as long as the two images share the same gray-scale range; see Section 10.7. Though this possibility presents an interesting feature for some applications, it is certainly not appropriate for others. The application of fluid registration for a physically elastic object, as introduced in Section 9.1, is doubtful. The physical motivation of the regularizer is based on fluid-like bodies such as honey. Hands and brains do not deform in general like honey.

11

DIFFUSION REGISTRATION

In this chapter we propose a novel gradient-based regularization term and devise a fast and stable implementation for a finite difference approximation of the underlying partial differential equation. Since this PDE may be viewed as a generalized diffusion equation, we call our new scheme *diffusion registration.*

In contrast to the physically motivated elastic and fluid registrations, see Chapters 9 and 10, the regularizer is motivated by smoothing properties of the displacement. Actually, another important motivation is that a registration step can be performed in $\mathcal{O}(N)$ floating point operations, where N denotes the number of unknowns. The main tool is the so-called additive operator splitting (AOS) scheme; see, e.g., Weickert (1998).

The idea is to split the original problem into a number of simpler problems, which allow for a fast numerical solution. In Section 11.3, we give a new proof for the accuracy of the AOS scheme. The proof is based purely on matrix analysis and therefore the result applies as well to more general situations; cf., Fischer & Modersitzki (1999).

Besides this, we also discuss Thirion's so-called *demons registration*; cf., Thirion (1998). Thirion proposed a method which works well in practice but its derivation is guided by intuition and not entirely understood. In the literature one may find several attempts to shed some light on his approach; see, e.g., Pennec et al (1999) or Bro-Nielsen & Gramkow (1996). Because Thirion's approach offers a variety of possible implementations, the underlying theory is widespread. However, the main point is that he calculates the deformations by regularizing certain driving forces by a Gaussian convolution filter. We show that this technique may be viewed as a special (low-order) approximation to the PDE connected to our new scheme and we thereby gain some new insight into Thirion's approach.

Diffusion registration was introduced by Fischer & Modersitzki (1999) and it is based on the distance measure $\mathcal{D}$ (cf., eqn (8.10)) and the regularizer

$$S[u] := \frac{1}{2}a[u,u], \quad \text{where} \quad a[u,v] = \sum_{\ell=1}^{d} \int_{\Omega} \langle \nabla u_\ell, \nabla v_\ell \rangle \, dx \tag{11.1}$$

and Neumann boundary conditions are imposed, i.e.,

$$\langle \nabla u_\ell(x), \vec{n}(x) \rangle_{\mathbb{R}^d} = \langle \nabla v_\ell(x), \vec{n}(x) \rangle_{\mathbb{R}^d} = 0 \quad \text{for } x \in \partial\Omega$$

and $\ell = 1, \ldots, d$. Here, $\vec{n}$ denotes the outer normal unit vector of $\partial\Omega$, $\Omega :=]0,1[^d$.

The idea behind this regularizer is to privilege smooth deformations while minimizing oscillations of the components of the displacement, see also Horn & Schunck (1981). Another important advantage of this regularization is that the Euler–Lagrange equations decouple with respect to the spatial directions. As a consequence, the matrix representation A of the finite difference-based discretized derivative of this smoother is a d-by-d block diagonal matrix.

11.1 Continuous and discrete Laplace equations

To begin with, we state the Euler–Lagrange equations for this particular registration.

Theorem 11.1 *The Euler–Lagrange equations for $\mathcal{J} = \mathcal{D} + \alpha\mathcal{S}$, where $\mathcal{D}$ is defined by eqn (8.10) and $\mathcal{S}$ is defined by eqn (11.1) are*

$$f(x, u(x)) + \alpha\Delta u(x) = 0, \quad x \in \Omega, \tag{11.2}$$

$\langle \nabla u_\ell(x), \vec{n}(x)\rangle_{\mathbb{R}^d} = 0$ *for* $x \in \partial\Omega$ *and* $\ell = 1, \dots, d$.

Proof Follows from Theorem 8.1, $d\mathcal{S}[u; v] = a[u, v]$, and, for a fixed ℓ, setting $\tilde{a}[u_\ell, v_\ell] := \int_\Omega \langle \nabla u_\ell, \nabla v_\ell\rangle \, dx$, from an application of Green's formula,

$$\begin{aligned}\tilde{a}[u_\ell, v_\ell] &= \int_\Omega \langle \nabla u_\ell, \nabla v_\ell\rangle \, dx \\ &= \int_{\partial\Omega} v_\ell \langle \nabla u_\ell, \vec{n}\rangle \, dA - \int_\Omega (\Delta u_\ell) v_\ell \, dx \\ &= -\int_\Omega (\Delta u_\ell) v_\ell \, dx.\end{aligned}$$

□

For a discretization of the Laplace operator we use the finite difference approximation introduced in Section 8.4 (eqn (3.4)),

$$\Delta u_\ell(X) \approx S^{\text{diff},d} * u_\ell(X). \tag{11.3}$$

For the most important cases $d = 2, 3$ the stencils $S^{\text{diff},d}$ are summarized in Table 11.1. Thus, using the grid notation introduced in Section 8.4 (cf., Definition 3.2), we may also write

$$\Delta[u_\ell](X) \approx S^{\text{diff},d} * u_\ell(X) \quad \text{or} \quad \Delta[u_\ell](\vec{X}) \approx A^{\text{diff},d}\, u_\ell(\vec{X}).$$

Note that the Neumann boundary conditions have to be taken into account; cf. Section 8.4. A discrete formulation of the Euler–Lagrange equations (cf., Theorem 11.1), then reads

$$f(\vec{X}, \vec{U}) + \alpha I_d \otimes A^{\text{diff},d}\, \vec{U} = 0. \tag{11.4}$$

Table 11.1 *Matrix stencils for the discrete Laplace operator.*

$$S^{\text{diff},d}_{n_1,\dots,n_2} = \begin{cases} -2d, & n_\ell = 2,\ \ell = 1, \dots, d \\ 1, & n_j = 1, 3, n_\ell = 2,\ \ell = 1, \dots, d,\ \ell \neq j. \end{cases}$$

$$S^{\text{diff},2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad S^{\text{diff},3}_{:,:,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$S^{\text{diff},3}_{:,:,2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -6 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad S^{\text{diff},3}_{:,:,3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

11.2 Factorization of the discrete Laplace operator

The matrix representation of the convolution with S, where S is an arbitrary symmetric, d-dimensional stencil, can be expressed recursively. An important building block is the matrix

$$M_m := \begin{pmatrix} 1 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{m\times m}. \tag{11.5}$$

To explain the following complex notation, we present an example for dimension $d = 2$.

Example 11.1 Let $d = 2$ and S be a symmetric matrix stencil,

$$S = \begin{pmatrix} S_{1,1} & S_{2,1} & S_{1,1} \\ S_{1,2} & S_{2,2} & S_{1,2} \\ S_{1,1} & S_{2,1} & S_{1,1} \end{pmatrix}.$$

For the convolution

$$\begin{aligned} v(x_i) &:= S * w(x_i) \\ &= \begin{cases} S_{1,1} w(x_{i_1-1,i_2-1}) + S_{2,1} w(x_{i_1-1,i_2}) + S_{1,1} w(x_{i_1-1,i_2+1}) \\ +S_{1,2} w(x_{i_1,i_2-1}) + S_{2,2} w(x_{i_1,i_2}) + S_{1,2} w(x_{i_1,i_2+1}) \\ +S_{1,1} w(x_{i_1+1,i_2-1}) + S_{2,1} w(x_{i_1+1,i_2}) + S_{1,1} w(x_{i_1+1,i_2+1}) \end{cases} \end{aligned}$$

with respect to Neumann boundary conditions, we may also write

$$\vec{V} = A^{(2)} \cdot \vec{U},$$

where

$$A^{(2)} = \begin{pmatrix} A_2^{(2,1)} + A_1^{(2,1)} & A_1^{(2,1)} & & & \\ A_1^{(2,1)} & A_2^{(2,1)} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & A_2^{(2,1)} & A_1^{(2,1)} \\ & & & A_1^{(2,1)} & A_2^{(2,1)} + A_1^{(2,1)} \end{pmatrix} \in \mathbb{R}^{N\times N},$$

$N = n_1 n_2$, and the matrices $A_1^{(2,1)}$ and $A_2^{(2,1)}$ are given by

$$\begin{aligned} A_q^{(2,1)} &= S_{2,q} I_{n_1} + S_{1,q} M_{n_1} \\ &= \begin{pmatrix} S_{2,q} + S_{1,q} & S_{1,q} & & & \\ S_{1,q} & S_{2,q} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & S_{1,q} & S_{2,q} & S_{1,q} \\ & & & S_{1,q} & S_{2,q} + S_{1,q} \end{pmatrix} \in \mathbb{R}^{n_1\times n_1}. \end{aligned}$$

Note that Neumann boundary conditions have been incorporated.

Using the Kronecker calculus, we may also write

$$A^{(2)} = I_{n_2} \otimes A_2^{(2,1)} + M_{n_2} \otimes A_1^{(2,1)}.$$

It turns out that the above construction can be applied to any dimension d. For $p_k = 1, 2$, $N_k := n_1 \cdots n_k$, and $k = 1, \ldots, d-1$, we have

$$\begin{aligned} A^{(d,d-1)}_{p_{d-1},\ldots,p_1} &:= I_{n_1} \otimes S_{2,p_{d-1},\ldots,p_1} + M_{n_1} \otimes S_{1,p_{d-1},\ldots,p_1} \qquad (11.6) \\ &\in \mathbb{R}^{n_1\times n_1}, \\ A^{(d,d-k)}_{p_{d-k},\ldots,p_1} &:= I_{n_k} \otimes A^{(d,d-k+1)}_{2,p_{d-k-1},\ldots,p_1} + M_{n_k} \otimes A^{(d,d-k+1)}_{1,p_{d-k-1},\ldots,p_1} \qquad (11.7) \\ &\in \mathbb{R}^{N_k\times N_k}, \\ A^{(d)} &:= I_{n_d} \otimes A_2^{(d,1)} + M_{n_d} \otimes A_1^{(d,1)} \qquad (11.8) \\ &\in \mathbb{R}^{N_d\times N_d}. \end{aligned}$$

In particular, for dimension $d = 2, 3$, we have

$$\begin{aligned} A_p^{(2,1)} &= I_{n_1} \otimes S_{2,p} + M_{n_1} \otimes S_{1,p}, \quad p = 1, 2, \\ A^{(2)} &= I_{n_2} \otimes A_2^{(2,1)} + M_{n_2} \otimes A_1^{(2,1)} \end{aligned}$$

(cf., Example 11.1) and

$$\begin{aligned} A^{(3,2)}_{p,q} &= I_{n_1} \otimes S_{2,p,q} + M_{n_1} \otimes S_{1,p,q}, \quad p, q = 1, 2 \\ A^{(3,1)}_{q} &= I_{n_2} \otimes A^{(3,2)}_{2,q} + M_{n_2} \otimes A^{(3,2)}_{1,q}, \quad q = 1, 2 \\ A^{(3)} &= I_{n_3} \otimes A_2^{(3,1)} + M_{n_3} \otimes A_1^{(3,1)}. \end{aligned}$$

Note that the matrix $-2I_m + M_m$ might be viewed as a discrete version of $\partial_{x,x}$ with Neumann boundary conditions.

If, in particular, the stencils defined in Table 11.1 are used, the description simplifies considerably. Here,

$$\begin{aligned} A^{\text{diff},2} &= S^{\text{diff},2}_{2,2} I_{n_1 n_2} + S^{\text{diff},2}_{1,2} I_{n_2} \otimes M_{n_1} + S^{\text{diff},2}_{2,1} M_{n_2} \otimes I_{n_1} \\ &= -4 I_{n_1 n_2} + I_{n_2} \otimes M_{n_1} + M_{n_2} \otimes I_{n_1}, \\ A^{\text{diff},3} &= S^{\text{diff},3}_{2,2,2} I_{n_1 n_2 n_3} + S^{\text{diff},3}_{1,2,2} I_{n_3} \otimes I_{n_2} \otimes M_{n_1} \\ &\quad + S^{\text{diff},3}_{2,1,2} I_{n_3} \otimes M_{n_2} \otimes I_{n_1} + S^{\text{diff},3}_{2,2,1} M_{n_3} \otimes I_{n_1} \otimes I_{n_1} \\ &= -6 I_{n_1 n_2 n_3} + I_{n_3} \otimes I_{n_2} \otimes M_{n_1} \\ &\quad + I_{n_3} \otimes M_{n_2} \otimes I_{n_1} + M_{n_3} \otimes I_{n_1} \otimes I_{n_1}. \end{aligned}$$

We are now ready to give an eigenvalue decomposition of the matrix $A^{(d)}$. The proof is based on the factorization of M_m and the recursive structure of $A^{(d)}$. For later usage, we introduce

$$C_m := \left(\cos \frac{(2j+1)k\pi}{2m} \right)_{j,k=0,\ldots,m-1} \in \mathbb{R}^{m\times m}, \qquad (11.9)$$

$$V_m := C_m \operatorname{diag}(\sqrt{1/m}, \sqrt{2/m}, \ldots, \sqrt{2/m}) \in \mathbb{R}^{m\times m}, \qquad (11.10)$$

$$D_m := 2 \operatorname{diag}\Big(\cos \frac{k\pi}{m},\ k = 0, \ldots, m-1 \Big) \in \mathbb{R}^{m\times m}. \qquad (11.11)$$

Lemma 11.2 *Let $M_m \in \mathbb{R}^{m\times m}$ be as in eqn (11.5) and C_m, V_m, and D_m be as in eqns (11.9), (11.10), and (11.11), respectively. Then $M_m V_m = V_m D_m$ and $V_m^\top V_m = I_m$.*

Proof Let $k \in \{0, \ldots, m-1\}$ be fixed and let $v := C_m e_{k+1}$ be the $(k+1)^{\text{th}}$ column of C_m. From

$$M_m v = (v_1 + v_2, v_1 + v_3, \ldots, v_{m-2} + v_m, v_{m-1} + v_m)^\top,$$

and the addition theorem for trigonometric functions,

$$2 \cos\alpha \cos\beta = \cos(\alpha - \beta) + \cos(\alpha + \beta),$$

we have

$$
\begin{aligned}
v_1 + v_2 &= \cos\frac{k\pi}{2m} + \cos\frac{3k\pi}{2m} = 2\cos\frac{k\pi}{m}\cos\frac{k\pi}{2m} = \lambda_k v_1,\\
v_j + v_{j+2} &= \cos\left(\frac{(2j+1)k\pi}{2m} - \frac{k\pi}{m}\right) + \cos\left(\frac{(2j+1)k\pi}{2m} + \frac{k\pi}{m}\right)\\
&= 2\cos\frac{(2j+1)k\pi}{2m}\cos\frac{k\pi}{m} = \lambda_k v_{j+1}, \quad j = 1,\ldots,m-2,\\
v_{m-1} + v_m &= \cos\frac{(2m-3)k\pi}{2m} + \cos\frac{(2m-1)k\pi}{2m} = 2\cos\frac{(m-1)k\pi}{m}\cos\frac{k\pi}{2m}\\
&= 2\cos\frac{k\pi}{m}\cos(k\pi)\cos\frac{k\pi}{2m} = 2\cos\frac{k\pi}{m}\cos\frac{(2m-1)k\pi}{2m} = \lambda_k v_m,
\end{aligned}
$$

and thus $M_m C_m = C_m D_m$. Since $M_m = M_m^\top$, the columns of C_m are orthogonal. With $\omega := e^{ik\pi/m}$,

$$
\begin{aligned}
e_{k+1}^\top C_m^\top C_m e_{k+1} &= \sum_{j=0}^{m-1}\cos^2\frac{(2j-1)k\pi}{2m} = m + \sum_{j=0}^{m-1}\cos\frac{(2j-1)k\pi}{m}\\
&= m + \sum_{j=0}^{m-1}\Re[e^{ik\pi(2j-1)/m}] = m + \omega\sum_{j=0}^{m-1}\omega^{2j}\\
&= \begin{cases} m, & \omega = 1\\ 2m, & \omega \neq 1, \end{cases}
\end{aligned}
$$

where, for $k \neq 0$ and thus $\omega \neq 1$, we used $\sum_{j=0}^{m-1}\omega^{2j} = (\omega^{2m}-1)/(\omega-1) = 0$. Hence, V_m is orthonormal. □

The next theorem states that the matrix $A^{(d)}$ (cf., eqn (11.8)) can be diagonalized in terms of discrete cosine transformations (DCTs). It is this property which enables an $\mathcal{O}(N \log N)$ implementation based on a fast DCT.

Theorem 11.3 *Let S be a d-dimensional, symmetric matrix stencil and $A^{(d)}$ be the matrix representation of the convolution with S with respect to Neumann boundary conditions. Then*

$$
\begin{aligned}
D^{(d)} &:= (V_{n_d}\otimes\cdots\otimes V_{n_1})^\top\, A^{(d)}\, (V_{n_d}\otimes\cdots\otimes V_{n_1})\\
&= \operatorname{diag}(d_{j_1,\ldots,j_d},\ j_q = 1,\ldots,n_1,\ q = 1,\ldots,d),
\end{aligned}
$$

where V_m is defined by eqn (11.10) and $d_{j_1,\ldots,j_d}$ is defined recursively by

$$
\begin{aligned}
d_{j_1}^{p_{d-1},\ldots,p_1} &= S_{2,p_{d-1},\ldots,p_1} + 2S_{2,p_{d-1},\ldots,p_1}\cos(j_1\pi/n_1),\\
d_{j_1,\ldots,j_k}^{p_{d-k},\ldots,p_1} &= d_{j_1,\ldots,j_k}^{2,p_{d-k},\ldots,p_1} + 2d_{j_1,\ldots,j_k}^{1,p_{d-k},\ldots,p_1}\cos(j_k\pi/n_k),\\
d_{j_1,\ldots,j_d} &= d_{j_1,\ldots,j_{d-1}}^{2} + 2d_{j_1,\ldots,j_{d-1}}^{1}\cos(j_d\pi/n_d).
\end{aligned}
$$

Proof From Lemma 11.2, we have for $A_{p_{d-1},\ldots,p_1}^{(d,d-1)}$ (cf., eqn (11.6))

$$
\begin{aligned}
D_{p_{d-1},\ldots,p_1}^{(d,d-1)} &:= V_{n_1}^\top\, A_{p_{d-1},\ldots,p_1}^{(d,d-1)}\, V_{n_1}\\
&= \operatorname{diag}\left(d_{j_1}^{p_{d-1},\ldots,p_1},\ j_1 = 1,\ldots,n_1\right),
\end{aligned}
$$

where $d_{j_1}^{p_{d-1},\ldots,p_1} = S_{2,p_{d-1},\ldots,p_1} + 2S_{2,p_{d-1},\ldots,p_1}\cos((j_1-1)\pi/n_1)$.

Using the recurrence formula (11.7), we have

$$
\begin{aligned}
D_{p_{d-k},\ldots,p_1}^{(d,d-k)} &:= (V_{n_k}\otimes\cdots\otimes V_{n_1})^\top\, A_{p_{d-k},\ldots,p_1}^{(d,d-k)}\, (V_{n_k}\otimes\cdots\otimes V_{n_1})\\
&= \operatorname{diag}\left(d_{j_1,\ldots,j_k}^{p_{d-k},\ldots,p_1},\ j_q = 1,\ldots,n_q,\ q = 1,\ldots,k\right),
\end{aligned}
$$

where $d_{j_1,\ldots,j_k}^{p_{d-k},\ldots,p_1} = d_{j_1,\ldots,j_k}^{2,p_{d-k},\ldots,p_1} + 2d_{j_1,\ldots,j_k}^{1,p_{d-k},\ldots,p_1}\cos((j_k-1)\pi/n_k)$.

Finally, from eqn (11.6),

$$
\begin{aligned}
D^{(d)} &:= (V_{n_d}\otimes\cdots\otimes V_{n_1})^\top\, A^{(d)}\, (V_{n_d}\otimes\cdots\otimes V_{n_1})\\
&= \operatorname{diag}(d_{j_1,\ldots,j_d},\ j_q = 1,\ldots,n_q,\ q = 1,\ldots,d)
\end{aligned}
$$

where $d_{j_1,\ldots,j_d} = d_{j_1,\ldots,j_{d-1}}^{2} + 2d_{j_1,\ldots,j_{d-1}}^{1}\cos((j_d-1)\pi/n_d)$. □

The following corollaries summarize the statement of Theorem 11.3 for the important cases $d = 2$ and $d = 3$, respectively.

Corollary 11.4 *Let $A^{\mathrm{diff},2}$ be the matrix associated with $S^{\mathrm{diff},2}$; cf., Table 11.1. Then*

$$
\begin{aligned}
D^{\mathrm{diff},2} &:= (V_{n_2}\otimes V_{n_1})^\top A^{\mathrm{diff},2}(V_{n_2}\otimes V_{n_1})\\
&= \operatorname{diag}(d_{j_1,j_2},\ j_1 = 1,\ldots,n_1,\ j_2 = 1,\ldots,n_2),
\end{aligned}
$$

where
$d_{j_1,j_2} = -4 + 2\cos((j_1-1)\pi/n_1) + 2\cos((j_2-1)\pi/n_2)$,
$j_1 = 1,\ldots,n_1$, $j_2 = 1,\ldots,n_2$.

Proof Follows directly from Theorem 11.3. □

Corollary 11.5 *Let $A^{\text{diff},3}$ be the matrix associated with $S^{\text{diff},3}$; cf., Table 11.1. Then*

$$\begin{aligned} D^{\text{diff},3} &:= (V_{n_3} \otimes V_{n_2} \otimes V_{n_1})^\top A^{\text{diff},3} (V_{n_3} \otimes V_{n_2} \otimes V_{n_1}) \\ &= \operatorname{diag}(d_{j_1,j_2,j_3},\ j_\ell = 1,\dots,n_\ell,\ \ell = 1,2,3), \end{aligned}$$

where

$$\begin{aligned} d_{j_1,j_2,j_3} = &-6 + 2\cos((j_1-1)\pi/n_1) + 2\cos((j_2-1)\pi/n_2) \\ &+ 2\cos((j_3-1)\pi/n_3),\ j_\ell = 1,\dots,n_\ell,\ \ell = 1,2,3. \end{aligned}$$

Proof Follows directly from Theorem 11.3. □

As a consequence of the decomposition stated in Theorem 11.3 and Corollaries 11.4 and 11.5, a fast DCT can be exploited for a numerical solution of the linear system (11.4). Thus, an $\mathcal{O}(N \log N)$ implementation based on real arithmetic can be deduced. However, in Section 11.3 we present an $\mathcal{O}(N)$ implementation of a time marching approach based on an additive operator splitting scheme.

11.3 Additive operator splitting (AOS)

A popular approach to solve a non-linear PDE like eqn (11.2) is to introduce an artificial time t and to compute the steady state solution $\partial_t u(x,t) = 0$ of the time-dependent PDE

$$\partial_t u(x,t) = f(x,u(x,t)) + \alpha \Delta u(x,t), \quad x \in \Omega,\ t \geq 0, \tag{11.12}$$

via a time marching algorithm, see also eqn (8.9). To overcome the non-linearity in f, we employ the following semi-implicit iterative scheme,

$$\partial_t u(x,t_{k+1}) - \alpha \Delta u(x,t_{k+1}) = f(x,u(x,t_k)), \quad k = 0,1,\dots, \tag{11.13}$$

where $u(x,t_0)$ is some initial deformation, typically $u(x,t_0) = 0$. In other words, the trick is to compute the driving force f for the previous solution $u(x,t_k)$ and subsequently to solve for $u(x,t_{k+1})$.

An important property of the system of equations (11.13) is that they are essentially decoupled. The coupling is only through the right hand side. The j^{th} equation reads

$$\partial_t u_j(x,t_{k+1}) - \alpha \Delta u_j(x,t_{k+1}) = f_j(x,u(x,t_k)), \quad k = 0,1,\dots. \tag{11.14}$$

Note that eqn (11.14) is nothing but an inhomogeneous heat equation and well understood; see, e.g., Folland (1995, Th 4.8).

There exist many schemes to solve (11.14) numerically. Here, we are interested in schemes which are on the one hand accurate and stable and on the other hand fast and efficient.

For the time discretization we introduce a time-step $\tau > 0$ and for the spatial discretization the grid $\vec{X}$; cf., Definition 3.2. For a fixed j, we set

$$\vec{V}^{(k)} := u_j(\vec{X}, k\tau) \tag{11.15}$$

and

$$\vec{F}_j^{(k)} := f_j(\vec{X}, u(\vec{X}, \tau k)). \tag{11.16}$$

Using these abbreviations the discrete version of eqn (11.14) reads

$$\frac{\vec{V}^{(k+1)} - \vec{V}^{(k)}}{\tau} - \sum_{\ell=1}^{d} A_\ell \vec{V}^{(k+1)} = \vec{F}_j^{(k)}, \tag{11.17}$$

where

$$\frac{\vec{V}^{(k)} - \vec{V}^{(k-1)}}{\tau} \approx \partial_t u_j(\vec{X}, k\tau)$$

is a forward difference approximation of the time derivative $\partial_t u_j$ with time-step τ and $A_\ell \in \mathbb{R}^{n \times n}$ is an appropriate finite difference approximation of the second order derivative of u_j with respect to the ℓ^{th} space coordinate,

$$A_\ell \vec{V}^k \approx \alpha\ \partial_{x_\ell, x_\ell} u_j(\vec{X}, \tau k). \tag{11.18}$$

In the case of image registration, we have chosen a simple three-point star leading to an essentially tridiagonal matrix A_ℓ; cf., Table 11.1. However, in the light of Theorem 11.6, there is no particular assumption on A_ℓ.

After rearranging eqn (11.17), we obtain the semi-implicit scheme for (11.14)

$$\vec{V}_{\text{IS}}^{(k+1)} := \left(I - \tau \sum_{\ell=1}^{d} A_\ell\right)^{-1} \left(\vec{V}_{\text{IS}}^{(k)} + \tau \vec{F}_j^{(k)}\right), \quad k = 0,1,\dots \tag{11.19}$$

It is known that this scheme is of order one with respect to the time-step τ, and of order two with respect to the spatial meshsize.

The iteration (11.19) requires the solution of a linear system with n unknowns at each time-step. Note that the systems connected to the individual matrices A_ℓ are essentially tridiagonal and may be solved by an $\mathcal{O}(n)$ direct scheme. On the other hand, the sum is not tridiagonal and therefore the system in (11.19) does not permit such a fast implementation, in general.

The idea of AOS is to replace the inverse of the sum by a sum of inverses. The corresponding iterates are defined by

$$\vec{V}_{\text{AOS}}^{(k+1)} := \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau A_\ell)^{-1}\left(\vec{V}_{\text{AOS}}^{(k)} + \tau\vec{F}_j^{(k)}\right), \quad k = 0, 1, \dots \tag{11.20}$$

This clever decomposition allows an $\mathcal{O}(n)$ implementation by employing the Thomas algorithm (1949).

11.3.1 *Matrix analysis of AOS*

The next theorem relates the iteration matrices of the semi-implicit and the AOS scheme to each other. It turns out that the distance between these two matrices is surprisingly small if the time-step has been chosen sufficiently small. Note that for image registration, we are not interested in time accuracy and τ may not be small. However, our result is based on matrix analysis, it is not restricted to matrices stemming from PDE discretization, and is therefore of interest independently of these facts. The following theorem is formulated for general matrices. Matrix distances are measured in the spectral norm: for $A \in \mathbb{C}^{n\times n}$,

$$\|A\|_2 := \max\{\sqrt{\lambda} : \ \lambda \text{ is eigenvalue of } A^{\mathrm{H}}A\}.$$

Theorem 11.6 *Let $d \in \mathbb{N}$, $\tau \geq 0$, and let $A_1, \dots, A_d \in \mathbb{R}^{n\times n}$ be simultaneously diagonalizable with eigenvalues in the left half plane. Then there exists a constant $C \in \mathbb{R}$ with*

$$\left\| \left(I - \tau\sum_{\ell=1}^{d} A_\ell\right)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau A_\ell)^{-1} \right\|_2 \leq C \cdot \tau^2.$$

Proof The idea is to employ a basis of eigenvectors such that the corresponding matrices become diagonal. To this end consider

$$WA_\ell W^{-1} = \Lambda_\ell = \operatorname{diag}(\lambda_{\ell,j},\ 1 \leq j \leq n),$$

where W is an eigenvector matrix of any A_ℓ and the Λ_ℓ's are the diagonal matrices based on the individual eigenvalues. Hence,

$$W(I - d\tau A_\ell)W^{-1} = I - d\tau\Lambda_\ell,$$

$$W\left(I - \tau\sum_{\ell=1}^{d} A_\ell\right)W^{-1} = \left(I - \tau\sum_{\ell=1}^{d}\Lambda_\ell\right),$$

and

$$W\left[\left(I - \tau\sum_{\ell=1}^{d} A_\ell\right)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau A_\ell)^{-1}\right]W^{-1}$$
$$= \left(I - \tau\sum_{\ell=1}^{d}\Lambda_\ell\right)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau\Lambda_\ell)^{-1}$$

is a diagonal matrix, where the k^{th} diagonal entry is given by

$$q_k = \varphi\left(\tau\sum_{\ell=1}^{d}\lambda_{\ell,k}\right) - \frac{1}{d}\sum_{\ell=1}^{d}\varphi(d\tau\lambda_{\ell,k}), \quad \varphi(x) = \frac{1}{1-x}.$$

A Taylor expansion of the analytic function φ at $x_0 = 0$ reads

$$\varphi(x) = 1 + x + \frac{2x^2}{(1-\xi)^3}, \quad \xi = \xi(x) \in [0, x].$$

This yields

$$\begin{aligned} q_k &= 1 + \tau\sum_{\ell=1}^{d}\lambda_{\ell,k} + \frac{2\tau^2(\sum_{\ell=1}^{d}\lambda_{\ell,k})^2}{(1-\xi)^3} \\ &\quad - \frac{1}{d}\sum_{\ell=1}^{d}\left(1 + d\tau\lambda_{\ell,k} + \frac{2(d\tau\lambda_{\ell,k})^2}{(1-\xi_\ell)^3}\right) \\ &= \tau^2\left(\frac{2(\sum_{\ell=1}^{d}\lamba_{\ell,k})^2}{(1-\xi)^3} - \frac{1}{d}\sum_{\ell=1}^{d}\frac{2(d\lambda_{\ell,k})^2}{(1-\xi_\ell)^3}\right) \\ &=: \tau^2 g(\lambda_{1,k}, \dots, \lambda_{d,k}). \end{aligned}$$

By assumption we can find compact sets Q_ℓ contained in the left complex half plane which enclose all eigenvalues of A_ℓ. Consequently, the function g is continuous on $Q := Q_1 \times \cdots \times Q_d$ and attains its maximum

$$\tilde{C} := \max\{|g(z)| \ : \ z \in Q\}.$$

We thus have $|q_k| \leq \tilde{C}\tau^2$, for $k = 1, \dots, n$, and the statement follows from

$$\left\| \left(I - \tau\sum_{\ell=1}^{d} A_\ell\right)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau A_\ell)^{-1} \right\|_2$$
$$\leq \|W\|_2\|W^{-1}\|_2 \cdot \left\| \left(I - \tau\sum_{\ell=1}^{d}\Lambda_\ell\right)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau\Lambda_\ell)^{-1} \right\|_2 \leq C \cdot \tau^2.$$

It is worth noticing that matrices are simultaneously diagonalizable if and only if they commute with each other, a proof of which can be found for example in Horn & Johnson (1990, Th. 1.3.19). It is this property which provides a convenient tool for checking the assumption of the above theorem.

In the statement of the theorem it is assumed that all eigenvalues are contained in the left half plane. This assumption ensures that, independent of the value of τ, the matrices

$$I - \tau \sum_{\ell=1}^{d} A_\ell \quad \text{and} \quad I - d\tau A_\ell$$

are non-singular. A close inspection of the proof shows that the theorem holds for arbitrary eigenvalues as well, as long as τ is small enough.

11.3.2 *Diffusion registration using AOS*

Now let us return to the PDE discretization (11.19) and (11.20), respectively. The next corollary relates the solutions of the two time marching processes to each other. In accordance with Theorem 11.6, the statement of the corollary is valid for general matrices.

Corollary 11.7 *Let $d, K \in \mathbb{N}$, $\tau \geq 0$, and let $A_1, \ldots, A_d \in \mathbb{R}^{n\times n}$ be simultaneously diagonalizable with eigenvalues in the left half plane. Moreover, let $\vec{V}_{\text{IS}}^{(k+1)}$ and $\vec{V}_{\text{AOS}}^{(k+1)}$ denote the solution of eqns (11.19) and (11.20), respectively. Then there exists a constant $C > 0$ with*

$$\left\| \vec{V}_{\text{IS}}^{(k+1)} - \vec{V}_{\text{AOS}}^{(k+1)} \right\|_2 \leq C \cdot \tau^2, \quad 0 \leq k \leq K.$$

The particular matrices introduced by (11.18) are given by

$$A_\ell = I \otimes \cdots \otimes I \otimes B_\ell \otimes I \otimes \cdots \otimes I,$$

where the ℓ^{th} factor B_ℓ is an approximation of the second order derivative in only one spatial direction and $\otimes$ denotes the Kronecker product of matrices. More precisely, we have

$$B_\ell = \begin{pmatrix} \alpha_{\ell,1} & \beta_{\ell,2} & & 0 \\ \gamma_{\ell,2} & \alpha_{\ell,2} & \ddots & \\ & \ddots & \ddots & \beta_{\ell,m} \\ 0 & & \gamma_{\ell,m} & \alpha_{\ell,m} \end{pmatrix},$$

with appropriate values of $\alpha_{\ell,j}$, $\beta_{\ell,j}$, and $\gamma_{\ell,j}$ satisfying $-\alpha_{\ell,j} \geq |\beta_{\ell,j+1}| + |\gamma_{\ell,j}|$. Using the particular stencils of Table 11.1, the entries become $\alpha_j = -2$, $\beta_j = \gamma_j = 1$.

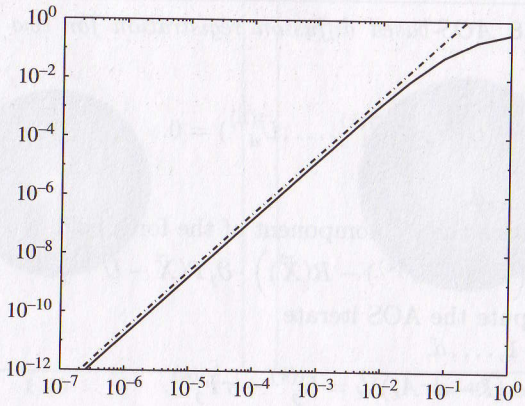

FIG. 11.1 $\|(I - \tau\sum_{\ell=1}^{d} A_\ell)^{-1} - \frac{1}{d}\sum_{\ell=1}^{d}(I - d\tau A_\ell)^{-1}\|_2$ (solid line) and $C \cdot \tau^2$ (dash–dotted line) versus τ, where $C = 25$ for the matrices arising in diffusion registration.

Obviously, the matrices A_ℓ commute and since the B_ℓ have negative eigenvalues, Theorem 11.6 and Corollary 11.7 apply. In other words, the iterates of the semi-implicit scheme and the AOS scheme differ by $\mathcal{O}(\tau^2)$.

It is apparent from the proof of Theorem 11.6 that the constant C depends on the eigenvalues and eigenvectors of the matrices A_ℓ. For the particular matrices arising in diffusion registration, it turns out that C is of moderate size and essentially independent of the number of grid points. To illustrate this fact we have computed the quantity

$$\left\| \left(I - \tau \sum_{\ell=1}^{d} A_\ell \right)^{-1} - \frac{1}{d} \sum_{\ell=1}^{d} (I - d\tau A_\ell)^{-1} \right\|_2$$

for various sizes of A_ℓ and various τ. Figure 11.1 shows a representative result for $d = 2$, $n_1 = n_2 = 1024$, where $C = 25$ serves as an upper bound. The plots for other sizes of A_ℓ are visually indistinguishable from the displayed one.

The overall algorithm for the AOS scheme is summarized in Algorithm 11.8. Note that the main steps in the algorithm are the computation of the force $\vec{F}^{(k)}$ related to the chosen distance measure and the solution of the linear systems related to our particular smoother. We use a standard $\mathcal{O}(n)$ d-linear interpolation technique for computation of the force. As already mentioned, the matrices $I - d\tau A_\ell$ are tridiagonal and strictly diagonally dominant. Hence, the $\mathcal{O}(n)$ Thomas algorithm (1949) is a numerically stable solution technique. In conclusion, we end up with a fast and efficient $\mathcal{O}(n)$ registration algorithm.

Moreover, the implementation offers a coarse grain parallelism based on the ℓ-loop in the algorithm. Due to the special Kronecker product structure of the matrices A_ℓ, a fine grain parallelism can be exploited, too. For example, in two

Algorithm 11.8 *AOS-based diffusion registration for two d-dimensional images R and T.*

Initialize $k = 0$, $\vec{U}^{(k)} = (\vec{U}_1^{(k)}, \ldots, \vec{U}_d^{(k)}) = 0$.
For $k = 0, 1, 2, \ldots$
  For $j = 1, \ldots, d$,
    % Compute the j^{th} component of the force field
    $\vec{F}_j^{(k)} = \left(T(\vec{X} - \vec{U}^{(k)}) - R(\vec{X})\right) \cdot \partial_j T(\vec{X} - \vec{U}^{(k)})$.
    % Compute the AOS iterate
    For $\ell = 1, \ldots, d$,
      Solve $(I - d\tau A_\ell)\vec{V}_\ell = \vec{U}_j^{(k)} + \tau \vec{F}_j^{(k)}$.
    End.
    Set $\vec{U}_j^{(k)} = \frac{1}{d} \sum_{\ell=1}^{d} \vec{V}_\ell$.
  End.
End.

dimensions we have to solve the two linear systems

$$((I - 2\tau B_1) \otimes I)\vec{V}_1 = \vec{U}_j^{(k)} + \tau \vec{F}_j^{(k)},$$

$$(I \otimes (I - 2\tau B_2))\vec{V}_2 = \vec{U}_j^{(k)} + \tau \vec{F}_j^{(k)}.$$

Each of these systems decouples into a number of small systems which can be solved independently in parallel.

11.4 Diffusion registration: • to C

We continue the registration of a • to a C; see Section 9.8 and Section 10.5.

In Figs 11.2 and 11.3 we show intermediate results of diffusion registration with $\alpha = 50$ and $\tau = 0.02$ using the AOS- and the DCT-based solvers, see Section 11.2, respectively. Finally, Fig. 11.4 directly compares the registration results with respect to the numerical convergence and the grids. These figures indicate that the registrations based on the AOS and the semi-implicit schemes are indistinguishable – at least for this example.

Note that it is not self-evident that the two methods are so close together. Particularly for large values of τ, the underlying models are different and thus one may expect different qualitative results.

Diffusion registration might be compared with elastic registration; see Chapter 9. In both techniques the registration is smoothed by regularizing the

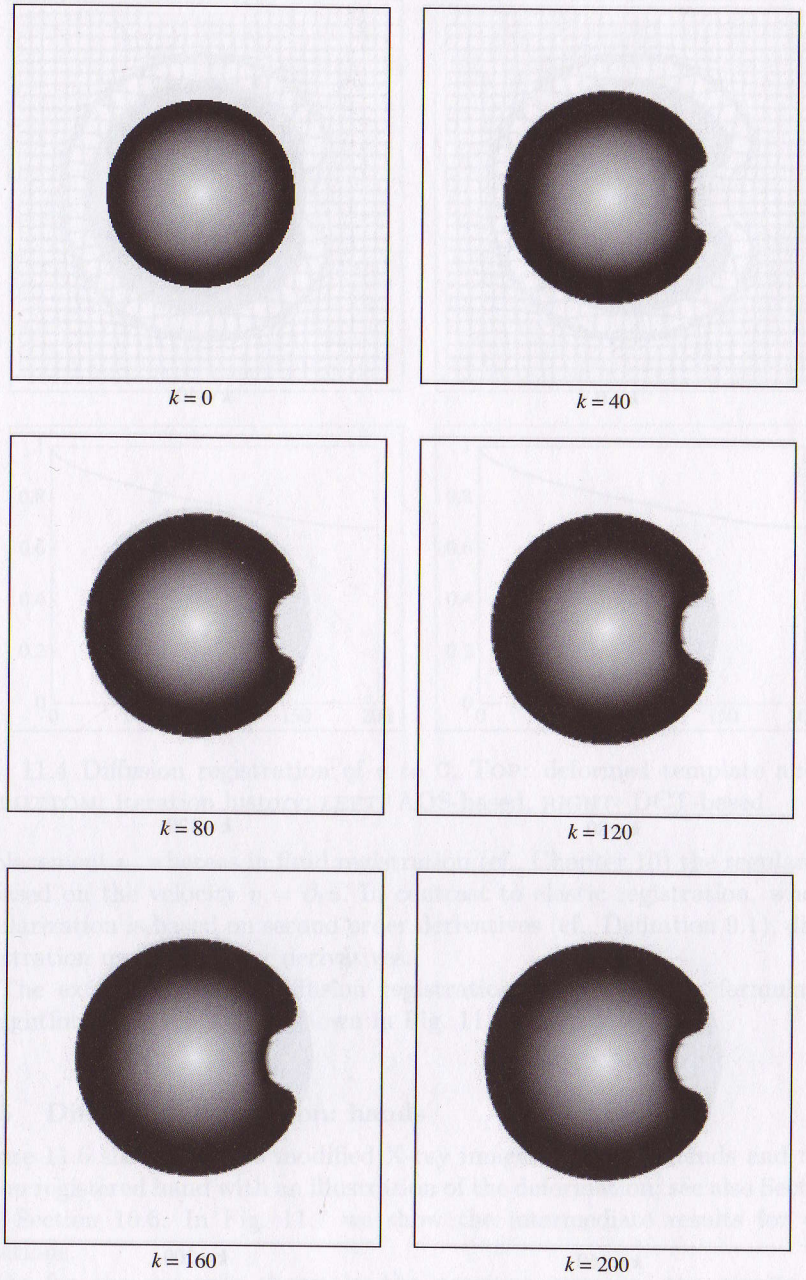

FIG. 11.2 Diffusion registration of • to C using AOS, $\alpha = 50$ and $\tau = 0.02$; intermediate results for $k = 0(40)200$.

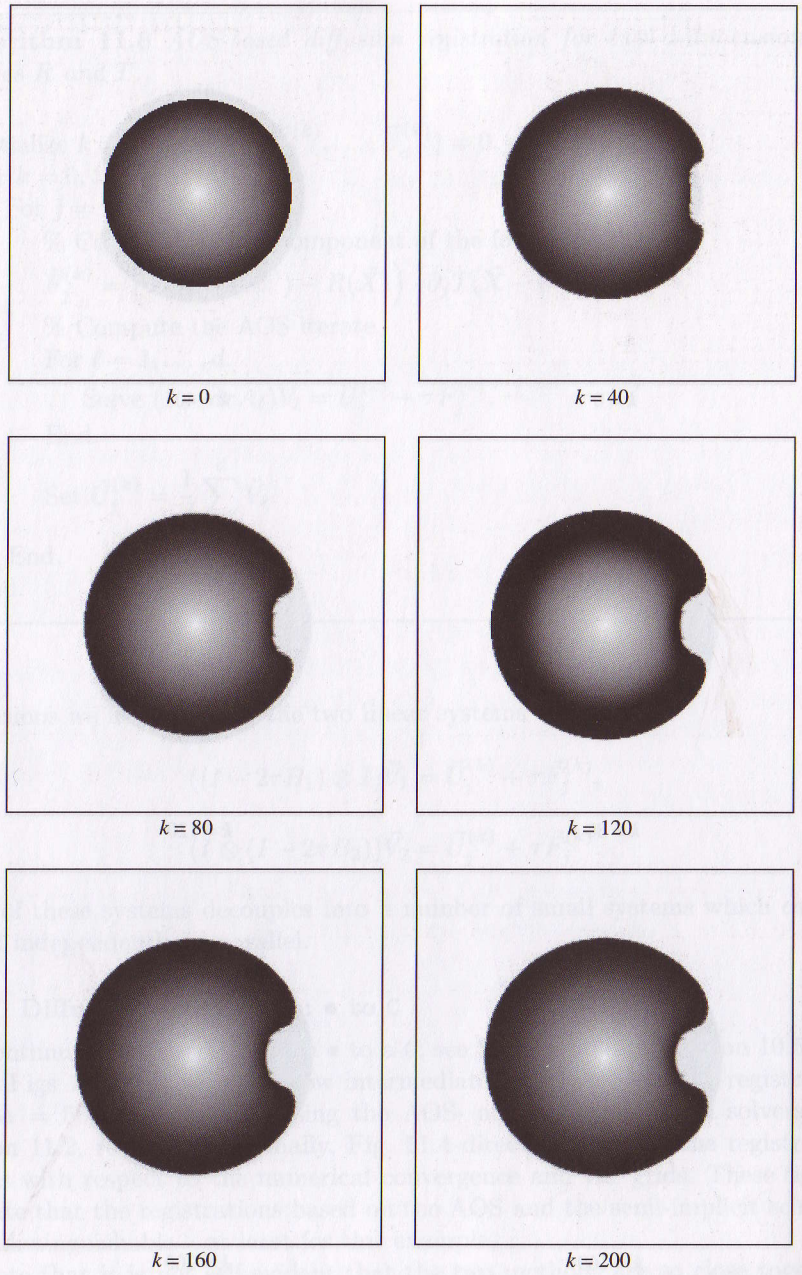

FIG. 11.3 Diffusion registration of • to C using DCT, $\alpha = 50$ and $\tau = 0.02$; intermediate results for $k = 0(40)200$.

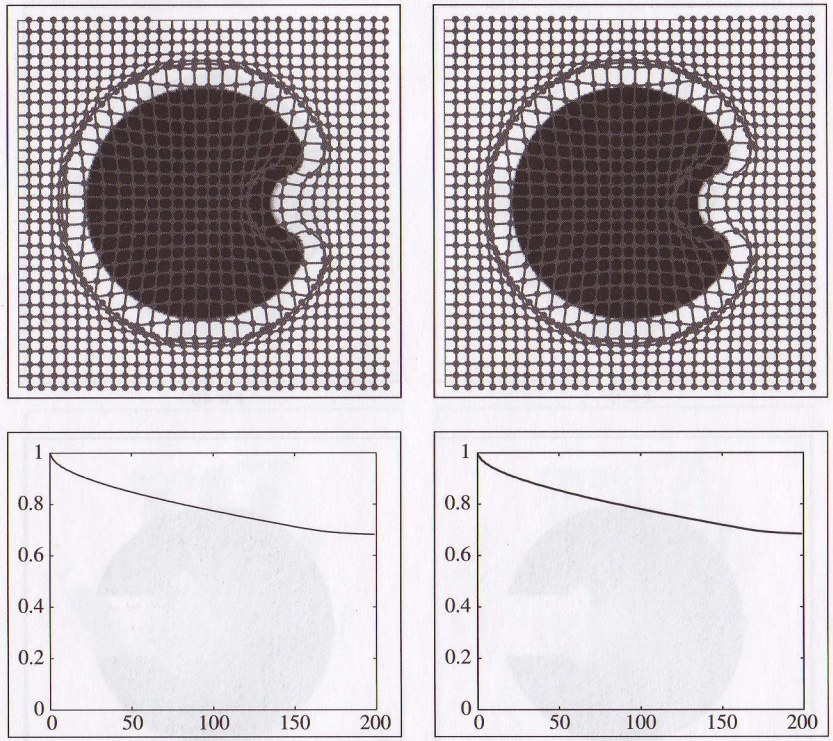

FIG. 11.4 Diffusion registration of • to C. TOP: deformed template and grid; BOTTOM: iteration history; LEFT: AOS-based, RIGHT: DCT-based.

displacement u, whereas in fluid registration (cf., Chapter 10) the regularization is based on the velocity $v = \partial_t u$. In contrast to elastic registration, where the regularization is based on second order derivatives (cf., Definition 9.1), diffusion registration uses first order derivatives.

The extension of our diffusion registration to a *fluid*-type formulation is straightforward. Results are shown in Fig. 11.5.

11.5 Diffusion registration: hands

Figure 11.6 shows the two modified X-ray images of human hands and the diffusion registered hand with an illustration of the deformation; see also Section 9.9 and Section 10.6. In Fig. 11.7 we show the intermediate results for various iterations.

As for the example shown in the previous sections, the results of the AOS-based (Figs 11.6 and 11.7) and the ones from the DCT-based diffusion registration (Figs 11.8 and 11.9) are almost indistinguishable.

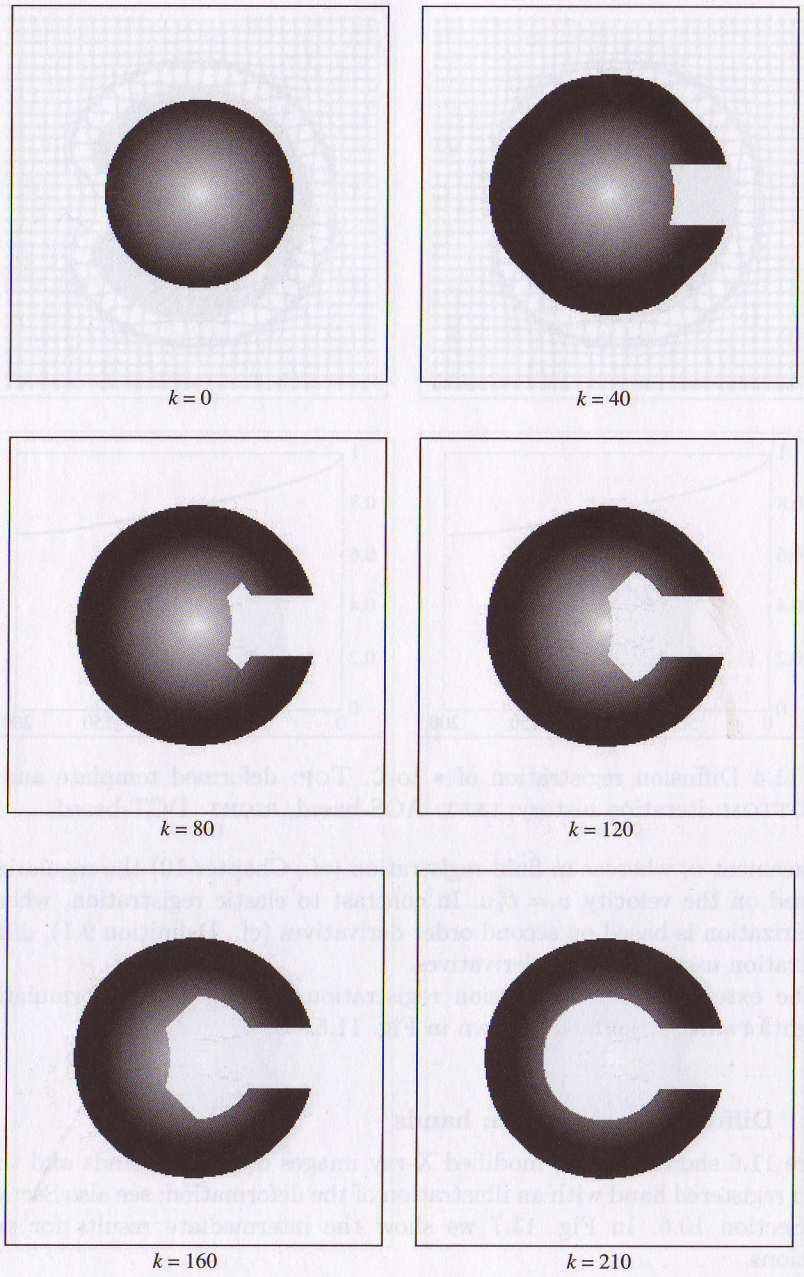

FIG. 11.5 Velocity-based diffusion registration of • to C using the AOS scheme, $\alpha = 1$ and $\tau = 0.2$; intermediate results for $k = 0(40)160, 210$.

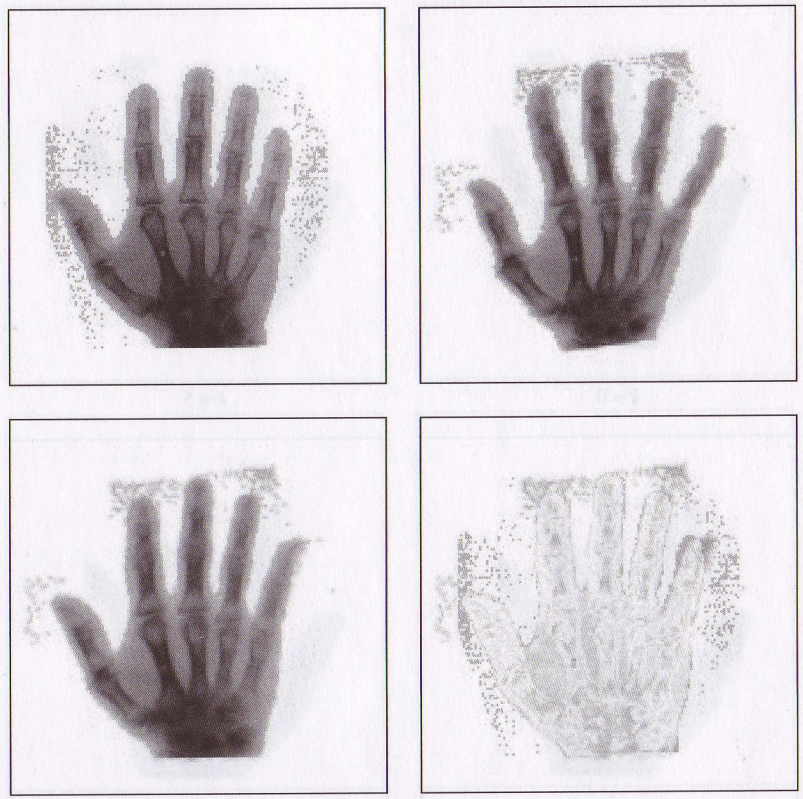

FIG. 11.6 Diffusion registration of hands using AOS, $\alpha = 5$, $\tau = 0.25$; TOP LEFT: reference image R, TOP RIGHT: template image T, BOTTOM LEFT: diffusion registered template T^{diff} with deformation, and BOTTOM RIGHT: difference $|R - T^{\text{diff}}|$.

11.6 Thirion's demons registration

In the landmark paper (1995), Thirion presents a new method to perform the non-rigid matching of two three-dimensional medical images. He describes his new method as based on so-called *demons*, a notation adapted from Maxwell's demons in thermodynamics.

The basic idea is to position demons at certain places in the image domain. These demons should then be able to decide whether or not a movement of one particle of the template image in a certain direction reduces the disparity between the reference and transformed images.

This intuitive idea is illustrated in Fig. 11.10. Roughly speaking, the task of the demons is to sort the particles. To do so they move the particles in- or outward depending on the relation between the scene and model. In order to

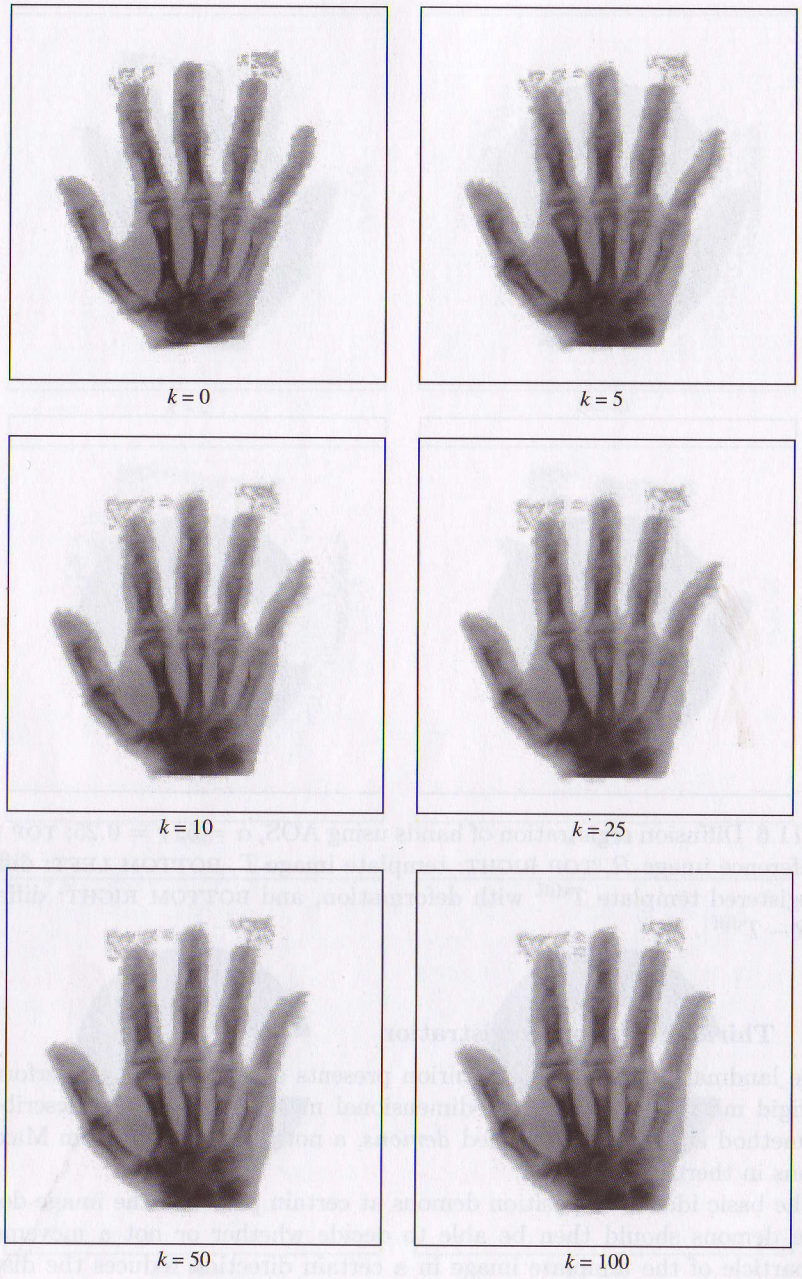

FIG. 11.7 Diffusion registration of hands using AOS ($\alpha = 5$, $\tau = 0.25$), intermediate results for $k = 0, 5, 10, 25, 50, 100$.

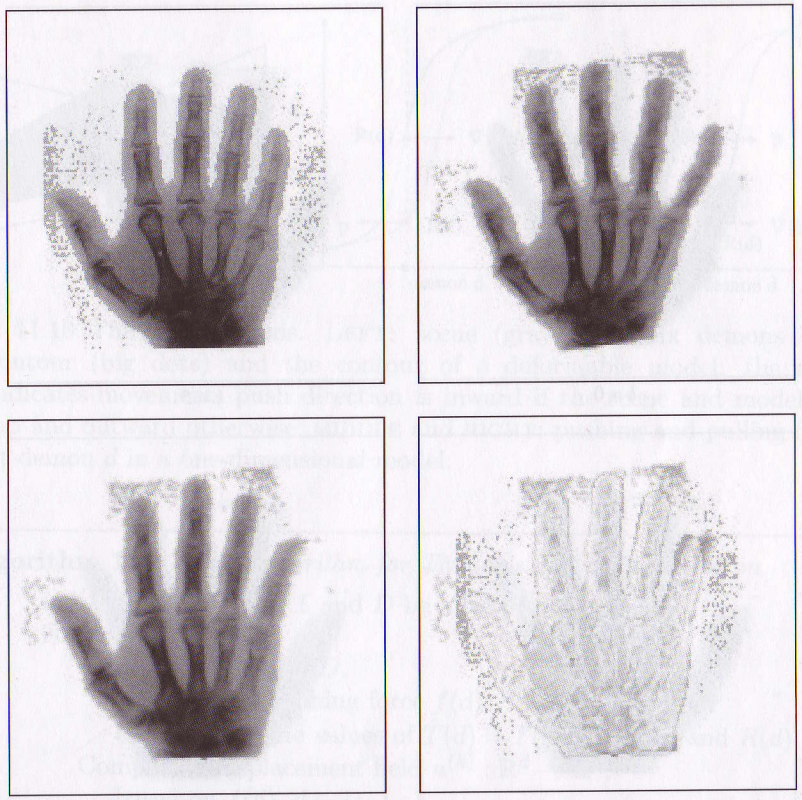

FIG. 11.8 Diffusion registration of hands using DCT ($\alpha = 5$, $\tau = 0.25$); TOP LEFT: reference image R, TOP RIGHT: template image T, BOTTOM LEFT: diffusion registered template T^{diff} with deformation, and BOTTOM RIGHT: difference $|R - T^{\text{diff}}|$.

illustrate this idea, two one-dimensional images R and T are considered, see Fig. 11.10. A demon d is located at a spatial position d, where $\nabla R(d) \neq 0$. Depending on the gradient $\nabla R(d)$ and the image difference $R(d) - T(d)$, the demons induce a pushing force p. The demon pushes the template according to $\nabla R(d)$, if $R(d) < T(d)$ (Fig. 11.10 middle), and according to $-\nabla R(d)$, if $R(d) > T(d)$ (Fig. 11.10 right).

A model algorithm is summarized in Algorithm 11.9, see also Thirion (1995). Thirion mentions many types of transformations. For the purpose of our considerations, we restrict ourselves to what Thirion called the *free-form case*. Following Thirion (1995), three questions remain open.

1. **Class of deformation**
   Demons are placed at each spatial position. The task then is to compute an appropriate displacement field from the typically rough force field. Here,

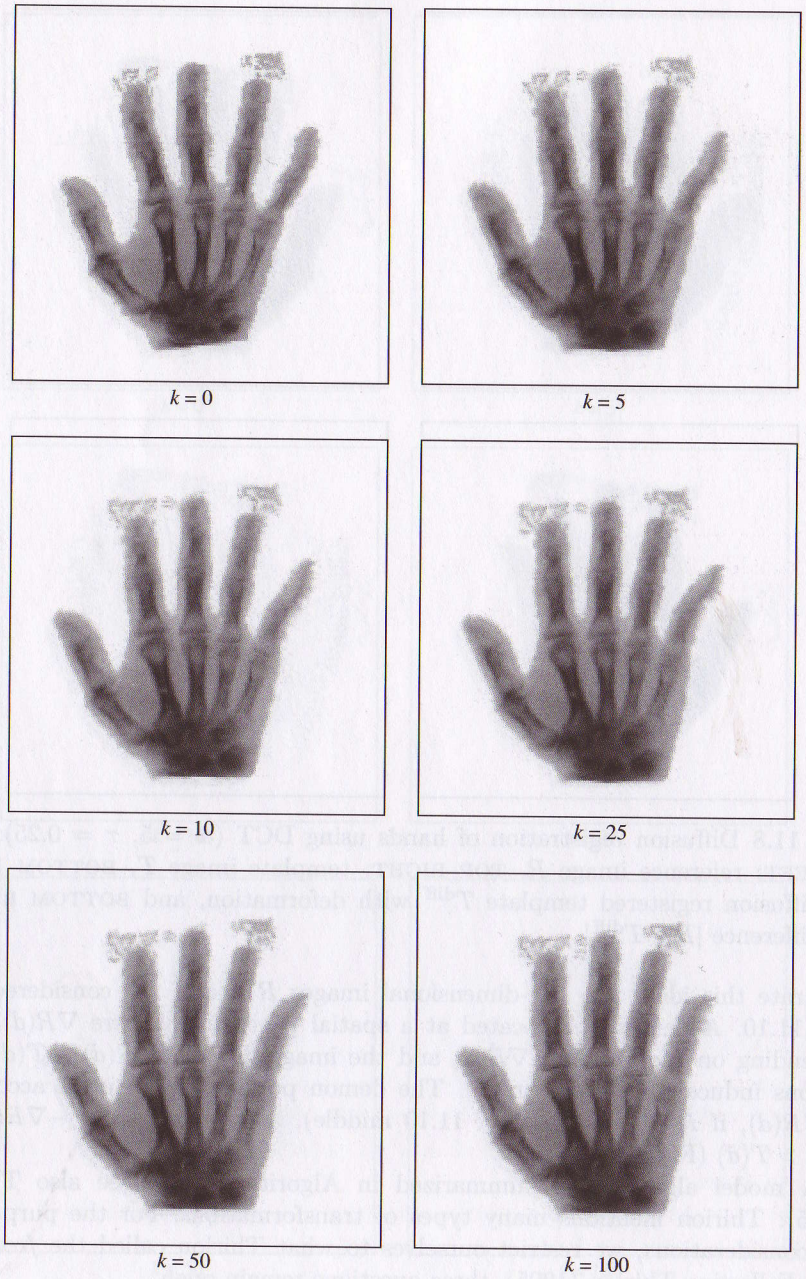

FIG. 11.9 Diffusion registration of hands with DCT ($\alpha = 5$, $\tau = 0.25$), intermediate results for $k = 0, 5, 10, 25, 50, 100$.

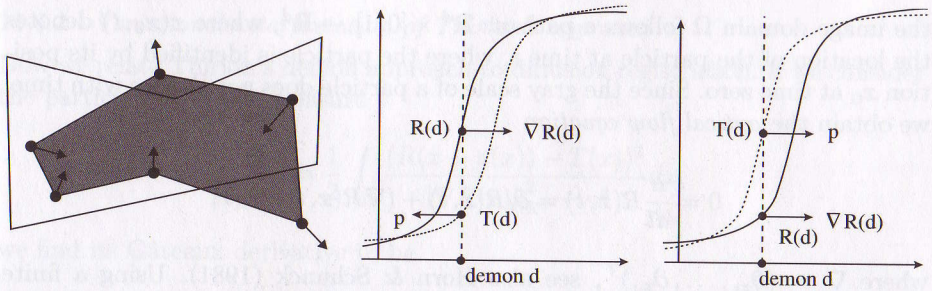

FIG. 11.10 Thirion's demons. LEFT: scene (gray) with six demons on its contour (big dots) and the contour of a deformable model; the arrows indicates movement, push direction is inward if the scene and model overlap and outward otherwise; MIDDLE and RIGHT: pushing and pulling force p at demon d in a one-dimensional model.

Algorithm 11.9 *Model algorithm for Thirion's demons registration.*

Let $k = 0$ and $\varphi^{(k)} = \mathcal{I}$ and D be a set of demons.
For $k = 0, 1, 2, \ldots$
  For each demon $d \in D$,
    compute the pushing force $f(d)$,
    according to the values of $\hat{T}(d) = T([\varphi^{(k)}]^{-1}(d))$ and $R(d)$.
  Compute a displacement field $u^{(k)} : \mathbb{R}^d \to \mathbb{R}^d$,
    based on $f(d)$, $d \in D$.
  Concatenate the transformations $\varphi^{(k)}$ and $u^{(k)}$,
    $\varphi^{(k+1)} = u^{(k)} \circ \varphi^{(k)}$.
End.

Thirion suggests a low-pass filter and in particular uses Gaussian filtering with a fixed and given covariance matrix σI_d.

2. **Interpolation scheme**
   An interpolation scheme has to be used. Here, we used a d-linear interpolation scheme; cf., Section 3.1.3.
3. **Type of demons**
   The demons type to be used is discussed in the next section.

11.6.1 *Demons type*

For simplicity we assume that the template T is a deformed version of the reference R, i.e., we assume a deformation process generating images $R(\cdot, t)$ for a time $t \in [0, 1]$, such that $R(x) = R(x, 0)$ and $T(x) = R(x, 1)$. Thus any particle P in

the image domain Ω follows a path $x : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, where $x(x_0, t)$ denotes the location of the particle at time t, where the particle is identified by its position x_0 at time zero. Since the gray scale of a particle does not change with time, we obtain the *optical flow equation*

$$0 = \frac{d}{dt} R(x,t) = \partial_t R(x,t) + (\nabla R(x,t))^\top \partial_t x,$$

where $\nabla = (\partial_{x_1}, \ldots, \partial_{x_d})^\top$, see also Horn & Schunck (1981). Using a finite difference approximation for the time derivative and setting $v := \partial_t x$, we obtain in particular

$$(\nabla R(x,t))^\top v \approx \frac{R(x, t+\tau) - R(x,t)}{\tau} \tag{11.21}$$

or

$$(\nabla R(x,t))^\top v \approx \frac{R(x,t) - R(x, t-\tau)}{\tau}, \tag{11.22}$$

where a forward and a backward finite difference approximation have been used. In particular for $\tau = 1$ we obtain

$$(\nabla R(x))^\top v = T(x) - R(x) \qquad (t = 0 \text{ in eqn (11.21)}) \tag{11.23}$$

and

$$(\nabla T(x))^\top v = T(x) - R(x) \qquad (t = 1 \text{ in eqn (11.22)}). \tag{11.24}$$

As in Thirion (1998) we focus on eqn (11.23) for the moment. Assuming that $\nabla R(x) \neq 0$, the general solution v^{general} is given by

$$v^{\text{general}} = \frac{T(x) - R(x)}{(\nabla R(x))^\top \nabla R(x)} \nabla R(x) + w,$$

where $w \in \text{span}(\nabla R(x))^\perp$.

Thirion suggests selecting v^{shortest}, the general solution with smallest norm, i.e., $w = 0$. Note that the computation of v becomes delicate whenever $\|\nabla R(x)\|_{\mathbb{R}^d}$ is close to zero. Small perturbations may lead to large errors. To avoid this phenomenon, an additional regularization parameter $\kappa \neq 0$ is introduced, such that

$$v_\kappa^{\text{shortest}} := \frac{T(x) - R(x)}{(\nabla R(x))^\top \nabla R(x) + \kappa^2} \nabla R(x). \tag{11.25}$$

In order to reduce the number of parameters, Thirion suggests taking $\kappa = T(x) - R(x)$, ignoring the fact that $T(x) = R(x)$ occasionally. Note that, if $T(x) = R(x)$, the smallest norm solution is given by $v = 0$. However, the smallest norm solution is not well-defined by formula (11.25).

11.6.2 *Variational interpretation of Thirion's approach*

Now we relate Thirion's demon approach to diffusion registration. If we consider the particular distance measure

$$\mathcal{D}[u] = \frac{1}{2} \int_\Omega \frac{(R(x + u(x)) - T(x))^2}{\|\nabla R(x)\|^2_{\mathbb{R}^d} + \kappa^2} dx,$$

we find its Gâteaux derivative to be

$$\begin{aligned} d\mathcal{D}[u; v] &= \lim_{h \to 0} \frac{1}{2} \int_\Omega \frac{(R(x + u + hv) - T(x))^2 - (R(x+u) - T(x))^2}{\|\nabla R(x)\|^2_{\mathbb{R}^d} + \kappa^2} dx \\ &= \int_\Omega \frac{(R(x+u) - T(x))}{\|\nabla R(x)\|^2_{\mathbb{R}^d} + \kappa^2} \nabla R(x)^\top v \; dx. \end{aligned}$$

Thus, Thirion's velocity $v_\kappa^{\text{shortest}}$ might be viewed as a force f in the variational setting.

If in addition the smoother $\mathcal{S}^{\text{diff}}$ introduced in eqn (11.1) is used to privilege smooth displacements, the Euler–Lagrange equations are given by Theorem 11.1.

A numerical scheme may be based on the semi-implicit scheme (8.8). If we had to solve (8.8) with respect to the whole space, i.e., $\Omega = \mathbb{R}^d$, then, under mild conditions on the driving force $f^{(k)}(x) := f(x, u(x, t_k))$, it would be possible to come up with an analytic solution. A representative result in this direction reads (see Folland (1995, Th. 4.8)): if $f^{(k)} \in L_1$, then the convolution

$$\begin{aligned} u^{(k+1)}(x,t) &= K_t(x) * f^{(k)}(x) \\ &= \int_{-\infty}^t \int_{\mathbb{R}^d} K_{t-s}(x-y) f^{(k)}(y) dy \; ds, \quad t > 0, \end{aligned}$$

is well-defined almost everywhere and is a distributional solution of (8.8). It will even be a classical solution if $f^{(k)} \in C^p$ for $p > 1$. Here

$$K_t(x) = (4\pi t)^{-d/2} \exp\left(- \|x\|^2_{\mathbb{R}^d} / (4t) \right)$$

denotes the Gauss kernel. In order to solve (8.8) with respect to the discretized bounded region $\Omega =]0,1[^d$, one may approximate the Gauss kernel by a Gauss filter of suitable length; that is, to compute at each time-step the force convolved with a Gauss filter K_σ of characteristic width σ. As is well-known, the Gauss filter-based scheme is less accurate than the outlined finite difference scheme.

This Gauss filter-based approach is essentially what Thirion calls "Demons 1: a complete grid of demons"; see Thirion (1998). However, he gives no hint on how to choose the parameter σ for a given application. It turns out in practice that a proper choice of this free parameter is a delicate matter. On the other hand, independent of the choice of σ, Thirion's approach is also of linear complexity.

11.6.3 *Steepest descent interpretation*

In this section we give another straightforward interpretation of Thirion's approach. The overall goal is to minimize the distance measure $\mathcal{D}[R, T; u]$; cf., eqn (8.2). Starting with a smooth displacement u one might use a steepest descent method $\partial_t u = -f(\cdot, u)$, see Theorem 8.1, or in a time discrete setting,

$$u^{(k+1)} = u^{(k)} + \tau f(\cdot, u^{(k)}).$$

The problem is that f is in general not smooth, such that the update and hence $u^{(k+1)}$ might be non-smooth. The trick is to project the update onto a smooth space, e.g., by convolving with a Gauss kernel. This gives

$$u^{(k+1)} = K_\sigma * (u^{(k)} + \tau f(\cdot, u^{(k)})).$$

11.7 Discussion of diffusion registration

The main drawback of diffusion registration is that although it is obvious to measure smoothness by oscillations of the gradients, it is not physical. Even though each component u_ℓ of the displacement can be viewed as a solution of a particular heat equation, a physical interpretation for the vector field u is missing. In our experience, however, the non-physical behavior of the method can hardly be detected in "real-life" applications.

In the registration process, the spatial directions are coupled only through the forces. It is this property that can be viewed as one major advantage of diffusion registration. The spatial decoupling allows for a block diagonalization. In addition, for each block the AOS scheme (cf., Section 11.3) presents a fast and stable solution technique of linear complexity. This makes diffusion registration a very attractive registration scheme, in particular for high-dimensional image data.

12

CURVATURE REGISTRATION

Elastic, fluid, and diffusion registration are sensitive with respect to affine linear displacements; see, e.g., Section 10.6. In particular, for dimension $d = 2$ and

$$u(x) = Cx + b, \quad \text{with} \quad C \in \mathbb{R}^{2\times 2}, \quad b \in \mathbb{R}^2,$$

for the elastic potential of u, we have

$$\mathcal{P}[u] = \mathcal{S}^{\text{elas}}[u] = \int_\Omega \mu(c_{1,1}^2 + (c_{1,2} + c_{2,1})^2/2 + c_{2,2}^2) + \lambda(c_{1,1} + c_{2,2})^2/2\, dx.$$

Hence, for $\mu \neq 0$,

$$\mathcal{P}[u] = 0 \iff c_{1,1} = c_{2,2} = 0 \wedge c_{1,2} = -c_{2,1}.$$

Note that in the derivation of elastic registration (see Section 9.1), we explicitly decomposed the transformation into rigid and non-rigid parts. Thus, from this modeling, the rigid parts need to be pre-registered.

For diffusion registration, we have

$$\mathcal{S}^{\text{diff}}[u] = \int_\Omega c_{1,1}^2 + c_{1,2}^2 + c_{2,1}^2 + c_{2,1}^2 dx$$

and $\mathcal{S}^{\text{diff}}[u] = 0 \iff C = 0$. Since diffusion registration penalizes the norm of the gradient, this property is a direct consequence of the regularizing term.

As a consequence, for all these non-linear registration techniques an affine linear pre-registration is unavoidable. In order to circumvent this additional pre-registration we introduce a novel regularizing term based on second order derivatives. Since the regularizer is related to *curvature*, the novel registration is called *curvature registration*; cf., Fischer & Modersitzki (2003).

The main point is not that the additional pre-registration becomes redundant but that the registration becomes less dependent on the initial position of the reference and template images.

Curvature registration is based on the distance measure $\mathcal{D}$ (cf., eqn (8.10)) and the regularizer

$$\mathcal{S}^{\text{curv}}[u] := \tfrac{1}{2} a[u, u], \tag{12.1}$$

where the bi-linear form a is defined by

$$a[u, v] = \sum_{\ell=1}^{d} \int_\Omega \Delta u_\ell \Delta v_\ell \, dx$$