UCLouvain
Institute for Information and Communication Technologies, Electronics and Applied Mathematics

ISP GROUP Image & Signal Processing

P.-Y. Gousenbourger, E. Massart, P.-A. Absil
ICTEAM Institute (UCLouvain) – Belgium
pierre-yves.gousenbourger@uclouvain.be

The problem

Find a C^1 curve $\mathbf{B} \in \mathcal{M}$, fitting data-points $d_i \in \mathcal{M}$ s.t.

$$\underset{\mathbf{B}\in\Gamma}{\operatorname{argmin}}\, E_\lambda(\mathbf{B}) := \int_{t_0}^{t_n} \left\| \frac{\mathrm{D}^2\mathbf{B}(t)}{\mathrm{d}t^2} \right\|^2_{\mathbf{B}(t)} \mathrm{d}t + \lambda \sum_{i=0}^{n} \mathrm{d}^2(\mathbf{B}(t_i), d_i),$$

Bézier spline! — regularizer — data attachment

Motivated by:
Denoising or resampling
Medical applications
Computer graphics

What it needs

A Riemannian manifold $\mathcal{M}$.
A set of m+1 manifold-valued data points $d_0, \dots, d_m \in \mathcal{M}$.
A set of m+1 time-parameters $t_0, \dots, t_m \in [0, n]$.
The Riemannian exponential "Exp"...
... and its inverse "Log".

And that's all!

What it returns

A composite C^1 curve made out of n pieces

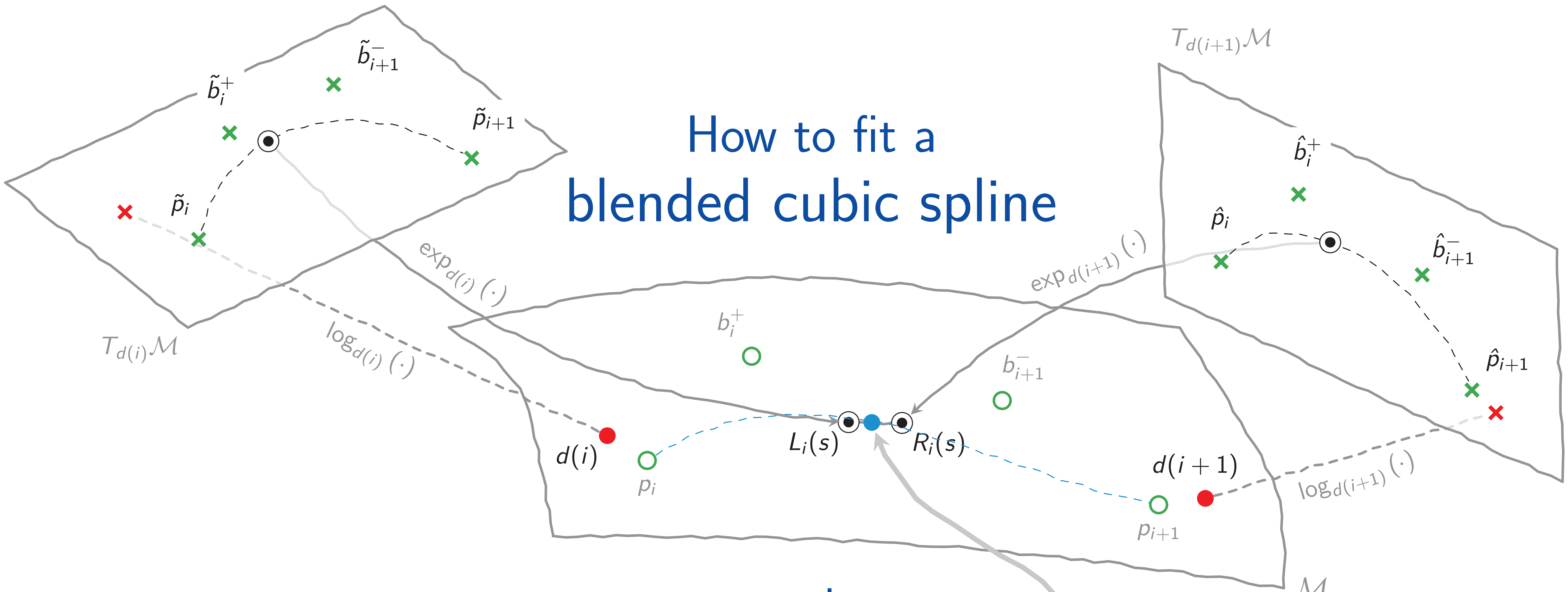
$$\mathbf{B} : [0, n] \to \mathcal{M} : t \mapsto \mathbf{B}(t) = \beta_i(t-i), \text{ with } i = \lfloor t \rfloor \dots$$

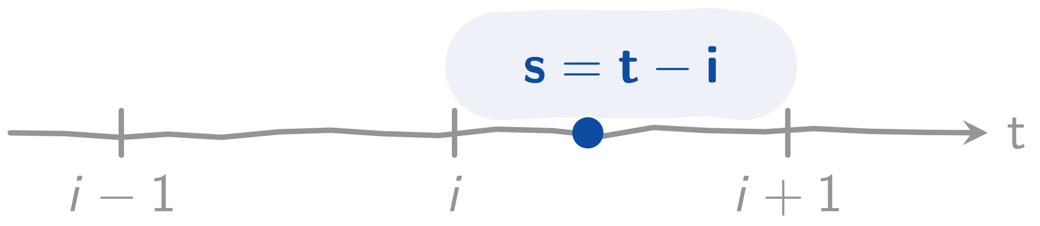
(i) ... differentiable on $[0, n]$,
(ii) ... that interpolates the data points if $m = n$ when $\lambda \to \infty$,
(iii) ... that is the natural cubic smoothing spline when $\mathcal{M} = \mathbb{R}^r$.

An efficient and easy-to-use method

(iv) ... that only uses Exp's and Log's on $\mathcal{M}$;
(v) ... that only stores $\mathcal{O}(n)$ tangent vectors;
(vi) ... that needs $\mathcal{O}(1)$ operations to compute $\mathbf{B}(t)$ at a given t.

How to fit a blended cubic spline to your data set on a manifold?





$\mathbf{B}(t) = \beta_i(s) = \mathrm{av}\,[L_i(s), R_i(s)]_w$ is a weighted geodesic averaging of $L_i(s)$ and $R_i(s)$ with a weight $w(s) = 3s^2 - 2s^3$

How it gets to it

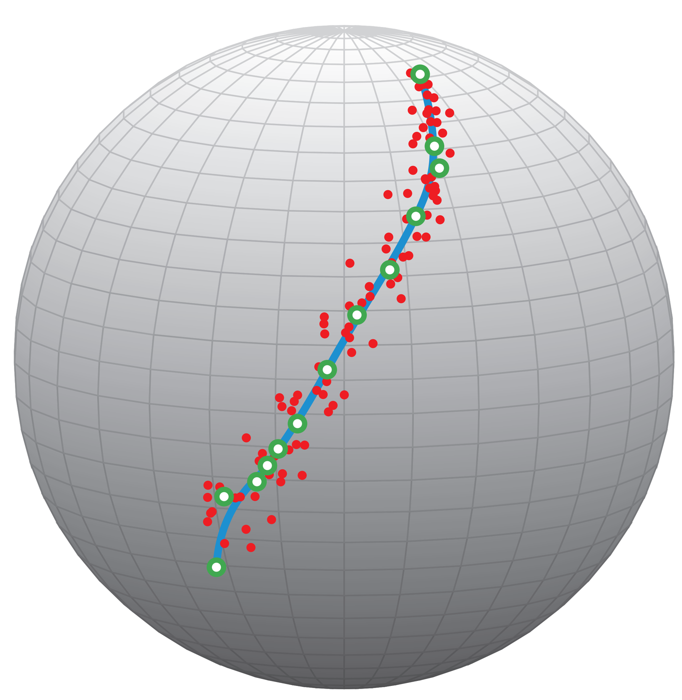
i → +1

Optimize on $T_{d(i)}\mathcal{M}$

1. Find the data point $d(i) = d_k$ whose associated t_k is the closest to $t = i$.
2. Map all data points $\{d_j\}_{j=0}^m$ to $T_{d(i)}\mathcal{M}$
3. Find the natural Bézier smoothing spline on $T_{d(i)}\mathcal{M}$. Store the control points $\tilde{p}_{i-1}, \tilde{b}_{i-1}^+, \tilde{b}_i^-, \tilde{p}_i, \tilde{b}_i^+, \tilde{b}_{i+1}^-, \tilde{p}_{i+1}$

What it looks like

$\lambda = 100,\ m = 100,\ n = 4$



$\lambda = 10^8,\ m = n = 10$