3

THE MATHEMATICAL SETTING

3.1 General remarks

3.1.1 *Introduction to images*

Here, we use a very simple model of an image, a mapping which assigns every spatial point x belonging to a certain set $\Omega \subset \mathbb{R}^d$ a *gray value* $b(x)$. Consequently, an image is viewed as a mapping. The dimension of its spatial domain is denoted by $d \in \mathbb{N}$ throughout this book.

Definition 3.1 *Let $d \in \mathbb{N}$. A function $b : \mathbb{R}^d \to \mathbb{R}$ is called a* d-dimensional image, *if*

1. *b is compactly supported,*
2. *$0 \le b(x) < \infty$ for all $x \in \mathbb{R}^d$,*
3. *$\int_{\mathbb{R}^d} b(x)^k dx$ is finite, for $k > 0$.*

The set of all images is denoted by

$$\mathrm{Img}(d) := \{\, b : \mathbb{R}^d \to \mathbb{R} \mid b \text{ is } d\text{-dimensional image} \,\}.$$

Here, the term image refers to a d-dimensional light-intensity function b; the value $b(x)$ gives the intensity of the image at a spatial position $x \in \mathbb{R}^d$. Although Definition 3.1 (1) looks restrictive at first glance, it appears to be natural for the registration of the histological sections within the HNSP. A human brain is bounded and the brain matter is completely embedded in paraffin wax. However, it might become crucial in other applications. Definition 3.1 (3) conforms naturally to the boundedness of light energy.

For all the applications we are interested in, the images are given in terms of discrete data and the intensity function has to be interpolated by some interpolation scheme; cf., Section 3.1.3. Thus, by choosing an appropriate interpolation scheme, we may even assume that the images are arbitrarily smooth. This assumption enables us to exploit fast numerical schemes which typically depend on second order derivatives.

One may also be interested in q-colored images, i.e., mappings $c : \mathbb{R}^d \to \mathbb{R}^q$, $c = (c^1, \dots, c^q)$, where $c^\ell \in \mathrm{Img}(d)$ for $\ell = 1, \dots, q$. The following concepts apply to any component of the q-colored image and can thus be extended in a straightforward way.

3.1.2 *From continuous to discrete images*

We now introduce *digital images* as a model for the discrete output of an imaging device. For an image $b \in \mathrm{Img}(d)$ we assume that the support of the image b is contained in a domain Ω, where for ease of presentation we assume

$$\Omega :=]0,1[^d \subset \mathbb{R}^d,$$

and denote the boundary of Ω by $\partial\Omega$.

Definition 3.2 *Let $d \in \mathbb{N}$, $\Omega :=]0,1[^d$, and $n_1, \dots, n_d \in \mathbb{N}$ be some given numbers. The points*

$$x_{j_1,\dots,j_d} = (x_{j_1}, \dots, x_{j_d})^\top \in \Omega \cup \partial\Omega, \tag{3.1}$$

where $1 \le j_\ell \le n_\ell$ and $1 \le \ell \le d$, are called grid points. *The array*

$$X := (x_{j_1,\dots,j_d})_{\substack{1 \le j_\ell \le n_\ell \\ \ell=1,\dots,d}} \in \mathbb{R}^{n_1 \times \cdots \times n_d} \tag{3.2}$$

is called the grid matrix.

Let $N := n_1 \cdots n_d$ and let the numbers $j \in \mathbb{N}$ $(1 \le j \le N)$ and $(j_1, \dots, j_d) \in \mathbb{N}^d$ $(1 \le j_\ell \le n_\ell, 1 \le \ell \le d)$ be related by the one-to-one lexicographical ordering, $j = \sum_{\nu=1}^{d-1} (j_{\nu+1} - 1) \prod_{\mu=1}^{\nu} n_\mu + j_1$. The vector $\vec{X} := (x_j)_{j=1,\dots,N} \in \mathbb{R}^N$, where $x_j = x_{j_1,\dots,j_d}$, is called the grid vector. *The set $\Omega_d := \{\, x_j, j = 1, \dots, N \,\}$ is called an $n_1 \times \cdots \times n_d$* grid.

In connection with the discretization of the partial differential equations to be discussed in Part II, we now introduce three different grids which are of particular interest.

Definition 3.3 *Let $d \in \mathbb{N}$, $\Omega :=]0,1[^d$, and $n_1, \dots, n_d \in \mathbb{N}$ be some given numbers. For $1 \le j_\ell \le n_\ell$, $1 \le \ell \le d$, let*

$$x_j^D := \left(\frac{j_1}{n_1 + 1}, \dots, \frac{j_d}{n_d + 1} \right)^\top, \tag{3.3}$$

$$x_j^N := \left(\frac{2j_1 - 1}{2n_1}, \dots, \frac{2j_d - 1}{2n_d} \right)^\top, \tag{3.4}$$

$$\text{and} \quad x_j^p := \left(\frac{j_1 - 1}{n_1}, \dots, \frac{j_d - 1}{n_d} \right)^\top \tag{3.5}$$

be grid points. The corresponding grids Ω_D, Ω_N, and Ω_p are called Dirichlet, Neumann, *and* periodic grids, *respectively.*

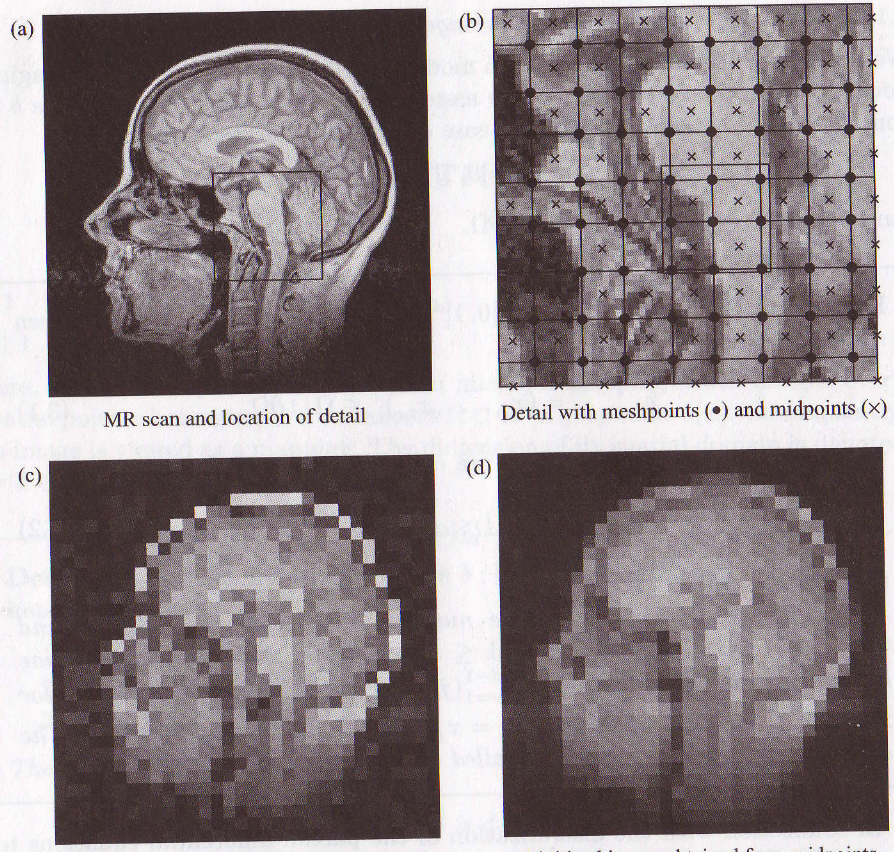

FIG. 3.1 (a) Image, (b) mesh- and midpoints in a detail of the image, (c) mesh and (d) midpoint digitized images (32×32 pixels).

For the discretization of a continuous image b on a discrete grid Ω_d two models are common; see also Fig. 3.1 for an illustration.

Meshpoint model:

$$b_j^{\bullet} := b_{j_1,\ldots,j_d}^{\bullet} := b(x_j) \quad \text{for all} \quad x_j \in \Omega_d. \tag{3.6}$$

Midpoint model:

$$b_j^{\times} := b_{j_1,\ldots,j_d}^{\times} := \int_{c_j} b(x)dx \quad \text{for all} \quad j = 1,\ldots,N. \tag{3.7}$$

Here, c_j denotes a small region with center x_j.

For both models we define $b(x) = 0$ for $x \notin \Omega$.

Definition 3.4 *Let $d \in \mathbb{N}$, $b \in \mathrm{Img}(d)$, and let Ω_d be a $n_1 \times \cdots \times n_d$ grid; cf., Definition 3.2. The arrays*

$$B^{\bullet} := (b_{j_1,\ldots,j_d}^{\bullet})_{\substack{j_\ell=1,\ldots,n_\ell,\\ \ell=1,\ldots,d}}, \quad B^{\times} := (b_{j_1,\ldots,j_d}^{\times})_{\substack{j_\ell=1,\ldots,n_\ell\\ \ell=1,\ldots,d}} \in \mathbb{R}^{n_1\times\cdots\times n_d},$$

where $b_j^{\bullet}$ and $b_j^{\times}$ are obtained respectively from the mesh- or the midpoint approach, are called d-dimensional digital images.

If the distinction between mesh- and midpoint discretization is not of importance, we may simply denote a d-dimensional discrete image by $B \in \mathbb{R}^{n_1\times\cdots\times n_d}$.

In the following, we assume that a digital image B is an appropriate model for the output of an imaging device. This is, of course, a considerable simplification. Different devices yield different outputs. Moreover, the output depends on additional parameters, e.g., light.

Many imaging devices, e.g., CCD cameras or flat bed scanners, perform a further image processing step by also digitizing the image values. In particular for the scans of the two-dimensional HNSP images we end up with values $\tilde{b}_i \in \{0,\ldots,255\}$, where

$$\tilde{b}_j = \begin{cases} 0, & b_j < 0, \\ 255, & b_j > 255, \\ [b_j], & \text{else}. \end{cases}$$

Here and later,

$$\lfloor x \rfloor := \max\{j \in \mathbb{Z} \mid j \le x\}, \tag{3.8}$$
$$\lceil x \rceil := \min\{j \in \mathbb{Z} \mid j \ge x\}, \tag{3.9}$$
$$[x] := \min\{j \in \mathbb{Z} \mid j - 0.5 \le x < j + 0.5\}. \tag{3.10}$$

We also use this notation for vectors, e.g., $[x] = ([x_1],\ldots,[x_d])^\top$ for $x \in \mathbb{R}^d$.

3.1.3 *From discrete to continuous images*

Given a discrete image B one may also want to assign image values at spatial positions that are not necessarily grid points. In other words, we are looking for an *interpolation scheme* $\mathcal{I}$,

$$\mathcal{I} : \mathbb{R}^{n_1 \times \cdots \times n_d} \times \mathbb{R}^d \to \mathbb{R}.$$

For an image B, the scheme assigns an intensity value $\mathcal{I}(B,x)$ to any position $x \in \mathbb{R}^d$.

We discuss different variants; see also Fig. 3.2 for a comparison. For ease of presentation, we focus on an $n_1 \times \cdots \times n_d$ periodic grid. The extension to other grids is straightforward. For $x \notin \Omega$ we set $\mathcal{I}(B,x) := 0$. Let $n = (n_1, \dots, n_d)$ and $x \in \Omega$ be an arbitrary point and $[x]$ the corresponding closest grid point and $\lfloor x \rfloor$ the corresponding closest grid point smaller then x, i.e.,

$$[x] = \left(\frac{[x_1 n_1]}{n_1}, \dots, \frac{[x_d n_d]}{n_d} \right)^\top \quad \text{and} \quad \lfloor x \rfloor = \left(\frac{\lfloor x_1 n_1 \rfloor}{n_1}, \dots, \frac{\lfloor x_d n_d \rfloor}{n_d} \right)^\top.$$

FIG. 3.2 TOP LEFT: Digital image 59×59 pixels, TOP RIGHT: next-neighbor interpolation, BOTTOM LEFT: bi-linear interpolation, and BOTTOM RIGHT: interpolation based on cosine basis functions.

Local methods

1. *Next-neighbor interpolation*

$$\mathcal{I}_{\mathrm{NN}}(B,x) := B([x]). \tag{3.11}$$

Note that the function $\mathcal{I}_{\mathrm{NN}}(B, \cdot\,)$ is in general not continuous.

2. *d-linear interpolation*

$$\begin{aligned} \mathcal{I}_{\mathrm{linear}}(B,x) & \\ := \sum_{k \in \{0,1\}^d} & B\left(\frac{\lfloor n_1 x_1 \rfloor + k_1}{n_1}, \dots, \frac{\lfloor n_d x_d \rfloor + k_d}{n_d} \right) \\ & \times \prod_{j=1}^{d} \left(\frac{(-1)^{k_j}}{n_j} (\lfloor x_j n_j \rfloor + 1 - k_j - x_j n_j) \right). \end{aligned} \tag{3.12}$$

Note that the function $\mathcal{I}_{\mathrm{linear}}(B, \cdot\,)$ is continuous but in general not continuously differentiable.

As a compromise between computational effort and accuracy, we typically exploit bi- ($d = 2$) and tri-linear ($d = 3$) interpolation schemes here.

Global methods

Let N be the number of grid points and $(\psi_j, j = 1, \dots, N)$ be a set of basis functions such that the interpolation problem

$$\sum_{j=1}^{N} \alpha_j \psi_j(i) = B(i), \quad i \in \Omega_d$$

has a unique solution for any $B \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. For a given discrete image B we define

$$\mathcal{I}_{\psi_1,\dots,\psi_N}(B,x) := \sum_{j=1}^{N} \alpha_j \psi_j(x). \tag{3.13}$$

Typical choices of basis function are tensor products of Lagrange interpolation polynomials, sinc functions (see, e.g., Shannon (1949)), B-splines (see, e.g., Thévenaz et al (2000)), a collection of sine and cosine functions, a collection of cosine functions, or wavelets (see, e.g., Aldroubi & Gröchening (2001)).

3.2 Mathematical notation

For later reference, we introduce some general notation. Let $\Omega \subset \mathbb{R}$ be an open set. By $L_2(\Omega)$ we denote the set of squared integrable functions,

$$L_2(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid \int_\Omega |f(x)|^2 \, dx < \infty \right\}$$

with the inner product

$$\langle f, g \rangle_{L_2} := \int_\Omega f(x)g(x)\,dx.$$

Furthermore, by $\mathcal{C}^k(\Omega)$ we denote the set of k-times differentiable functions and by $\mathcal{C}_c^k(\Omega)$ we denote the set of k-times differentiable functions with compact support.

Definition 3.5 *Let $d \in \mathbb{N}$. Moreover, let $\mathcal{D} := \mathcal{C}_c^\infty(\mathbb{R}^d)$ denote the space of arbitrary often differentiable functions with compact support and $\mathcal{D}^*$ its dual. For $z \in \mathbb{R}^d$ we define the* point evaluation functional *or* Dirac δ functional *by*

$$\delta_z[\varphi] = \varphi(z) \quad \text{for all} \quad \varphi \in \mathcal{D}.$$

Let $f \in L_2(\mathbb{R}^d)$ be such that $\int_{\mathbb{R}^d} f(x)\,dx = 1$ and for $h > 0$, let $f_h(x) := h^{-d} f(x/h)$. For any $\varphi \in \mathcal{D}$, we have

$$\lim_{h \downarrow 0} \int_{\mathbb{R}^d} f_h(x)\varphi(x)\,dx = \varphi(0) = \delta_0[\varphi].$$

Thus, as usual, we make use of the notion "δ-function" for the limit, i.e.,

$$\delta_z[\varphi] = \int_{\mathbb{R}^d} \delta_z(x)\varphi(x)\,dx = \varphi(z).$$

For the various differential operators used throughout this book, we provide the overview given in Table 3.1.

Table 3.1 *Notation for derivatives.*

∂_{x_j}	partial derivative with respect to the j^{th} component
∇	gradient, $\nabla f = (\partial_{x_1} f, \dots, \partial_{x_d} f)^\top$ for $f : \mathbb{R}^d \to \mathbb{R}$, $\nabla\varphi = (\partial_{x_k}\varphi_j)_{j,k} \in \mathbb{R}^{d\times d}$ for $\varphi : \mathbb{R}^d \to \mathbb{R}^d$
∇_p	gradient with respect to the variable p
div	divergence, $\operatorname{div}\varphi = \langle \nabla, \varphi \rangle_{\mathbb{R}^d} = \partial_{x_1}\varphi_1 + \dots + \partial_{x_d}\varphi_d$ for $\varphi : \mathbb{R}^d \to \mathbb{R}^d$
∇^2	Hessian matrix $\nabla^2 f = (\partial_{x_j,x_k} f)_{j,k} \in \mathbb{R}^{d\times d}$ for $f : \mathbb{R}^d \to \mathbb{R}$,
Δ	Laplace operator, $\Delta f = \partial_{x_1,x_1} f + \dots + \partial_{x_d,x_d} f$ for $f : \mathbb{R}^d \to \mathbb{R}$, $\Delta\varphi = (\Delta\varphi_1, \dots, \Delta\varphi_d)^\top$ for $\varphi : \mathbb{R}^d \to \mathbb{R}^d$
D^κ	$D^\kappa f = (\partial/\partial_{x_1})^{\kappa_1} \cdots (\partial/\partial_{x_d})^{\kappa_d} f$ for $f : \mathbb{R}^d \to \mathbb{R}$ and $\kappa \in \mathbb{N}^d$
$d\mathcal{F}[\varphi; \psi]$	Gâteaux derivative of a functional $\mathcal{F}$ at φ with respect to a perturbation ψ

Finally, we use the Kronecker calculus; see, e.g., Horn & Johnson (1991, §4) or Brewer (1978). In particular, for $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$, we have

$$A \otimes B := \begin{pmatrix} a_{1,1}B & \cdots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \cdots & a_{m,n}B \end{pmatrix} \in \mathbb{C}^{mp\times nq}.$$

The identity matrix is denoted by $I_n \in \mathbb{R}^{n\times n}$.

3.3 The registration problem

Image registration means finding a suitable spatial transformation such that a transformed image becomes similar to another one. The registration problem typically occurs when two images exhibit essentially the same object, but the object or the position of the imaging device is different for the two images. Thus, the images are spatially not *aligned* or not *registered*, i.e., there is no direct spatial correspondence between them. Typically this problem occurs if the images are taken from different perspectives, times, or imaging devices.

Usually, one of the images is viewed as a *reference* R and the other one as a deformable *template* T or *study* . Ideally, we are looking for a *transformation* $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, such that the reference image R and the deformed template image T_φ are similar, where

$$T_\varphi(x) := T \circ \varphi(x) = T(\varphi(x)). \tag{3.14}$$

In many practical applications the problem is even harder. This is because the reference and template images show different but similar objects. Typical examples include the registration of images taken before and after surgery for a patient or the registration of consecutive tissue sections from a histological sectioning process. Here, the aim of the registration is to remove the *artificial* differences introduced for example by movement, but to retain the *real* differences due to the variations of the different objects.

For a mathematical treatment of the problem, similarity needs to be measured in an appropriate way. If the images are taken from different imaging devices, there might not be a correspondence between the intensities $R(x)$ and $T_\varphi(x)$ for an optimal φ. Thus, we may also allow for an additional function $g : \mathbb{R} \to \mathbb{R}$ and compare $R(x)$ and $g \circ T_\varphi(x) = g(T(\varphi(x)))$. Moreover, one may also consider distance measures which are not related to intensities but are based on the so-called *mutual information* of the images; cf., Section 6.4. Different similarity measures $\mathcal{D}$ are introduced in the forthcoming sections. The general registration problem reads as follows.

Problem 3.1 *Given a distance measure $\mathcal{D} : \operatorname{Img}(d)^2 \to \mathbb{R}$ and two images $R, T \in \operatorname{Img}(d)$, find a mapping $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ and a mapping $g : \mathbb{R} \to \mathbb{R}$ such that $\mathcal{D}(R, g \circ T \circ \varphi) = \min$.*

As we will see in the following, Problem 3.1 is ill-posed. Thus, a direct approach is impossible. Different regularization approaches are discussed subsequently.

3.3.1 *Restricted transformations*

In many applications there exist implicit requirements with respect to the transformation. A typical example is that the transformation has to be smooth or even diffeomorphic. Approaches based on regularization are investigated in Part II. Explicit parametric requirements are even more popular in the literature and typical example are given below.

Rigid transformation, $\varphi(x) = Qx + b$, where $Q \in \mathbb{R}^{d\times d}$ is orthogonal with $\det Q = 1$ and $b \in \mathbb{R}^d$. The transformation is called rigid because only rotations and translations of the coordinates are permitted.

(Affine) linear transformation, $\varphi(x) = Ax + b$, $A \in \mathbb{R}^{d\times d}$ with $\det A > 0$, $b \in \mathbb{R}^d$. Note that in contrast to other authors we allow for individual scalings.

Polynomial transformation, $\varphi \in \Pi_q^d(\mathbb{R}^d)$; cf., Definition 3.6.

B-spline transformation,

$$\varphi = (\varphi_1, \ldots, \varphi_d)^\top : \mathbb{R}^d \to \mathbb{R}^d, \quad \text{where} \quad \varphi_\ell \in \text{Spline}_{y,q}(\mathbb{R}^d),$$

where $\text{Spline}_{y,q}(\mathbb{R}^d)$ is a d-dimensional spline space spanned by B-splines of degree q with respect to the knots y_k, $k = 1, \ldots, K$; cf., e.g., De Boor (1978).

For reference purposes we also present a formal definition of polynomials.

Definition 3.6 *Let $d, q \in \mathbb{N}$ and $\kappa = (\kappa_1, \ldots, \kappa_d) \in \mathbb{N}^d$. The* polynomials *of degree q are defined by*

$$\Pi_q(\mathbb{R}^d) := \left\{ \psi : \mathbb{R}^d \to \mathbb{R} \mid \psi(x) = \textstyle\sum_{|\kappa|\le q} \alpha_\kappa x^\kappa, \ \alpha_\kappa \in \mathbb{R} \right\}$$

where $|\kappa| = \kappa_1 + \cdots + \kappa_d$ and for $x \in \mathbb{R}^d$ we set $x^\kappa := x_1^{\kappa_1} \cdots x_d^{\kappa_d}$. The set of d-dimensional polynomials *of degree q is defined by*

$$\Pi_q^d(\mathbb{R}^d) := \left\{ \varphi : \mathbb{R}^d \to \mathbb{R}^d \mid \varphi_\ell \in \Pi_q(\mathbb{R}^d), \ \ell = 1, \ldots, d \right\}.$$

3.3.2 *The Lagrange and Euler frames*

For the image interpolation there are two reference frames, the Lagrange coordinates and the Euler coordinates. If we assume the transformation φ to be invertible, we can write $\tilde{x} := \varphi(x)$ or equivalently $x = \tilde{\varphi}(\tilde{x}) := \varphi^{-1}(\tilde{x})$.

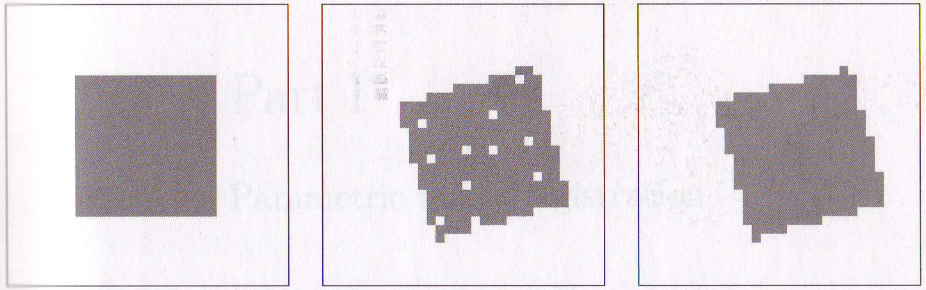

FIG. 3.3 Image and affine linear mapped image. LEFT: Input image 32 × 32 pixels, MIDDLE: next-neighbor interpolation using the Lagrange coordinates, RIGHT: next-neighbor interpolation using the Euler coordinates. Image has been rotated by 20 degrees.

From a computational point of view we have

$$B^{\text{Lagrange}}(\varphi(i,j)) := B(i,j) \quad \text{and} \quad B^{\text{Euler}}(i,j) := B(\overbrace{\varphi^{-1}(\overbrace{i,j}^{\tilde{x}})}^{\tilde{\varphi}(\tilde{x})}).$$

Figure 3.3 illustrates the difference between the two frames and indicates that the Euler frame is preferable. Thus, we take advantage of the Euler coordinates.