a regular non closed curve in the plane. The curve is in black, the unit normal vectors associated with four points are represented as red arrows, and the resulting unit normal bundle is represented in blue, with its canonical orientation. Note that this representation is only illustrative, as the true normal bundle belongs to the space $\mathbb{R}^2 \times \mathbb{S}^1$ in this case.

The unit normal bundle of a smooth curve with extremities is depicted in Figure 0.6. Note that extremities have non-negligible contributions in the normal bundle and thus are taken into account in the normal cycle. This is a major difference with currents or varifolds. Note also that since the unit normal bundle can be depicted as a closed hypersurface (as in Figure 0.6), it has a canonical orientation that is independent of the orientation of the initial shape.

5.3. Normal cycles for discrete curves or surfaces

The unit normal bundle is well-defined for smooth enough sets, specifically sets with positive reach (see [Fed59], section 4). This class of sets encompasses compact C^2 submanifolds with boundary (which we will refer to as C^2 sets in the following). However, it is, a priori, not well defined in

general for union of C^2 sets, as for instance with discrete curves or surfaces. Fortunately, it is possible to extend the notion of normal cycles for such sets using the following *additive property*:

Proposition 1 (Additive property, [Zä87]). *Consider $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ and $\mathcal{M}_1 \cap \mathcal{M}_2$ are C^2 sets (possibly with boundary). Then we can define*

$$N(\mathcal{M}) := N(\mathcal{M}_1) + N(\mathcal{M}_2) - N(\mathcal{M}_1 \cap \mathcal{M}_2)$$

This property allows to define the normal cycles for shapes that are the union of compact C^2 sets with boundaries. This case encompasses the one of discrete curves or surfaces. Such a normal cycle corresponds to a generalized unit normal bundle which is illustrated in Figures 0.7 and 0.8.

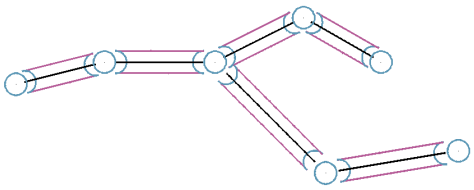


Figure 0.7 Representation of the generalized unit normal bundle of a blue discrete curve. For curves we observe two parts for the normal bundle: a cylindrical part (in purple), associated with the edges and a spherical part (in blue) associated with the vertices.

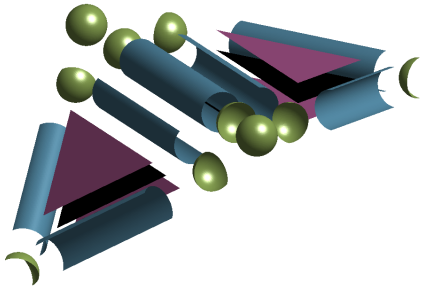


Figure 0.8 Representation of the generalized unit normal bundle of triangles (in black). It comprises three parts: a planar part (in purple) associated with the interior, a cylindrical part (in blue) associated to the edges and a spherical part (in yellow) associated with the vertices.

5.4. Kernel metrics on normal cycles

Since normal cycles are currents associated with unit normal bundles, we can again rely on the theory of RKHS presented in Sections 3.2 and 4.2. It is left to design a scalar kernel K on $\mathbb{R}^3 \times \mathbb{S}^2$, that will be a product of a spatial kernel K_p and a spherical kernel K_s. This has already been presented in Section 4.2.

In this framework, the inner product between two curves or two surfaces represented as normal cycles takes the form

$$\langle N(\mathcal{M}_1), N(\mathcal{M}_2)\rangle_{W'} = \int_{\mathcal{N}_{\mathcal{M}_1}} \int_{\mathcal{N}_{\mathcal{M}_2}} K_p(x,y) K_s(\vec{u},\vec{v}) \langle \tau_1(x,\vec{u}), \tau_2(y,\vec{v})\rangle dS(x,\vec{u}) dS(y,\vec{v}) \quad (0.17)$$

where $\tau_1(x,\vec{u})$ is the tangent vector of the unit normal bundle at point $(x,\vec{u})$. Notice that this expression is true only for compact C^2 curves or surfaces (possibly with boundary). It can be extended to discrete curves or surfaces through the previous additivity property. Let us add a few remarks. First of all, one can notice that this expression is similar for curves or surfaces: the normal bundle is indeed 2-dimensional in $\mathbb{R}^3 \times \mathbb{S}^2$, independently of the actual dimension of the shape. Secondly, the contribution of normal vectors through the kernel K_s but also through the tangent vector to the normal bundle τ_1 shows that curvature is taken into account by those metrics in contrast to the previous approaches of currents or varifolds.

The choice of the spherical kernel is mainly driven by the possibility to explicitly compute such inner products between two discrete shapes. With discrete curves, those metrics have been

implemented with constant (as in example 1), linear (example 2) or Sobolev kernels (*i.e* the reproducing kernel of some Sobolev space on the sphere $\mathbb{S}^2$ which has explicit expansion in the spherical harmonics basis). In the case of surfaces however, current implementations are restricted to the constant spherical kernel, even though it provides less theoretical guarantees (namely, the space of test functions W is not rich enough to ensure that the associated pseudo-distance $d_W(\mathcal{M}, \mathcal{M}') = ||N(\mathcal{M}) - N(\mathcal{M}')||^2_{W'}$ is a real distance, see again [RG16] for a precise discussion on these topics).

5.5. Discrete inner product

In the case of discrete curves or surfaces, as shown in Figure 0.8, the unit normal bundle consists of several components: spherical (in green) associated to vertices, cylindrical (in red) associated to edges, and for surfaces planar components (in light blue) associated to faces. The expression of the full inner product is greatly simplified thanks to the following result:

Proposition 2. *The spherical, cylindrical and planar components are orthogonal with respect to the kernel metric on normal cycles defined in Section 5.4 by formula* (0.17)

Numerical computation involves integrating over the planar, cylindrical and spherical parts. We approximate the spatial integration as for currents or varifolds. The integration over the normal part is explicitly calculated. For the sake of simplicity, we here only express it for the constant spherical kernel ($K_s(\vec{u}, \vec{v}) = 1$), equivalent derivations for the linear or Sobolev kernels may be found in [RG16]. In the case of curves, it reads:

$$\langle N(\mathcal{M}_1), N(\mathcal{M}_2)\rangle_{W'} = \frac{\pi^2}{4} \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} K_p(x_i, y_j) \left\langle \vec{A}_i, \vec{B}_j \right\rangle \tag{0.18}$$

where $\vec{A}_i = \sum_k f^i_k/|f^i_k|$ is the sum of the normalized edges with x_i as vertex, and oriented outward from x_i. For surfaces:

$$\begin{aligned}\langle N(\mathcal{M}_1), N(\mathcal{M}_2)\rangle_{W'} = &\frac{\pi^2}{4} \sum_{i=1}^{n_e} \sum_{j=1}^{m_e} K_p(c_i, d_j) \left\langle \vec{f}_i, \vec{g_j} \right\rangle \left\langle \sum_{\{T | f_i \text{ edge of } T\}} \vec{n}_{T,\vec{f}_i}, \sum_{\{T' | g_j \text{ edge of } T'\}} \vec{n}_{T',\vec{g_j}} \right\rangle \\ &+ \frac{\pi^2}{4} \sum_{\substack{x_i \text{ vertex} \\ \text{of } \partial\mathcal{M}_1}} \sum_{\substack{y_j \text{ vertex} \\ \text{of } \partial\mathcal{M}_2}} K_p(x_i, y_j) \left\langle \vec{A}_i, \vec{B}_j \right\rangle\end{aligned} \tag{0.19}$$

where the first double sum is a double loop on the edges (there are n_e edges in $\mathcal{M}$ and m_e in $\mathcal{M}_2$) and where $\vec{A}_i = \sum_k f^i_k/|f^i_k|$ is the sum of the normalized edges of the border, with x_i as vertex, and oriented outward from x_i, c_i is the middle of the edge $\vec{f}_i$. $\vec{n}_{T_i,\vec{f}_i}$ is the normal vector of the triangle T_i such that $n_{T_i,\vec{f}_i} \times \vec{f}_i$ is oriented inward for the triangle T. B_j is the similar notation as A_i for the triangulation $\mathcal{M}_2$. Note that with the constant normal kernel, the inner product involves only the cylindrical part, *i.e* quantities associated with the edges of the triangles (and thus, with the discrete mean curvature).

Since the expression of the inner product (and thus the metric) is explicit in the discrete case, it is easy to obtain the gradient with respect to the vertices of $\mathcal{M}_1$ by a chain rule.

Remark 4. With the linear spherical kernel, the inner product involves the spherical part (associated with the vertices and thus the Gaussian curvature) and the planar part. Interestingly, this planar part then corresponds to the inner product obtained from the varifold representation with the Binet spherical kernel introduced in Section 4.2.

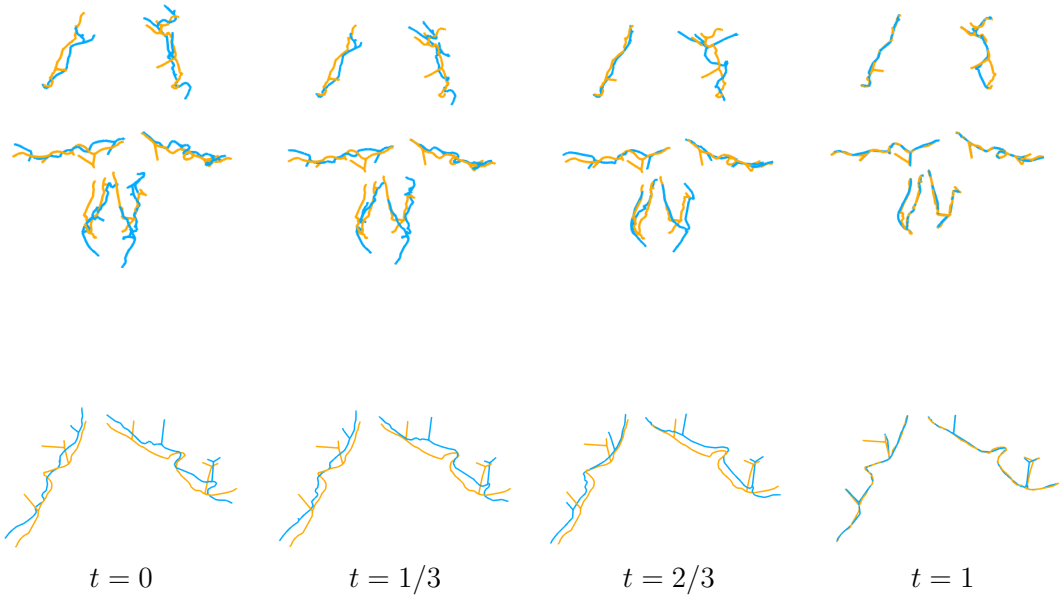


Figure 0.9 Top row: registration of blue to orange brain sulci. The deformation kernel is a Cauchy kernel of width $\sigma_V = 20$. The data attachment term is normal cycles with a Sobolev kernel of order 3 for K_s and Gaussian spatial kernel for K_p. Bottom row: zoom on the registration of two sulci. Each row shows the evolution of the deformation with time. One can observe that since normal cycles take into account the extremities, the matching is convincing.

The crucial difference between normal cycles and currents or varifolds is that any part of the discrete shape has a non negligible component in the unit normal bundle and will be taken into account by the metric. Hence, there is an explicit term associated with the boundaries of shapes. For registration purpose, this feature will enforce the matching of boundaries, corners but also of branching points.

5.6. Examples and applications

This section aims to illustrate some of the properties of normal cycles.

First, we show an example on curves. The data consist of brain sulcal curves that were automatically segmented and labelled from anatomical Magnetic Resonance Imaging (MRI) brain images, following the method described in [ACG+11]. We chose two individuals and six labelled corresponding sulcal curves for each individual. We thank Guillaume Auzias for extracting and providing us the dataset of sulcal curves used in our experiments. The data fit in a box of size $120 \times 140 \times 110$.

The matching is performed with a single deformation, but 6 data attachment terms with normal cycles: one for each pair of corresponding sulci. The details of the registration procedure are precised in Figure 0.9. The matching is complex since the number of branching points is not necessarily the same for corresponding curves, and two curves to match can be really twisted from one to another. Moreover, the fact that a single deformation is required for the whole brain implies high local variations. In Figure 0.9, we present the registration with normal cycles and Sobolev normal kernel. The visualization of this three-dimensional configuration is not easy, but the end points and corresponding branching points are well matched when possible (we recall that there is not always a corresponding branching point). Moreover, the registration driven by normal cycles allows complex local deformation (even though it is expensive) to reduce the data attachment term.

In the bottom row, we present a zoom on two sulci to showcase the properties of a registration with normal cycles. This specific example shows all the benefit that one can expect from the metric on normal cycles. The natural consideration of the extremities and the branching points provides

a convincing registration, even though this implies a deformation with high local variation. This is even more striking on the left sulci. The deformation is a good compromise, even though there are not the same number of branching points in the source and target.

As a second real data example, we consider the registration of retinal surfaces which are shown in Figure 0.10. Those retina are surfaces in 3D segmented from Optical Coherence Tomography images. The typical size is 8×8mm and each retina is sampled with approximately 5000 points. The data acquisition and preprocessing is described in [LFF+13]. In our simulations, we use a Gaussian deformation kernel K_V with $\sigma_V = 0.6$. All the details of the matching are in Figure 0.10. These surfaces have boundaries and corners which will be seen as region with singularities for the kernel metric on normal cycles. This is not the case for the varifolds metric, which makes the registration of the corresponding corners more involved. The matching of the borders is more accurate with normal cycles while providing a much more regular deformation (see Figure 0.10).

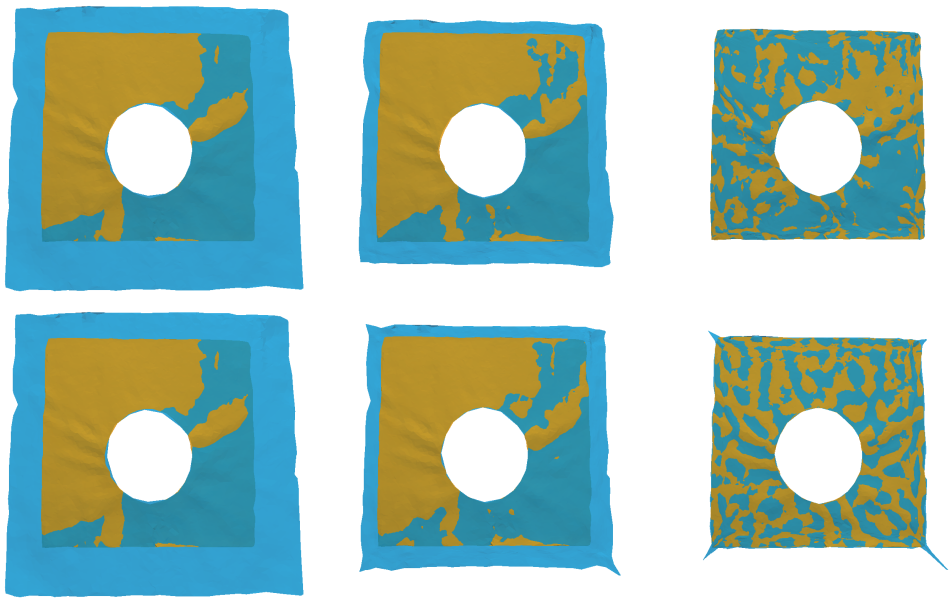


Figure 0.10 Each row represents the matching of two retina surfaces in 3D with kernel metric on normal cycles (top) and varifolds (bottom). The target shape is in orange and the source shape is in blue. Each shape has 5000 points. For the varifolds metric, the geometric kernel is Gaussian. The kernel K_s is $K_s(\vec{u}, \vec{v}) = \langle \vec{u}, \vec{v} \rangle^2$ as in example 4, so that no additional parameter is involved. The same parameters are used for each data attachment term. K_V is a Gaussian kernel with $\sigma_V = 0.6$.

6. Computational aspects

6.1. Fast Kernel Computations

The metrics presented in this chapter all use embeddings of shapes in some RKHS. Such "kernels methods" are convenient tools as they allow to define simple yet intuitive distances between shapes by choosing meaningful kernels tailored to particular applications. Moreover, kernel methods lead to natural and efficient discrete approximations of the continuous models which are amenable to treat real datasets.

As we have seen in the previous sections, the considered metrics are then defined based on the inner product resulting from the RKHS structure. The explicit formulas (0.7), (0.8) or (0.14),

```
1  import numpy as np # load numpy library
2
3  def squared_distances(x, y):# matrix of the squared norms
4    return np.sum((x[:,np.newaxis,:] - y[np.newaxis,:,:]) ** 2, axis=2)
5
6  M = np.exp(-squared_distances(c_f1, c_f2) / (sigma*sigma))
7  S =  np.sum(tau_f1  * (np.matmul(M,tau_f2)) # final results
```

Figure 0.11 An implementation of Gaussian kernel sum (0.21) in Python using `numpy`

(0.15) or (0.18), (0.19), of these various inner products all involve computations of the form:

$$S = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} F_{i,j}, \tag{0.20}$$

where $F_{i,j}$'s are real numbers depending (possibly non linearly) on some known quantities such as points position (the variables $c_{f_i} \in \mathbb{R}^k$) or tangent spaces orientation (variables $\vec{\tau}_{f_i}, \vec{\nu}_{f_i} \in \mathbb{R}^k$), *etc.*... Computing a single $F_{i,j}$ in (0.20) requires, in general, several operations such as kernel evaluations. It yields to a total computational complexity of $O(m_1 m_2)$ to evaluate a single distance between two shapes (m_1 and m_2 being the number of segments/triangles in the shapes). This quadratic complexity is one of the main obstacles to apply kernel based distances on real data. Indeed, many practical cases involve data where m_1 and m_2 can reach an order of magnitude up to 10^7 or more.

There are two main strategies to efficiently perform computations of kernel based distances: "brute force" exact computations using parallel architectures and methods computing an approximate value of S but with a much lower complexity (typically quasi-linear instead of quadratic). We discuss some possible methods in the remainder of this section.

6.1.1. Exact computations

By exact computations, we here mean that all $F_{i,j}$'s are evaluated up to the machine precision before computing the full double sum in (0.20). The methods described in this section can be used to compute any kernels.

Linear Algebra library

The first natural way to perform exact kernel computations is to use one of the standard linear algebra libraries (such as BLAS, Armadillo, *etc.*...). Every scientific programming language now has bindings to state-of-the-art linear algebra libraries that are able to take advantage of multiple cores in modern central processing unit (CPU).

The basic idea here is to create an array of size $m_1 \times m_2$ containing all values $F_{i,j}$ and perform two summation (reduction) steps. For instance, the current inner product (0.21) below may be written in Python as in Figure 0.11 where, under the hood, `numpy` calls BLAS optimized library to perform the matrix multiplication and the final reduction step of line 7.

This solution is competitive in term of speed when the sizes m_1 and m_2 of the data at hand are small (up to 10^3 typically as shown Figure 0.12). The bottleneck is the memory footprint since storing the $m_1 \times m_2$ array `M` of floating-points numbers (line 6 of Figure 0.11) may be impossible.

Graphics Processing Unit (GPU)

Brute force computations using massively parallel architectures such as GPU are a very efficient way to compute kernel based distances when the sizes m_1 and m_2 are moderate (in the range of 10^3 to 10^6). Using a GPU for very small values of m_1 and m_2 is in general counter productive as there may be some overhead due to memory transfers between the standard memory and the GPU memory. This being said, the performances of an implementation highly depend on the

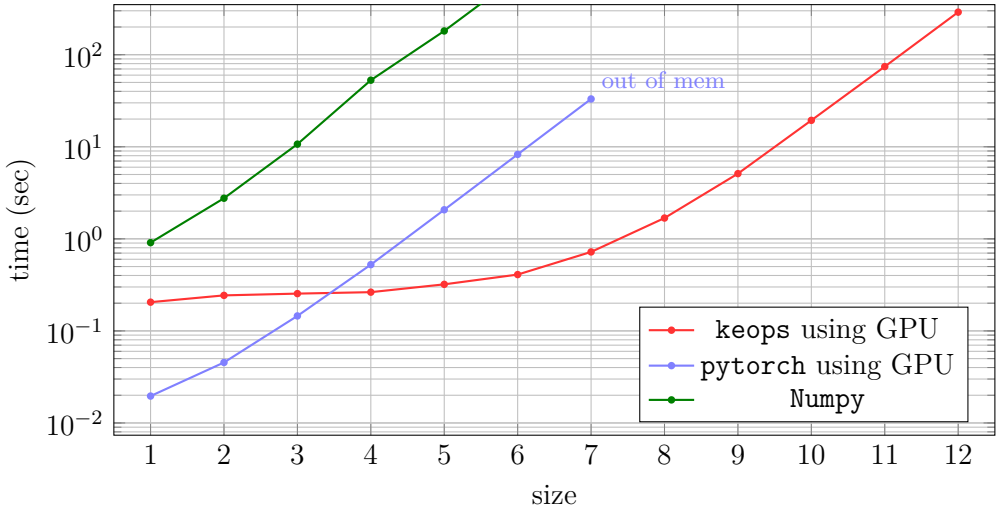


Figure 0.12 Average time to compute 200 Gaussian kernel sums depending on the size. The size increase exponentially: at size i we have $m_1 = 200 \times 2^i$ and $m_2 = 300 \times 2^i$.

environment and are subject to possible changes in the future.

Some high-level languages allow to execute (transparently for the end-user) the code of Figure 0.11 on a GPU. Nevertheless, the memory limitations remain the same. In fact, they are even worst as the size of the memory on a GPU is usually much smaller than the standard memory size. In Figure 0.12 we can see that a Nvidia GPU Tesla P100 with 16Go of RAM ran out of memory for $m_1, m_2 > 26000$.

To overcome this problem, it is possible to use a so-called "tile implementation" method. The idea is to divide the computations of the $m_1 m_2$ terms $F_{i,j}$ in small batches (the tiles) and aggregate the results on the fly. In practice, the computations of the $F_{i,j}$ and the reduction step are then made on a single batch at a time reducing the amount of memory needed to compute S and without loosing accuracy. The memory architecture of GPUs is particularly well suited for this kind of job as it is possible to use a low latency memory called shared memory. Although, this tile implementation is well documented, there is currently few high level libraries that can be used for general matrix multiplications or operations as formula (0.20). A notable exception being `keops` [CFG18] which is a Cuda/C++ software with Python, Matlab and R bindings designed to compute operations on kernels. Performances are shown Figure 0.12.

6.1.2. Approximate methods

Even with a strong parallelized implementation using GPU, exact methods to compute kernel based distances may be impossible to apply when m_1 and m_2 are greater than 10^7. The quadratic complexity being simply too high in this range of values.

We present here two methods that are able to compute an approximated value of S of formula (0.20) with a controlled error in at most a sublinear complexity. Unfortunately, approximation methods may not be used with any general kernel. We will then assume here that the problem has the following form:

$$S = \sum_{i=1}^{m_1} \langle \vec{\tau}_{f_{1,i}}, [M\vec{\tau}_{f_2}]_i \rangle \quad \text{where} \begin{cases} M = [\rho(\|c_{f_{1,i}} - c_{f_{2,j}}\|^2)] \in \mathrm{M}_{(m_1,m_2)} \\ \text{and} \\ \vec{\tau}_{f_2} = [\tau_{f_{2,j}}] \in \mathrm{M}_{(m_2,k)} \end{cases} \tag{0.21}$$

where ρ may be a Gaussian or Cauchy function. In formula (0.21), for any $1 \leq i \leq m_1$ we have denoted $[M\vec{\tau}_{f_2}]_i$ the vector in $\mathbb{R}^k$ whose coordinates are given by the i-th row of $[M\vec{\tau}_{f_2}]$. Thence, computing S can be done at the price of a convolution (the matrix product $M\vec{\tau}_{f_2}$ which is the

costly part) followed by reduction step (the sum over the scalar product in $\mathbb{R}^k$).

Grid method and Non-uniform Fast Fourier transform (NFFT)
These two methods can be employed to compute approximations of convolutions with radial scalar kernels as used in equation (0.21). Both approaches rely on the fact that convolutions can be written as a pointwise product in the frequency domain. The difficulty being here that points do not *a priori* lie on an equispaced grid and the Fast Fourier Transform (FFT) can not be used directly. The idea is then to interpolate the data on a regular grid (first step known as gridding step) to be able to apply the FFT for the computation of convolutions (second step). Finally, the results are evaluated at the initial data location (third step). The total cost of the method is then quasi linear in terms of number of points in the shape and grid size. Nevertheless, a major drawback is that the grid size explodes with the dimension of the ambient space limiting the range of applications to $k \leq 3$.

The grid method [Dur09] has been implemented in the software `Deformetrica` [DC18] for Gaussian kernel and is used to compute current and varifolds norms with Binet kernels. The NFFT could be used for similar tasks and two possible choices for implementations are [KKP09] and [GL04].

Fast Multipole Methods (FMM)
FMM [GR87] are numerical methods coming from physics to approximate computations involving kernels. Original motivations were to compute numerical solutions of the n-body problem or solutions of some partial differential equations. Although, to the best of the authors' knowledge, these methods are not being currently used in our context, it is worth mentioning FMM as a potential approach to make computations of formulas similar to (0.21) with m_1, m_2 above 10^7 numerically tractable.

The idea of FMM is to perform calculations in a hierarchical way by splitting the ambient space $\mathbb{R}^k$ into adaptive sets of sub-regions whose sizes depend on the density of points in the space. This partition is then used to make a batch computation of the sum. The evaluation of an admissible kernel in each batch can be approximated with a controlled precision. Finally, the approximated results are aggregated in a divide-and-conquer fashion reducing the overall computational complexity to a sub-linear complexity in terms of the number of points. Moreover, the memory needed to perform the overall calculation is also significantly decreased.

Let us finally mention a useful particular case of FMM: the Fast Gauss Transform [YDG03] which was specifically developed to compute sums of the form (0.21) with Gaussian kernel. FGT can then be used to compute current distances or varifold distances with a Binet kernel on the orientation part.

6.2. Compact approximations

The approximation quality of the discrete models presented so far depends on the resolution of the meshes, namely the number m_1, m_2 (and size) of segments and triangles for curves and surfaces respectively. Due to the use of kernels, those representations may be redundant and thus simplified prior to further processing. To focus the discussion, we will here restrict to the model of currents presented in Section 3.

Take as example a mesh composed of two segments, which can be modeled as two Dirac currents $\delta_{c_1}^{\tau_1}$ and $\delta_{c_2}^{\tau_2}$ as in equation (0.6). Using for instance a Gaussian kernel with scale parameter equal to σ, one can easily verify that if $\|c_1 - c_2\| \ll \sigma$, $\delta_{c_1}^{\tau_1} + \delta_{c_2}^{\tau_2}$ can be well approximated for the $\|\cdot\|_{W'}$ norm by the single Dirac $\delta_{\frac{c_1+c_2}{2}}^{\tau_1+\tau_2}$. Exploiting this idea, one could approximate all the segments in a neighbourhood of size σ (or cells for a surface mesh) with a single "average" segment (*i.e.* point + orientation) modeled as a Dirac current. This representation would be less redundant since the resulting Dirac currents would be (almost) orthogonal to each other (*i.e.* the

distance between their centers would be greater than σ). In order to be an optimal decomposition, the small set of Diracs should also accurately approximate the original shape.

A greedy approach of this idea is the one introduced in [DPTA09] and based on the *matching pursuit algorithm*. It results in a set of N Dirac currents $\sum_{i=1}^{N} \delta_{x_i}^{\alpha_i}$ that well approximate the original shape composed of m Dirac currents with a very high compression ratio (*i.e.* $N \ll m$). Each Dirac current locally integrates the redundancy of the data at the scale of σ. An example of this approximation approach is shown in Fig.0.13, where we employ a white matter fiber tract resulting from a tractography algorithm applied on diffusion MRI. For more information, the reader is referred to [DPTA09] and [DFP$^+$11a].

The previous technique produces a very concise representation that works quite well in practice for both curves and surfaces. However, it accurately approximates only the areas of the shape characterized by a high density of segments (or cells). Moreover, it results in an ensemble of *disconnected* oriented points that do not preserve the original connectivity of the shape. This can complicate the interpretation and impede the studies where the connectivity of the meshes is important as, for instance, when working with white matter fiber bundles. For this reason, the authors in [GCMK$^+$16] proposed to approximate an ensemble of curves, as a white matter tract, by selecting a small set of *weighted prototypes*. Prototypes are chosen among the curves and they approximate their neighbour and similar curves. Their weights are related to the number of curves approximated. All curves, prototypes included, are modelled as weighted currents, an extension of the framework of currents. This computational model takes into consideration both the pathway of the curves and the location of their endpoints. Using the same notation as for currents, and calling (a_1, b_1) and (a_2, b_2) the endpoints of two curves $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, their inner product is:

$$\langle [\mathcal{M}_1], [\mathcal{M}_2] \rangle_{W'} \approx K_a(a_1, a_2) K_b(b_1, b_2) \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} K_g(c_{f_{1,i}}, c_{f_{2,j}}) \langle \vec{\tau}_{f_{1,i}}, \vec{\tau}_{f_{2,j}} \rangle \tag{0.22}$$

where K_a, K_b and K_g are three radial Gaussian kernels. Two curves are thus considered similar if their endpoints are close to each other and if their trajectories are similar. As for currents, a bundle $\mathcal{B}$ of two curves $\mathcal{M}_1$ and $\mathcal{M}_2$ is represented as a sum in this framework $[\mathcal{B}] = [\mathcal{M}_1] + [\mathcal{M}_2]$. If the two curves are similar, $[\mathcal{B}]$ could be well approximated by $2[\mathcal{M}_1]$ or $2[\mathcal{M}_2]$. Using this idea, an ensemble of similar curves can be represented with a single *weighted prototype* where the weight is related to the number of curves approximated. A weighted prototype can be visualized as a constant-radius tube where the curve chosen as prototype is the central axis and the radius is proportional to the weight (see Fig.0.13). In order to find the best prototypes based on the metric of weighted currents, the authors in [GCMK$^+$16] proposed a greedy approach divided into two steps. They first divide the bundle of curves into smaller subsets, called fascicles, and then select the prototypes in each fascicle independently. Every fascicle is defined as a small subset of curves which are considered similar in the framework of weighted currents. This subdivision is based on the maximization of a quality function called *modularity*. The selection of prototypes is even in this case based on an iterative algorithm inspired by orthogonal matching pursuit. The main differences with respect to the previous algorithm are that the prototypes are chosen from the original curves, thus preserving the connectivity, and that all parts of the original bundle, even the small fascicles, are well approximated. However, it can only be used with curves and not with surfaces. More details can be found in [GCMK$^+$16].

6.3. Available implementations

Several open-source codes incorporate implementations of the distances that are described in this chapter. We refer to Table 0.1 for a summary of some of these.

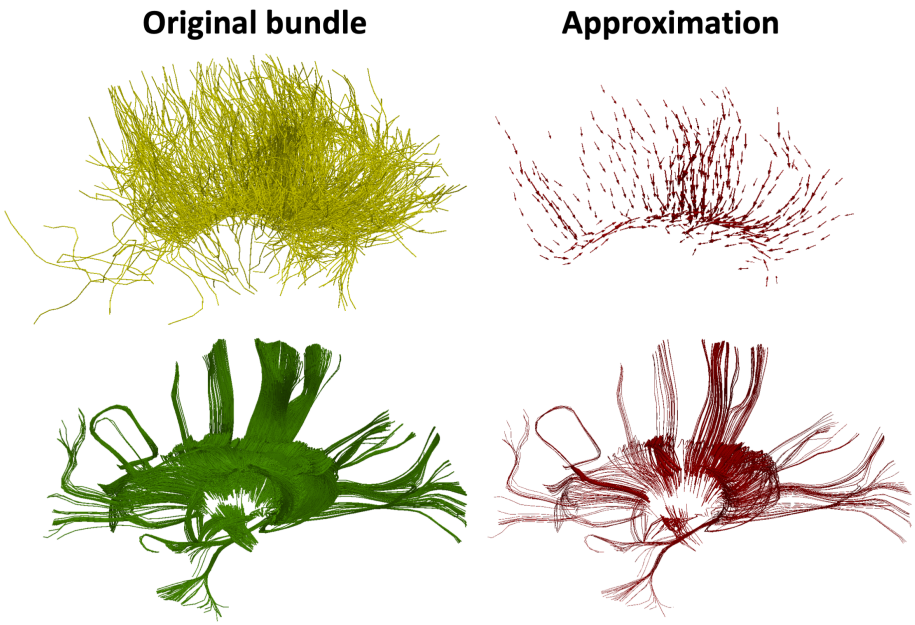


Figure 0.13 Compact approximations of white matter tracts resulting from tractography algorithms applied on diffusion MRI. The first row shows an approximation based on Dirac currents [DFP$^+$11a]. The second row presents a different tract approximated with weighted prototypes [GCMK$^+$16].

Name	language	GPU	Currents		Varifolds		Normal
			Gauss	Rad	Binet	Rad	Cycles
Deformetrica [DC18]	C++	✓	✓		✓		
FshapesTk [CCT18]	Matlab	✓	✓	✓	✓	✓	
LDDMM/NormalCycles [GR18]	Matlab	✓	✓	✓	✓(*)		✓
MRICloud [JTL$^+$14]	C++	✓	✓				

Table 0.1 Available implementations of LDDMM registration algorithms described in this chapter. Gauss stands for Gaussian radial kernel, Rad means any radial kernel, Binet is the varifold distance with Binet kernel for K_s. (*) Varifolds in this code are only available as special case of normal cycles, which gives unefficient implementation.

7. Conclusion

In this chapter, we presented different mathematical and numerical models to quantify the discrepancy between two curves or two surfaces. All these approaches rely on embedding shapes into certain distribution or measure spaces on which kernel-based metrics are constructed. We showed that the various possible choices of spaces and kernels provide in turn a large family of fidelity metrics. One important advantage of this framework is its versatility and simplicity of use that allows taking advantage of the properties of one particular metric over another one, tailored to the specific data structure under study.

In particular, we emphasized that metrics based on the model of currents, while requiring a consistent orientation of shapes, provide robustness to certain noisy features in the geometry. Varifolds on the other hand prevent certain undesirable effects of currents like flattening or insensitivity to some thin geometric structures, and with the adequate choice of kernels can also spare the user from the need to orient the given meshes. Finally, the normal cycle approach leads to a higher-order shape representation with metrics that incorporate comparison of the curvatures. Although more costly to compute from a numerical point of view, normal cycles are particularly well-suited for data involving branching structures, complex boundaries and corners...

In terms of applications, we mainly focused here on the use of such metrics as fidelity terms

in LDDMM registration problems but this framework can be readily combined with other models for inexact registration, as for instance Sobolev elastic curve matching in the recent work of [BBCMA18]. It can also be embedded in similar fashion within template/atlas estimation procedures on populations of shapes. As a final note, we also want to emphasize that the interest of these methods goes beyond the sole issue of constructing fidelity terms for those problems. We briefly mentioned in Section 6.2 the applications to sparse approximations of shapes, authors in [KCC17] have also investigated clustering algorithms based on varifold metrics while several works such as [CCT17, LCC$^+$17] have proposed extensions of some of this chapter's methods for the purpose of joint analysis of shape and function in structuro-functional datasets.

Acknowledgements

Nicolas Charon acknowledges support from the National Science Foundation, through grant n^o 1819131.

[ACG+11] G. Auzias, O. Colliot, J. A. Glaunès, M. Perrot, J. F. Mangin, A. Trouvé, and S. Baillet. Diffeomorphic brain registration under exhaustive sulcal constraints. *IEEE Transactions on Medical Imaging*, 30(6):1214–1227, June 2011.

[All72] W. Allard. On the first variation of a varifold. *Annals of Mathematics*, 95(3), 1972.

[Alm66] F. Almgren. *Plateau's Problem: An Invitation to Varifold Geometry*. Student Mathematical Library, 1966.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[BBCMA18] M. Bauer, M. Bruveris, N. Charon, and J. Moeller-Andersen. A relaxed approach for curve matching with elastic metrics. *ESAIM: Control, Optimisation and Calculus of Variations*, 2018.

[BMTY05] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(139-157), 2005.

[CA04] P. Cachier and N. Ayache. Isotropic energies, filters and splines for vector field regularization. *Journal of Mathematical Imaging and Vision*, 20(3):251–265, May 2004.

[CCT17] B. Charlier, N. Charon, and A. Trouvé. The fshape framework for the variability analysis of functional shapes. *Journal of Foundations of Comput. Maths*, 17(2):287–357, 2017.

[CCT18] B. Charlier, N. Charon, and A. Trouvé. Fshapestk: The functional shapes toolkit (software). `https://github.com/fshapes/fshapesTk/`, 2013–2018.

[CDVTU10] C. Carmeli, E. De Vito, A. Toigo, and V. Umanita. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

[CFG18] B. Charlier, J. Feydy, and J.A. Glaunès. Keops (software). `https://plmlab.math.cnrs.fr/benjamin.charlier/keops`, 2017–2018.

[CT13] N. Charon and A. Trouvé. The varifold representation of non-oriented shapes for diffeomorphic registration. *SIAM journal of Imaging Science*, 6(4):2547–2580, 2013.

[DC18] S. Durrleman and Contributors. Deformetrica (software). `www.deformetrica.org`, 2009–2018.

[DFP+11a] S. Durrleman, P. Fillard, X. Pennec, A. Trouvé, and N. Ayache. Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents. *NeuroImage*, 55(3):1073–1090, 2011.

[DFP+11b] S. Durrleman, P. Fillard, X. Pennec, A. Trouvé, and N. Ayache. Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents. *NeuroImage*, 55(3):1073 – 1090, 2011.

[DPC+14] S. Durrleman, M. Prastawa, N. Charon, J.R Korenberg, S. Joshi, G. Gerig, and A. Trouvé. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *Neuroimage*, 101:35–49, 2014.

[DPT+09] S. Durrleman, X. Pennec, A. Trouvé, G. Gerig, and N. Ayache. Spatiotemporal atlas estimation for developmental delay detection in longitudinal datasets. *Medical Image Computing and Computer-Assisted Intervention*, 12(Pt 1):297–304, 2009.

[DPTA09] S. Durrleman, X. Pennec, A. Trouvé, and N. Ayache. Statistical models of sets of curves and surfaces based on currents. *Medical Image Analysis*, 13(5):793–808, 2009.

[DR04] Fabrice Dodu and Christophe Rabut. Irrotational or divergence-free interpolation. *Numerische Mathematik*, 98(3):477–498, Sep 2004.

[Duc77] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, *Constructive Theory of Functions of Several Variables*, pages 85–100, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg.

[Dur09] S. Durrleman. *Statistical models of currents for measuring the variability of anatomical curves, surfaces and their evolution*. PhD thesis, Inria Sophia Antipolis, 2009.

[DVUV13] E. De Vito, V. Umanità, and S. Villa. An extension of mercer theorem to matrix-valued measurable kernels. *Applied and Computational Harmonic Analysis*, 34(3):339 – 351, 2013.

[FCVP17] J. Feydy, B. Charlier, F-X. Vialard, and G. Peyré. Optimal Transport for Diffeomorphic Registration. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, pages 291–299, 2017.

[Fed59] H. Federer. Curvature measures. *Trans. Amer. Maths. Soc.*, 93, 1959.

[Fed69] H. Federer. *Geometric measure theory*. Springer, 1969.

[GCMK+16] P. Gori, O. Colliot, L. Marrakchi-Kacem, Y. Worbe, F. De Vico Fallani, M. Chavez, C. Poupon, A. Hartmann, N. Ayache, and S. Durrleman. Parsimonious Approximation of Streamline Trajectories in White Matter Fiber Bundles. *IEEE Transactions on Medical Imaging*, 35(12):2609–2619, 2016.

[GJ06] Joan Glaunès and Sarang Joshi. Template estimation from unlabeled point set data and surfaces for computational anatomy. In X. Pennec and S. Joshi, editors, *Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006)*, pages 29–39, 1st of October 2006.

[GL04] L. Greengard and J.-Y. Lee. Accelerating the nonuniform fast fourier transform. *SIAM Review*, page 443, 2004.

[Gla05] J. A. Glaunès. *Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie numérique*. PhD thesis, Université Paris 13, 2005.

[GQMY08] J. A. Glaunès, A. Qiu, M. I. Miller, and L. Younes. Large deformation diffeomorphic metric curve mapping. *International Journal of Computer Vision*, 80(3):317–336, 2008.

[GR87] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 135(2):280 – 292, 1987.

[GR18] J. A. Glaunès and P. Roussillon. Lddmm algorithms using currents and normal cycles (software). `http://w3.mi.parisdescartes.fr/~glaunes/LDDMM_Currents_and_NormalCycles.zip`, 2014–2018.

[GTY04] J.A. Glaunès, A. Trouvé, and L. Younes. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:712–718, 2004.

[GV06] J.A. Glaunès and M. Vaillant. Surface matching via currents. *Proceedings of Information Processing in Medical Imaging (IPMI), Lecture Notes in Computer Science*, 3565(381-392), 2006.

[HSS08] T. Hofmann, B. Schölkopf, and A.J. Smola. Kernel methods in machine learning. *Ann. Statist.*, 36(3):1171–1220, 06 2008.

[JM00] S.C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9(8):1357–1370, 2000.

[JTL+14] S. Jain, D. Tward, D. Lee, A. Kolasny, T. Brown, T. Ratnanather, M. Miller, and L. Younes. Computational anatomy gateway: Leveraging xsede computational resources for shape analysis. In *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, XSEDE '14, pages 54:1–54:6, 2014.

[KCC17] I. Kaltenmark, B. Charlier, and N. Charon. A general framework for curve and surface comparison and registration with oriented varifolds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[KKP09] J. Keiner, S. Kunis, and D. Potts. Using nfft 3 - a software library for various nonequispaced fast fourier transforms. *ACM Trans. Math. Software*, pages 2013–2037, 2009.

[LCC+17] S. Lee, N. Charon, B. Charlier, K. Popuri, E. Lebed, P.R Ramana, M. Sarunic, A. Trouvé, and M.F Beg. Atlas-based shape analysis and classification of retinal optical coherence tomography images using the functional shape (fshape) framework. *Medical Image Analysis*, 35:570–581, 2017.

[LFF+13] S. Lee, N. Fallah, F. Forooghian, A. Ko, K. Pakzad-Vaezi, A.B. Merkur, A.W. Kirker, D.A. Albiani, M. Young, M.V. Sarunic, and M.F. Beg. Comparative analysis of repeatability of manual and automated choroidal thickness measurements in nonneovascular age-related macular degeneration. *Invest. Ophthalmol. Vis. Sci.*, 54(4):2864–2871, 2013.

[MG14] M. Micheli and J. A. Glaunès. Matrix-valued kernels for shape deformation analysis. *Geom. Imaging Comput.*, 1(1):57–139, 2014.

[MKB+11] C. McGann, E. Kholmovski, J. Blauer, S. Vijayakumar, T. Haslam, J. Cates, E. DiBella, N. Burgon, B. Wilson, A. Alexander, M. Prastawa, M. Daccarett, G. Vergara, N. Akoum, D. Parker, R. MacLeod, and N. Marrouche. Dark regions of no-reflow on late gadolinium enhancement magnetic resonance imaging result in scar formation after atrial fibrillation ablation. *Journal of the American College of Cardiology*, 58(2):177–185, 2011.

[MMY10] J. Ma, M.I. Miller, and L. Younes. A bayesian generative model for surface template estimation. *Journal of Biomedical Imaging*, 2010:16:1–16:14, January 2010.

[Mor95] F. Morgan. *Geometric measure theory, a beginner's guide*. Academic Press, 1995.

[MP05] C.A. Micchelli and M. Pontil. On Learning Vector-Valued Functions. *Neural Computation*, 17(1):177–204, 2005.

[PCD+16] Y. Pan, G. Christensen, O. Durumeric, S. Gerard, J. Reinhardt, and G. Hugo. Current- and Varifold-Based Registration of Lung Vessel and Airway Trees. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 566–573, 2016.

[RBT+12] J. Ratnanather, T. Brown, H. Trinh, L. Younes, M.I. Miller, S. Mori, and M. Albert. Shape analysis of hippocampus and amygdala in BIOCARD. *Alzheimer's & Dementia*, 8:P63, 2012.

[RG16] P. Roussillon and J. A. Glaunès. Kernel metrics on normal cycles and application to curve matching. *SIAM J. Imaging Sciences*, 9:1991–2038, 2016.

[RLLS16] I. Rekik, G. Li, W. Lin, and D. Shen. Multidirectional and Topography-based Dynamic-scale Varifold Representations with Application to Matching Developing Cortical Surfaces. *NeuroImage*, 135:152–162, 2016.

[RVW+11] L. Risser, F-X. Vialard, R. Wolz, M. Murgasova, D. Holm, and D. Rueckert. Simultaneous Multi-scale Registration Using Large Deformation Diffeomorphic Metric Mapping. *IEEE Transactions on Medical Imaging*, 30(10):1746–59, 2011.

[Sch64] L. Schwartz. Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (noyaux reproduisants). *Journal d'Analyse Mathématique*, 13(1):115–256, Dec 1964.

[VQGM07] M. Vaillant, A. Qiu, J.A. Glaunès, and M.I. Miller. Diffeomorphic metric surface mapping in subregion of the superior temporal gyrus. *NeuroImage*, 34(3):1149 – 1159, 2007.

[YDG03] C. Yang, R. Duraiswami, and N. A. Gumerov. Improved fast gauss transform. Technical report, 2003.

[Zä86] M. Zähle. Integral and current representation of Federer's curvature measure. *Arch. Maths.*, 23:557–567, 1986.

[Zä87] M Zähle. Curvatures and currents for unions of set with positive reach. *Geometriae Dedicata*, 23:155–171, 1987.