IMODAL: creating learnable user-defined deformation models

Leander Lacroix
INSERM U1299, Université Paris-Saclay
leander.lacroix@protonmail.com

Benjamin Charlier
IMAG, Univ. Montpellier, France
benjamin.charlier@umontpellier.fr

Alain Trouvé
Centre Borelli, ENS Paris-Saclay
alain.trouve@ens-paris-saclay.fr

Barbara Gris
LJLL, Sorbonne Université, CNRS
gris@ljll.math.upmc.fr

Abstract

A natural way to model the evolution of an object (growth of a leaf for instance) is to estimate a plausible deforming path between two observations. This interpolation process can generate deceiving results when the set of considered deformations is not relevant to the observed data. To overcome this issue, the framework of deformation modules allows to incorporate in the model structured deformation patterns coming from prior knowledge on the data. The goal of this article is twofold. First defining new deformation modules incorporating structures coming from the elastic properties of the objects. Second, presenting the IMODAL library allowing to perform registration through structured deformations. This library is modular: adapted priors can be easily defined by the user, several priors can be combined into a global one and various types of data can be considered such as curves, meshes or images. It can be downloaded at https://github.com/imodal.

1. Introduction

The last decade have witnessed important progress in shape analysis fuelled by the increasing availability of data sets produced by a wide range of imaging devices. However, biological and medical shape analysis stand at a particular place where a range of constraints are limiting the size of available databases and usually further down-scaled to few tens when filtered under specific conditions. Visual inspection by experts is still playing an important role in nowadays analysis especially when dealing with shape evolution. This is for different reasons: difficulty to separate the target objects from background, difficulty to define population models when dealing with geometrical object (cross-sectional aspect) and to aggregate them into longitudinal profiles [11]. The computational anatomy and large deformation setting [12, 24, 3, 20, 18, 17] have helped a lot to build principle ways to generate digital atlases from data sets, perform registrations, compute deformation paths using paths of diffeomorphisms transporting wide range a geometrical features. However, the ubiquity of the all-purpose approaches is often limited in two ways: first the reliance of weakly structured deformation models preventing the introduction of high level prior knowledge (as deformation properties and constraints attached to the objects) in the analysis and second, their inability to provide the user with an automatic separation of the different non-linear superposition of biologically relevant dynamic components.

The paper is devoted to the exposition of a mathematical and computational framework IMODAL built upon the modular approach pioneered in [14] and answering the two previous limitations. We present in Figure 2 two results of registration between the initial and final states of a *basipetal* leaf growth ($i.e.$ growth from the base) extracted from Figure 1a. We used unstructured deformations (LDDMM) [4] and modular ones computed with IMODAL with the prior knowledge that the deformation should be *basipetal*. The basipetal growth is well recovered by IMODAL (see Section 3.1 for more detail) while unstructured deformations fail modelling this growth.

2. Modelling evolution through constrained deformations

2.1. Large deformations

We recall here the notion of large deformations as defined in [2]. Let $d, \ell \geq 1$ be two integers and $\mathcal{C}_0^\ell(\mathbb{R}^d)$ be the space of vector fields of class $\mathcal{C}^\ell$ of $\mathbb{R}^d$ whose derivatives $k \leq \ell$ tend to zero at infinity. For $v \in \mathcal{C}_0^\ell(\mathbb{R}^d)$, $\|v\|_{\ell,\infty}$ denotes the supremum norm: $\|v\|_{\ell,\infty} = \sum_{|\beta|\leq\ell} \|D^\beta v\|_\infty$.

Proposition 1 *Let* $v_\cdot$ *in* $L^1([0,1], \mathcal{C}_0^\ell(\mathbb{R}^d))$, i.e. *a time-dependent vector field such that* $\int_0^1 \|v_t\|_{\ell,\infty} dt < \infty$. *Then there exists a unique absolutely continuous solution* $\varphi_\cdot^v$,

called the flow of $v_\cdot$, such that $\varphi_0^v = \mathrm{Id}$ and $\dot{\varphi}_t^v = v_t \circ \varphi_t^v$. In addition, for any t in $[0,1]$, φ_t^v is a C^ℓ diffeomorphism.

This notion of large deformation can be used to model the evolution of a *shape* between two observations. Here, we use the concept of shapes as defined in [1], which covers point clouds, curves, meshes and any other geometrical object that can be transformed by sufficiently smooth diffeomorphisms and vector fields. A space of shapes $\mathcal{O}$ is in particular defined through its *infinitesimal action* $(q, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \longrightarrow v \cdot q \in T\mathcal{O}$ which specifies how vector fields (and then diffeomorphisms *via* time integration) act on its elements. Given two observations a and b of a shape at two different times, the idea is to reconstruct the time evolution between them *via* the *registration* of the first one a (called the *source*) into the second one b (called the *target*) using large deformations [4]. It amounts to minimizing: $J : v \in L^2([0,1], V) \mapsto \int_0^1 |v(t)|_V^2 \mathrm{d}t + D(\varphi_{t=1} \cdot a, b)$ where V is a space of smooth vector fields continuously embedded in $\mathcal{C}_0^\ell(\mathbb{R}^d)$ ($V \hookrightarrow \mathcal{C}_0^\ell(\mathbb{R}^d)$) and D a distance between shapes measuring how close the deformation of a is from b. In the example presented in Figure 2, the registration is performed between *unparametrized* curves (without point correspondence) and we use the varifold framework [8] to define the distance D. The result of this registration is satisfying from a fitting point of view but from a modelling point of view, this result raises two questions.

First, the path $t \mapsto \varphi_t \cdot a$ does not correspond to a basipetal growth and then does not help modelling the growth. It would be necessary to impose, as a prior, the fact that the growth is basipetal and then search, amongst all the possible basipetal growths, the one fitting the observed data. A first attempt to address this issue would be to enrich the RKHS V [10, 21, 25] so that more complex deformation patterns can be generated. However, it would not be possible allow to impose shape-dependent priors since the space of allowed vector fields is not shape-dependent. A natural way to incorporate such prior is to constrain, at each time t, the vector field to be of a certain type, formulated as a function of the shape at t and corresponding to a *basipetal infinitesimal growth*. The second question that arises is about the interpretability of the result of the registration: how can it be used to understand the growth ? Here this result is given as a dense vector field, parametrized in large dimension, which cannot be directly interpreted. Several frameworks allow to build particular deformations adapted to specific priors [22, 23] and their evolution [13, 16, 26]. However building and analyzing complex structured deformation models evolving naturally with the deformed object is still a challenge.

2.2. Framework of deformation modules

The notion of deformation module, introduced in [14], makes possible to incorporate *structural priors* in the vector fields, while leaving some parameters free to enable to fit the data. In the leaf growth's example, if we first assume that it follows a basipetal growth pattern ($i.e.$ that a certain growth at the tip necessary comes with a larger growth at the base), this knowledge can be incorporated in the deformation framework via *a generator of vector fields* encoding structural relations between local growth factors. This generator takes as input the current *geometrical descriptors* of the leaf (in order to locate the base for instance) as well as a growth intensity, and returns an adapted vector field. The growth intensity acts here as a *control* parameter that can be adjusted at each time of the growth trajectory to fit the data. The "geometrical descriptors" of the leaf can be seen as a special case of the general notion of *shape* introduced in [1]. It gathers all the geometric parameters that are relevant to model the growth trajectory and are flown by the deformation of the ambient space ($i.e.$ at each time, their speeds are given by the generated vector field). We give here a

(a) Basipetal (b) Acropetal

Figure 1: Growth from the base 1a and from the tip 1b of the leaf. Data from [15] of leaves before (left) and after (right) growth. Black dots were drawn before growth.

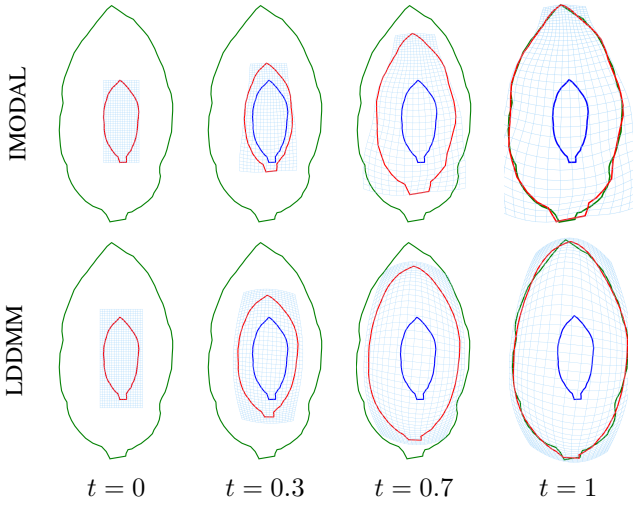


Figure 2: Basipetal growth: curves registration using IMODAL and LDDMM (unstructured large deformation). The source is blue, target is green and the deformed source is red.

slightly simplified definition of deformation modules:

Definition 1 *We say that $M = (\mathcal{O}, H, \zeta, c)$ is a **deformation module** with **geometrical descriptors** in $\mathcal{O}$, **controls** in H, **field generator** ζ and **cost** c, if*

- *$\mathcal{O}$ is a shape space of $\mathbb{R}^d$, $H = \mathbb{R}^{dimH}$,*
- *$\zeta : (q, h) \in \mathcal{O} \times H \to \zeta_q(h) \in C_0^3(\mathbb{R}^d)$ is continuous, linear with respect to h and C^2 with respect to q,*
- *$c : (q, h) \in \mathcal{O} \times H \to c_q(h) \in \mathbb{R}^+$ is a continuous mapping such that $q \mapsto c_q$ is smooth and for all $q \in \mathcal{O}$, $h \mapsto c_q(h)$ is a positive quadratic form on H.*

Note that considering a parameter being part of the geometrical descriptor is a modeling choice: one could define another deformation module, generating similar fields but built upon a different space of geometrical descriptors. In this case, the generated modular large deformations (see Section 2.4) would be different. The cost function determines how "desirable" a particular choice is with respect to the prior. In the following, we will always assume that the cost satisfies the Uniform Embedding Condition (UEC): $|\zeta_q(h)|_V^2 \leq M c_q(h)$ with V a Reproducing Kernel Hilbert Space (RKHS) of vector fields continuously embedded in $\mathcal{C}_0^3(\mathbb{R}^d)$ ($V \hookrightarrow \mathcal{C}_0^3(\mathbb{R}^d)$) and $M > 0$ independent from (q, h). This condition is necessary to build large deformations from a deformation module (see Section 2.5). An important feature of this framework is the possibility to combine several deformation modules into a more complex one.

Definition 2 (Combination) *Let $M^i = (\mathcal{O}^i, H^i, \zeta^i, c^i)$, $i = 1, \dots, N$ be N deformation module, the compound module $M = C(M^1, \dots, M^N)$ is defined by $M = (\mathcal{O}, H, \zeta, c)$ where $\mathcal{O} = \prod_{i=1}^N \mathcal{O}^i$, $H = \prod_{i=1}^N H^i$, $\zeta : (q, h) \in \mathcal{O} \times H \mapsto \sum_i \zeta_{q^i}^i h^i$ and $c : (q, h) \in \mathcal{O} \times H \mapsto \sum_i c_{q^i}^i h^i$ with $q = (q_1, \dots, q_N)$, $h = (h_1, \dots, h_N)$.*

This compound deformation module generates vector fields that sums of vector fields generated by the initial deformation modules: the field generator combines the different prior of each module into a new, compound, prior.

2.3. Examples

We present here several examples of deformation modules. We denote by V_σ the scalar Gaussian RKHS of scale $\sigma > 0$ and K_σ its kernel.

2.3.1 Unstructured deformation module

A first simple example of deformation module allows to build sums of $N \in \mathbb{N}$ local translations in $\mathbb{R}^d$ similarly to [6, 9]. We set $\mathcal{O} = (\mathbb{R}^d)^N$ (space of points), $H = (\mathbb{R}^d)^N$ (space of vectors) and $\zeta_q(h) = \sum_{i=1}^N K_\sigma(x_i, \cdot) h_i$ with $\sigma > 0$, $q = (x_1, \dots, , x_n) \in \mathcal{O}$ and $h = (h_1, \dots, h_N) \in H$. This module is said to be *unstructured* as any speed for the points $(x_1, \dots, , x_n)$ can be retrieved as the application of a field of the type $\zeta_q(h)$. It is used to *correct* the model of deformation in order to better fit the data.

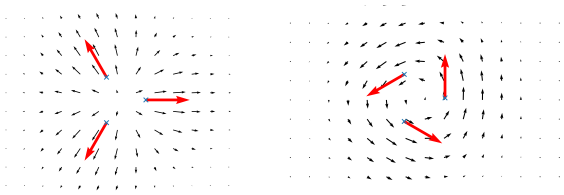


Figure 3: Local scaling (left) and local rotation (right) vector fields approximated by constrained translations generated by 3 control points each.

2.3.2 Constrained translations

By imposing some links between the different points and vectors of the translations, one can produce more constrained fields, as illustrated in Figure 3.

2.3.3 Implicit deformation modules

The deformation prior may not be given in the form of an explicit type of vector field that one wants to use, but more in some properties that the vector fields should satisfy. In order to address this problem, we define *implicit deformation modules*, where the field generator ζ is defined as the solution of a minimization problem:

$$\zeta_q(h) = \text{argmin}\{|v|_V^2 + \frac{1}{\nu}|S_q(v) - A_q(h)|^2\} \quad (1)$$

with V a RKHS of vector field, $S_q : V \to Y$ a linear surjective constraint operator on vector fields that takes values in the space of constraints Y (vector space of finite dimension) and $A_q : H \to Y$ a linear operator which defines the value that one wants to observe. The spaces of geometrical descriptors $\mathcal{O}$ and controls H will be specific to each type of implicit deformation module. The cost of the deformation module is defined by $c_q(h) = |\zeta_q(h)|_V^2 + \frac{1}{\nu}|S_q(\zeta_q(h)) - A_q(h)|^2$. From equation (1), one can compute the explicit expression of the field generator ζ_q as detailed in the Appendix. Note that if $V = V_\sigma$, with $\sigma > 0$, a consequence of Equation (1), is that the field $\zeta_q(h)$ is located around the points on which depends $S_q(v)$, in an area depending on σ. Then, the support of the generated field follows naturally the evolution of q.

Order 0 As a first example we set $\sigma > 0$, $\mathcal{O} = (\mathbb{R}^d)^N$, $H = (\mathbb{R}^d)^N$, $S_q(v) = (v(x_1), \dots, v(x_N))$ and $A_q(h) =$

$(K_q + \nu Id)h$ where $q = (x_1, \dots, x_N)$ and K_q is the matrix $(K_\sigma(x_i, x_j)I_d)_{i,j}$. We call the corresponding module *implicit deformation module of order 0* as it is defined from constraints on the values of the field (evaluation of order 0). It can be shown easily (see Appendix) that the field generator is given by $\zeta_q(h) = \sum_{i=1}^N K_\sigma(x_i, \cdot)h_i$ with $h = (h_1, \dots, h_N)$. The cost is $c_q(h) = h^T(K_q + \nu Id)h$ and, as we need to invert the cost operator to perform registration (see Section 2.5), this module is a *regularized version* of the one presented in Section 2.3.1.

Order 1 We give now a second example of implicit deformation modules allowing to incorporate objects-related elastic properties in the deformation model. More specifically, we constrain the local changes of lengths induced by the action of the vector field at some specified locations x_i and along some specified directions *attached to the object*. This amounts to constraining the *infinitesimal strain tensor*

$$\epsilon_{x_i}(v) \doteq \frac{Dv(x_i) + Dv^*(x_i)}{2} \qquad (2)$$

which is a symmetric tensor capturing the local metric changes (expansion or dilation along given directions). A natural way is to introduce a family $q = ((x_i, R_i))_{i\in I}$ of affine orthonormal frames (here coded by its origin x_i and an associated rotation R_i) and to ask $\epsilon_{x_i}(v)$ to be diagonalizable into the orthonormal basis given by R_i. This means that there exists a diagonal matrix D_i such that $\epsilon_{x_i}(v) = R_i D_i R_i^T$. Structural relations among the different D_i's can be captured saying that $D_i = D_i(h)$ where h is a control parameter $h \in H$. The value $(R_i D_i(h) R_i^T)_i$ imposed on the infinitesimal strain tensor is designed by *growth factor* and the operator $h \mapsto (D_i(h))_i$ (defining the model) is called the *growth model operator*. We present in Figures 4, 5, 6 three examples of growth factors $((R_i D_i R_i^T)_i)$. They are represented by ellipsoids whose axes are $|D_{i,j}|R_{i,j}$, $1 \leq j \leq d$ where $D_{i,j}$ is the $j-th$ diagonal coefficient of D_i and $R_{i,j}$ is the $j-th$ column of R_i.

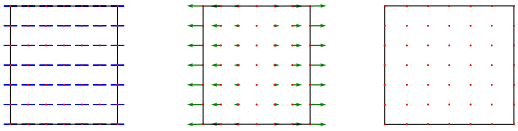


Figure 4: Horizontal uniform elongation. Left: the blue (degenerate) ellipses represent the growth factor $R_i D_i R_i^T$ with $D_i = \mathrm{diag}(1, 0)$ and $R_i = \mathrm{Id}$ for each x_i (red points). Middle: generated vector field in green. Right: infinitesimal deformation of the rectangle induced by this field.

The formal derivation of module of order 1 is straightforwardly described choosing

$$\mathcal{O} = \{q = ((x_i, R_i))_{i\in I} \in (\mathbb{R}^d \times SO_d(\mathbb{R}))^I\} \qquad (3)$$

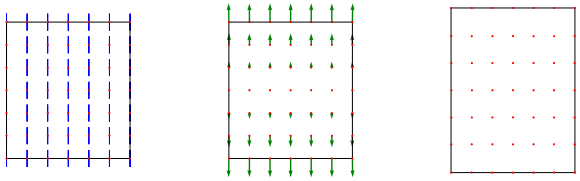


Figure 5: Vertical uniform elongation. Left: the blue (degenerate) ellipses represent the growth factor $R_i D_i R_i^T$ with $D_i = \mathrm{diag}(0, 1)$ and $R_i = \mathrm{Id}$ for each x_i (red points). Middle: generated vector field in green. Right: infinitesimal deformation of the rectangle induced by this field.

where I is the indexing set, $S_q(v) = (\epsilon_{x_i}(v))_{i\in I}$, $A_q(h) = (R_i D_i(h) R_i^T)_{i\in I}$ with $\epsilon_x(v)$ given by (2) and for each i, $D_i(h)$ is a diagonal matrix that depends linearly on the control h. The growth model operator $h \mapsto (D_i(h))_{i\in I}$ specifies the prior on the changes of lengths. Note that $S_q : V \mapsto (\mathcal{S}(\mathbb{R}^d))^I$ (with $\mathcal{S}(\mathbb{R}^d)$ the space of symmetric matrices in $\mathbb{R}^d$) is surjective if $x_i \neq x_j$ for $i \neq j$. The space of control is $H = \mathbb{R}^p$ with p an integer which is in general small (between 1 and 4 in the examples). Since the constraints are now on the derivative of v, we coin the name of *module of order 1*.

Remark 1 *The space of $\mathcal{O}$ given by equation (3) is not strictly speaking a shape space as defined in [1] where objects can be deformed under the action of diffeomorphisms* $\mathrm{Diff}^1(\mathbb{R}^d)$. *However, it is a generalized shape space defined as a set of objects that can be deformed under the action of* $G = \mathrm{Diff}^1(\mathbb{R}^d) \times C^0_{Id}(\mathbb{R}^d, SO_d(\mathbb{R}))$ *(the subscript Id is to enforce the convergence to Id at infinity). Let (x, R) be an affine orthonormal frame and $(\phi, \psi) \in G$, then $(\phi, \psi) \cdot (x, R) = (\phi(x), \psi(x)R)$. The tangent space of G at $e_G = (Id, x \in \mathbb{R}^d \mapsto I_d)$ is $W = C^1(\mathbb{R}^d, \mathbb{R}^d) \times C^0_0(\mathbb{R}^d, \mathrm{Skew}_d(\mathbb{R}))$ (with* $\mathrm{Skew}_d(\mathbb{R})$ *the space of skew-symmetric matrices of $\mathbb{R}^d$) and then, by differentiating $(\phi, \psi)\cdot(x, R)$ with respect to (ϕ, ψ) at e_G, we can define the action of an element $(v, A) \in W$ on (x, R) by $(v, A) \cdot (x, R) = (v(x), A(x)R)$.*

2.3.4 Silent module

A last example of deformation module is the one that generates a null vector field: $\mathcal{O}$ is a given shape space, $H = \{0\}$, $\zeta_q(h) = 0$ and $c_q(h) = 0$. We call it the *silent deformation module induced by $\mathcal{O}$*. When it is combined with other ones (see Definition 2), it does not contribute to the generated vector field but its geometrical descriptors are deformed by it. Silent deformation modules are particularly interesting for registration problems where the deformation modules used to generate the large deformation have geometrical descriptors different from the shape data (see Remark 4).

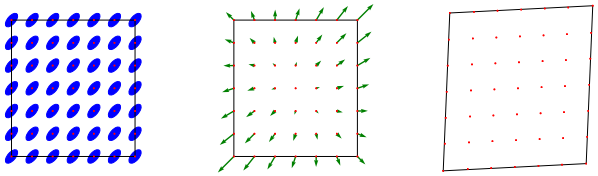


Figure 6: Uniform shearing. Left: the blue ellipses represent the growth factor $R_i D_i R_i^T$ with $D_i = \mathrm{diag}(2,1)$ and $R_i = \begin{pmatrix} \cos\pi/4 & -\sin\pi/4 \\ \sin\pi/4 & \cos\pi/4 \end{pmatrix}$. Middle: generated vector field in green. Right: infinitesimal deformation of the rectangle induced by this field.

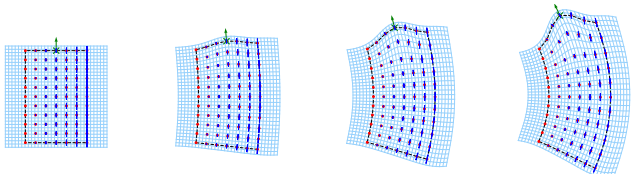


Figure 7: Example of large deformation generated by the combination of an implicit module M^1 of order 1 and a module M^2 generating a local translation. The red points are the point-component of the geometrical descriptor of M^1, the blue (flat) ellipses represent the growth factor . The green point is the geometrical descriptor of M^2 and the green vector is its control.

2.4. Modular large deformations

We will now see how priors on vector fields can be transferred to large deformations. We consider *controlled path of finite energy* where the integrated vector fields (see Proposition 1) are generated by the deformation module with geometrical descriptors following the deformation:

Definition 3 *Let $M = (\mathcal{O}, H, \zeta, c)$ be a deformation module and let $a, b \in \mathcal{O}$. We denote by $\Omega_{a,b}$ the set of measurable curves $t \mapsto (q_t, h_t) \in \mathcal{O} \times H$ where q is absolutely continuous, satisfying $q_{t=0} = a$, $q_{t=1} = b$, $\dot{q}_t = v_t \cdot q_t$ for almost every $t \in [0,1]$ with $v_t = \zeta_{q_t}(h_t)$, and $E(q,h) = \int_0^1 c_{q_t}(h_t)\mathrm{d}t < \infty$, it is called the **energy** of (q,h).*

From trajectories of $\Omega_{a,b}$, we can now build **modular large deformations**:

Proposition 2 *Let $M = (\mathcal{O}, H, \zeta, c)$ be a deformation module satisfying the UEC. Let $a, b \in \mathcal{O}$ and $(q,h) \in \Omega_{a,b}$. We define for each $t \in [0,1]$, $v_t = \zeta_{q_t}(h_t)$. Then, the flow φ^v exists and for each $t \in [0,1]$, $q_t = \varphi_t \cdot a$. The final diffeomorphism $\varphi^v_{t=1}$ is called a **modular large deformation**.*

We present in Figure 7 the example of a rectangle deformed by a large deformations obtained via the combination (see Definition 2) of a implicit deformation module M^1 of order 1 (see Section 2.3.3) and a deformation module M^2 generating a local translation (see 2.3.1). The growth model operator of M^1 is defined for each i by $h \in \mathbb{R} \mapsto \left(\begin{smallmatrix} 0 & 0 \\ 0 & u_i \end{smallmatrix}\right)$ and the rotation matrices of the geometrical descriptor are initialized at $R_i(t=0) = Id$.It generates a *bending* deformation because of the non uniform vertical elongation. Simultaneously, the local translation deforms the upper part of the rectangle. This example illustrates two important features of the framework. First, the geometrical descriptors naturally follow the deformation of the object and then the structure incorporated via the choice of deformation module is updated at each time according to the current geometrical descriptors. In particular the rotation matrices R_i (components of the geometrical descriptor of M^1) rotate so that the 'vertical elongation' remains coherent with the new geometrical descriptors of the 'rectangle'. Second, at each time the vector field is a sum of two vector fields, generated respectively by M^1 and M^2 so that the total deformation is a natural combination of the two priors.

Remark 2 *In order to define the evolution $v_t \cdot q$ in Definition 3 when q is an affine frame, we need to associate, to each vector field $v \in \mathcal{C}^1_0(\mathbb{R}^d, \mathbb{R}^d)$, an element of $\mathcal{C}^0_0(\mathbb{R}^d, \mathrm{Skew}_d(\mathbb{R}))$ (see Remark 1). We defined it as $\frac{\mathrm{d}v - \mathrm{d}v^*}{2}$ which corresponds to the 'infinitesimal rotation' created by v at each point (its coefficients are given by the curl of v).*

2.5. Registration

In order to reconstruct the evolution between two observations the idea is to estimate a large deformations transporting the initial observation (source) as close as possible to the final observation (target) and with the lowest energy possible. It can be shown [14] that the minimizing trajectories (q,h) are such that there exists $p : t \in [0,1] \mapsto T^*_{q_t}\mathcal{O}$ so that for each time:

$$\left\{ \begin{array}{rcl} \dot{q}_t & = & \frac{\partial H}{\partial p}(q_t, p_t, h_t) = \zeta_{q_t}(h_t) \cdot q_t \\ \dot{p}_t & = & -\frac{\partial H}{\partial q}(q_t, p_t, h_t) \\ h & = & Z_q^{-1} \rho_q^* p \end{array} \right. \quad (4)$$

where $H : (q,p,h) \in T^*\mathcal{O} \times H \mapsto (p|\zeta_q(h) \cdot q) - \frac{1}{2} c_q(h)$ is the *Hamiltonian*, $\rho_q(h) = \zeta_q(h) \cdot q$ and $Z_q : H \to H^*$ is an invertible operator so that for each h in H, $c_q(h) = (Z_q h | h)$. The variable p is called the *momentum*.

Remark 3 *Given $(q_0, p_0) \in T^*\mathcal{O}$, there exists a unique solution to equations (4) starting at $(q_0, p_0, Z_{q_0}^{-1} \rho^*_{q_0} p_0)$.*

Registering a source shape a into a target shape b amounts then to minimizing the following functional:

$$J_{a,b} : p_0 \in T^*_a\mathcal{O} \mapsto \int_0^1 c_{q_t}(h_t)\mathrm{d}t + D(\varphi_{t=1} \cdot a, b) \quad (5)$$

with $q_{t=0} = a$, $p_{t=0} = p_0$ and (q,p,h) satisfying (4).

Remark 4 *In practice, the shapes $q_s \in \mathcal{O}_s$ that will be registered do not contain directly the specific geometrical information that is necessary to express the prior on deformation. We then consider an augmented shape $\tilde{q} = (q_s, q)$ where q contains the relevant geometrical information. This practical case of registration can also be studied by minimizing Equation (5) with the module $\tilde{M} = C(M, M_s)$ with M the deformation module corresponding to the prior and M_s the silent deformation module induced by the shape space $\mathcal{O}_s$ (see Section 2.3.4). The non silent part of the initial geometrical descriptor can also be optimized if necessary as it is in general not observed.*

3. Examples

This framework is implemented in IMODAL[1] (Implicit Modular Object Deformation Analysis Library), in Python, using the librairies PyTorch [19] for automatic differentiation and KeOps [7] to avoid memory overflow if computations are done on GPU. It is a *modular* implementation in the sense that it relies on several abstract classes that one can combine in order to design a deformation model adapted to the observed data. Several deformation modules are implemented so that the user only needs to specify their parameters. They can be fixed, estimated or defined as functions of meta-parameters that can themselves be estimated (see examples of Section 3.1). All the optimizations are performed using the L-BFGS optimizer of the librairy PyTorch [19]. A more detailed presentation is given in the Appendix. A key parameter in our examples is the growth model operator for implicit modules of order 1. This operator $h \mapsto (D_i(h))_{i \in I}$, is implemented as a tensor C of size $N \times d \times p$ with N the number of points x_i on which the field has to be constrained, d the dimension of the ambient space and p the dimension of the control space so that $D_i(h) = \mathrm{diag}(C[i]h)$ for $1 \leq i \leq N$. We detail in each situation how it can be built in order to model the observed evolutions.

3.1. Growth modelling

We first present how the framework of deformation modules can help modelling the growths of the leaf presented in Figure 1a. As explained previously, we extract the boundaries of the leaves so that it amounts in registering two unparametrized curves (represented by varifolds [8]). Performing this registration with unstructured large deformation (LDDMM [4]) enables to obtain a good fit of these curves but the grid deformation does not correspond to a basipetal growth (see Figure 2). Even when we register simultaneously theses curves and the dots black dots extracted from Figure1a (which were drawn on the leaves before their growth), the grid deformations is not satisfying (see Figure 8 second row).

[1] https://github.com/imodal

We present here how IMODAL allows to model this growth. We use an implicit module of order 1 (see Section 2.3.3) and we first need to set its parameters. The initial geometrical descriptors are geometrical frames (see Section 2.3.3): their origins (dots) are set *automatically*, given a chosen point density and the source shape, using the tools implemented in IMODAL (see Figure 9b). We initialized the rotation matrices attached to these points to identity $i.e.$ we assume here that the initial principle axes of local metric change are the horizontal and vertical ones. These initial parameters are not optimized but they evolve during the integration of the flow. We also set the scale of the scalar Gaussian RKHS of vector fields, see Equation (1), to 30. The last parameter to determine here is the growth model operator, and, as we do not know it, we need to *estimate* it from the observed data. We choose the dimension of the control space P equal to 1 as we do not search to decompose the growth in several sub-growth models. In addition, instead of estimating independently the components $C[i]$ of the growth model tensor, we suppose that they depend smoothly on the x_i's. We assume that this dependence can be modelled with a polynomial function of degree 3 depending only on the y-axis (so that the point-correspondence of the left-part between the observed ink dots influences C everywhere). The parameters of the polynomial can be estimated by using a callback function which is triggered at each evaluation. The output of the registrations is then in three parts: (*i*) this polynomial, defining the growth model tensor, (*ii*) the matching of the leaves using the estimated growth model tensor, (*iii*) trajectories of controls $t \mapsto h(t)$ quantifying how each module is used at each time.

In order to obtain a better fit, we combine this implicit deformation module of order one with a module generating a global translation and an implicit module of order 0 (see Section 2.3.3) which will produce *small corrections* of the deformation model produced by the implicit module of order 1. Its geometrical descriptors are initialized by points on the boundary of the leaf (see Figure 9b).

We present in Figures 8 (first row) the result of these registrations. The initialized and estimated growth factor are represented by ellipses in Figures 9c and 9d. In the estimated one, the ellipses are larger at the base of the leaf: the estimated growth model is basipetal.

If we use this estimated growth model tensor, we can also register only the curves (boundaries of the leaves) and reconstruct a basipetal growth even without the point-correspondence of the ink dots as shown in Figure 2 first row. We follow the same strategy to register the curves extracted from Figure 1b with unstructured deformations (Figure 10b) and with IMODAL (Figure 10c): the acropetal growth pattern is only recovered with IMODAL.

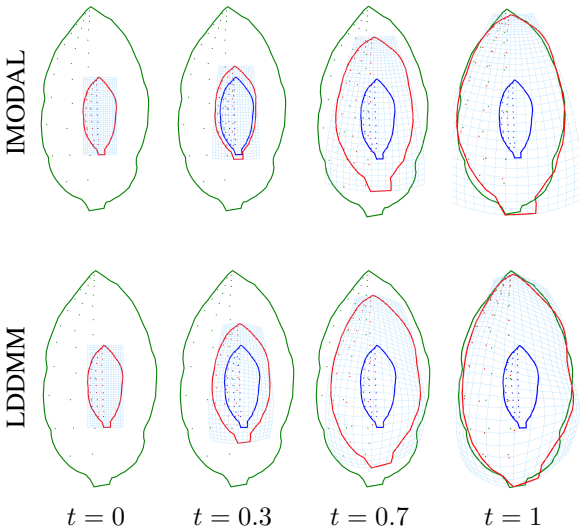


Figure 8: Basipetal growth: registration of curves and dots using IMODAL or unstructured large deformations. The source is blue, target is green and deformed source is red.

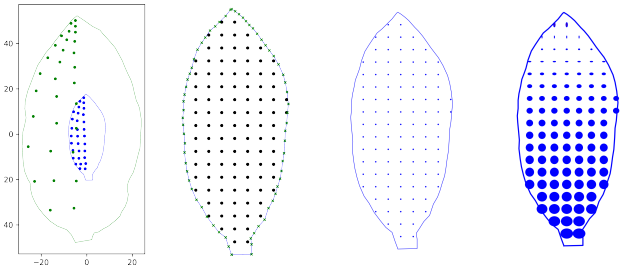


Figure 9: (a) Data extracted from Figure 1a ; (b) Initial geometrical descriptors: black (order 1) and green (order 0) ; Initialization (c) and estimation (d) of the growth factor.

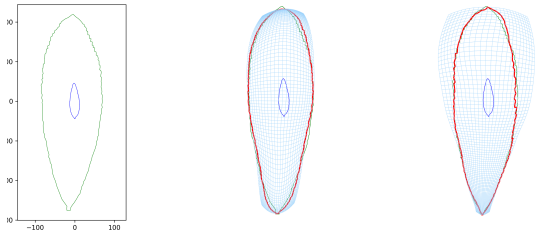


Figure 10: Modelling acropetal growth through curves registration (extracted from Figure 1b). Source in blue, target in green, deformed source in red.

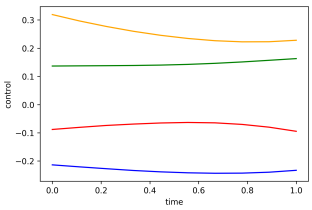


Figure 11: Estimated controls for the registration of Figure 13. Red (horizontal) and orange (vertical) for the trunk elongations, blue (horizontal) and green (vertical) for the crown ones.

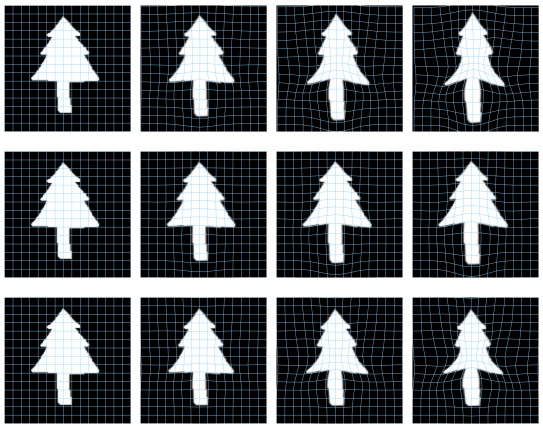
3.2. Analyzing differences in images

In a second example, we present how our framework allows to analyse the differences between two images, focusing on particular, user-chosen, aspects. The interpolation of images used in the algorithm is based on the code developed in Deformetrica [6]. Let us consider the two images presented in Figure 12 (from the MPEG-7 Core Experiment CE-Shape-1 database [5]): they show two pine trees that differ in the height and width of their trunks and crowns, and as well on the shape of the crown. In addition, the source image is segmented so that the trunk and the crown are localized. We consider the case where we are interested in the variation of these widths and heights, and then we design an adapted implicit deformation module of order 1 (see Section 2.3.3). The geometrical descriptor is initialized using the segmentation: points are uniformly positioned around the trunk and the crown (see Figure 12), we emphasize again here that they are *automatically* positioned given the segmentation and a point density. The rotation matrices attached to these points (defining local frames, see Section 2.3.3) are initialized to identity. In order to study the variation of width and height within the trunk and the crown, we define the growth model tensor C in the following manner: equal to $C[i] = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{smallmatrix}\right)$ if the i-th point is in the trunk and $C[i] = \left(\begin{smallmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$ if it is in the crown. This means that the control variable h is of dimension 4 at each time, the first two components control the growth factor at the trunk (horizontal and vertical stretching), and the last two control it at the crown. In the trunk and crown areas, vertical and horizontal stretching can be used *independently* but *uniformly* in the two areas. The result of the registration is presented in Figure 13 (first row). Here the goal is not to obtain the best fit as possible, but to obtain the best one using only the *allowed vocabulary* imposed by the choice of deformation module and then to *analyse* it by studying the values of the controls. They are presented in Figure 11, positive controls correspond to an elongation while negative ones correspond to shrinking. By analysing them one can deduce that the width of the trunk is almost constant, the crown becomes larger and the whole tree becomes taller.

An interesting feature of IMODAL is the possibility to follow the action of each part of the total deformation, we present in Figure 13 (last two rows) the deformation obtained by following the action of the vertical elongation of

(a) Source (b) Target (c) Modules

Figure 12: Image analysis. Points in 12c are the implicit module's geometrical descriptors (trunk in orange, crown in green).



$t = 0$ $t = 0.3$ $t = 0.7$ $t = 1$

Figure 13: First row: registration; Second row: Following trunk elongation; Third row: Following crown shrinking.

the trunk and the horizontal shrinking of the crown.

3.3. Learning the intrinsic local frames

In all the previous examples, the local frames attached to the object to define the principal directions of the growth factor are supposed to be known. However, this is not always the case and we show here that this parameter can be estimated. We consider the registration of the Stanford bunny and a 'sheared' version of it (see Figure 14). These surfaces have no point-correspondence and then we use the varifold [8] setting to compute the data attachment term. We use an implicit deformation module of order one, and initialize the point-components of the geometrical descriptor in a convex envelop of the source shape (this is done automatically given a point density). The rotation matrices attached to these points and coding for the local orientation of the object are initialized to the identity matrix. We *estimate* them while performing the registration by using a callback function. Given the data, we assume that the growth model tensor, is the same at each point (it is then of dimension 3), and we estimate it. We combine this implicit module with a module generating a global translation, one

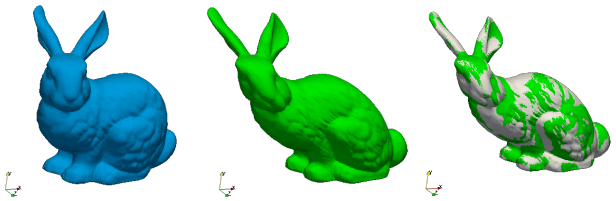


Figure 14: (left) Source ; (middle) Target ; (right) Match

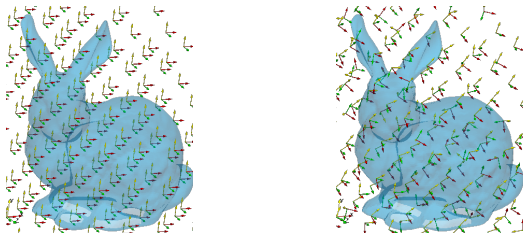


Figure 15: Local frames at $t = 0$: initialisation (left) and optimisation (right).

generating a global rotation because the data are not rigidly registered and an implicit module of order 0 (with a high cost penalization). We present in Figure 14 the result of the registration. We also present in Figure 15 the initialization and optimisation of the local frames defining the initial geometrical descriptor of the implicit module. One can see that they align themselves with the 'stretching' directions. The estimated growth factor, is $(3.2, -2.4, 0.3)$: there is an elongation in the 'updated' x-axis (in red) and a shrinking in the 'updated' y-axis (in yellow). Here the shearing is built as an elongation along an axis, a contraction along an orthogonal one and a global rotation.

4. Discussion

As we have seen, the implicit deformation module approach implemented in the IMODAL library offers a versatile environment for the user to incorporate prior knowledge for shape and image analysis through large deformations, and provides tools to disentangle the variability between objects. The implementation scales from 2D to 3D shapes and we illustrated how some parameters of the deformation model can be easily learnt when unknown paving the way, beyond to above proofs of concept, to a more in-depth exploitation in biological and medical contexts.

References

[1] Sylvain Arguillere. *Géométrie sous-riemannienne en dimension infinie et applications à l'analyse mathématique des formes*. PhD thesis, 2014.

[2] Sylvain Arguillere, Emmanuel Trélat, Alain Trouvé, and Laurent Younes. Shape deformation analysis from the op-

timal control viewpoint. *Journal de mathématiques pures et appliquées*, 104(1):139–178, 2015.

[3] J Ashburner. Computational anatomy with the spm software. *Magnetic Resonance Imaging*, 27:1163–1174, October 2009.

[4] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computational Anatomy: Computing metrics on anatomical shapes. In *Proceedings of the International Symposium on Biomedical Imaging*, pages 341–344. IEEE, 2002.

[5] M Bober, JD Kim, HK Kim, YS Kim, WY Kim, and K Muller. Summary of the results in shape descriptor core experiment. mpeg-7, iso. Technical report, IEC JTC1/SC29/WG11/MPEG99.

[6] Alexandre Bône, Maxime Louis, Benoît Martin, and Stanley Durrleman. Deformetrica 4: an open-source software for statistical shape analysis. In *International Workshop on Shape in Medical Imaging*, pages 3–13. Springer, 2018.

[7] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *arXiv preprint arXiv:2004.11127*, 2020.

[8] N. Charon and A. Trouvé. The varifold representation of nonoriented shapes for diffeomorphic registration. *SIAM Journal on Imaging Sciences*, 6(4):2547–2580, 2013.

[9] Stanley Durrleman, Marcel Prastawa, Nicolas Charon, Julie R Korenberg, Sarang Joshi, Guido Gerig, and Alain Trouvé. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage*, 101:35–49, 2014.

[10] Joan Glaunes and Mario Micheli. Matrix-valued kernels for shape deformation analysis. geometry. *Imaging and Computing*, 1(1):57–139, 2014.

[11] Anupama Goparaju, Alexandre Bone, Nan Hu, Heath B Henninger, Andrew E Anderson, Stanley Durrleman, Matthijs Jacxsens, Alan Morris, Ibolya Csecs, Nassir Marrouche, et al. Benchmarking off-the-shelf statistical shape modeling tools in clinical applications. *arXiv preprint arXiv:2009.02878*, 2020.

[12] Ulf Grenander and Michael I Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, 56(4):617–694, 1998.

[13] Ulf Grenander, Anuj Srivastava, and Sanjay Saini. A pattern-theoretic characterization of biological growth. *IEEE Transactions on Medical Imaging*, 26(5):648–659, 2007.

[14] Barbara Gris, Stanley Durrleman, and Alain Trouvé. A sub-riemannian modular framework for diffeomorphism-based analysis of shape ensembles. *SIAM Journal on Imaging Sciences*, 11(1):802–833, 2018.

[15] Mainak Das Gupta and Utpal Nath. Divergence in patterns of leaf growth polarity is associated with the expression divergence of mir396. *The Plant Cell*, 27(10):2785–2799, 2015.

[16] Dai-Ni Hsieh, Sylvain Arguillère, Nicolas Charon, Michael I Miller, and Laurent Younes. A model for elastic evolution on foliated shapes. In *International Conference on Information Processing in Medical Imaging*, pages 644–655. Springer, 2019.

[17] Michael I. Miller, Sylvain Arguillère, Daniel J. Tward, and Laurent Younes. Computational anatomy and diffeomorphometry: A dynamical systems model of neuroanatomy in the soft condensed matter continuum. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 0(0):e1425.

[18] Michael I Miller, Alain Trouvé, and Laurent Younes. Hamiltonian systems and optimal control in computational anatomy: 100 years since d'arcy thompson. *Annual Review of Biomed Engineering*, (17):447–509, November 4 2015.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[20] X Pennec. From Riemannian Geometry to Computational Anatomy. *Elements*, 2011.

[21] Laurent Risser, François-Xavier Vialard, Robin Wolz, Darryl D Holm, and Daniel Rueckert. Simultaneous fine and coarse diffeomorphic registration: application to atrophy measurement in Alzheimer's disease. *Medical Image Computing and Computer-Assisted Intervention*, 13(Pt 2):610–617, 2010.

[22] Christof Seiler, Xavier Pennec, and Mauricio Reyes. Capturing the multiscale anatomical shape variability with polyaffine transformation trees. *Medical image analysis*, 16(7):1371–1384, 2012.

[23] Stefan Sommer, Mads Nielsen, Sune Darkner, and Xavier Pennec. Higher-order momentum distributions and locally affine lddmm registration. *SIAM Journal on Imaging Sciences*, 6(1):341–367, 2013.

[24] Paul M Thompson and Arthur W Toga. A framework for computational anatomy. *Computing and Visualization in Science*, 5(1):13–34, 2002.

[25] F-X Vialard, L Risser, D Rueckert, and CJ Cotter. 3d image registration via geodesic shooting using and efficient adjoint calculation. *Journal International Journal of Computer Vision*, 97(2):229–241, April 2012.

[26] Laurent Younes. Constrained diffeomorphic shape evolution. *Foundations of Computational Mathematics*, 12(3):295–325, 2012.