

Chapter 2

Motion Estimation

In the introduction of this book we have seen that PET image acquisition is susceptible to motion artifacts. The first step to overcome this problem is the separation of the measured data into different motion states via gating as described in Sect. 1.3. The next step is the estimation of motion between these gated reconstructions. This step is crucial as its accuracy highly influences the quality of the final motion corrected image.

In this chapter we discuss different algorithmic paradigms in terms of *image registration* and *optical flow* for motion estimation. For both paradigms, a focus is laid on the preservation of radioactivity (mass-preservation) to incorporate a priori information. For image registration, different data terms are introduced which are suitable for monomodal PET registration in combination with mass-preservation. In particular, a data term adopted to noisy PET data is proposed with the SAD measure. The preservation of mass requires diffeomorphic transformations to be a valid assumption. Thus, different regularization schemes will be introduced and discussed. In addition to the non-parametric transformation model we also present motion estimation using the popular parametric transformation model based on B-splines for image registration. For optical flow, classical local and global approaches are introduced and advanced methods are briefly discussed. In particular, mass-preserving optical flow in combination with non-quadratic penalization is addressed. The last part of this chapter compares the two algorithmic paradigms and intends to provide some guidance to finding an appropriate method for motion estimation and its configuration in practice.

2.1 Image Registration

Image registration is a versatile approach to motion estimation and provides a full range of application-specific options. We give an introduction to image registration including suitable options for the data term in Sect. 2.1.1 and the regularization functional in Sect. 2.1.2. The alternative to non-parametric image registration in terms of B-spline transformations is discussed in Sect. 2.1.3. One of the main messages of this book is the mass-preserving transformation model introduced in Sect. 2.1.4 which is tailored for gated PET. We refer to the comprehensive reviews of image registration for further reading [18, 49, 63, 66, 88, 90, 94, 95, 139].

The aim of image registration is to spatially align two corresponding images. In order to formulate this definition of image registration mathematically, we need to find a way to measure the similarity of two images. The objective is then to find a meaningful spatial transformation that, applied to one of the images, maximizes the similarity.

For motion estimation, a so-called *template image* $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ is registered onto a *reference image* $\mathcal{R} : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^3$ is the *image domain*. This yields a *spatial transformation* $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ representing point-to-point correspondences between $\mathcal{T}$ and $\mathcal{R}$. To find $\mathbf{y}$, the following functional has to be minimized

$$\arg \min_{\mathbf{y}} \mathcal{J}(\mathbf{y}) := \mathcal{D}(\mathcal{M}(\mathcal{T}, \mathbf{y}), \mathcal{R}) + \alpha \mathcal{S}(\mathbf{y}) . \quad (2.1)$$

Here $\mathcal{D}$ denotes the *distance functional* measuring the dissimilarity between the transformed template image and the fixed reference image. $\mathcal{M}$ is the *transformation model* specifying how the transformation $\mathbf{y}$ should be applied to the template image $\mathcal{T}$. $\mathcal{S}$ is the *regularization functional* which penalizes non-smooth transformations and thus enforces meaningful solutions.

The standard transformation model is simply given by an interpolation of $\mathcal{T}$ at the transformed grid $\mathbf{y}$

$$\mathcal{M}^{\text{std}}(\mathcal{T}, \mathbf{y}) := \mathcal{T} \circ \mathbf{y} = \mathcal{T}(\mathbf{y}) . \quad (2.2)$$

We will discuss an alternative transformation model for mass-preserving image registration in Definition 2.11. A discussion of different options for the data term $\mathcal{D}$ is given in the next Sect. 2.1.1 and for the regularization term $\mathcal{S}$ in Sect. 2.1.2.

2.1.1 Data Terms

The *data term* measures the dissimilarity, i.e., reversal of the similarity, of two input images. In Eq. (2.1) the transformed template image is compared to the reference image. For simplicity we will use the original template image (without transformation) for the following definitions without loss of generality.

(Dis)similarity measures for monomodal registration tasks are best suited for motion correction of gated PET data. We thus restrict the discussion in the rest of this section to these monomodal measures. For a more detailed discussion of similarity measures for multimodal studies, such as mutual information (which measures the amount of shared information of two images) or normalized gradient fields (which measures deviations in the gradients of the input images), we refer to [95].

A common dissimilarity measure for monomodal image registration is SSD, which is introduced in the following Definition. Differences of the reference and the template image are measured quadratically.

Definition 2.1 ($\mathcal{D}^{\text{SSD}}$ – Sum of squared differences). The *sum of squared differences* (SSD) of two images $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ and $\mathcal{R} : \Omega \rightarrow \mathbb{R}$ on a domain $\Omega \subset \mathbb{R}^3$ is defined as

$$\mathcal{D}^{\text{SSD}}(\mathcal{T}, \mathcal{R}) := \frac{1}{2} \int_{\Omega} (\mathcal{T}(\mathbf{x}) - \mathcal{R}(\mathbf{x}))^2 d\mathbf{x}. \quad (2.3)$$

SSD measures the point-wise distances of image intensities. For images with locally similar intensities the measure gets low. SSD has to be minimized and has its optimal value at 0.

Large differences in the input images are often induced by noise which consequently leads to a high energy in the data term. L_1 -like distance measures are often used in optical flow techniques [19] to overcome this problem. The L_1 distance of two images $\mathcal{T}$ and $\mathcal{R}$ is defined as

$$L_1(\mathcal{T}, \mathcal{R}) := \int_{\Omega} |\mathcal{T}(\mathbf{x}) - \mathcal{R}(\mathbf{x})| d\mathbf{x}. \quad (2.4)$$

Using this functional as a data term in Eq. (2.1) raises problems as we require differentiability of all components during optimization. As the absolute value function is not differentiable at zero, the idea is thus to create a differentiable version of the L_1 distance by adding a differentiable outer function ψ to the difference in Eq. (2.4) with a similar behavior to the absolute value function.

Definition 2.2 ($\mathcal{D}^{\text{SAD}}$ – Sum of absolute differences). The (approximated) *sum of absolute differences* (SAD) of two images $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ and $\mathcal{R} : \Omega \rightarrow \mathbb{R}$ on a domain $\Omega \subset \mathbb{R}^3$ is defined as

$$\mathcal{D}^{\text{SAD}}(\mathcal{T}, \mathcal{R}) := \int_{\Omega} \psi(\mathcal{T}(\mathbf{x}) - \mathcal{R}(\mathbf{x})) d\mathbf{x}, \quad (2.5)$$

(continued)

Definition 2.2 (continued)

where $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is the continuously differentiable Charbonnier penalizing function [28] with a small positive constant $\beta \in \mathbb{R}^{>0}$

$$\psi(x) = \sqrt{x^2 + \beta^2} . \quad (2.6)$$

Remark 2.1. Different choices for the function ψ in Definition 2.2 are possible, cf. [19].

Remark 2.2. The function ψ in Eq. (2.6) is always greater than zero. Consequently, $\mathcal{D}^{\text{SAD}}(\mathcal{T}, \mathcal{T}) \neq 0$. This issue could be fixed by subtracting β

$$\hat{\psi}(x) := \sqrt{x^2 + \beta^2} - \beta . \quad (2.7)$$

For minimization of the registration functional we need to compute the derivatives of the functional and hence the first

$$\partial \psi(x) = \frac{x}{\sqrt{x^2 + \beta^2}} \quad (2.8)$$

and second derivative of ψ

$$\partial^2 \psi(x) = \frac{\beta^2}{(x^2 + \beta^2)^{3/2}} \quad (2.9)$$

are given here for completeness.

The behavior of the penalizing function ψ for different values of β is shown in Fig. 2.1. It can be observed that ψ becomes more quadratic around zero for higher β values. This might prevent abrupt jumps in the first derivative between -1 and 1 during optimization, thus stabilizing the optimization process.

2.1.2 Regularization

According to Hadamard [61] a problem is called *well-posed* if there

1. *Exists* a solution that is
2. *Unique* and when
3. Small changes in data lead only to small changes in the result.

In image registration, small changes in the data can lead to large changes in the results. Further, as the uniqueness is generally not given, image registration is *ill-posed* [48]. This makes regularization inevitable and essential to find feasible transformations.

Regularization restricts the space of possible transformations to a smaller set of reasonable functions. Regularization should be chosen depending on the application and can either be implicit or explicit. If the transformation is regularized by the properties of the space itself, as in parametric image registration, we speak of *implicit regularization*. For example, for a rigid 3D transformation only a total of three rotation and three translation parameters need to be estimated instead of a 3D motion vector for each voxel. In *explicit regularization*, a penalty is added to the registration functional to avoid non-smooth transformations. The penalty is often based on a model that is physically sound and fits the processed data.

As the regularization for parametric transformations is implicitly given we will only analyze penalties for non-linear image registration in this section. In the rest of this section we will introduce the following regularizers:

- Diffusion regularization (Sect. 2.1.2.1),
- Elastic regularization (Sect. 2.1.2.2), and
- Hyperelastic regularization (Sect. 2.1.2.3).

These regularizers are specifically chosen as diffusion regularization is typically used for optical flow applications. Further, VAMPIRE relies on hyperelastic regularization, which is a non-linear generalization of linear elastic regularization. More information about regularization (e.g., curvature or log-elasticity) and additional constraints, which are beyond the scope of this book, can be found in [3, 48, 94, 99, 107].

Hereinafter, we assume a given transformation $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which is composed of the spatial position $\mathbf{x} \in \mathbb{R}^3$ and a deformation (or displacement) $\mathbf{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ at position $\mathbf{x}$, i.e., $\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$.

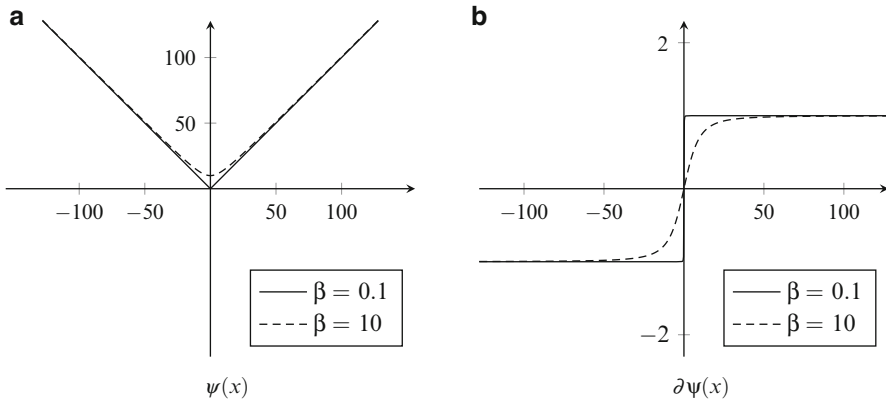


Fig. 2.1 The Charbonnier penalizing function (a) from Eq. (2.6) and its derivative (b) from Eq. (2.8) are plotted for $\beta = 0.1$ and $\beta = 10$. They both give an approximation to the absolute value function

2.1.2.1 Diffusion Regularization

One of the simplest ways to control the behavior of the transformation is *diffusion regularization* [47]. The basic idea is to disallow large variations in the gradient of the motion vector between neighboring voxels, i.e., oscillations in the deformation $\mathbf{u}$ are penalized.

The motivation is thus simply a demanded smoothness of the transformation and no physical one. An advantage of diffusion regularization over other methods is its low computational complexity which allows fast computations and the treatment of high-dimensional data [94]. This becomes important when striving for real-time computations, which is why diffusion regularization is popular for optical flow techniques, cf. Sect. 2.2.

Definition 2.3 ($\mathcal{S}^{\text{diff}}$ – Diffusion regularization). For a transformation $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with $\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathbf{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the *diffusion regularization energy* is defined as

$$\mathcal{S}^{\text{diff}}(\mathbf{u}) := \int_{\Omega} \|\nabla \mathbf{u}(\mathbf{x})\|_2^2 d\mathbf{x} = \int_{\Omega} \sum_{i=1}^3 |\nabla \mathbf{u}^i(\mathbf{x})|^2 d\mathbf{x}, \quad (2.10)$$

where the (squared) Frobenius norm is defined as $\|A\|_2^2 := \text{tr}(A^T A)$ for matrices $A \in \mathbb{R}^{3 \times 3}$. $\mathbf{u}^i$ denotes the i -th component of $\mathbf{u}$ with $i \in \{1, 2, 3\}$.

2.1.2.2 Elastic Regularization

As indicated in the introduction of this section, regularization methods can be motivated physically. This is the case for *linear elastic regularization* which measures the forces of a transformation applied to an elastic material [17] (Lagrange frame). In the language of elasticity we would say that the *stress* due to *strain* is measured. Or vice versa, in a more physical motivation, the deformation (*strain*) is a reaction of force (*stress*) [94] (Euler frame).

Definition 2.4 ($\mathcal{S}^{\text{elastic}}$ – Elastic regularization). For a transformation $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with $\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathbf{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the *elastic regularization energy* is defined as

(continued)

Definition 2.4 (continued)

$$\begin{aligned} \mathcal{S}^{\text{elastic}}(\mathbf{u}) &:= \int_{\Omega} \mu \|\nabla \mathbf{u}(\mathbf{x})\|_2^2 + (\lambda + \mu)(\nabla \cdot \mathbf{u}(\mathbf{x}))^2 d\mathbf{x} \\ &= \int_{\Omega} \mu \left(\sum_{i=1}^3 |\nabla \mathbf{u}^i(\mathbf{x})|^2 \right) + (\lambda + \mu) \left(\sum_{i=1}^3 \mathbf{u}_i^i(\mathbf{x}) \right)^2 d\mathbf{x}, \end{aligned} \quad (2.11)$$

where $\mu \in \mathbb{R}^{>0}$ and $\lambda \in \mathbb{R}^{>0}$ are the Lamé constants according to Ciarlet [30]. Again, $\|\cdot\|_2$ is the Frobenius norm (see Eq. (2.10)) and $\mathbf{u}^i$ denotes the i -th component of $\mathbf{u}$ with $i \in \{1, 2, 3\}$.

To better understand the physical interpretation of the elastic energy we will take a look at some useful properties of elasticity in terms of the *Poisson ratio* and *Young's modulus*.

Definition 2.5 (ν – Poisson ratio). The *Poisson (contraction) ratio* is defined as

$$\nu = \frac{\lambda}{2(\lambda + \mu)}, \quad (2.12)$$

for the Lamé constants $\mu \in \mathbb{R}^{>0}$ and $\lambda \in \mathbb{R}^{>0}$ in Definition 2.4.

The Poisson ratio ν measures the amount of compression for the linear elastic model. If a material is compressed in one direction, the Poisson ratio measures the quotient of this value and the expansion in the other directions. For compressible materials we thus have a small quotient and for incompressible materials a large one. To model incompressible materials in the registration setting, the Poisson ratio needs to be weighted high. Note that this interpretation only holds for small displacements since we are dealing with a linear model.

Definition 2.6 (E – Young's modulus). *Young's modulus* is defined as

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad (2.13)$$

for the Lamé constants $\mu \in \mathbb{R}^{>0}$ and $\lambda \in \mathbb{R}^{>0}$ in Definition 2.4.

Young's modulus E is the quotient of stress (tension) and strain (stretch) in the same direction. The connection between the Lamé constants, Poisson ratio, and Young's modulus is given by

$$\lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}, \quad (2.14)$$

$$\mu = \frac{E}{2(1+\nu)}, \quad (2.15)$$

$$\lambda > 0 \text{ and } \mu > 0 \iff 0 < \nu < \frac{1}{2} \text{ and } E > 0. \quad (2.16)$$

The linear dependency of Young's modulus E and the Lamé constants can be seen in Eqs. (2.14) and (2.15). Together with Eq. (2.11) it gets obvious that E simply scales the regularization functional. For practical reasons it can thus be set to $E = 1$ [73].

2.1.2.3 Hyperelastic Regularization

In many medical applications of image registration the user has a priori knowledge about the processed data concerning the allowed deformations. For example the invertibility of the transformation is a general requirement. More specific knowledge, like, e.g., the degree of compressibility of tissue can also be controlled. We have seen in the linear elastic case that the Poisson ratio (Definition 2.6) gives a measure for compressibility of small deformations. In the non-linear case, the determinant of the Jacobian of the transformation measures the volume change and can thus be used to control the compression behavior – even for large deformations. This is done with polyconvex *hyperelastic regularization* [22, 30, 40]. The regularization functional $\mathcal{S}^{\text{hyper}}$ controls changes in length, area of the surface, and volume of $\mathbf{y}$ and guarantees thereby in particular the invertibility of the estimated transformation.

Definition 2.7 ($\mathcal{S}^{\text{hyper}}$ – Hyperelastic regularization). Let $\alpha_l, \alpha_a, \alpha_v \in \mathbb{R}^{>0}$ be constants and $p, q \geq 2$. Further, let $\Gamma_a, \Gamma_v : \mathbb{R} \rightarrow \mathbb{R}$ be positive and strictly convex functions, with Γ_v satisfying $\lim_{z \rightarrow 0^+} \Gamma_v(z) = \lim_{z \rightarrow \infty} \Gamma_v(z) = \infty$. The *hyperelastic regularization* energy caused by the transformation $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with $\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathbf{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is defined as

$$\mathcal{S}^{\text{hyper}}(\mathbf{y}) = \alpha_l \cdot \mathcal{S}^{\text{length}}(\mathbf{y}) + \alpha_a \cdot \mathcal{S}^{\text{area}}(\mathbf{y}) + \alpha_v \cdot \mathcal{S}^{\text{vol}}(\mathbf{y}). \quad (2.17)$$

The three summands individually control changes in length, area of the surface, and volume

$$\mathcal{S}^{\text{length}}(\mathbf{y}) := \int_{\Omega} \|\nabla(\mathbf{y}(\mathbf{x}) - \mathbf{x})\|_2^p d\mathbf{x} = \int_{\Omega} \|\nabla \mathbf{u}(\mathbf{x})\|_2^p d\mathbf{x} \quad (2.18)$$

$$\mathcal{S}^{\text{area}}(\mathbf{y}) := \int_{\Omega} \Gamma_a(\|\text{Cof}(\nabla \mathbf{y}(\mathbf{x}))\|_2^q) d\mathbf{x} \quad (2.19)$$

(continued)

Definition 2.7 (continued)

$$\mathcal{S}^{\text{vol}}(\mathbf{y}) := \int_{\Omega} \Gamma_v(\det(\nabla \mathbf{y}(\mathbf{x}))) d\mathbf{x}, \quad (2.20)$$

where $\|\cdot\|_2$ is the Frobenius norm (see Eq. (2.10)) and $\text{Cof}(\cdot)$ denotes the cofactor matrix.

Remark 2.3. A typical choice of p and q in Definition 2.7 is $p = q = 2$ [51].

In the formulation in Eq. (2.1), the positive real number α balances between minimizing the data driven energy term (maximizing the image similarity) and retaining smooth and realistic transformations, which is controlled by the regularization energy. As each term of the hyperelastic regularizer has an individual weighting factor, α can be set to 1 and only α_f , α_a , and α_v need to be determined.

For $\alpha_v > 0$, the conditions for Γ_v claimed above ensure $\det(\nabla \mathbf{y}) > 0$, i.e., $\mathbf{y}$ is a diffeomorphism [40]. This allows us to omit the absolute value bars later in Eq. (2.45) of the mass-preserving transformation model. Hyperelastic regularization has the ability of modeling tissue characteristics like compressibility, improves the robustness against noise and thus enforces realistic cardiac and respiratory motion estimates.

2.1.2.4 Relations

To deepen the understanding of the regularization energies described in this section, we highlight some similarities and differences. This helps us to understand the relations between the different regularization variants. As we will see in Sect. 2.1.4, controlling volume changes plays an important role in connection with the mass-preserving motion estimation approach VAMPIRE. Accordingly, this aspect is highlighted in particular.

Diffusion $\leftrightarrow$ Elastic

As we have seen, the elastic energy consists of the two terms $\|\nabla \mathbf{u}\|_2^2$ and $(\nabla \cdot \mathbf{u})^2$. The interpretation of the latter is revealed as a measure of compressibility, i.e., of volume changes. The first term is the known diffusion regularization term, which is hence integrated into the elastic energy.

Diffusion $\leftrightarrow$ Hyperelastic

Diffusion regularization is included into hyperelastic regularization. The length term $\mathcal{S}^{\text{length}}$ in Eq. 2.17 equals the diffusion regularization term. The hyperelastic regularization energy has two additional energy terms – $\mathcal{S}^{\text{area}}$ measuring changes in area and $\mathcal{S}^{\text{vol}}$ measuring changes in volume.

Elastic $\leftrightarrow$ Hyperelastic

The Jacobian determinant $\det(\nabla \mathbf{y})$ represents the exact volume change due to a transformation $\mathbf{y}$. Consequently, a value of one characterizes incompressibility. Let us recall, the Poisson ratio ν in the linear elastic model (Definition 2.5) also controls the amount of incompressibility. Large values of ν (incompressible) results from large values of λ . For large values of λ compared to μ , the elastic energy is dominated by $(\nabla \cdot \mathbf{u})^2$. The Jacobian determinant can be approximated by $\nabla \cdot \mathbf{u}$ if the $|\partial_j \mathbf{u}^i|$ are sufficiently small:

$$\det(\nabla \mathbf{y}) \approx 1 + \nabla \cdot \mathbf{u} . \quad (2.21)$$

This shows the approximate behavior of linear elastic regularization. To conclude, hyperelastic regularization holds for large deformations whereas linear elastic regularization is restricted to small deformations. Further, the area energy term in hyperelastic regularization, which is missing in elastic regularization, serves as a link between the length and volume term [22].

2.1.3 B-Spline Transformation

So far only non-parametric transformations were discussed. However, in some cases the search space for the desired transformation can be restricted by using a suitable parametric transformation model. As an example, for many brain alignment tasks the transformation can be assumed to be rigid. This can increase the robustness of the registration task. The most prominent parametric transformations are scaling, rigid, affine, and spline-based transformations [95]. For non-linear transformations, as required for PET motion estimation, the parametric B-spline transformation is of particular interest.

In parametric image registration the transformation is given in terms of a parametric function $\mathbf{y}_p : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. $\mathbf{y}_p$ is defined by a parameter vector p . The task is to find the parameter set p minimizing the distance between $\mathcal{R}$ and the transformed input image $\mathcal{M}(\mathcal{T}, \mathbf{y}_p)$

$$\mathcal{J}(p) := \mathcal{D}(\mathcal{M}(\mathcal{T}, \mathbf{y}_p), \mathcal{R}) \stackrel{!}{=} \min , \quad (2.22)$$

where $\mathcal{D}$ is a data term according to Sect. 2.1.1 and $\mathcal{M}$ is a transformation model according to Eq. (2.2) or (2.46).

Definition 2.8 (Spline transformation). Given a number $n = (n^1, n^2, n^3) \in \mathbb{N}^3$ of spline coefficients and a spline basis function $b : \mathbb{R} \rightarrow \mathbb{R}$, the *spline transformation* or *free-form transformation* $\mathbf{y}_p(\mathbf{x}) = (\mathbf{y}_p^1(\mathbf{x}), \mathbf{y}_p^2(\mathbf{x}), \mathbf{y}_p^3(\mathbf{x}))$ for a point $\mathbf{x} = (x, y, z)^T \in \Omega \subset \mathbb{R}^3$ is defined as

$$\mathbf{y}_p^1(\mathbf{x}) = x + \sum_{i=1}^{n^1} \sum_{j=1}^{n^2} \sum_{k=1}^{n^3} p_{i,j,k}^1 b_i(x) b_j(y) b_k(z), \quad (2.23)$$

$$\mathbf{y}_p^2(\mathbf{x}) = y + \sum_{i=1}^{n^1} \sum_{j=1}^{n^2} \sum_{k=1}^{n^3} p_{i,j,k}^2 b_i(x) b_j(y) b_k(z), \quad (2.24)$$

$$\mathbf{y}_p^3(\mathbf{x}) = z + \sum_{i=1}^{n^1} \sum_{j=1}^{n^2} \sum_{k=1}^{n^3} p_{i,j,k}^3 b_i(x) b_j(y) b_k(z), \quad (2.25)$$

where

$$p = \left\{ (p_{i,j,k}^1, p_{i,j,k}^2, p_{i,j,k}^3)^T \in \mathbb{R}^3 \mid \right. \\ \left. i \in \{1, \dots, n^1\}, j \in \{1, \dots, n^2\}, k \in \{1, \dots, n^3\} \right\} \quad (2.26)$$

are the spline coefficients and $b_i(x) = b(x - i)$ for $i \in \{1, \dots, n^1\}$, $b_j(y) = b(y - j)$ for $j \in \{1, \dots, n^2\}$, $b_k(z) = b(z - k)$ for $k \in \{1, \dots, n^3\}$. Note that this notation is only valid for $\Omega = \{(x, y, z)^T \in \mathbb{R}^3 \mid 0 \leq x < n^1 + 1, 0 \leq y < n^2 + 1, 0 \leq z < n^3 + 1\}$.

A possible choice for the basis function $b : \mathbb{R} \rightarrow \mathbb{R}$, often called *mother spline* (cf. Fig. 2.2), is

$$b(x) = \begin{cases} (x+2)^3, & -2 \leq x < -1, \\ -x^3 - 2(x+1)^3 + 6(x+1), & -1 \leq x < 0, \\ x^3 + 2(x-1)^3 - 6(x-1), & 0 \leq x < 1, \\ (-x+2)^3, & 1 \leq x < 2, \\ 0, & \text{else.} \end{cases} \quad (2.27)$$

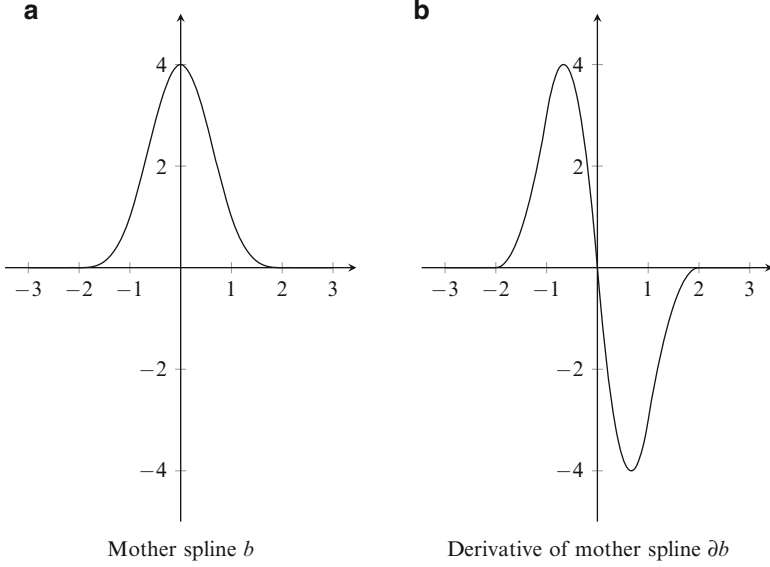


Fig. 2.2 The mother spline b defined in Eq. (2.27) is shown in (a). b is continuously differentiable and its derivative is plotted in (b)

Parametric transformations are implicitly regularized by the reduced number of parameters compared to non-parametric transformations, as discussed in Sect. 2.1.2. Hence, regularization of, e.g., rigid or affine transformations, is usually not necessary. However, regularization becomes of importance for spline transformations due to the non-linearity [48]. Possible regularization variants for spline transformations are discussed in the following. For parametric transformations the functional in Eq. (2.22) becomes

$$\mathcal{J}(p) := \mathcal{D}(\mathcal{M}(\mathcal{T}, \mathbf{y}_p), \mathcal{R}) + \alpha \cdot \mathcal{S}_M(p) \stackrel{!}{=} \min, \quad (2.28)$$

for a regularization term $\mathcal{S}_M$ and a scalar weighting factor $\alpha \in \mathbb{R}^{>0}$ balancing between the data and regularization term.

2.1.3.1 Hyperelastic Regularization

Hyperelastic regularization, introduced for non-parametric transformations in Definition 2.7, can also be applied to the B-spline transformation model. The transformation $\mathbf{y}_p(\mathbf{x}) = \mathbf{x} + \mathbf{u}_p(\mathbf{x})$ is written in the discrete setting of [95] as

$$\mathbf{y} = \mathbf{x} + \mathbf{Q} \cdot \mathbf{p}. \quad (2.29)$$

The deformation $\mathbf{u}$ is represented by a projection of the spline coefficients, given by the parameter vector p , into the image space by the projection matrix Q

$$\mathbf{u} = Q \cdot p . \quad (2.30)$$

For the hyperelastic regularization the transformation $\mathbf{y}$ is first determined according to Eq. (2.29) and is then regularized as described in Eq. (2.17). The regularization functional in Eq. (2.28) is then defined as

$$\mathcal{S}_M(p) := \mathcal{S}^{\text{hyper}}(\mathbf{x} + Q \cdot p) . \quad (2.31)$$

2.1.3.2 Coefficient-Based Regularization

The regularization term $\mathcal{S}_M(p)$ in Eq. (2.28) is based on the coefficient vector p in the case of coefficient-based regularization and is a weighted norm of the form

$$\mathcal{S}_M(p) := \|p\|_M^2 := p^T M p . \quad (2.32)$$

The matrix M determines the regularization type. Instead of regularizing the deformation $\mathbf{u}_p(\mathbf{x})$ for each spatial position $\mathbf{x} \in \mathbb{R}^3$ as in the non-parametric case (cf. Sect. 2.1.2) we now regularize directly on p . We speak of coefficient-based regularization in *image space* if the regularization on p considers the projection matrix Q and thus regularizes $Q \cdot p$. If only p is taken into account we speak of coefficient-based regularization in *coefficient space*.

With Tikhonov regularization we will discuss one option for coefficient-based regularization in coefficient space. Tikhonov regularization punishes gradients in the spline coefficients and is the coefficient-based analog of diffusion regularization introduced in Definition 2.3.

Definition 2.9 (Tikhonov regularization (coefficient space)). *Tikhonov regularization* in coefficient space is defined by the choice of

$$M^{\text{tc}} = D^T D \quad (2.33)$$

in Eq. (2.32). The superscript tc denotes (t)ikhonov in (c)oefficient space. D is defined by

$$D = I_3 \otimes \begin{bmatrix} D^1 \\ D^2 \\ D^3 \end{bmatrix} . \quad (2.34)$$

(continued)

Definition 2.9 (continued)

Here I_3 is the 3×3 identity matrix and $\otimes$ denotes the Kronecker product [16]. The 1D derivative operators are defined by

$$D^1 = I_{n^3} \otimes I_{n^2} \otimes d_1^1, \quad (2.35)$$

$$D^2 = I_{n^3} \otimes d_1^2 \otimes I_{n^1}, \quad (2.36)$$

$$D^3 = d_1^3 \otimes I_{n^2} \otimes I_{n^1}. \quad (2.37)$$

The I_{n^i} denote identity matrices with the size $n^i \times n^i, i \in \{1, 2, 3\}$, and the n^i defined according to Definition 2.8. The discrete 1D first derivative operators for each dimension are defined by

$$d_1^i = \begin{bmatrix} -1 & 1 & 0 \\ & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n^i-1, n^i}, i \in \{1, 2, 3\}. \quad (2.38)$$

The analog of the above definition in image space is given with the following Definition 2.10. As the real displacements and not the corresponding coefficients are processed, this version is even more related to the diffusion regularization energy in Definition 2.3.

Definition 2.10 (Tikhonov regularization (image space)). Let the number of spline coefficients be $n = (n^1, n^2, n^3) \in \mathbb{N}^3$ and $b_i(x) = b(x - i)$ for $i \in \{1, \dots, n^d\}$, $d \in \{1, 2, 3\}$, be the translated spline basis function b defined in Eq. (2.27). The *Tikhonov regularization* matrix in image space is defined as

$$M^{\text{ti}} = I_3 \otimes T^3 \otimes T^2 \otimes T^1. \quad (2.39)$$

The matrices $T^d \in \mathbb{R}^{n^d, n^d}$ are defined by

$$T_{i,j}^d = \int_{\Omega^d} \partial b_i(x) \cdot \partial b_j(x) dx, d \in \{1, 2, 3\}, \quad (2.40)$$

where $i \in \{1, \dots, n^d\}$, $j \in \{1, \dots, n^d\}$, and $\Omega^d \subset \mathbb{R}$ is the 1D subspace of Ω in dimension $d \in \{1, 2, 3\}$.

(continued)

Definition 2.10 (continued)

The concrete numbers for the matrix T^d (including boundary treatment) are

$$T^d = \begin{bmatrix} 15.1 & -5.0 & -7.2 & -0.3 & & & & & 0 \\ -5.0 & 23.9 & -4.5 & -7.2 & -0.3 & & & & \\ -7.2 & -4.5 & 24.0 & -4.5 & -7.2 & -0.3 & & & \\ -0.3 & -7.2 & -4.5 & 24.0 & -4.5 & -7.2 & -0.3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -0.3 & -7.2 & -4.5 & 24.0 & -4.5 & -7.2 & -0.3 \\ & & & & -0.3 & -7.2 & -4.5 & 24.0 & -4.5 & -7.2 \\ & & & & & -0.3 & -7.2 & -4.5 & 23.9 & -5.0 \\ 0 & & & & & & -0.3 & -7.2 & -5.0 & 15.1 \end{bmatrix} \in \mathbb{R}^{n^d, n^d} \quad (2.41)$$

for $d \in \{1, 2, 3\}$. The entries of the matrix T^d are obtained by calculating the integral in Eq. (2.40) for each index $i \in \{1, \dots, n^d\}$ and $j \in \{1, \dots, n^d\}$, $d \in \{1, 2, 3\}$.

2.1.4 Mass-Preserving Image Registration

The standard transformation model for image registration in Eq. (2.2) does not guarantee the preservation of mass, i.e., in general

$$\int_{\Omega} \mathcal{T}(\mathbf{x}) d\mathbf{x} \neq \int_{\Omega} \mathcal{T}(\mathbf{y}(\mathbf{x})) d\mathbf{x}. \quad (2.42)$$

Further, the distance functional $\mathcal{D}$ typically entails the assumption of similar intensities at corresponding points. This assumption is not valid for the standard transformation model $\mathcal{M}^{\text{std}}$ in case of dual gated PET as discussed in connection with Fig. 1.11. Consequently, the standard transformation model needs some modification to express this feature.

The requirement for the desired transformation model $\mathcal{M}^{\text{MP}}$ is the preservation of mass

$$\int_{\Omega} \mathcal{T}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}(x)) d\mathbf{x}. \quad (2.43)$$

From the integration by substitution theorem for multiple variables we know that the following equation holds

$$\int_{\mathbf{y}(\Omega)} \mathcal{T}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathcal{T}(\mathbf{y}(\mathbf{x})) |\det(\nabla \mathbf{y}(\mathbf{x}))| d\mathbf{x}. \quad (2.44)$$

With respect to PET, Eq.(2.44) guarantees already the same total amount of radioactivity before and after applying the transformation $\mathbf{y}$ to $\mathcal{T}$. The Jacobian determinant $|\det(\nabla \mathbf{y}(\mathbf{x}))|$ accounts for the volume change induced by the transformation $\mathbf{y}$, representing the mass-preserving component. As $\mathbf{y}$ should reflect cardiac and respiratory motion, transformations that are not bijective are anatomically not meaningful and have therefore to be excluded. For example, the hyperelastic regularization functional in Sect. 2.1.2.3 guarantees $\mathbf{y}$ to be diffeomorphic and orientation preserving, which allows us to drop the absolute value bars

$$\int_{\mathbf{y}(\Omega)} \mathcal{T}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathcal{T}(\mathbf{y}(\mathbf{x})) \det(\nabla \mathbf{y}(\mathbf{x})) d\mathbf{x}. \quad (2.45)$$

We derived the VAMPIRE (Variational Algorithm for Mass-Preserving Image Registration) based on the above considerations in our previous work [51]. The source code of VAMPIRE can be downloaded at [52].

Definition 2.11 ($\mathcal{M}^{\text{MP}}$ – Mass-preserving transformation model). For an image $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ on the domain $\Omega \subset \mathbb{R}^3$ and a transformation $\mathbf{y} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ the *mass-preserving transformation model* is defined as

$$\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}) := (\mathcal{T} \circ \mathbf{y}) \cdot \det(\nabla \mathbf{y}) = \mathcal{T}(\mathbf{y}) \cdot \det(\nabla \mathbf{y}). \quad (2.46)$$

In the mass-preserving transformation model of VAMPIRE the template image $\mathcal{T}$ is transformed by interpolation on the deformed grid $\mathbf{y}$ with an additional multiplication by the volume change. The multiplication by the Jacobian is a physiological and realistic modeling for density-based images [128, 138]. It guarantees similar intensities at corresponding points after transformation with a simultaneous preservation of the total amount of radioactivity.

A simple 2D example for the mass-preserving transformation model is shown in Fig. 2.3 for illustration. We will use the same notation as in the 3D case for $\mathcal{T}$, $\mathcal{R}$, Ω , $\mathbf{y}$, and $\mathbf{x}$. Thus, we redefine $\Omega \subset \mathbb{R}^2$, $\mathbf{y} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, and $\mathbf{x} = (x, y)^T \in \mathbb{R}^2$ for the 2D example. $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ and $\mathcal{R} : \Omega \rightarrow \mathbb{R}$ are defined in the same way but for the new domain Ω .

The images of the example are constructed to be similar to SA views of the heart to a certain degree, cf. Fig. 1.14b. Further, they have the same total mass (by construction) and thus exactly fulfill the conditions for mass-preservation. The template and reference image represent smooth signals based on Gaussian distributions. The template image is created by subtracting two Gaussian distributions with different standard deviation $\sigma_1, \sigma_2 \in \mathbb{R}^{>0}$ and $\sigma_1 > \sigma_2$

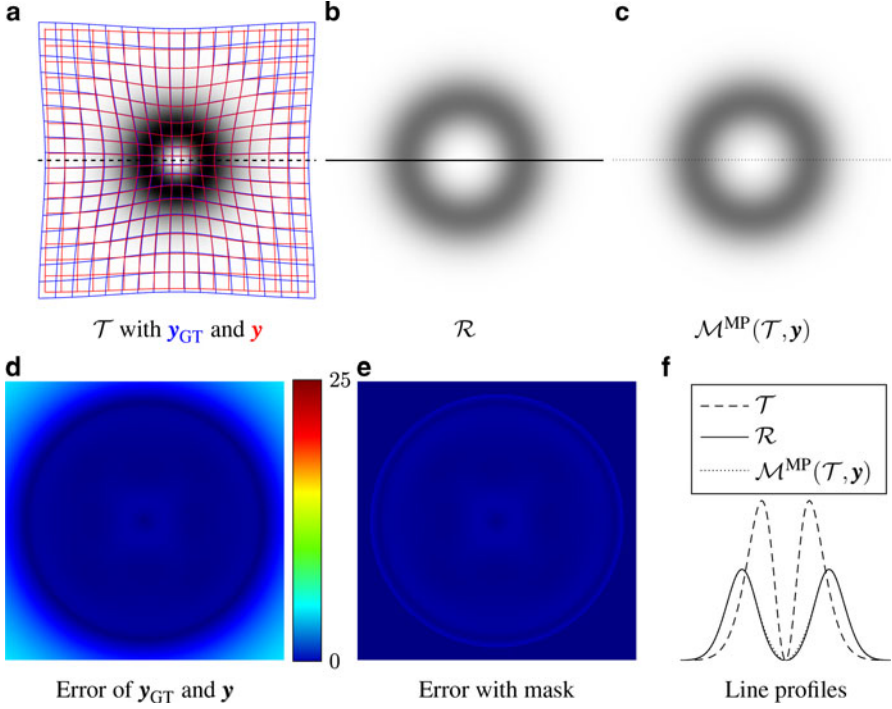


Fig. 2.3 2D example for the mass-preserving transformation model. The template image $\mathcal{T}$ in (a) is registered to the reference image $\mathcal{R}$ in (b) with VAMPIRE resulting in (c). The average endpoint error e between the estimated transformation $\mathbf{y}$ and the ground-truth transformation $\mathbf{y}_{\text{GT}}$ is visualized in (d). The same plot restricted to a mask where $\mathcal{T}$ and $\mathcal{R}$ have an intensity above 1 is shown in (e). The color map for both error plots is given in-between. Line profiles are shown in (f). (Colored figures are only available in the online version.)

$$\mathcal{T}(\mathbf{x}) := g(\mathbf{x}, \sigma_1) - g(\mathbf{x}, \sigma_2), \quad (2.47)$$

with

$$g(\mathbf{x}, \sigma) := \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (2.48)$$

The intensities are finally scaled between 0 and 255. The reference image $\mathcal{R}$ is created by applying a known transformation $\mathbf{y}_{\text{GT}}$ to $\mathcal{T}$ according to the mass-preserving transformation model $\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}_{\text{GT}})$ in Eq. (2.46). Hence, $\mathcal{T}$ and $\mathcal{R}$ have the same global mass. The ground-truth transformation is defined by

$$\mathbf{y}_{\text{GT}}(\mathbf{x}) := \begin{pmatrix} x \cdot (1 - \frac{1}{2}g(\mathbf{x}, \sigma_3)) \\ y \cdot (1 - \frac{1}{2}g(\mathbf{x}, \sigma_3)) \end{pmatrix}, \quad \sigma_3 \in \mathbb{R}^{>0}. \quad (2.49)$$

The template image is shown with the ground-truth transformation (blue) and an estimated transformation (red) in Fig. 2.3a. The average endpoint error e from Eq. (2.89) between the two grids is shown in Fig. 2.3d (scaled between 0 and 25 in pixel units). It can be seen that the estimated transformation differs from the ground-truth transformation at the corners of the images, where the intensities are almost zeros in the images $\mathcal{T}$ and $\mathcal{R}$. We thus restrict the quantitative error analysis to the region where the images have an intensity value above 1. The error e restricted to this area is shown in Fig. 2.3e and is 0.47 pixels on average with a maximum error $e^{\max}$ from Eq. (2.90) of 1.39 pixels. The reference image is shown in Fig. 2.3b and the resulting image after applying the estimated transformation is shown in Fig. 2.3c. Line profiles are given in Fig. 2.3f. The dotted line profile after mass-preserving motion estimation is almost indistinguishable from the reference line profile.

The same experiment is performed for the standard transformation model defined in Eq. (2.2) and is shown in Fig. 2.4. The estimated grid in Fig. 2.4a is subject to major errors which results in an erroneous transformed image in Fig. 2.4c. The error

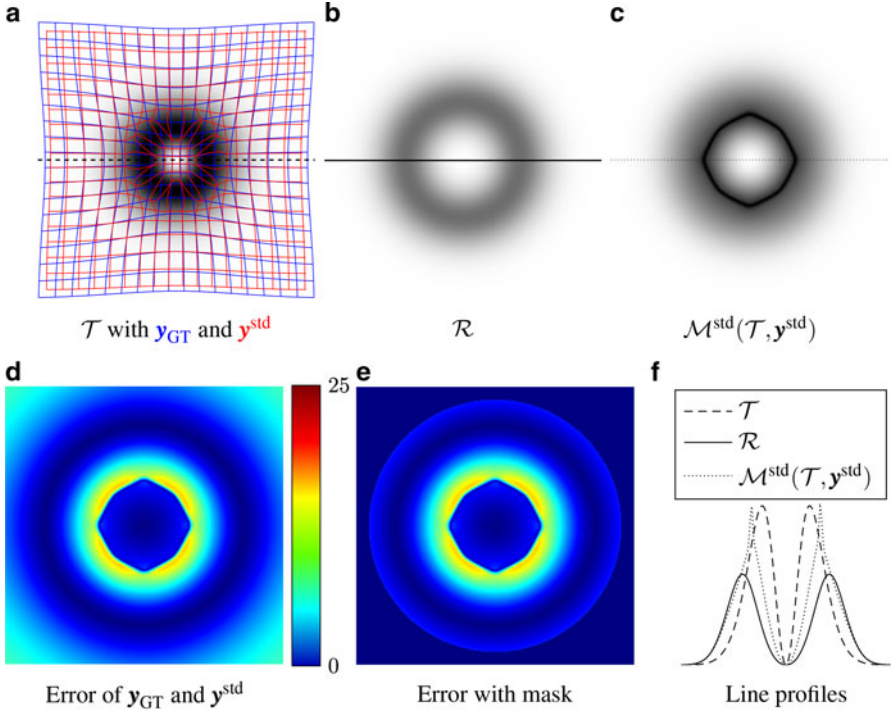


Fig. 2.4 2D example for the standard transformation model. The template image $\mathcal{T}$ in (a) is registered to the reference image $\mathcal{R}$ in (b) with the standard transformation model in Eq. (2.2) resulting in (c). The average endpoint error e between the estimated transformation $\mathbf{y}^{\text{std}}$ and the ground-truth transformation $\mathbf{y}_{\text{GT}}$ is visualized in (d). The same plot restricted to a mask where $\mathcal{T}$ and $\mathcal{R}$ have an intensity above 1 is shown in (e). The color map for both error plots is given in-between. Line profiles are shown in (f). (Colored figures are only available in the online version.)

e (restricted to the same area as discussed above) is shown in Fig. 2.4e (scaled between 0 and 25) and is 4.08 pixels on average with a maximum endpoint error $e^{\max}$ of 17.11 pixels. This simple 2D example thus shows that the mass-preserving motion model is mandatory for density-based images with varying intensities at corresponding points.

2.1.5 Multi-level Optimization

The image registration problem, in terms of minimizing the functional $\mathcal{J}$ in Eq. (2.1), is solved following a discretize-then-optimize approach as in [95]. Discrete objective functionals are obtained by discretizing the objective functional on a coarse to fine hierarchy of levels. On each level the discrete problem can be solved using standard optimization methods, like for instance Gauss-Newton, (l-)BFGS or Trust Region [100]. A short introduction to these methods is given in Sects. 2.1.5.1–2.1.5.3.

It has to be stressed that careful discretization is required to ensure that numerical solutions are meaningful. This is of particular importance for mass-preserving transformations where it is assumed that $\det(\nabla \mathbf{y}(\mathbf{x})) > 0$ for all $\mathbf{x} \in \Omega$. Discretization of these constraints is known to be extremely challenging and has been discussed in detail in [59, 60]. Using the discretization of the hyperelastic regularization functional derived in [22] we can assure positive Jacobean determinants.

We next discuss the *multi-level* strategy which is similar to cascadic multi-grid [15]. We begin by solving a discrete optimization on a very coarse level which thus has only a small number of unknowns. The coarse transformation is then interpolated to the next finer level and used as a starting guess for the minimization of the next objective function. The procedure is repeated until the transformation is of desired resolution. Typically, the image data is simplified as well. Multiple versions of the original images are constructed by iteratively down-sampling the data by a factor around $\frac{1}{2}$. Thus, for 3D data a voxel on a lower level is, e.g., created by averaging 8 neighboring voxels of the next higher level. Variants like using a Gaussian smoothing during down-sampling are also common. An example for multi-level data is given in Fig. 2.5.

There are several advantages of multi-level schemes. First, the danger of being trapped in a local minimum is reduced as only major image features drive the registration on a coarse level. Second, methods such as Gauss-Newton converge faster for a starting guess close to the minimum. Third, solving the coarse grid problems is computationally dramatically cheaper and in most cases only a small number of correction steps are required on the finer levels. The principle of multi-level image registration is summarized in Algorithm 2.1.

An artificial example for the potential behavior of the objective function is given in Fig. 2.6. The objective function on the lowest level (coarse level) shows a smooth behavior and thus allows a fast and robust initial guess of the global minimum. With increasing resolution level the additional information in the images leads to a less smooth behavior of the function.

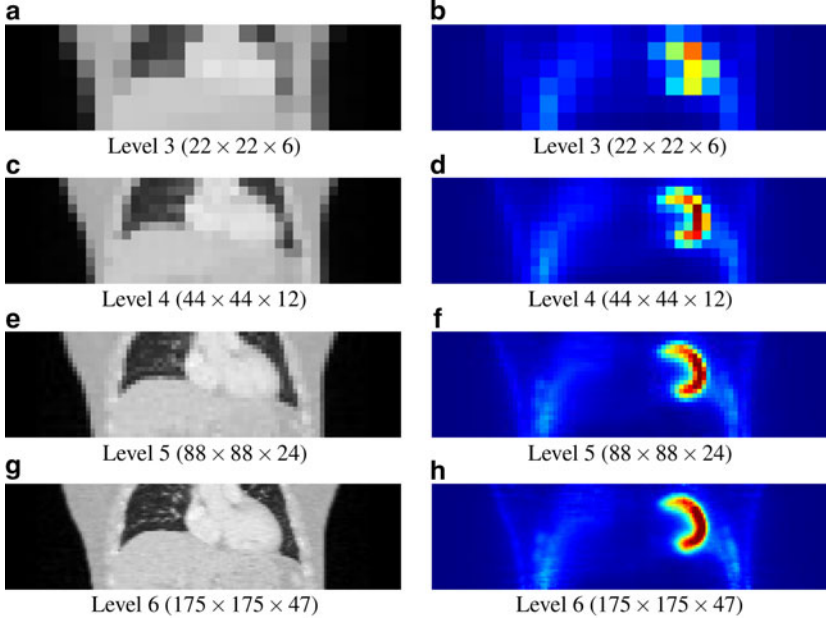


Fig. 2.5 Multi-Level: The central coronal slice in y -direction is shown for different resolution levels. The different levels of a CT scan are shown in the *left column* and corresponding PET images are shown in the *right column*. (a) Level 3 ($22 \times 22 \times 6$) (b) Level 3 ($22 \times 22 \times 6$) (c) Level 4 ($44 \times 44 \times 12$) (d) Level 4 ($44 \times 44 \times 12$) (e) Level 5 ($88 \times 88 \times 24$) (f) Level 5 ($88 \times 88 \times 24$) (g) Level 6 ($175 \times 175 \times 47$) (h) Level 6 ($175 \times 175 \times 47$)

Algorithm 2.1 Multi-level image registration

Input: Template image $\mathcal{T}$ and reference image $\mathcal{R}$

Output: Transformation $\mathbf{y}$

```

 $\mathbf{y}_{\min\text{Level}} \leftarrow \mathbf{1}$  // initialize transformation with identity
for level = minLevel  $\rightarrow$  maxLevel do
     $\mathcal{T}_{\text{level}} \leftarrow \mathcal{T}$  // down-sampling of template image
     $\mathcal{R}_{\text{level}} \leftarrow \mathcal{R}$  // down-sampling of reference image
     $\mathbf{y}_{\text{level}} \leftarrow \text{argmin} \mathcal{J}(\mathbf{y}_{\text{level}})$  // update motion by minimizing the functional
    if level < maxLevel then
         $\mathbf{y}_{\text{level}+1} \leftarrow \mathbf{y}_{\text{level}}$  // prolongate transformation to next level
    end if
end for
 $\mathbf{y} \leftarrow \mathbf{y}_{\max\text{Level}}$  // output final transformation

```

We will meet the general multi-level idea again in optical flow motion estimation in Sect. 2.2, as it allows to estimate large deformations with optical flow. However, the approach presented here still differs from the optimize-then-discretize approach that is typically used in optical flow, cf. Sect. 2.2. Most importantly, the discretization used here guarantees that the solution fulfills the properties assumed in the theory on each discretization level and that large steps can be taken in the numerical optimization. Also optimization techniques are well understood and clear rules for stopping [55], step lengths [1, 100], etc. exist.

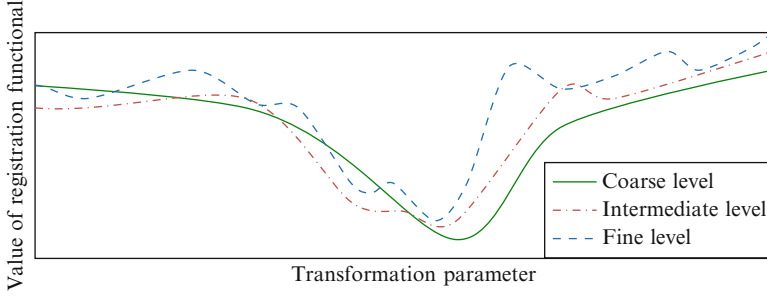


Fig. 2.6 Artificial example for the behavior of an objective function in a multi-level setting. For each level, going from the coarse to the fine level, the value of the registration functional is plotted against the (artificial) transformation parameter. With increasing level, the function is getting less smooth due to the increased level of details in the images. Note the slight shift of the global minimum between the different levels

2.1.5.1 Gauss-Newton

The minimization of the discretized registration functional $\mathcal{J}$ can be solved by means of the Gauss-Newton method. To this end, we can approximate the functional in Eq. (2.1) by a Taylor expansion (leaving the regularization term aside)

$$\begin{aligned}\mathcal{J}(\mathbf{y} + \nabla \mathbf{y}) &\approx \mathcal{J}(\mathbf{y}) + \nabla \mathcal{J}(\mathbf{y}) \nabla \mathbf{y} + \frac{1}{2} (\nabla \mathbf{y})^T (\nabla^2 \mathcal{J}(\mathbf{x})) (\nabla \mathbf{y}) \\ &= \mathcal{J} + \nabla \mathcal{J} \nabla \mathbf{y} + \frac{1}{2} (\nabla \mathbf{y})^T H (\nabla \mathbf{y}),\end{aligned}\quad (2.50)$$

where H is the (approximated) Hessian matrix. Let us assume the functional $\mathcal{J}$ is given by a residual $r(\mathbf{y})$ and an outer function $\psi(\cdot)$. In the case of SSD this is $r(\mathbf{y}) = \mathcal{T} \circ \mathbf{y} - \mathcal{R}$ and $\psi(x) = \frac{1}{2}x^2$. The Hessian matrix H is then approximated by

$$H = \nabla^2 \mathcal{J}(\mathbf{x}) = \nabla r(\mathbf{x})^T \nabla^2 \psi \nabla r(\mathbf{x}). \quad (2.51)$$

According to [95] a minimizer of $\mathcal{J}$ is then given by

$$H \nabla \mathbf{y} = -\nabla \mathcal{J}. \quad (2.52)$$

The last step in the Gauss-Newton method is the search for an adequate step size for the motion update $\nabla \mathbf{y}$. The updated transformation should reduce the objective function's value compared to the previous guess. A simple way to do this is the Armijo line search. The idea is to add the previously determined search direction to the transformation with decreasing step size (it is halved in each step) until a reduction of the objective function's value occurs.

2.1.5.2 BFGS and l-BFGS

The BFGS (named after Broyden, Fletcher, Goldfarb, and Shanno) method is a quasi-Newton method, which means that the Hessian matrix is estimated at runtime. The Hessian matrix is approximated from function and gradient values. So only first derivatives are necessary for optimization [100].

The l in l-BFGS stands for *limited memory* [85, 95]. In l-BFGS the Hessian matrix is never built or stored explicitly. Vectors representing the Hessian matrix implicitly are used instead. These implementations are particularly appropriate for large problems.

2.1.5.3 Trust Region

In trust region optimization the objective function is treated only in a small (trusted) region where it is approximated by a quadratic model. This is a somehow reverse approach to line search methods. In trust region methods the step size is chosen first (in terms of the region size) and the search direction is computed subsequently. The proceeding in line search methods is vice versa: given a search direction, an appropriate step size is searched.

2.1.6 Results

Software phantoms like the XCAT phantom [123] are helpful to validate the proposed motion estimation algorithms as they provide morphological (anatomy) and functional (physiology) ground-truth information. The XCAT phantom is popular in emission tomography for human anatomy as it provides a model of the subject's anatomy and physiology. It is possible to include respiratory and cardiac motion resulting in 4D/5D images. In addition to the generation of PET images and attenuation maps, the XCAT phantom provides information about the ground-truth motion being used to generate the different motion states. This motion information can thus be used to quantitatively evaluate the estimated motion fields of the motion estimation approaches. Some results based on the ground-truth motion are given in Sect. 2.3.

For the evaluation in the following, a 5-by-5 dual gating data set is generated with the XCAT software phantom. The maximum diaphragm motion is set to 20 mm and the dual gates cover the whole range of the breathing and cardiac cycle. The total size of each volume is $175 \times 175 \times 47$ with a voxel size of 3.375 mm in each dimension, simulating the Siemens BiographTM Sensation 16 PET/CT scanner (Siemens Medical Solution).

The XCAT phantom data is trimmed to be comparable to real patient data in terms of tracer distribution, noise, and the scanning system, cf. [51]. To simulate nearly realistic PET images the following steps are performed:

1. Blurring with a Gaussian kernel (simulation of the PVE),
2. Forward projection,
3. Simulation of Poisson noise, and
4. Reconstruction.

The Gaussian kernel is chosen with a FWHM of 3.85 mm according to the simulated scanner [51]. The blurred images are forward projected into measurement space, where Poisson noise is simulated. The amount of noise is adjusted approximately equal to real patient data. In a final step, the sinograms are reconstructed with the EMRECON software [78]. It should be mentioned that the image acquisition process could alternatively be simulated using the simulation tool *GATE* [69].

We applied the proposed VAMPIRE approach, once with the SSD and once with the SAD distance measure, to the XCAT data. The results of the SSD and SAD VAMPIRE approach are visualized in Figs. 2.7 respectively 2.8 for one representative slice. The gate in maximum inspiration and systole, shown in each case in Fig. 2.7a, is chosen as the template image $\mathcal{T}$, as it is the most different gate compared to the reference image $\mathcal{R}$ in maximum expiration and diastole, shown in Fig. 2.7b. Additionally, the transformed image according to the estimated transformation, i.e., $\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}_{\text{SSD}})$ and $\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}_{\text{SAD}})$, and a vector plot are given in Fig. 2.7c, d. Some quantitative numbers based on the ground-truth transformation are further given at the end of Sect. 2.3 for the VAMPIRE results and the Mass-Preserving Optical Flow (MPOF) approach described in the next section.

2.2 Optical Flow

The motion between two images can not only be estimated by means of image registration. A common alternative is the so-called *optical flow*. The objective is indeed the same, but the problem formulation and implementation are different. Optical flow algorithms are used to estimate a dense flow field between two images/volumes.

In this section we will first derive the basic equation of optical flow. It will directly lead us to the early works on this topic: the local approach of Lucas and Kanade in Sect. 2.2.1 and the global approach of Horn and Schunck in Sect. 2.2.2. Advanced optical flow methods will be discussed in Sect. 2.2.3. A mass-preserving optical flow algorithm will be presented in Sect. 2.2.4. After a brief discussion of a multi-level implementation in Sect. 2.2.5 we finally present some results in Sect. 2.2.6.

The basic idea behind optical flow estimation is that the intensity of a voxel $\mathbf{x}^T = (x, y, z) \in \mathbb{R}^3$ remains constant over time. While changing its spatial position, the intensity of a moving point does not change. This constraint is called *brightness constancy constraint* and can be formulated for 3D volumes as

$$\mathcal{I}(x, y, z, t) = \mathcal{I}(x + u, y + v, z + w, t + 1), \quad (2.53)$$

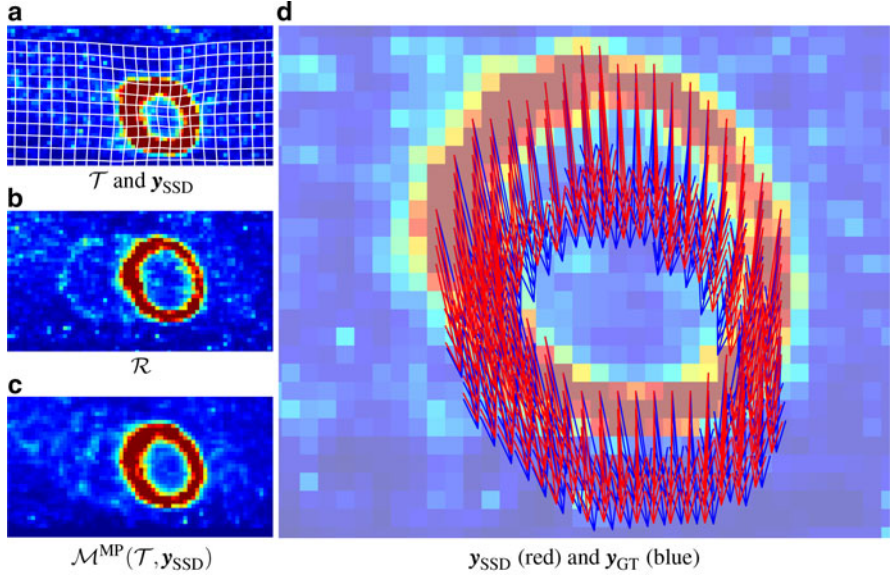


Fig. 2.7 SSD VAMPIRE: $\mathcal{T}$ in (a) is registered to $\mathcal{R}$ in (b) leading to the overlaid transformation grid in (a). The transformed template image is shown in (c). A magnification with a comparison of the estimated vectors (red) and the ground-truth vectors (blue) is shown in (d). (Colored figures are only available in the online version.)

where $\mathcal{I} : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$ denotes the *image* on the *domain* $\Omega \subset \mathbb{R}^3$ and $t \in \mathbb{R}$ the *time*. The *flow* or *displacement* is given by $\mathbf{u}^T = (u, v, w, 1) \in \mathbb{R}^4$ for the three spatial dimensions and one time step.

A Taylor linearization in the point $\mathbf{x}$ is applied to $\mathcal{I}(x+u, y+v, z+w, t+1)$ which is valid for small displacements, resulting in

$$\begin{aligned}
 \mathcal{I}(x+u, y+v, z+w, t+1) &= \mathcal{I}(x, y, z, t) \\
 &+ \frac{\partial}{\partial x} \mathcal{I}(x, y, z, t) \cdot u \\
 &+ \frac{\partial}{\partial y} \mathcal{I}(x, y, z, t) \cdot v \\
 &+ \frac{\partial}{\partial z} \mathcal{I}(x, y, z, t) \cdot w \\
 &+ \frac{\partial}{\partial t} \mathcal{I}(x, y, z, t) \\
 &+ \text{higher order terms.}
 \end{aligned} \tag{2.54}$$

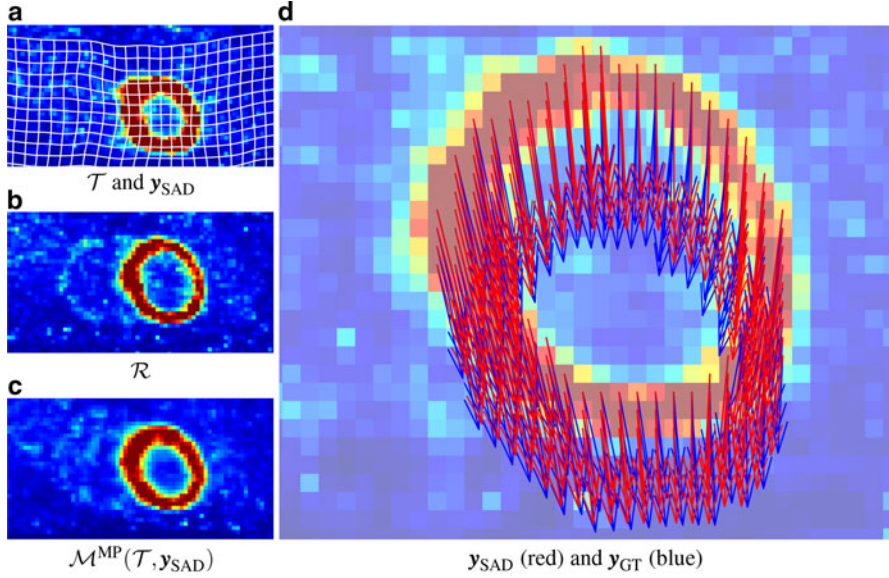


Fig. 2.8 SAD VAMPIRE: $\mathcal{T}$ in (a) is registered to $\mathcal{R}$ in (b) leading to the overlaid transformation grid in (a). The transformed template image is shown in (c). A magnification with a comparison of the estimated vectors (*red*) and the ground-truth vectors (*blue*) is shown in (d). (Colored figures are only available in the online version.)

Inserting Eq. (2.54) into (2.53) leads to

$$\begin{aligned}
 0 &= \frac{\partial}{\partial x} \mathcal{I}(x, y, z, t) \cdot u + \frac{\partial}{\partial y} \mathcal{I}(x, y, z, t) \cdot v + \frac{\partial}{\partial z} \mathcal{I}(x, y, z, t) \cdot w + \frac{\partial}{\partial t} \mathcal{I}(x, y, z, t) \\
 &= \mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t \\
 &= \mathbf{u}^T \nabla \mathcal{I}.
 \end{aligned} \tag{2.55}$$

For the sake of notation simplicity we will use the shorthand $\mathcal{I}$ for $\mathcal{I}(\mathbf{x}, t)$ in the following. The terms $\mathcal{I}_x$, $\mathcal{I}_y$, and $\mathcal{I}_z$ above represent the derivatives of $\mathcal{I}$. Similarly, $\mathbf{u}(\mathbf{x})$ is shortened to $\mathbf{u}$. Equation (2.55) immediately gives us the data term for standard optical flow algorithms.

$$\mathbf{D}(\mathcal{I}, \mathbf{u})^2 := (\mathbf{u}^T \nabla \mathcal{I})^2 = (\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t)^2. \tag{2.56}$$

As we now have an equation in three variables (the motion along the three directions) additional conditions have to be formulated to make the problem solvable. Different conditions have been proposed to this end.

2.2.1 Lucas-Kanade Algorithm

As mentioned above, the flow vector $\mathbf{u}^T = (u, v, w, 1)$ with three unknowns has to be estimated at each spatial position $\mathbf{x}^T = (x, y, z)$. To obtain an equation system so that the problem can be solved, Lucas and Kanade [86] made the assumption that the motion field is constant in a small area around $\mathbf{x}$. With Eq. (2.56) we have

$$\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t = 0 \quad (2.57)$$

or

$$\mathbf{u}^T \nabla \mathcal{I} = 0. \quad (2.58)$$

Assuming a constant flow $\mathbf{u}$ in a small window of size $m \times m \times m$, $m > 1$, centered at the voxel position $\mathbf{x}$ and denoting the neighbors as (x_i, y_i, z_i) , $i = 1, \dots, n$, we get a set of equations

$$\begin{aligned} \mathcal{I}_{x_1} u + \mathcal{I}_{y_1} v + \mathcal{I}_{z_1} w + \mathcal{I}_{t_1} &= 0 \\ \mathcal{I}_{x_2} u + \mathcal{I}_{y_2} v + \mathcal{I}_{z_2} w + \mathcal{I}_{t_2} &= 0 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots & \\ \mathcal{I}_{x_n} u + \mathcal{I}_{y_n} v + \mathcal{I}_{z_n} w + \mathcal{I}_{t_n} &= 0 \end{aligned} \quad (2.59)$$

where $\mathcal{I}_{x_i}$, $\mathcal{I}_{y_i}$, $\mathcal{I}_{z_i}$, and $\mathcal{I}_{t_i}$, $i = 1, \dots, n$, represent the shortterm for $\mathcal{I}_x(x_i, y_i, z_i, t)$, $\mathcal{I}_y(x_i, y_i, z_i, t)$, $\mathcal{I}_z(x_i, y_i, z_i, t)$, and $\mathcal{I}_t(x_i, y_i, z_i, t)$, respectively. Usually the central voxel is given more weight to suppress noise and thus a Gaussian weighting function is applied to obtain a modified data term [135]

$$\mathbf{D}_{\text{LK}}(\mathcal{I}, \mathbf{u})^2 := \mathbf{u}^T (G * (\nabla \mathcal{I} \nabla \mathcal{I}^T)) \mathbf{u}, \quad (2.60)$$

where G is a Gaussian smoothing kernel which is convolved component-wisely with the image gradients $\nabla \mathcal{I} \nabla \mathcal{I}^T$ of the neighboring voxels. This minimization problem is solved in a standard least-squares manner.

The optical flow estimated with the Lucas-Kanade algorithm fades out quickly with increasing distance from the motion boundaries. The method is comparatively robust in presence of noise [9, 19].

Algorithm 2.2 Lucas-Kanade optical flow method

Input: Images $\mathcal{I}(\mathbf{x}, t)$ and $\mathcal{I}(\mathbf{x}, t + 1)$

Output: Motion estimate $\mathbf{u}$ (for all voxels)

estimate $\nabla \mathcal{I}$ of each voxel for $\mathcal{I}(\mathbf{x}, t)$

for each voxel $\mathbf{x}$ **do**

compute $\mathbf{u}(\mathbf{x})$ by least-squares based minimization of Eq. (2.60)

end for

2.2.2 Horn-Schunck Algorithm

Another option to make the ill-posed problem solvable is to assume an additional condition of smoothness in flow. This condition is used in the seminal work of Horn and Schunck [64]. Globally smooth motion field means that it varies only little between neighbouring voxels. Thus, they apply a homogeneous diffusion regularization term to the optical flow functional in Eq. (2.56), cf. Sect. 2.1.2.1. In this way a new energy functional consisting of a data term and a smoothness term is built which is to be minimized to get the optical flow

$$\mathcal{J}_{\text{HS}}(\mathbf{u}) = \int_{\Omega} \mathbf{D}(\mathcal{I}, \mathbf{u})^2 d\mathbf{x} + \alpha \int_{\Omega} \mathbf{S}(\mathbf{u}) d\mathbf{x}, \quad (2.61)$$

where $\mathbf{S}(\mathbf{u})$ is the diffusion regularization energy

$$\mathbf{S}(\mathbf{u}) = |\nabla u|^2 + |\nabla v|^2 + |\nabla w|^2. \quad (2.62)$$

The parameter α is a regularization constant. Larger values of α lead to a smoother flow field.

The functional in Eq. (2.61) can be minimized by solving the corresponding Euler-Lagrange equations

$$\begin{aligned} \Delta u - \frac{1}{\alpha^2} \mathcal{I}_x(\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t) &= 0 \\ \Delta v - \frac{1}{\alpha^2} \mathcal{I}_y(\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t) &= 0 \\ \Delta w - \frac{1}{\alpha^2} \mathcal{I}_z(\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}_t) &= 0 \end{aligned} \quad (2.63)$$

where Δ denotes the Laplace operator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

Using the standard approximation of the Laplace operator in image processing: $\Delta f = \bar{f} - f$, $f = u, v, w$, where $\bar{f}$ is the average of f in the neighborhood of the current voxel position, we arrive at the equation system

$$\begin{pmatrix} \alpha^2 + \mathcal{I}_x^2 & \mathcal{I}_x \mathcal{I}_y & \mathcal{I}_x \mathcal{I}_z \\ \mathcal{I}_x \mathcal{I}_y & \alpha^2 + \mathcal{I}_y^2 & \mathcal{I}_y \mathcal{I}_z \\ \mathcal{I}_x \mathcal{I}_z & \mathcal{I}_y \mathcal{I}_z & \alpha^2 + \mathcal{I}_z^2 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \alpha^2 \bar{u} - \mathcal{I}_x \mathcal{I}_t \\ \alpha^2 \bar{v} - \mathcal{I}_y \mathcal{I}_t \\ \alpha^2 \bar{w} - \mathcal{I}_z \mathcal{I}_t \end{pmatrix}.$$

Solving this system leads to the following iterative scheme

$$\begin{aligned}
 u^{k+1} &= \bar{u}^k - I_x \cdot \frac{I_x \bar{u}^k + I_y \bar{v}^k + I_z \bar{w}^k + I_t}{\alpha^2 + I_x^2 + I_y^2 + I_z^2} \\
 v^{k+1} &= \bar{v}^k - I_y \cdot \frac{I_x \bar{u}^k + I_y \bar{v}^k + I_z \bar{w}^k + I_t}{\alpha^2 + I_x^2 + I_y^2 + I_z^2} \\
 w^{k+1} &= \bar{w}^k - I_z \cdot \frac{I_x \bar{u}^k + I_y \bar{v}^k + I_z \bar{w}^k + I_t}{\alpha^2 + I_x^2 + I_y^2 + I_z^2}
 \end{aligned} \tag{2.64}$$

where the superscript k denotes the iteration number.

An advantage of the Horn-Schunck algorithm is that it yields a high density of flow vectors, i.e., the flow information missing in inner parts of homogeneous objects is *filled in* from the motion boundaries. However, it is more sensitive to noise than local methods [19, 93].

Algorithm 2.3 Horn-Schunck optical flow method

Input: Images $\mathcal{I}(\mathbf{x}, t)$ and $\mathcal{I}(\mathbf{x}, t + 1)$

Output: Motion estimate $\mathbf{u}$ (for all voxels)

compute $\nabla \mathcal{I}$ for each voxel $\mathbf{x}$ of $\mathcal{I}(\mathbf{x}, t)$

initialize $\mathbf{u}$ for all voxels, e.g., null vector

while change in $\mathbf{u} > \text{threshold}$ **do**

for each voxel $\mathbf{x}$ **do**

 compute $\mathbf{u}(\mathbf{x})$ as given in Eqs. (2.64)

end for

end while

2.2.3 Advanced Optical Flow Algorithms

In the literature exists a substantial amount of recent work that improve the basic local and global optical flow approaches in various ways. As stated before, local methods are often more robust in the case of noise, while global techniques yield dense flow field. Bruhn et al. [19] propose a method that combines important advantages of local and global approaches and thus yields dense flow fields that are robust against noise. By inserting Eq. (2.60) into (2.61) we get

$$\mathcal{J}_B(\mathbf{u}) = \int_{\Omega} \mathbf{D}_{LK}(\mathcal{I}, \mathbf{u})^2 d\mathbf{x} + \alpha \int_{\Omega} \mathbf{S}(\mathbf{u}) d\mathbf{x}. \tag{2.65}$$

In addition, they add an L_1 -like penalizing function ψ to the individual terms in Eq. (2.65)

$$\mathcal{J}_B(\mathbf{u}) = \int_{\Omega} \psi_1(\mathbf{D}_{LK}(\mathcal{I}, \mathbf{u})^2) d\mathbf{x} + \alpha \int_{\Omega} \psi_2(\mathbf{S}(\mathbf{u})) d\mathbf{x}. \tag{2.66}$$

A possible choice for ψ functions is given in Eq.(2.6), and ψ_1 and ψ_2 vary only in the choice of the parameter β . From a statistical viewpoint this can be regarded as applying methods from robust statistics where outliers are penalized less severely than in quadratic approaches. In general, such methods give better results at locations with flow discontinuities.

In a variety of applications the fundamental brightness constancy constraint in Eq.(2.53) may not apply. In the literature several alternatives have thus been proposed. One possibility is to demand constancy of image gradient or gradient magnitude. Other options include higher-order derivatives like the Hessian, Laplacian, or the determinant of the Hessian, photometric invariants, and texture features. Multiple features can also be combined towards complex constancy constraints (see [135] for an overview). In addition to the general development in computer vision, special attention has also been given to novel approaches motivated by medical imaging, e.g., [126].

While the accuracy of optical flow algorithms has improved steadily, the typical formulation has changed little since the seminal work of Horn and Schunck [64]. By means of a thorough analysis Sun et al. [125] attempt to uncover the “secrets” that have made recent advances possible. They analyze how the objective function, the optimization method, and modern implementation practices influence the accuracy. It is found that classical optical flow formulations perform surprisingly well when combined with modern optimization and implementation techniques (in particular, median filtering of intermediate flow fields during optimization).

2.2.4 Mass-Preserving Optical Flow

The aforementioned optical flow methods are all based on the brightness constancy constraint given in Eq.(2.53). However, this condition is not fulfilled in some imaging modalities, e.g., PET, due to partial volume effects, cf. Fig. 1.11. For such cases an optical flow method independent of this constraint is required.

As the total brightness of an organ can be considered constant for the duration of the image formation process, the continuity equation for the conservation of the total mass can be used as an alternative model, representing a Mass Preserving Optical Flow (MPOF) method [33]. The continuity equation for mass conservation is given as [10, 31]

$$\nabla \cdot (\mathcal{I}\mathbf{u}) = 0, \quad (2.67)$$

where $\nabla \cdot$ is the divergence. Again, a functional can be defined based on Eq.(2.67) and minimized to get the optical flow estimate

$$\mathcal{J}(\mathbf{u}) := \int_{\Omega} (\nabla \cdot (\mathcal{I}\mathbf{u}))^2 d\mathbf{x} \stackrel{!}{=} \min. \quad (2.68)$$

Definition 2.12 ($\mathbf{D}_{\text{mp}}$ – Data term for mass-preserving optical flow). For two images $\mathcal{I}(\mathbf{x}, t)$ and $\mathcal{I}(\mathbf{x}, t + 1)$ and a deformation (or velocity) field $\mathbf{u}$ the data term for mass-preserving optical flow is defined as

$$\begin{aligned}\mathbf{D}_{\text{mp}}(\mathcal{I}, \mathbf{u}) &= \nabla \cdot (\mathcal{I}\mathbf{u}) \\ &= \mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_z w + \mathcal{I}(u_x + v_y + w_z) + \mathcal{I}_t .\end{aligned}\quad (2.69)$$

As usual a regularization term is added to the functional, e.g., the diffusion regularization term of the Horn-Shunck method in Eq. (2.62). The mass-preserving optical flow functional to be minimized is therefore

$$\mathcal{J}_{\text{mp}}(\mathbf{u}) = \int_{\Omega} \mathbf{D}_{\text{mp}}(\mathcal{I}, \mathbf{u})^2 d\mathbf{x} + \alpha \int_{\Omega} \mathbf{S}(\mathbf{u}) d\mathbf{x} , \quad (2.70)$$

where α is the regularization parameter. The corresponding Euler-Lagrange equations are given by

$$\begin{aligned}0 &= \mathbf{D}_x \mathcal{I} + \alpha \Delta u , \\ 0 &= \mathbf{D}_y \mathcal{I} + \alpha \Delta v , \\ 0 &= \mathbf{D}_z \mathcal{I} + \alpha \Delta w ,\end{aligned}\quad (2.71)$$

where $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_z$ are the derivatives of $\mathbf{D}_{\text{mp}}$ in the corresponding directions. Solving this system leads to the following iterative scheme

$$\begin{aligned}u^{k+1} &= u^k + \mathbf{D}_x \mathcal{I} + \alpha \Delta u , \\ v^{k+1} &= v^k + \mathbf{D}_y \mathcal{I} + \alpha \Delta v , \\ w^{k+1} &= w^k + \mathbf{D}_z \mathcal{I} + \alpha \Delta w .\end{aligned}\quad (2.72)$$

Algorithm 2.4 Mass-preserving optical flow method

Input: Images $\mathcal{I}(\mathbf{x}, t)$ and $\mathcal{I}(\mathbf{x}, t + 1)$

Output: Motion estimate $\mathbf{u}$ (for all voxels)

initialize $\mathbf{u}$ for all voxels, e.g., null vector

while change in $\mathbf{u} > \text{threshold}$ **do**

for each voxel $\mathbf{x}$ **do**

 compute $\mathbf{u}(\mathbf{x})$ as given in Eqs. (2.72)

end for

end while

As shown above in Sect. 2.2.3, the motion discontinuities can be better modeled if an L_1 -like penalization term is applied. Using the aforementioned function ψ in Eq. (2.6) we get

$$\mathcal{J}_{\text{mpn}}(\mathbf{u}) = \int_{\Omega} \psi_1(\mathbf{D}_{\text{mp}}(\mathcal{I}, \mathbf{u})^2) d\mathbf{x} + \alpha \int_{\Omega} \psi_2(\mathbf{S}(\mathbf{u})) d\mathbf{x}, \quad (2.73)$$

where ψ_1 and ψ_2 vary only in the choice of the parameter β in Eq. (2.6). The corresponding Euler-Lagrange equations can be found in [37]. The optical flow can be estimated by using an iterative scheme similar to that given above in Algorithm 2.4 for the quadratic case.

2.2.5 Multi-level Optimization

As in the case of image registration, the proposed optical flow methods should be applied in a multi-level framework to avoid local minima during optimization, cf. Sect. 2.1.5. The multi-level framework becomes particularly important for optical flow methods as it allows the treatment of large motion with the Taylor approximation, which only holds for small displacements. The motion is thus calculated on a coarse level, where it is still small in terms of voxel size. Starting at the lowest level we get a rough estimate of the flow vectors. These vectors are prolonged (interpolated) to the next higher level. On that level the original template image is interpolated with the current transformation. Again, the flow is estimated, but on the finer resolution, to figure out the residual motion. This process is repeated until the maximal resolution is reached.

As the motion vectors are estimated with the mass-preserving motion model, the transformation of the template image needs to be mass-preserving as well. The approximation in Eq. (2.67) is used in [33] for the mass-preserving transformation of the template image. For this approximation the time derivative in the equation is calculated as $\mathcal{I}_t = \mathcal{T} - \mathcal{R}$. As the optical flow $\mathbf{u}$ in Eq. (2.67) is known after motion estimation the motion corrected image can be calculated according to

$$\mathcal{T}^{\text{MC}}(\mathbf{x}) = \mathcal{T}(\mathbf{x}) + \nabla \cdot (\mathcal{T}(\mathbf{x})\mathbf{u}(\mathbf{x})), \quad \forall \mathbf{x} \in \Omega. \quad (2.74)$$

This equation is accurate for small displacements. However, this assumption does not hold for potentially large non-linear cardiac displacements. A better, non-approximated, method is to use the mass-preserving transformation model in Definition 2.11 based on the exact computation of the Jacobian determinant. Accordingly, the motion corrected image is calculated as

$$\mathcal{T}^{\text{MC}}(\mathbf{x}) = \mathcal{T}(\mathbf{x} + \mathbf{u}(\mathbf{x})) \cdot \det(\nabla(\mathbf{x} + \mathbf{u}(\mathbf{x}))), \quad \forall \mathbf{x} \in \Omega. \quad (2.75)$$

This image is then used as the template image for the next level. The proceeding is summarized in Algorithm 2.5.

Algorithm 2.5 Mass-preserving multi-level optical flow

Input: Template image $\mathcal{T}$ ($= \mathcal{I}(\mathbf{x}, t)$) and reference image $\mathcal{R}$ ($= \mathcal{I}(\mathbf{x}, t + 1)$)
Output: Motion estimate $\mathbf{u}$ (for all voxels)
 initialize $\mathbf{u}_{\min\text{Level}}$ for all voxels, e.g., null vector
for level = minLevel $\rightarrow$ maxLevel **do**
 $\mathcal{R}_{\text{level}} \leftarrow \mathcal{R}$ // down-sampling
 $\mathcal{T}_{\text{level}} \leftarrow \mathbf{u}$ // apply motion with mass-preservation $\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{x} + \mathbf{u}(\mathbf{x}))$ & down-sampling
 // NOTE: the exact computation of the Jacobian is used
 $\mathbf{u}_{\text{level}} \leftarrow \text{argmin} \mathcal{J}_{\text{mp}}(\mathbf{u}_{\text{level}})$ // update motion by minimizing the functional
 if level < maxLevel **then**
 $\mathbf{u}_{\text{level}+1} \leftarrow \mathbf{u}_{\text{level}}$ // prolongate motion to next level
 end if
end for
 $\mathbf{u} \leftarrow \mathbf{u}_{\max\text{Level}}$ // output final motion estimate

2.2.6 Results

The MPOF approach is applied to the same XCAT software phantom data as described in Sect. 2.1.6 in connection with the VAMPIRE approach. The results of MPOF are visualized in Fig. 2.9 for the same slice used in Figs. 2.7 and 2.8. The gate in maximum inspiration and systole is chosen here as the template image $\mathcal{T}$ (shown in Fig. 2.9a), as it is the most different gate compared to the reference image $\mathcal{R}$ in maximum expiration and diastole (shown in Fig. 2.9b). Additionally, the transformed template image according to the estimated transformation, i.e., $\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}_{\text{OF}})$, and a vector plot are given in Fig. 2.9c, d. Some quantitative numbers based on the ground-truth transformation will be given at the end of following Sect. 2.3.

2.3 Comparison and Evaluation

In the previous sections we have seen two different approaches to mass-preserving motion estimation. The variant based on image registration is given in Sect. 2.1.4 and the optical flow variant is derived in Sect. 2.2.4. Technically, optical flow and image registration approaches provide different approaches to motion estimation. On the one hand, the image registration framework simplifies the incorporation of physically meaningful regularization like (non-linear) hyperelasticity. On the other hand, optical flow has also several advantages as it is, e.g., computationally efficient due to the linear Taylor approximation. Both approaches have in common that they

are generally well understood. That means that they are versatile with a variety of different possible configurations (e.g., intensity or gradient distance measures).

One of the main differences between image registration and optical flow, as presented in this book, is the numerical realization. While image registration is based on a discretize-then-optimize strategy, optical flow follows an optimize-then-discretize strategy. The optimization in the case of optical flow includes a Taylor approximation whereas image registration relies on the exact computation of all its components. The Taylor approximation makes, e.g., the incorporation of non-linear hyperelastic regularization into the optical flow functional in Eq. (2.70) challenging. Hyperelasticity would thus transform into something like linear elasticity, which is only suitable for small deformations.

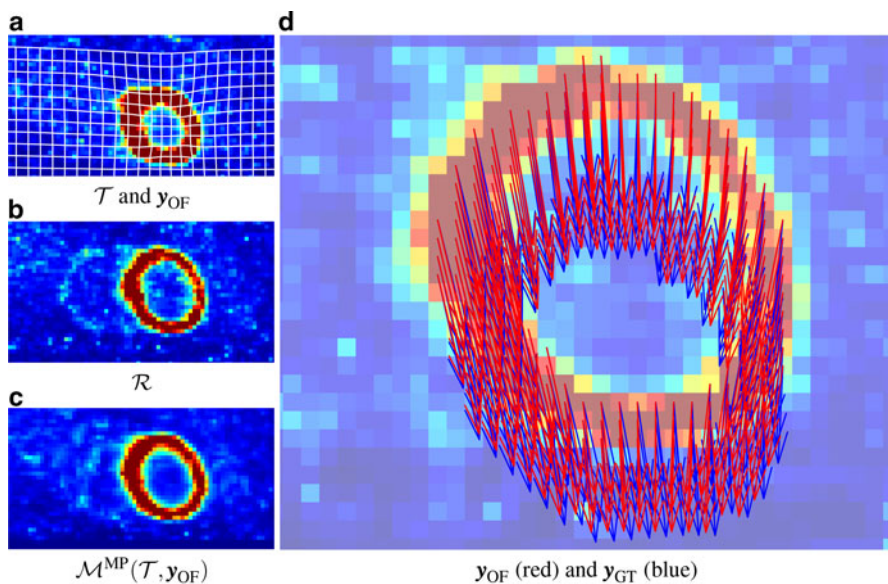


Fig. 2.9 MPOF: $\mathcal{T}$ in (a) is registered to $\mathcal{R}$ in (b) leading to the overlaid transformation grid in (a). The transformed template image is shown in (c). A magnification with a comparison of the estimated vectors (red) and the ground-truth vectors (blue) is shown in (d). (Colored figures are only available in the online version.)

2.3.1 Comparison

Despite the fundamental differences, the basic idea of MPOF and VAMPIRE is the same. Hence, we will derive the mass-preserving optical flow equations from the VAMPIRE model in the following. For this purpose, let us recall the concept of mass-preservation as introduced in Sect. 2.1.4. The idea of mass-preserving image registration is the preservation of the total intensity of an image due to a

transformation model $\mathcal{M}$

$$\int_{\mathbf{y}(\Omega)} \mathcal{T}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathcal{M}(\mathcal{T}, \mathbf{y}(\mathbf{x})) d\mathbf{x}, \quad (2.76)$$

which is exactly given by the integration by substitution theorem for multiple variables

$$\int_{\mathbf{y}(\Omega)} \mathcal{T}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathcal{T}(\mathbf{y}(\mathbf{x})) \det(\nabla \mathbf{y}(\mathbf{x})) d\mathbf{x}. \quad (2.77)$$

This is the same equation that was used for the derivation of VAMPIRE in Eq. (2.45). Since optical flow is generally speaking a Taylor approximation of image registration, the exact computation of the Jacobian determinant in the VAMPIRE approach is approximated for MPOF as well. The transformation model for mass-preserving image registration is given by

$$\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}) = (\mathcal{T} \circ \mathbf{y}) \cdot \det(\nabla \mathbf{y}). \quad (2.78)$$

We will restrict the time variable t of $\mathcal{I}$, introduced in Eq. (2.53), to the interval $[0, 1]$ to make the transition from image registration to optical flow

$$\begin{aligned} \mathcal{I}(\mathbf{x}, t) : \mathbb{R}^3 \times [0, 1] &\rightarrow \mathbb{R} \text{ with} \\ \mathcal{I}(\cdot, 0) &= \mathcal{T} \text{ and} \\ \mathcal{I}(\cdot, 1) &= \mathcal{R}. \end{aligned} \quad (2.79)$$

The time variable is also introduced for the corresponding transformations

$$\mathbf{y}(\mathbf{x}, t) : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3. \quad (2.80)$$

The objective is to find the set of transformations that align the image data to the reference frame

$$\mathcal{I}(\mathbf{y}(\cdot, t), t) \stackrel{!}{=} \mathcal{R} = \mathcal{I}(\mathbf{x}, 1), \forall \mathbf{x} \in [0, 1]. \quad (2.81)$$

Note that $\mathbf{y}(\cdot, 1)$ is the identity transformation.

The equation known from image registration

$$\int_{\Omega} \mathcal{T}(\mathbf{y}(\mathbf{x})) \cdot \det(\nabla \mathbf{y}(\mathbf{x})) d\mathbf{x} \quad (2.82)$$

at time t now reads

$$\int_{\Omega} \mathcal{I}(\mathbf{y}(\mathbf{x}, t), t) \cdot \det(\nabla \mathbf{y}(\mathbf{x}, t)) d\mathbf{x}. \quad (2.83)$$

According to [10] we compute the temporal derivative

$$\int_{\Omega} \frac{\delta}{\delta t} (\mathcal{I}(\mathbf{y}(\mathbf{x}, t), t) \cdot \det(\nabla \mathbf{y}(\mathbf{x}, t))) d\mathbf{x}, \quad (2.84)$$

which yields (approximately)

$$\int_{\Omega} \left(\nabla \mathcal{I}(\mathbf{y}(\mathbf{x}, t), t) \cdot \frac{\delta}{\delta t} (\mathbf{y}(\mathbf{x}, t)) + \mathcal{I}(\mathbf{y}(\mathbf{x}, t), t) \cdot \left(\nabla \cdot \frac{\delta}{\delta t} (\mathbf{y}(\mathbf{x}, t)) \right) \right) \det(\nabla \mathbf{y}(\mathbf{x}, t)) d\mathbf{x}, \quad (2.85)$$

by using the chain rule and the approximation

$$\frac{\delta}{\delta t} \det(\nabla \mathbf{y}(\mathbf{x}, t)) \approx \left(\nabla \cdot \frac{\delta}{\delta t} (\mathbf{y}(\mathbf{x}, t)) \right) \det(\nabla \mathbf{y}(\mathbf{x}, t)). \quad (2.86)$$

Like in standard optical flow we set the integral in Eq. (2.85) equal to zero and get

$$0 = (\nabla \mathcal{I})^T \mathbf{u} + \mathcal{I}_t + \mathcal{I} \nabla \cdot \mathbf{u} \quad (2.87)$$

$$\iff 0 = \nabla \cdot (\mathcal{I} \mathbf{u}) + \mathcal{I}_t, \quad (2.88)$$

where $\mathbf{u}^T = \left(\frac{\delta}{\delta t} \mathbf{y}_x(\mathbf{x}, t), \frac{\delta}{\delta t} \mathbf{y}_y(\mathbf{x}, t), \frac{\delta}{\delta t} \mathbf{y}_z(\mathbf{x}, t) \right) =: (u, v, w)$ is the velocity. As we are only dealing with the registration of two images, $\mathcal{I}$ consists only of two images ($\mathcal{T}$ and $\mathcal{R}$) and $\mathbf{u}$ is reduced to one velocity field connecting these two images. This finally leads to the data term for mass-preserving optical flow that we know from Definition 2.12.

One important assumption of both, optical flow and image registration, is a positive Jacobian determinant, i.e., $\det(\nabla \mathbf{y}) > 0$. As the exact values of the Jacobian determinant are used for image registration, this can be ensured by simply disallowing negative values. However, optical flow requires more effort to guarantee diffeomorphic transformations as, e.g., proposed by Yang et al. [137]. The idea is to restrict the motion update to a vector less than 0.4 voxel which prevents foldings and preserves diffeomorphic motion estimates. This, however, might reduce the convergence speed of the overall algorithm. In practice we even observe a similar run-time compared to image registration where in each Gauss-Newton iteration a large and complicated linear system needs to be solved. Another approach by Vercauteren et al. [132] uses an approximation of the exponential of the deformation to ensure diffeomorphisms. As the proposed implementation is only based on an approximation of the exponential of the deformation it finally gives no real guarantee for $\det(\nabla \mathbf{y}) > 0$ [137].

In conclusion, mass-preserving optical flow and mass-preserving image registration are based on the same assumption, i.e., preservation of the overall radioactivity. Both approaches just differ in their technical realization. On experimental and clinical data we could show that both approaches yield satisfying (i.e., realistic)

results with comparable accuracy for PET motion correction, see [37, 51] and the evaluation at the end of this section. It is thus difficult to give preference to one particular method in practice for all applications. The discussion above is intended to provide some orientation in the reader's decision-making process for one method. In order to give more assistance we perform a quantitative evaluation of both methods in the following.

2.3.2 Evaluation

The following evaluation is carried out on *software phantom data* and *patient data*. While the analysis based on the software phantom data is performed to quantitatively compare the image registration and optical flow based approaches, the patient study shows the general clinical applicability of the proposed methods.

2.3.2.1 Software Phantom Data

The images of the software phantom evaluation are based on the results presented in Fig. 2.8 for VAMPIRE and Fig. 2.9 for MPOF. More information about the XCAT software phantom data and in particular the image generation process can be found in Sect. 2.1.6.

Results

For a visual comparison we show some results of the proposed methods for motion estimation in Fig. 2.10. Cardiac slices are shown for a single gate in Fig. 2.10a for the reference gate $\mathcal{R}$, showing a high noise level. The combination of all available gates, representing a reconstruction without any motion correction, is given in Fig. 2.10b. This image features severe motion induced artifacts compared to the reference gate $\mathcal{R}$, thus demanding for motion correction techniques. However, the SNR is significantly higher compared to the single gate in Fig. 2.10a. The combination of all single gates using the ground-truth motion information, provided by XCAT, can be found in Fig. 2.10c. Accordingly, this image represents the best possible, and thus desired, image quality as it combines the high SNR of Fig. 2.10b and the reduced motion of Fig. 2.10a. The resulting images based on estimated motion can be found in Fig. 2.10d for the image registration based SAD VAMPIRE approach and in Fig. 2.10e for the MPOF approach. The image quality of both approaches is comparable with the image based on the ground-truth motion in Fig. 2.10c.

Validation

The results which are shown in Fig. 2.10 and the underlying motion estimates are quantitatively evaluated. The results are given in Table 2.1. The endpoint error e is used to determine the Euclidean distance of the estimated and the ground-truth deformation. For two transformations $\mathbf{y}_1, \mathbf{y}_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and a non-empty set $\Omega \subset \mathbb{R}^3$ the endpoint error is defined as

$$e(\mathbf{y}_1, \mathbf{y}_2) := \frac{1}{|\Omega|} \int_{\Omega} |\mathbf{y}_1(\mathbf{x}) - \mathbf{y}_2(\mathbf{x})| d\mathbf{x}, \quad (2.89)$$

where $|\Omega| := \int_{\Omega} d\mathbf{x}$ and for vectors $|\cdot|$ is the Euclidean norm. The values given in the table are averaged values over all gates. The maximum endpoint error is considered to analyze the worst case of the transformation mismatch and is defined as

$$e^{\max}(\mathbf{y}_1, \mathbf{y}_2) := \max_{\mathbf{x} \in \Omega} |\mathbf{y}_1(\mathbf{x}) - \mathbf{y}_2(\mathbf{x})|. \quad (2.90)$$

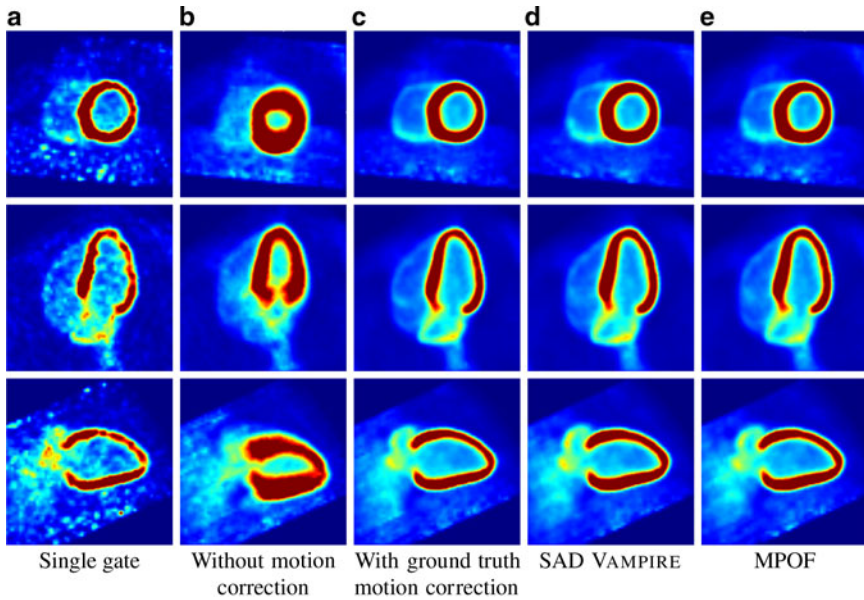


Fig. 2.10 The cardiac planes (from *top to bottom*: SA, HLA, and VLA) are shown for the single reference gate $\mathcal{R}$ in (a) out of a 5×5 dual gating. The combination of all gates without any motion correction can be seen in (b). Motion artifacts in terms of blurring can be clearly seen. The motion corrected image based on the ground-truth motion is shown in (c). The image shows a reduced blurring, e.g., in the blood pool. The results of the image registration based SAD VAMPIRE approach and the optical flow based MPOF approach are shown in (d) and (e). The image quality of the proposed VAMPIRE and MPOF approach is comparable to the best possible result based on ground-truth motion and considerably better than the resulting image without any motion correction. All images are corrected for attenuation

The values given in the table denote the maximum error over all dual gates. The transformation's regularity is analyzed via the range of the Jacobian map

$$\left[\min_{\mathbf{x} \in \Omega} \{\det(\nabla \mathbf{y}(\mathbf{x}))\} , \max_{\mathbf{x} \in \Omega} \{\det(\nabla \mathbf{y}(\mathbf{x}))\} \right]. \quad (2.91)$$

Orientation preserving and diffeomorphic transformations, i.e., transformations free of foldings, can be assured by controlling the maximum decrease or increase of a voxel's volume or intensity. Finite values greater than 0 represent appropriate values for the Jacobian map. A value of 1 indicates no volume change.

The Normalized Cross-Correlation (NCC) measure is suitable for general intensity-based validation. For two images $\mathcal{T} : \Omega \rightarrow \mathbb{R}$ and $\mathcal{R} : \Omega \rightarrow \mathbb{R}$ on a domain $\Omega \subset \mathbb{R}^3$ the NCC is defined as

$$\text{NCC}(\mathcal{T}, \mathcal{R}) := \frac{\int_{\Omega} \mathcal{T}_{\mu}(\mathbf{x}) \mathcal{R}_{\mu}(\mathbf{x}) d\mathbf{x}}{\sqrt{\int_{\Omega} \mathcal{T}_{\mu}(\mathbf{x})^2 d\mathbf{x}} \sqrt{\int_{\Omega} \mathcal{R}_{\mu}(\mathbf{x})^2 d\mathbf{x}}}, \quad (2.92)$$

where $\mathcal{T}_{\mu} = \mathcal{T} - \mu(\mathcal{T})$ and $\mathcal{R}_{\mu} = \mathcal{R} - \mu(\mathcal{R})$ are the unbiased versions of $\mathcal{T}$ and $\mathcal{R}$. For an image $\mathcal{I}$, $\mu(\mathcal{I})$ is the expected value. It should be emphasized that the validation based on image intensities can only give certain hints on the accuracy of the registration process. Indeed, it was shown in [115] that even inaccurate registration results can produce good scores on intensity-based measures. Accordingly, the combination with ground-truth transformation-based measures discussed before is mandatory. In Table 2.1 $\text{NCC}(\mathcal{T}, \mathcal{R})$ represents the similarity before motion correction and $\text{NCC}(\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}), \mathcal{R})$ after motion correction.

The values in Table 2.1 indicate that all proposed methods have a comparable accuracy. The average endpoint errors e and the maximum endpoint errors $e^{\max}$ are very close to each other and reveal a sub-voxel accuracy for each method, given a voxel size of 3.375 mm. The Jacobian range demonstrates that all results are diffeomorphic and thus free of foldings. The correlation values also increase after motion correction. The reason why the values for $\text{NCC}(\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}), \mathcal{R})$ after motion correction are not 1 is that the random noise in the image prevents a perfect match. With other words, a too high correlation would indicate an unwanted match of the random noise in the images during motion estimation.

2.3.2.2 Patient Data

In Fig. 1.1 we have already seen the effect of motion correction on real patient data as a motivation. The shown data is part of a set of 21 patients (19 male, 2 female; between 37 and 76 years old) with known coronary artery disease. One hour after injection of ^{18}F -FDG (4 MBq/kg) a 20 min list mode scan was acquired for evaluation of myocardial viability prior to revascularization. All patients underwent a hyperinsulinemic euglycemic clamp technique before and during the scan to enhance FDG uptake in the heart. In addition, β -blockers were administered to all

Table 2.1 Evaluation of SSD VAMPIRE, SAD VAMPIRE, and MPOF based on common validation criteria. The definitions of the validation criteria are given in this section. The numbers for $e(\mathbf{y}, \mathbf{y}_{GT})$ and NCC are average values over all 25 dual gates. The range of the Jacobian map represents the global minimum and maximum for all images. The maximum end-point error is also the global maximum error for all images

	SSD VAMPIRE	SAD VAMPIRE	MPOF
$e(\mathbf{y}, \mathbf{y}_{GT})$	1.50 mm	1.46 mm	1.45 mm
$e^{\max}(\mathbf{y}, \mathbf{y}_{GT})$	8.19 mm	6.00 mm	5.64 mm
$\min_{\mathbf{x}}\{\det(\nabla \mathbf{y}(\mathbf{x}))\}$	0.64	0.58	0.39
$\max_{\mathbf{x}}\{\det(\nabla \mathbf{y}(\mathbf{x}))\}$	1.19	1.44	1.45
$NCC(\mathcal{T}, \mathcal{R})$	0.61	0.61	0.61
$NCC(\mathcal{M}^{\text{MP}}(\mathcal{T}, \mathbf{y}), \mathcal{R})$	0.87	0.88	0.88

patients to lower and keep constant the heart rate during examination. All scans were performed on a Siemens BiographTM Sensation 16 PET/CT scanner (Siemens Medical Solution) with a spatial resolution of around 6–7 mm [41]. During list mode acquisition an ECG signal was recorded for cardiac gating.

A phase-based cardiac gating with variable widths throughout all cycles and equal widths within each cycle is performed. The respiratory signals are estimated on basis of list mode data without auxiliary measurements [23]. An amplitude-based gating is applied to the respiratory signal, deduced from changes in the center of mass. For each of the 21 patients we choose 25 gates (five respiratory and five cardiac gates).

Image reconstruction is performed with the 3D EM software EMRECON [77, 78]. Twenty iterations with one subset are chosen. The output images are sampled with $175 \times 175 \times 47$ voxels. Given an isotropic voxel size of 3.375 mm, this results in a Field Of View (FOV) of $590.625 \times 590.625 \times 158.625$ mm. To prevent the attenuation correction artifacts discussed above, cf. [34], motion correction is performed on data that is not corrected for attenuation during reconstruction.

Results

We have seen in Table 2.1 that the performance of VAMPIRE and MPOF is comparable in practice. Thus we only show results from one method in the following analysis. Specifically, the presented results were generated with VAMPIRE and the SSD distance measure. We chose the gate in mid-expiration and diastole as our reference image $\mathcal{R}$ as diastole is the most common cardiac phase and the maximum motion of mid-expiration to all other respiratory phases is on average the smallest. We set the hyperelastic regularization parameters in Eq. (2.17) to $\alpha_l = 5$, $\alpha_a = 1$, and $\alpha_v = 10$ according to [51].

The effects of motion correction are illustrated for one patient in Fig. 2.11. The reference phase $\mathcal{R}$ in mid-expiration and diastole is shown in Fig. 2.11a. One particular template image $\mathcal{T}$ out of the 24 remaining gates is shown in Fig. 2.11b. We chose the gate in maximum inspiration and systole which is very dissimilar to $\mathcal{R}$. The grid representing the estimated transformation is overlaid on $\mathcal{T}$. Volume changes induced by this transformation are visualized in Fig. 2.11c. After applying the estimated transformation to $\mathcal{T}$ with the mass-preserving transformation model we obtain the image shown in Fig. 2.11d. The absolute difference images before and after applying the estimated transformation are given in Fig. 2.11e, f. Combining all 25 motion corrected gates yields the final image in Fig. 2.11g. A clear reduction of motion induced blurring can be observed compared to the image without motion correction in Fig. 2.11h.

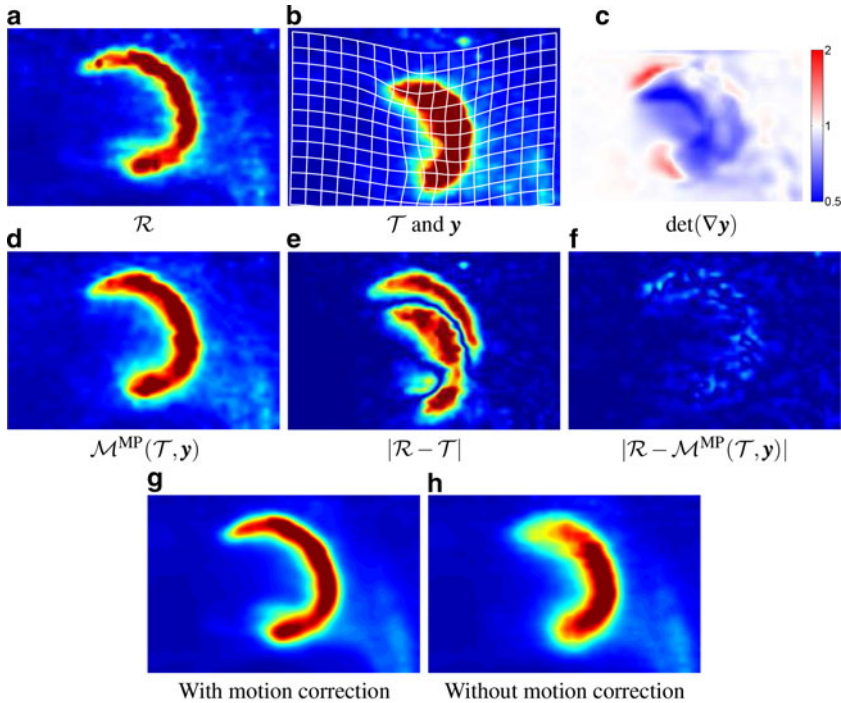


Fig. 2.11 Effects of motion correction illustrated for one patient. (b) is registered to $\mathcal{R}$ (a) using VAMPIRE resulting in (d). The absolute difference image before (e) and after motion correction (f) illustrate the accuracy. The estimated transformation $\mathbf{y}$ (b) is smooth and free of foldings which can also be seen at the plot of the Jacobian map (c). The logarithmic color map ranges from 0.5 (blue) over 1 (white) to 2 (red). The bottom row shows the final image with (g) and without (h) motion correction. (The images are taken from [51]. © 2011 IEEE. Colored figures are only available in the online version.)

Validation

We perform a validation of the patient data based on the range of the Jacobian determinant and the NCC defined in Eqs. (2.91) and (2.92). The minimum of all Jacobian maps (for all patients and all gates) is 0.340 and the maximum is 2.375. As no negative or too large values appear we can ensure that no unwanted foldings occur in the estimated transformations. For the NCC we can observe an improvement from 0.88 ± 0.07 to 0.97 ± 0.002 on average across all patients. For the most challenging gates in relation to the reference gate, i.e., heart in systole and heart in maximum inspiration, we found an increase from 0.80 ± 0.06 to 0.97 ± 0.001 and 0.79 ± 0.06 to 0.97 ± 0.002 , respectively. Figure 2.12 illustrates the average NCC values for the cardiac and respiratory phases individually. The two most challenging phases are gate two in Fig. 2.12a, representing the systole, and gate five in Fig. 2.12b, representing the gate in maximum inspiration.

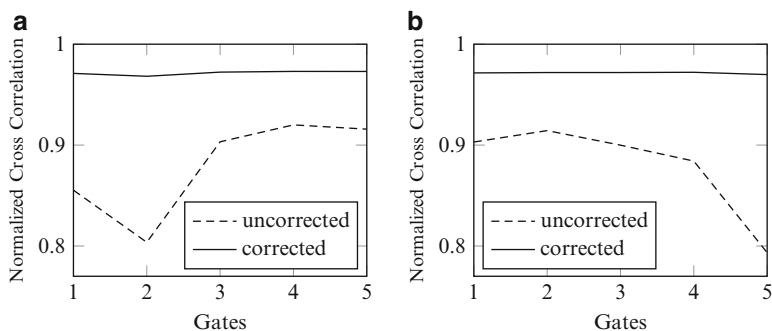


Fig. 2.12 The average NCC before and after motion correction is plotted for all cardiac (a) and respiratory (b) gates for the patient data (The plots are taken from [51]. © 2011 IEEE)

Attenuation Correction

Additional corrections like scatter and attenuation correction are necessary to make the presented results quantitative [38]. As attenuation correction is not the focus of this chapter we give only a short outlook. According to the method proposed by Bai and Brady [8] we perform an image-based attenuation correction. The idea of their attenuation correction approach is to reconstruct the PET reference gate $\mathcal{R}$ twice – once *with* attenuation correction ($\mathcal{R}^{\text{AC}}$, AC = Attenuation Corrected) and once *without* ($\mathcal{R}^{\text{NAC}}$, NAC = Non-Attenuation Corrected). Hence, an additional CT scan that matches the reference PET gate $\mathcal{R}$ is required. The Attenuation Coefficient Factor (ACF) image $\mathcal{I}^{\text{ACF}} = \frac{\mathcal{R}^{\text{AC}}}{\mathcal{R}^{\text{NAC}}}$ of the two reference images is computed. For the reference gate it holds

$$\mathcal{R}^{\text{AC}} = \mathcal{I}^{\text{ACF}} \cdot \mathcal{R}^{\text{NAC}} = \frac{\mathcal{R}^{\text{AC}}}{\mathcal{R}^{\text{NAC}}} \cdot \mathcal{R}^{\text{NAC}}. \quad (2.93)$$

As the average image

$$\mathcal{T}_{\text{avg}} = \mathcal{T}_{\text{avg}}^{\text{NAC}} = \frac{1}{25} \sum_{r=1}^5 \sum_{c=1}^5 \mathcal{M}^{\text{MP}}(\mathcal{T}_r^c, \mathbf{y}_r^c) \quad (2.94)$$

matches the reference phase, we can approximate the attenuation corrected version through

$$\mathcal{T}_{\text{avg}}^{\text{AC}} = \mathcal{I}^{\text{ACF}} \cdot \mathcal{T}_{\text{avg}}^{\text{NAC}} = \frac{\mathcal{R}^{\text{AC}}}{\mathcal{R}^{\text{NAC}}} \cdot \mathcal{T}_{\text{avg}}^{\text{NAC}} = \mathcal{R}^{\text{AC}} \cdot \frac{\mathcal{T}_{\text{avg}}^{\text{NAC}}}{\mathcal{R}^{\text{NAC}}}. \quad (2.95)$$

The results of the patient data with attenuation correction are illustrated in Fig. 2.13. The top row shows the *non*-attenuation corrected data and the middle row the attenuation corrected data. From left to right: the reference image $\mathcal{R}$, the reconstructed image without motion correction, the averaging result of VAMPIRE, and line profiles. The ACF image and the μ -map are additionally shown in the bottom row.

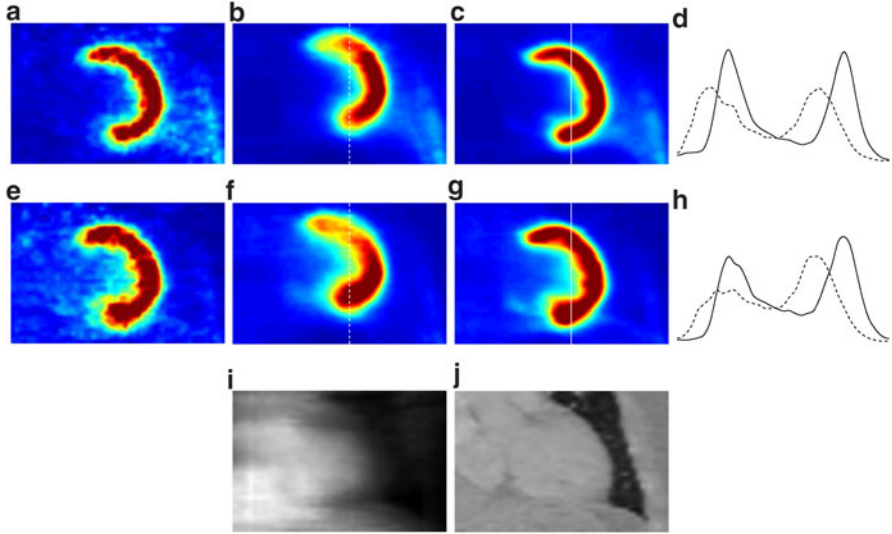


Fig. 2.13 Comparison of Non-Attenuation Corrected (NAC) and Attenuation Corrected (AC) patient data. (a) $\mathcal{R}^{\text{NAC}}$ (b) No MC, NAC (c) VAMPIRE, NAC (d) Line profiles NAC (e) $\mathcal{R}^{\text{AC}}$ (f) No MC, AC (g) VAMPIRE, AC (h) Line profiles AC (i) ACF image $\mathcal{I}^{\text{ACF}}$ (j) μ -map

The main message of Fig. 2.13 is that the application of attenuation correction may induce severe artifacts to cardiac PET data which can be overcome by our proposed motion estimation and motion correction approaches. The mentioned

artifacts can be observed in the upper region of the left ventricle in Fig. 2.13f. The radioactivity distribution is strongly underestimated due to the misalignment of the μ -map in Fig. 2.13j and the PET data in Fig. 2.13b. However, as attenuation correction is essential for quantitative evaluations we can choose the reference image $\mathcal{R}$ as the respiratory and cardiac phase representing the same motion state as the μ -map and perform motion correction. The resulting image with motion correction and attenuation correction in Fig. 2.13g is no longer affected by the artifacts present in Fig. 2.13f and can thus be used for further quantitative clinical evaluations.

Motion Correction in Thoracic Positron Emission
Tomography

Gigengack, F.; Jiang, X.; Dawood, M.; Schäfers, K.P.

2015, XI, 88 p. 33 illus., 24 illus. in color., Softcover

ISBN: 978-3-319-08391-9