Year: 2014

Variational image registration using inhomogeneous regularization

Jud, Christoph and Lüthi, Marcel and Albrecht, Thomas and Schönborn, Sandro and Vetter, Thomas

Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Variational Image Registration using Inhomogeneous Regularization

Christoph Jud · Marcel Lüthi · Thomas Albrecht · Sandro Schönborn · Thomas Vetter

Received: date / Accepted: date

Abstract We present a generalization of the convolution-based variational image registration approach, in which different regularizers can be implemented by conveniently exchanging the convolution kernel, even if it is nonseparable or nonstationary. Nonseparable kernels pose a challenge because they cannot be efficiently implemented by separate 1D convolutions. We propose to use a low-rank tensor decomposition to efficiently approximate nonseparable convolution. Nonstationary kernels pose an even greater challenge because the convolution kernel depends on, and needs to be evaluated for, every point in the image. We propose to pre-compute the local kernels and efficiently store them in memory using the Tucker tensor decomposition model. In our experiments we use the nonseparable exponential kernel and a nonstationary landmark kernel. The exponential kernel replicates desirable properties of elastic image registration, while the landmark kernel incorporates local prior knowledge about corresponding points in the images. We examine the trade-off between the computational resources needed and the approximation accuracy of the tensor decomposition methods. Furthermore, we obtain very smooth displacement fields even in the presence of large landmark displacements.

Keywords variational image registration · separable filter approximation · nonstationary filtering · Gaussian process regression

1 Introduction

Image registration is a common problem that arises in many applications of medical image analysis. The problem is to find a non-rigid transformation which aligns two images. In this paper we focus on the variational principle to formulate this problem, which has found broad acceptance in the literature [4,5,33]. The sought transformation corresponds to the minimum of a functional, which finds a compromise between the image similarity of the transformed target and the reference image $I_T, I_R : \Omega \to \mathbb{R}$, defined on some domain Ω, and the smoothness of the image transformation $\varphi : \Omega \to \Omega$. Formally, the problem can be written as

$$\mathcal{J}[\varphi] := \mathcal{D}[I_R, I_T, \varphi] + \mathcal{S}[\varphi], \tag{1}$$

where $\mathcal{D}$ is an image distance measure and $\mathcal{S}$ a regularizer. Using calculus of variation, the optimum of (1) can be found by solving a system of partial differential equations.

An elegant framework to minimize Equation (1) was proposed by Beutien *et al.* [4], where a minimum is reached using a convolution-based approach. Different regularization properties can be achieved by choosing different convolution kernels. However, their approach only addresses stationary kernels and only works efficiently if the kernel is separable.

In this paper, we present a generalization of this framework, where the regularization kernel are even nonseparable or nonstationary. By employing low-rank tensor decomposition [18], we approximate nonseparable convolution kernels by separable 1D kernels in order to perform the convolution separately in each space dimension by successive 1D

C. Jud [1]
E-mail: christoph.jud@unibas.ch

M. Lüthi [1]
E-mail: marcel.luethi@unibas.ch

T. Albrecht [1]
E-mail: thomas.albrecht@unibas.ch

S. Schönborn [1]
E-mail: sandro.schoenborn@unibas.ch

T. Vetter [1]
E-mail: thomas.vetter@unibas.ch

[1] Department of Mathematics and Computer Science
University of Basel, CH-4056 Basel, Switzerland

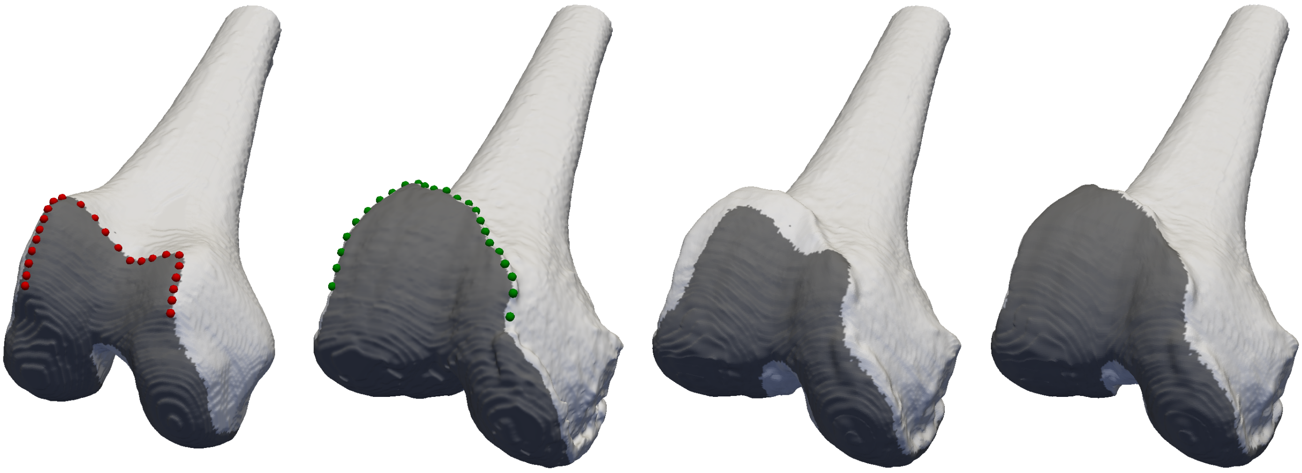

Fig. 1 The first two figures from the left show the reference and target femur surfaces. The landmark points are depicted in color, red and green respectively. The patellar surface is marked in dark gray. In the right two figures the reference is transformed to the target with two different methods. First, using standard diffeomorphic Demons [39], which does not incorporate the landmarks. The patellar surface is clearly misaligned. In the last figure, the registration was performed with our hybrid approach, which considers the landmarks. This results in an accurate patellar surface alignment.

convolutions. Compared to nonseparable convolution, this dramatically reduces the computational complexity and, additionally, it accurately retains the regularization properties. For instance, in 3D the separable Gaussian kernel requires three 1D convolutions, while a rank-4 nonseparable kernel needs $4 \times 3 = 12$ separable 1D convolutions.

Furthermore, we extend the framework to nonstationary regularizers, *i.e.* regularizers which depend on the spatial location. The corresponding spatially varying local filter kernels require an efficient handling, which we approach by a powerful caching scheme. In addition, we use an extension of the framework in order to ensure that the resulting transformations are diffeomorphic. As in [28,37,38], in each iteration, we compute an efficient approximation of the exponential mapping that keeps the transformations diffeomorphic.

While this generalization is useful in a wide variety of registration tasks, our original motivation was to integrate landmarks into the regularization. In Figure 1, we illustrate the practical importance of hybrid landmark and image registration. In this example, the landmarks help to greatly improve the registration accuracy of the patellar surface of a human femur. Using our nonstationary filtering approach, we can efficiently compute a solution to the hybrid registration problem as formulated by Lüthi *et al.* [21], which is conceptually appealing, but has so far been computationally infeasible. The idea is to integrate landmarks directly into the regularization, which in our terms means to minimize a slightly different functional

$$\hat{\mathcal{J}}[\boldsymbol{\varphi}] := \mathcal{D}[I_R, I_T, \boldsymbol{\varphi}] + \hat{\mathcal{S}}[\boldsymbol{\varphi}, X], \tag{2}$$

where X is a list of landmark displacements. Since the regularizer $\hat{\mathcal{S}}$ considers the given displacements at the landmark positions, the resulting kernel is nonstationary. Therefore, simple convolution approaches are not applicable, since the filter kernel varies for each image location. However, in our framework, we are able to efficiently handle the regularizer's local dependency. Hence, performing hybrid image registration in terms of Equation (2) becomes computationally feasible even for 3D images.

Non-rigid image registration has been extensively studied in literature and several attempts have been made to reach a general framework for different regularizers [5,33]. For an overview of image registration methods, we refer to the survey paper of Sotiras *et al.* [30], and more specifically to the book of Modersitzki [23] for a deeper discussion about variational methods for image registration. For the diffeomorphic regularization we refer to [1–3,28,37,38]. Nonstationary filtering methods have been used *e.g.* in [6,31], where the regularization is locally adjusted depending on local image features as *e.g.* curvature or local transformation properties like stiffness. Different hybrid methods, which combine landmarks and image features have been proposed in literature. For example in [14,16,26], the landmarks are treated as additional constraints. The methods require a perfect interpolation of the landmarks resulting in numerical problems during optimization. Other methods do not enforce the landmark constraints strictly, but add the landmark differences as another cost term to the functional in Equation (1) [2,17, 20,25,29]. In our work, we were inspired by the Bayesian approach of Lüthi *et al.* [21], where these two kinds of methods are combined by integrating the landmarks into the regularization. This makes the uncertainty on the landmarks independent from the data term and reduces the search space to transformations, agreeing with the landmark displacements. Contrary to the approach of Lüthi *et al.* [21] however, our nonstationary filtering technique is able to efficiently handle large 3D images.

Our paper is structured as follows: in the background section we briefly introduce the variational image registration framework and present ideas about the hybrid image registration framework of Lüthi *et al.* [21] adapted to our method. In the subsequent method section we present our separate and nonstationary filter approach. In our experiments, we show registration results using different positive definite kernel functions such as Gaussian and exponential kernels. By using low-rank approximations of nonseparable kernels, we show an accuracy gain as well as an improved convergence property during the registration process. Applying a nonstationary kernel, we show the memory savings we reach with our caching scheme using tensor decomposition. Furthermore, we discuss in more detail the introductory patellar surface example, where the incorporation of the landmarks leads to better registration performance. Finally, we discuss the advantages and challenges related to the landmark based transformation.

2 Background

2.1 Variational Image Registration Framework

Considering dense image features, for instance gray scale values of a CT image, formulating the registration problem as a variational optimization problem turned out to be very useful in literature [4,5,28,33,37,38]. A mapping that registers the two images I_R and I_T is sought as the minimum of the joint functional (1). Using methods from the calculus of variations, the functional is differentiated with respect to the mapping $\boldsymbol{\varphi}$. If we denote the space of all admissible mappings as $\varPhi$, the functional derivatives of the two terms are $d\mathcal{D} =: \mathbf{f} : \varPhi \to \mathrm{I\!R}^d$ and $d\mathcal{S} =: \mathcal{A} : \varPhi \to \varPhi$. $\mathcal{A}$ is typically a differential operator and a minimum of the functional has to satisfy the system of partial differential equations

$$\mathcal{A}[\boldsymbol{\varphi}] = \mathbf{f}(\boldsymbol{\varphi}). \tag{3}$$

Many different strategies to solve this PDE and the associated minimization problem have been put forward, including finite difference methods, finite element methods B-spline based methods *etc.*

The fastest method to solve (3), on which our work is based, is a convolution approach [28,33]. This approach is possible if the fundamental solution or "Green's function" for the operator $\mathcal{A}$ is known. The Green's function is then a positive definite kernel function $k : \varOmega \times \varOmega \to \mathrm{I\!R}$, and the PDE can be minimized by the iteration scheme

$$\boldsymbol{\varphi}^{i+1} = k * (\boldsymbol{\varphi}^i \circ \mathbf{f}(\boldsymbol{\varphi}^i)), \tag{4}$$

where $*$ denotes the convolution operation.

The classical example is the Demons algorithm where $\mathcal{A} = -\varDelta$ and k is the Gaussian kernel. In principle it is also possible to choose the kernel directly, without actually specifying the corresponding operator. Long *et al.* [19] proposed to use the exponential kernel $k(x, y) = \frac{1}{C_d} \cdot e^{-\frac{\|x-y\|}{\alpha}}$, where C_d is a normalization constant for the number of dimensions d. $C_1 = 2\alpha, C_2 = 2\pi\alpha^2, C_3 = 8\pi\alpha^3$. They showed that compared to the Gaussian kernel, this kernel better approximates the linear elasticity regularization, which leads to better transformation properties (see also Steinke *et al.* [32]). Since the kernel is not separable, standard computational requirements are exceeded if it is directly applied in this framework. In our experiments later in this paper, we also show registration results using this regularizer.

2.2 Neighborhood Preservation

Common simple regularizers, as the ones discussed in the previous section, penalize unsmooth transformations. They cannot avoid foldings, nor do they lead to invertible transformations. But in medical applications invertible transformations are preferred. Various authors [1,8,28,35,37,38] have therefore proposed an additional restriction of the mappings $\boldsymbol{\varphi}^i$ by modeling $\boldsymbol{\varphi}$ using geodesic flows of diffeomorphisms [3]. Following the diffeomorphic Demons [37] approach, the restriction of an optimal displacement field to be diffeomorphic $\boldsymbol{\varphi} \in \mathit{Diff}(\varOmega)$ can be achieved by mapping the current transformation $\boldsymbol{\varphi}^i$ back onto the Lie group of diffeomorphisms [1,28,37,38]. This ensures the transformation to be invertible and neighborhood preserving. A possible way to do that, which only marginally changes the optimization scheme in Equation (4), is by calculating the group exponential map exp of the Lie group of diffeomorphisms of $\boldsymbol{\varphi}^i$ after evaluating the field update

$$\boldsymbol{\varphi}^{i+1} = k * (\boldsymbol{\varphi}^i \circ \exp(\mathbf{f}(\boldsymbol{\varphi}^i))). \tag{5}$$

The exponential mapping can be efficiently approximated by a scaling and squaring algorithm [1,37].

2.3 Hybrid Image Registration

In hybrid image registration, both dense image features and landmarks are available. In addition to the images I_R and I_T, we are given lists $X_R = \{x_i^R\}_{i=1}^n$ and $X_T = \{x_i^T\}_{i=1}^n$

of n corresponding landmark points for each image. The displacements induced by these landmarks are given by

$$\begin{aligned} X &= \{(x_1^R, x_1^T - x_1^R), \ldots, (x_n^R, x_n^T - x_n^R)\} \\ &=: \{(x_1, y_1), \ldots, (x_n, y_n)\}. \end{aligned}$$

Further, let $u : \Omega \to \mathrm{I\!R}^d$ be an unknown displacement field, such that the warp $\boldsymbol{\varphi}(x) = x + u(x), x \in \Omega$ and the target image is warped by $I_T(x + u(x))$. In [21], Lüthi *et al.* modeled the prior knowledge about u using a Gaussian process $\mathcal{GP}(\mu, k)$, which is defined by a mean function $\mu : \Omega \to \mathrm{I\!R}^d$ and a covariance function $k : \Omega \times \Omega \to \mathrm{I\!R}$. In our terms, this means to minimize

$$\mathcal{J}_{GP}[\boldsymbol{\varphi}] := \mathcal{D}_{GP}[I_R, I_T, \mu, u] + \mathcal{S}_{GP}[k, u],$$

where the target image is warped by $I_T(x + \mu(x) + u(x))$. Assuming a zero mean, this functional is equivalent to the functional (1) introduced at the beginning of this paper. However, the strength of this interpretation is that we can now formulate the hybrid registration problem by conditioning the Gaussian process on the n given landmark displacements. The resulting posterior process $\mathcal{GP}_X(\mu_X, k_X)$ is given in closed form by

$$\begin{aligned} \mu_X(x) &= \mu(x) + K_X(x)^T (K + \sigma^2 Id)^{-1} Y \qquad (6) \\ k_X(x, y) &= k(x, y) - K_X(x)^T (K + \sigma^2 Id)^{-1} K_X(y)^T, \end{aligned}$$

where $K_X(x) = (k(x, x_i))_{i=1}^n \in \mathrm{I\!R}^n, K \in \mathrm{I\!R}^{n \times n}$ is the kernel matrix with entries $K_{i,j} = k(x_i, x_j)$, $Y = (y_1 - \mu(x_1), \ldots, y_n - \mu(x_n))^T \in \mathrm{I\!R}^n$ are the mean free landmark displacements and σ^2 models the uncertainty about matching accuracy of the landmarks (see also *e.g.* Rasmussen in [27], Chapter 2.2). Hence, our functional which we minimize becomes

$$\mathcal{J}_{GP_X}[\boldsymbol{\varphi}] := \mathcal{D}_{GP_X}[I_R, I_T, \mu_X, u] + \mathcal{S}_{GP_X}[k_X, u], \qquad (7)$$

using μ_X as landmark based mean transformation and the kernel function k_X for regularization.

Since k_X depends on the landmark displacements, it is not stationary and cannot directly be handled by the optimization scheme (4) introduced above. In the following, we present the necessary adjustments to still benefit from the advantages of the framework.

3 Methods

In this paper, we generalize the variational image registration framework to kernel functions, which possibly are nonseparable and nonstationary. While the optimization scheme in (4) is conceptually not restricted to separable convolution kernels, nonseparable filtering exceeds standard computational requirements. In Section 3.1, we present a separable 1D filter approximation for nonseparable filters, based on low-rank tensor approximation, which enables us to perform the convolution efficiently.

For the generalization to nonstationary kernels, which vary depending the spatial location, the Equation (4) has to be rewritten. We explicitly write the convolution integral but with a kernel which is not stationary (*cf.* the work of McOwen in [22], Chapter 2.3 and Evans [11], Chapter 2.3). This becomes the following integral equation

$$\boldsymbol{\varphi}^{i+1} = \int_\Omega k_X(\cdot, s)(\boldsymbol{\varphi}^i \circ \mathbf{f}(\boldsymbol{\varphi}^i))\mathrm{d}s. \qquad (8)$$

Similar, with $\boldsymbol{\varphi}$ restricted to be diffeomorphic, Equation (5) becomes

$$\boldsymbol{\varphi}^{i+1} = \int_\Omega k_X(\cdot, s)(\boldsymbol{\varphi}^i \circ \exp(\mathbf{f}(\boldsymbol{\varphi}^i)))\mathrm{d}s. \qquad (9)$$

In Section 3.2, we further introduce a nonstationary filter approach, which makes the approximation of the integral (8) and (9) computationally feasible.

3.1 Separable Filter Decomposition

Applying the proposed optimization scheme to image registration requires the discretization of the formulation in Equation (9). To make it more clear, we start by writing the spatially discretized version of Equation (5), where the kernel k becomes a 3D tensor H_0:

$$\boldsymbol{\varphi}^{i+1}(x) = H_0 * (\boldsymbol{\varphi}^i \circ \exp(\mathbf{f}(\boldsymbol{\varphi}^i)))_x. \qquad (10)$$

H_0 is the discrete unit impulse response of k with elements $H_{0_{ijq}} = k(0, (i, j, q)^T)$ and i, j, q cover the neighborhood around 0, while the subscript x of the second term indicates the equally large discrete neighborhood around the point x.

If the kernel is separable the iteration scheme of Equation (10) can be accelerated greatly by performing the convolution separately in each space dimension by successive 1D convolutions. The Gaussian kernel has this nice property of separability. Therefore without any further effort, the convolution with this kernel can be performed separately. To still benefit from this performance gain for nonseparable kernels, like the exponential kernel [19], their separability has to be approximated. In 2D, this can be achieved by standard singular value decomposition. However in 3D, this leads to mathematical challenges that go beyond standard linear algebra, since a filter kernel in 3D is a third order tensor. In contrast to 2D matrices, it is an NP-hard problem to determine the rank of a specific given higher order tensor (see Kolda and Bader in [18]). Hence, the rank R becomes a parameter which has to be estimated. Nevertheless, we are able to compute the approximation using a CANDECOMP/PARAFAC (CP) decomposition model [18]. This gives us separable 1D approximations of the discrete unit impulse

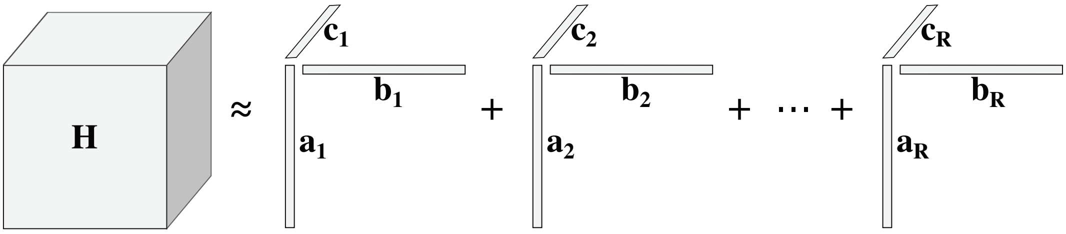

Fig. 2 CP tensor decomposition model of the third-order tensor H.

response H_0. In Figure 2 the decomposition model is visually illustrated. Such a decomposition can be formulated as a minimization problem

$$\min_{\tilde{H}_0} \|H_0 - \tilde{H}_0\| \text{ with } \tilde{H}_0 = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r, \tag{11}$$

where the operation $\otimes$ denotes the three-way outer product $\tilde{H}_{0_{ijk}} = \sum_r^R a_{ir} b_{jr} c_{kr}$. Standard algorithms to optimize (11) are based on the alternating least squares (ALS) method [15], which is explained in more detail in the following section. The parameter R is estimated by testing the approximation performance for different ranks (see Section 4.1).

Once the decomposition is performed, the distributivity (12) and the associativity (13) of the convolution operation can be exploited to perform the convolution separately in each dimension with a_r, b_r and c_r

$$\begin{aligned} H_0 * \mathbf{f} \approx \tilde{H}_0 * \mathbf{f} &= \left[\sum_r^R a_r \otimes b_r \otimes c_r\right] * \mathbf{f} \\ &= \sum_r^R [a_r \otimes b_r \otimes c_r * \mathbf{f}] \qquad (12) \\ &= \sum_r^R a_r * (b_r * (c_r * \mathbf{f}))\,. \qquad (13) \end{aligned}$$

For cubic filter kernels, having a filter width m, the computational cost for each output pixel reduces significantly from m^3 to $3Rm$.

3.1.1 Alternating Least Squares Method

The low-rank approximation of H_0 is calculated by minimizing the optimization problem (11). The minimizers a_r, b_r and c_r are obtained using the alternating least squares (ALS) method [15]. To this purpose, we introduce a notation to represent a tensor in a matrix form.

Let $H \in \mathbb{R}^{P\times Q\times R}$ be a third-order tensor. By fixing one index the tensor is sliced into two-dimensional sections which have horizontal *(mode-*1*)*, lateral *(mode-*2*)* and frontal *(mode-*3*)* orientation for the indices $\{1,2,3\}$ respectively. The mode-n *unfolding* denoted as $H_{(n)}$ concatenates the mode-n slices horizontally to a matrix.

Following Kolda [18], the CANDECOMP/PARAFAC model can be expressed as

$$H = [\![A,B,C]\!] \equiv \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r,$$

while $H_{(1)} = A(C \odot B)^T$, with $A = (a_1, a_2, \ldots, a_R)$ and likewise B and C. $\odot$ is the Khatri-Rao product (see A.1).

The matrices A, B and C, which minimize (11) can be calculated by alternately fixing all but one matrix *e.g.* A. This is followed by minimizing

$$\min_A \|H_{(1)} - A(C \odot B)^T\|_F,$$

which has the optimum at

$$A = H_{(1)}[(C \odot B)^T]^\dagger.$$

Using the special property that

$$(A \odot B)^\dagger = (A^TA \star B^TB)^\dagger (A \odot B)^T,$$

where $\star$ is the Hadamard product (see A.2) and $A^\dagger$ the Moore-Penrose pseudo-inverse, the equations can be iteratively solved for A, B and C

$$\begin{aligned} A &= H_{(1)}(C \odot B)(B^TB \star C^TC)^\dagger \\ B &= H_{(1)}(C \odot A)(A^TA \star C^TC)^\dagger \\ C &= H_{(1)}(B \odot A)(A^TA \star B^TB)^\dagger. \end{aligned}$$

until the values of A, B and C converge. The convergence speed depends on the initialization of the fixed matrices. A common choice for the initialization is to use the Higher-order SVD [10] discussed in Section 3.2.2.

Since H_0 is now decomposed, the convolution in (10) can be performed separately.

3.2 Efficient Nonstationary Filtering

While stationary kernels $k(x-y)$ only depend on the difference of x and y, nonstationary kernels $k(x,y)$ are dependent on both arguments. Therefore, for such kernels, separable filtering is not possible since the associativity no longer holds. If we spatially discretize the integral Equation in (9)

$$\boldsymbol{\varphi}^{i+1}(x) = H_x * (\boldsymbol{\varphi}^i \circ \exp(\mathbf{f}(\boldsymbol{\varphi}^i)))_x, \tag{14}$$

where H_x is the discrete impulse response of k at location x, *i.e.* $H_{x_{ijq}} = k(x, (i, j, q)^T)$, we see that H now depends on x, which makes the problem nonstationary. In general, the calculation of all the local impulse responses makes the problem computationally unscalable. However, in the particular case where we minimize the hybrid functional (7) we can exploit the following properties of the landmark kernel k_X to reach an algorithm which is computationally feasible.

3.2.1 Landmark kernel properties

The landmark kernel k_X consists out of the kernel k subtracted by a landmark dependent term. The difference between k and the full landmark kernel k_X becomes negligible if

$$\forall x_i \in X \; k(x, x_i) < \xi \tag{15}$$

i.e. if x is not in the neighborhood of any landmark. This property is exploited to approximate the integral of Equation (9) by only considering k, the first part of the landmark kernel k_X, if the value of its second part goes to zero. We perform the approximation in two steps:

1. At first, the whole image is filtered separately using the stationary part k.
2. Subsequently, the nonseparable and nonstationary filtering with the full kernel k_X is performed, but only for pixels where (15) is not fulfilled.

The second part is the most expensive step, because for each point in the vicinity of the landmarks its discrete local impulse response H_x has to be calculated. This means a cubically increasing amount of kernel evaluations, which covers the neighborhood of all points having landmark support, and this in each iteration of Equation (9). To reduce the computational demands we propose the following caching scheme.

3.2.2 Local filter caching

Since the landmark kernel is nonstationary, but still time-invariant, it is reasonable to keep the computed filter kernels in memory to save computational time for the following iterations. Jumping out of the frying pan into the fire, the amount of memory to cache all the filter kernels grows rapidly depending on the filter width and the number of landmarks. Therefore, we compress these local filter kernels by again taking advantage of tensor decomposition, before we cache them in the memory.

As we saw in Section 3.1.1, the CP decomposition is obtained by the ALS method, which is quite costly due to its iterative nature. Because H_0 has to be decomposed only once, it is still well suited to approximate the separability of the stationary filter. However, it is too slow to decompose all the local impulse responses H_x.

Compared to the CP decomposition the Tucker decomposition [34] (see Figure 3) is significantly faster. It is an alternative model to decompose a tensor. Similar to the CP model the tensor is decomposed into triplets of vectors, but they are weighted by a full so called “core” tensor.

$$\tilde{H}_x = [\![G; A, B, C]\!] \equiv \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_p \otimes b_q \otimes c_r, \tag{16}$$

where g_{pqr} are the elements of the core tensor G and P, Q, R are the ranks for each space dimension. In this model the *unfolded* tensor H is represented as

$$H_{(1)} = AG_{(1)}(C \bullet B)^T,$$

where $\bullet$ is the Kronecker product (see A.3). Using the Higher-order SVD algorithm of De Lathauwer and De Moor [10]

$$\min \|H_{(1)} - AG_{(1)}(C \bullet B)^T\|$$

can be very efficiently minimized by setting A, B and C to the leading left singular vectors of the corresponding mode-n *unfolding* $H_{(n)}$

$$A = U_P^{(1)}, \; B = U_Q^{(2)}, \; C = U_R^{(3)},$$

where $U_l^{(n)}$ is the matrix consisting out of the leading l singular vectors of $H_{(n)}$ and $G_{(1)} = AH_{(1)}(C \bullet B)^T$.

Compared to the CP model the Tucker decomposition is a less restricted model where the core G can be dense while in the CP model the core is a super diagonal tensor[1] with ones on the diagonal. Although it cannot be used for separable filter approximation due to the weighting with the dense G, the memory savings are similar to the CP model. Setting $P = Q = R$ and having a filter length m, the memory consumption reduces from m^3 to $R^3 + 3Rm$ per voxel in the support of the landmarks. In this paper we chose P, Q and R by testing the resulting approximation performance.

3.3 Multi-resolution vs. Multi-scale

We presented a method to minimize the registration functional (7). It is mainly based on the local iterative minimization scheme (4). As such, it relies on a reasonable initialization and is prone to get “stuck” in local minima. In order to deal with that, we adopt a multiresolution strategy [40]. The support in voxels of the kernel function k implicitly increases towards the lower resolution levels. Therefore, in combination with the posterior mean function μ_X, we use a

[1] A super diagonal tensor is the generalization of a diagonal matrix to higher order tensors, where the entries outside the main diagonal are zero.

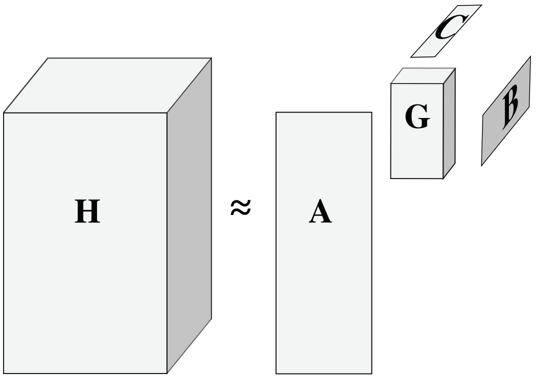

Fig. 3 Tucker tensor decomposition model.

multiscale kernel $\tilde{k}$ (*cf.* Opfer in [24]), which combines kernels with different support, to compute the landmark based mean transformation μ_X (6):

$$\tilde{k}(x,y) = \sum_{l=0}^{L} \lambda_l k^l(x,y),$$

where λ_l are positive weights and k^l correspond to k with adjusted kernel parameters per scale level l and L is the number of scale levels. The parameter *e.g.* for the Gaussian kernel becomes $\sigma_g^l = \sigma_g \cdot 2^{L-l}$. We have set the weights λ_l to 10^{-l}.

3.4 The Algorithm

By joining all the previously described building blocks, we have obtained a non-rigid image registration framework, in which different regularizers can be implemented by conveniently exchanging the regularization kernel, even if it is nonseparable or nonstationary. Specifically, the landmark kernel is supported by our framework. The diffeomorphic regularization is also approximated, as shown in Equation (14). Moreover, we showed a multiscale approach that brings the landmark mean together with the image-based optimization on different resolution levels. The full algorithm which maximizes Equation (7) by joining all the presented concepts is provided in Listing 1.

In the following, the performance of our filter approximation techniques is evaluated in detail, while we also provide a qualitative hybrid registration example.

4 Results

We presented a method which enables an efficient approximation of the optimization scheme in Equation (9). In this section, we perform registration experiments for validating our method. First, we provide a detailed study about the separable filter approximation and discuss its approximation performance in terms of accuracy and computational aspects. Second, we analyze the local filter compression with respect to memory consumption, computational demands as well as approximation accuracy. We compare our method with Elastix [17], where the landmarks are incorporated as an additional cost term to the functional (1). This is followed by a qualitative result of the introductory patellar surface example. Likewise, we compare these results with Elastix. In an additional section we discuss the landmark based mean transformation μ_X in more detail.

As quality measurement, we use the target registration error (TRE), the dice coefficient (DICE) and we count singularities of the displacement fields, which is the number of voxels where the determinant of the Jacobian is smaller than zero. To compare two displacement fields A and B for each vector pair, we consider the magnitude differences and the vectors directional discrepancy. Following that, we define the accuracy loss:

$$\tau(A,B) = \int_{\Omega} \left(\|A(x)\| - \|B(x)\|\right)^2 + \left\| \frac{A(x)}{\|A(x)\|} \times \frac{B(x)}{\|B(x)\|} \right\|^2 \mathrm{d}x, \tag{17}$$

where $\tau(A,A) = 0$ and greater than zero for dissimilar displacement fields.

Since we only compare different regularizers and their approximations, we use for all experiments, the sum of squared differences similarity measure

$$\mathcal{D}[I_R, I_T, \boldsymbol{\varphi}] := \int_{\Omega} \left(I_T \circ \boldsymbol{\varphi}(x) - I_R(x)\right)^2 \mathrm{d}x.$$

Following Thirion [33], we perform second order gradient descent on $\mathcal{D}$ and obtain the forces

$$\mathbf{f}(\boldsymbol{\varphi}(x)) = -\frac{(I_T \circ \boldsymbol{\varphi}(x) - I_R(x))\nabla I_T \circ \boldsymbol{\varphi}(x)}{\kappa^2 \left(I_T \circ \boldsymbol{\varphi}(x) - I_R(x)\right)^2 + \|\nabla I_T \circ \boldsymbol{\varphi}(x)\|^2},$$

with κ^2 the reciprocal of the mean squared image spacing.

Listing 1 Variational Image Registration Algorithm

```
Input: $I_R, I_T, X, k, \sigma, n, L$
Output: $\varphi^n$
Build image pyramid -> $I_R^0, ..., I_R^L, I_T^0, ..., I_T^L$
Initialize $\varphi^0 = Id$

foreach: resolution level $l$
  perform mean transformation of $I_R^l, I_T^l$ using $\mu_X$
  compute and decompose $H_0 \to A_0, B_0, C_0$ (Equation 11)
  for $i = 0...n$:
    compute force field: $\mathbf{f}^l(\varphi^i)$
    perform separable filtering: $\varphi^{i+1} = H_0 * (\varphi^i \circ \exp(\mathbf{f}^l(\varphi^i)))$ (Equation 10)
    for $x$ near a landmark in $X$:
      if i==0:
        decompose and cache $H_x \to g_x, A_x, B_x, C_x$ (Equation 16)
      else:
        reconstruct $\tilde{H}_x$ using $g_x, A_x, B_x, C_x$
      perform nonstationary filtering for neighborhood of $x$ (Equation 14)
  set $\varphi^0 = upsample(\varphi^n)$ for the next resolution level
```

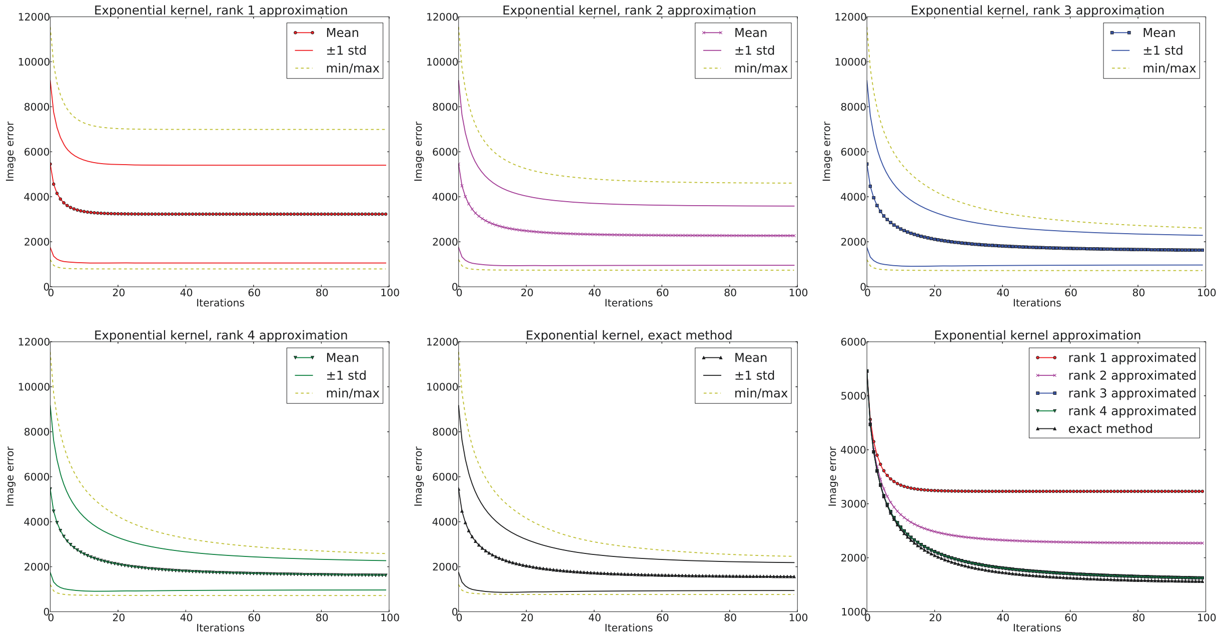

Fig. 4 This figure shows the image error averaged over all 9 experiments for approximation rank one to four as well as for the exact method. For each experiment the mean error is plotted as well as ± one standard deviation in a solid style and the max/min as a dashed curve. In the last subfigure the averages for all variants are again shown in one plot.

Generally, we set the prior mean function μ always to a rigid pre-alignment of the images.

We implemented our algorithm by extending the finite difference solver framework of the Insight Toolkit [39] and performed the experiments on an Intel Xeon CPU @ 3 GHz on 12 cores.

Elastix Configuration For the registration with Elastix [17] we used the B-spline transformation model combined with the mean squares metric and an LBFGS optimizer. For the landmark examples we combined the mean squares metric with the "Corresponding Points Euclidean Distance Metric" which is equivalent to the target registration error.

4.1 POPI Breathing Thorax Model

In this first experiment, we show quantitative results by different approximation ranks of the nonseparable exponential kernel ($\alpha = 1$) without considering landmarks. The filter H_0 has been discretized in a 23^3 voxel neighborhood. We

Approximation rank	$R = 1$	$R = 2$	$R = 3$	$R = 4$	None
iterations to converge	20	44	68	68	63
final average image error	3588	2524	1808	1799	1737
accuracy loss	1.0033	0.6407	0.2299	0.2250	0.0000
relative CPU time	0.08	0.17	0.25	0.33	1.00

Table 1 This table provides information about convergence properties with different approximation ranks. The iterations to convergence in the first row were computed as the number of iterations until the average image error change deceed the value 3. The accuracy loss is calculated by using $\tau(A, B)$ in Equation (17) between A the resulting and B the exact method transformation.

compare the results to the exact method, which is obtained with the same kernel, but without separable filtering. We used the POPI dataset provided by the Léon Bérard Cancer Center & CREATIS lab, Lyon, France [36], which contains 10 CT images of a breathing lung. The images have a resolution of $482 \times 360 \times 141$ voxels and a spacing of $0.98 \times 0.98 \times 2mm^3$. For our experiment, the images have been resampled to $235 \times 175 \times 141$ voxels and scaled to isotropic spacing at $2mm^3$. In the experiment, we have chosen the image number 0 to be the reference image, while we calculated the experiments on a single scale. We repeated the experiment by increasing the rank R of the separable filter approximation from one to four. $R = 1$ corresponds to the rank-one approximation used in Beuthien and Kamen [4] which serves us as baseline. The exact method corresponds to the algorithm of Long *et al.* [19] extended to 3D.

In Figure 4, we illustrate the image error averaged over the 9 registrations changing during the optimization for each experiment. In the first three experiments the convergence rate decreases with increasing rank R, while the resulting image error is getting smaller. One can also observe that for $R \geq 3$ the image error stays nearly the same and is close to the exact method. Moreover, the variance of the image error is getting more narrow with higher R. It can be assumed that for $R > 4$ no significantly improved approximation can be achieved. For a better comparison, all mean curves are again shown together in the last plot. Furthermore, in Table 1, the results of the experiments are summarized in numerical terms. Evaluating the accuracy loss (17) between the approximations and the exact method, higher rank approximations reach greater accuracy. The CPU time is considerable high for the exact method. With a third of the effort, we achieve a good approximation, accepting only a very small loss of accuracy. For a more detailed comparison, we repeated the whole experiment again on different scale levels. The results are listed in Table 2 and the upper part of Table 3. Note that all quantities are averaged over the 9 experiments.

The results show that for nonseparable kernels, a one-rank approximation is not accurate enough to approximate the filter's regularization property. With increasing rank, the calculation time gets larger. The increase is linear in R. Since the resulting image error as well as the convergence properties using $R = 4$ do not significantly differ from the exact method, we think that 4 ranks are sufficient to approximate the exponential kernel separably.

For a meaningful comparison to Elastix, we performed three experiments. First, the smoothness parameter $\sigma_{\text{B-spline}}$ of the B-spline transform has been tuned to a small TRE ($\sigma_{\text{B-spline}} = 4$). Second, $\sigma_{\text{B-spline}}$ was tuned in order that no singularities are present in the result, but simultaneously for a TRE which is as small as possible ($\sigma_{\text{B-spline}} = 16$). Finally, the parameter was chosen for a resulting transformation, which are approximately as smooth as the ones obtained by our method ($\sigma_{\text{B-spline}} = 64$). To quantify the smoothness of a displacement field A we integrate over the local displacement changes:

$$v(A) = \int_{\Omega} \int_{\mathfrak{B}_x} \|A(s) - A(x)\| \, \mathrm{d}s \, \mathrm{d}x,$$

where $\mathfrak{B}_x$ is the neighborhood around x with radius 1. The results in Table 3 show, that you can't have your cake and eat it too. In Elastix there is a trade-off between the TRE and the smoothness of the transformation. $\sigma_{\text{B-spline}}$ can be tuned for a small TRE accepting a less smooth transformation or it is chosen such, that the resulting transformation is smooth, but with a higher TRE. However, our method reaches significantly smoother transformations compared to Elastix with a similar TRE. Since we regularize for diffeomorphic transformations it was expected, that compared to Elastix using a small smoothness parameter, no singularities will be present in the results. As soon as we increase $\sigma_{\text{B-spline}}$ such that the transformations are as smooth as in our method, the TRE and DICE performance drops dramatically for Elastix.

To quantify the efficiency of our filter caching approach, we performed the experiments once more, but included 21 landmarks provided in the POPI dataset. The landmark uncertainty was set to $\sigma = 0.02$. For comparison, the exact method, which combines the separable filtering and the landmarks, does no compression. In Table 4, the average resources needed for each experiment are listed. As expected, for the Tucker decomposition, slightly more CPU time is needed. However, it is negligible compared to the memory savings reached with this compression. Furthermore, the approximation of the local filter kernels is nearly perfect resulting in a very small loss of accuracy. The most CPU intensive part in each experiment is the 1st iteration, because initially, all local filter responses have to be calculated. Without the

	$R = 1$	$R = 2$	$R = 3$	$R = 4$	exact method	Voxels
CPU time for level 0 (h)	0.21	0.41	0.61	0.83	3.69	87290
CPU time for level 1 (h)	1.81	3.54	5.18	7.01	25.71	712530
CPU time for level 2 (h)	13.88	27.28	40.23	53.19	159.13 ≈ 1 week	5798625

Table 2 This table provides information about the computational time needed for the different experiments. To show the time complexity depending on the number of voxels in the image, we calculated the experiments on 3 different scale levels and performed 100 iterations per level. The timing is given in average CPU hours needed to perform one registration.

method	landmarks	TRE	singularities	$\upsilon(A)$	DICE	CPU time (h)
rigid alignment	n/a	1.673 ± 1.062	0	n/a	0.959	n/a
exponential kernel ($R = 1$)	excl.	1.419 ± 1.059	0	0.094	0.966	15.51
exponential kernel ($R = 2$)	excl.	1.027 ± 0.801	0	0.117	0.974	30.24
exponential kernel ($R = 3$)	excl.	0.588 ± 0.158	0	0.146	0.981	45.99
exponential kernel ($R = 4$)	excl.	0.584 ± 0.149	0	0.148	0.981	59.49
exponential kernel (exact)	excl.	0.579 ± 0.103	0	0.198	0.981	188.47
elastix ($\sigma_{\text{B-spline}} = 4$)	excl.	0.540 ± 0.104	6143.8	0.861	0.987	2.24
elastix ($\sigma_{\text{B-spline}} = 16$)	excl.	0.601 ± 0.150	0	0.398	0.981	2.25
elastix ($\sigma_{\text{B-spline}} = 64$)	excl.	0.819 ± 0.308	0	0.154	0.974	2.23
exponential kernel (Tucker)	incl.	0.484 ± 0.127	0.4	0.144	0.981	80.1
elastix ($\sigma_{\text{B-spline}} = 4, w = 1$)	incl.	0.411 ± 0.086	6255.8	0.859	0.987	2.42
elastix ($\sigma_{\text{B-spline}} = 4, w = 64$)	incl.	0.024 ± 0.001	2565.9	0.624	0.987	2.41
elastix ($\sigma_{\text{B-spline}} = 16, w = 1$)	incl.	0.535 ± 0.089	869.8	0.544	0.984	2.95
elastix ($\sigma_{\text{B-spline}} = 16, w = 64$)	incl.	0.159 ± 0.042	948.0	0.534	0.984	2.96
elastix ($\sigma_{\text{B-spline}} = 64, w = 1$)	incl.	0.814 ± 0.303	0	0.156	0.974	2.35
elastix ($\sigma_{\text{B-spline}} = 64, w = 64$)	incl.	0.796 ± 0.300	0	0.154	0.974	2.34

Table 3 In this table we show the performance evaluation of the POPI experiment with and without landmarks. In our method we used the exponential kernel as regularizer. For comparison, in elastix we used the B-spline transformation model.

Ξ	Scale level	Overall CPU time (h)	CPU time 1st iteration (h)	Memory (GB)	Masked voxels	Accuracy loss
without	0	4.05	2.32	10.17	29249	0
	1	9.85	2.63	18.58	54933	0
	2	58.89	5.60	28.61	84424	0
Tucker	0	5.12	2.51	1.11	29249	$6.46 \cdot 10^{-5}$
	1	12.48	3.22	1.44	54933	$2.32 \cdot 10^{-5}$
	2	66.84	6.48	2.55	84424	$9.69 \cdot 10^{-6}$

Table 4 The table provides information about resources needed to perform the experiments. For the upper part of the table, no Tucker decomposition of the local filter kernels was performed. In the lower part, the local filters were decomposed using the Tucker decomposition to save memory during caching. H_x was discretized on a 23^3 neighborhood. The approximation ranks were set to $P = Q = R = 5$. The masked voxels are the ones, which are in the support of the landmarks. The support limit ξ was set to 10^{-6}.

caching scheme therefore, the overall CPU time would explode to CPU weeks.

To compare our hybrid results with Elastix, we performed the hybrid B-spline registration twice, using a small resp. a large weight w for the landmark cost term (see Table 3). A large weight results in a smaller TRE while the overall smoothness decreases. Several singularities are present in the Elastix results, while the singularities in our method are negligible. The major advantage of our method becomes apparent with the overall smoothness. Despite the landmark consideration it is much higher than in the Elastix experiment.

The TRE could be decreased, but regarding the small uncertainty on the landmarks we would have expected a smaller landmark error. This discrepancy originates from the discretization of the mean transformation μ_X, which in this experiment leads to a TRE drift of 0.264 ± 0.121,

$$\text{TRE}_{\text{drift}} = \text{TRE}(\mu_{X\,\text{discretized}}) - \text{TRE}(\mu_{X\,\text{exact}}).$$

Compared to the experiment in Section 4.2 where the resolution is about twice as high the discretization error results in a TRE drift of 0.056 ± 0.002, which is negligible. Therefore, the mean transformation should be discretized on a finer grid.

A note on the parallelization Since our method is based on image filtering, it is well suited to perform the filtering for each voxel in parallel. Hence, the standard parallelization framework of ITK could directly be used to speed up the

Level	speedup (landmarks excl.)	speedup (landmarks incl.)
0	13.8	11.3
1	17.7	12.4
2	17.9	14.9

Table 5 The speedups were calculated by $\frac{\text{Time (1 process)} - \text{Time (24 processes)}}{\text{Time (24 processes)}}$. Therefore, an optimal full parallelizable algorithm would have a speedup of 23, using 24 processes.

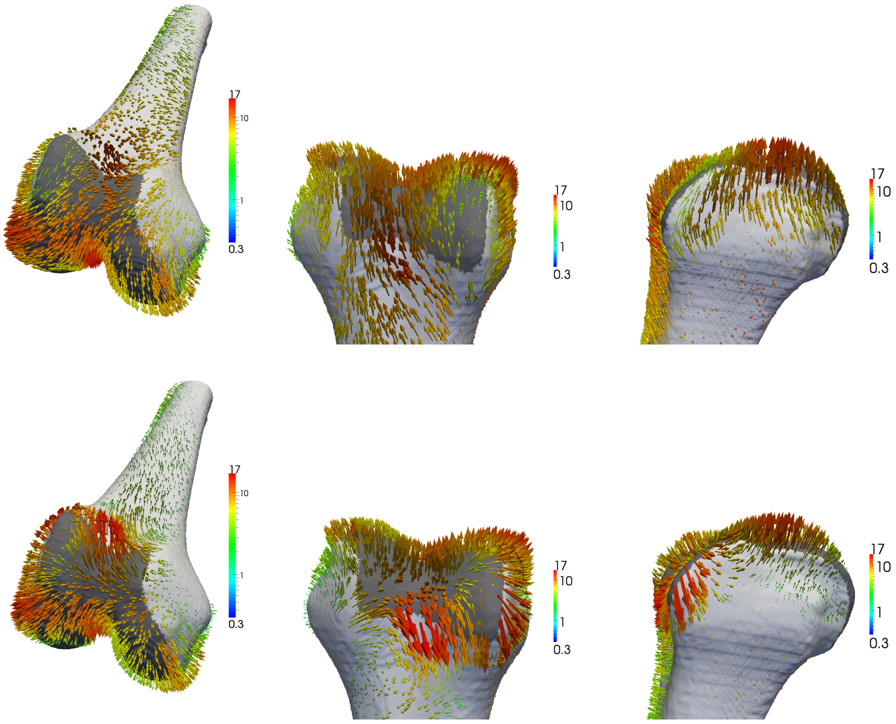

Fig. 5 The figures show the warp field resampled on the reference's surface, depicted as arrows. First row: registration was performed without landmarks. Second row: registration was performed including the landmarks.

calculations. We performed the experiments with 24 processes and reached an average speedup between 15 and 18 as listed in Table 5. Because the landmarks are not evenly distributed over the image domain, the work load is also not evenly distributed to the processes. Therefore, we reached a lower speedup in the hybrid registration experiment. The actual time needed to perform the experiments is the CPU time listed in Table 1 and 2 divided by the speedups listed in Table 5. For example, calculating the 9 registrations on level 2, the exact method took us 2.5 days instead of 8.5 weeks.

4.2 Patellar Surface Registration

We further performed a 3D experiment registering two femur shapes. The challenge with this kind of data is that the border of the patellar surface is potentially hard to recognize and its variation can be quite large, such that an accurate registration of the patellar surfaces using fully automatic algorithms is difficult. We obtained the patellar surfaces of the target and reference bone from an expert. By incorporating well-chosen landmarks, we can force our algorithm to register even the patellar surface correctly. The shapes were represented as signed distance images of $353 \times 327 \times 491$ voxels (isotropic spacing 0.57mm^3) and registered on 5 scale levels. For k we used the Gaussian kernel with $\sigma_g = 1$ and a landmark uncertainty of $\sigma = 0.3 \cdot 10^{-3}$. We approximated the landmark kernel k_X with $P = Q = R = 5$. For illustration, we performed the experiment once without landmarks and once including the landmarks.

In Figure 5, the warp fields are shown resampled on the bone surface depicted as arrows. Especially at the upper border of the patellar surface, one can see the strong impact of the landmarks. In Figure 6, we plotted the warped reference shape including the dark gray marked part. Without considering the landmarks, the border of the patellar surface is clearly misaligned, while it is correctly registered when the landmarks are incorporated.

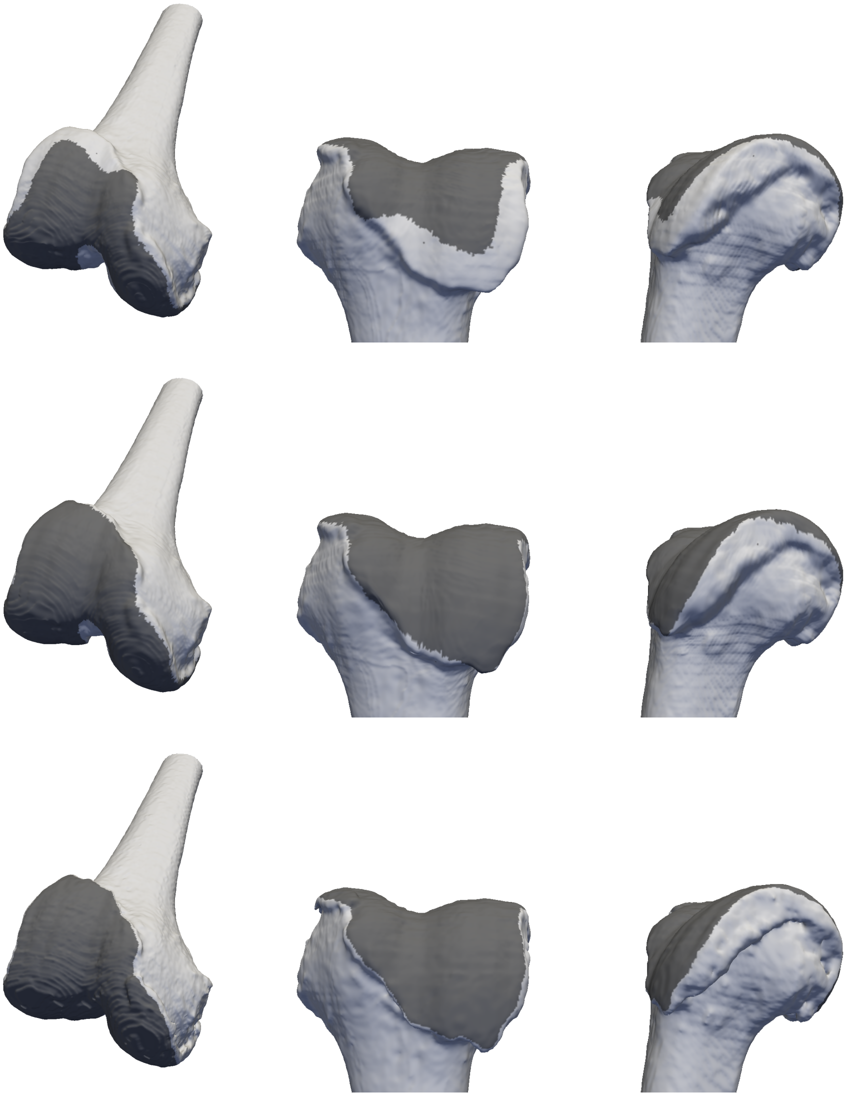

Fig. 6 The figures show the warped reference shape including the colored patellar surface. First row: registration was performed without landmarks. Second row: registration was performed including the landmarks. Third row: ground truth target shape.

method	TRE	singularities	$v(A)$	DICE
rigid aligned	12.268 ± 13.627	0	0	0.797
diffeomorphic demons	13.722 ± 31.546	0	0.250	0.989
elastix ($\sigma = 1, w = 1$)	**0.005 ± 1.199**	20703	0.314	0.973
our approach	1.225 ± 1.328	**48**	**0.251**	**0.990**

Table 6 Quantitative measures of the femur example.

We performed the same experiment with Elastix and summarized the results in Table 6. The parameters of the hybrid B-spline registration were tuned concerning the TRE and DICE performance measures. While Elastix brings the TRE down to nearly zero a very large amount of singularities are present in the resulting transformation and the dice coefficient is rather low. Our method reaches a small TRE as well. Furthermore, the singularity count is very low, the DICE quite high and the displacement field smooth.

4.3 Smooth Mean Displacement

Since we can force multiple reference landmarks to match one single target landmark by setting σ equal to zero, μ_X is

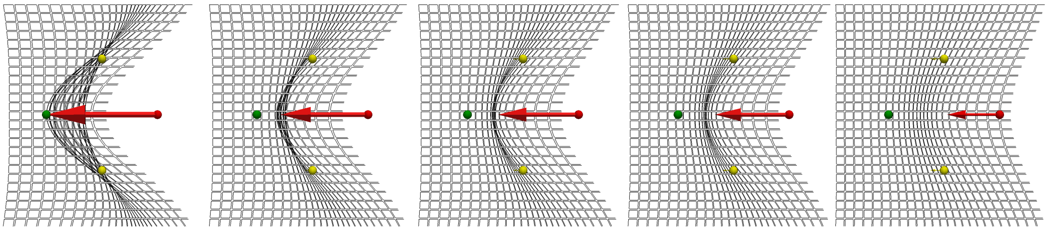

Fig. 7 Transformed grid (200px^2 and isotropic spacing of $0.1mm^2$) with mean displacement using the Gaussian kernel ($\sigma_g = 6$). There are 3 landmarks defined as reference and target points (red=reference, green=target, the yellow ones are equal for both). The uncertainty on the landmarks is increased for the experiments from left to right ($\sigma = 0, 0.5 \cdot 10^{-3}, 0.75 \cdot 10^{-3}, 0.1 \cdot 10^{-2}, 0.25 \cdot 10^{-2}$). The arrows illustrate to which location a point is transformed by the displacement field.

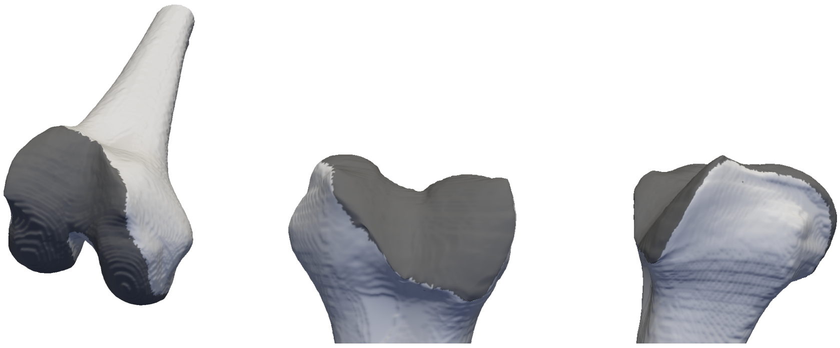

Fig. 8 The figures show the warped reference shape by the mean transformation using the multiscale kernel $\tilde{k}$ with 5 scale levels. Overall the shape looks the same as the reference except in the regions of the landmarks. There we have a smooth transformation to the target.

not guaranteed to be invertible. In Figure 7, an artificial example is shown where a grid is transformed by the mean displacement using different σ. Setting σ equal to zero, or too small, results in unfavorable folds and barely make sense in real world medical problems. Therefore, in our patellar surface experiment, we have chosen the parameters such that folds in μ_X hardly ever occur. The mean transformed reference shape is shown in Figure 8, where no holes can be identified on the surface.

Nevertheless, an inverse transformation can be obtained using the fixed-point approach of Chen *et al.* [7], where the inverse is iteratively approximated. An entirely different approach could be to perform diffeomorphic point matching [2,9,12,13] for obtaining a invertible mean displacement. This will be addressed in future work.

5 Conclusion

In this paper, we implemented an efficient variational image registration framework, where a large variety of positive definite kernels can be used for regularization. Compared to standard approaches, we are able to accurately approximate separable filters for nonseparable regularizers in order to relax the computational demands. With less than a third of the computational effort, we approximate the true regularization with a very small loss of accuracy, while the rank-1 approximation is three times faster but results in a accuracy loss which is one order of magnitude larger. Furthermore, using an efficient nonstationary filtering scheme, we allow for location-dependent regularization. This enables us to perform hybrid landmark and image registration by utilizing the landmark kernel which incorporates landmark displacements as prior knowledge. For this purpose, accepting little more computational time, we can reduce the memory usage by at least one order of magnitude. Additionally, we added the diffeomorphic constraint on the resulting transformation. Its approximation does not significantly change the optimization scheme. The comparison with the hybrid B-spline registration shows, that our method results in smoother displacement fields even if landmark displacements are incorporated. We also discussed challenges associated with the invertibility of the landmark based transformation. An additional prior on this transformation, which ensures invertibility, similar to [2,12], would further improve the registration. This will be addressed in future work.

A Matrix Products

A.1 Khatri-Rao product

Given a matrix $A \in \mathrm{I\!R}^{m \times q}$ and a matrix $B \in \mathrm{I\!R}^{n \times q}$, the Khatri-Rao product of A and B is the matching column-wise Kronecker product

$$A \odot B = (a_1 \bullet b_1, a_2 \bullet b_2, \cdots a_q \bullet b_q) \in \mathrm{I\!R}^{mn \times q}.$$

A.2 Hadamard product

Given a matrix $A \in \mathrm{I\!R}^{m \times n}$ and a matrix $B \in \mathrm{I\!R}^{m \times n}$, the Hadamard product of A and B is the point-wise matrix product

$$A \star B = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \cdots & a_{mn}b_{mn} \end{pmatrix} \in \mathrm{I\!R}^{m \times n}.$$

A.3 Kronecker product

Given a matrix $A \in \mathrm{I\!R}^{m} \times \mathrm{I\!R}^{n}$ and a matrix $B \in \mathrm{I\!R}^{q} \times \mathrm{I\!R}^{r}$, the Kronecker product of A and B is given as

$$A \bullet B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \in \mathrm{I\!R}^{mq \times nr}.$$

References

1. Arsigny, V., Commowick, O., Pennec, X., Ayache, N.: A log-euclidean framework for statistics on diffeomorphisms. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006 pp. 924–931 (2006)
2. Avants, B., Schoenemann, P., Gee, J.: Lagrangian frame diffeomorphic image registration: Morphometric comparison of human and chimpanzee cortex. Medical Image Analysis **10**(3), 397–412 (2006)
3. Beg, M., Miller, M., Trouvé, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. International Journal of Computer Vision **61**(2), 139–157 (2005)
4. Beuthien, B., Kamen, A., Fischer, B.: Recursive green's function registration. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010 pp. 546–553 (2010)
5. Bro-Nielsen, M., Gramkow, C.: Fast fluid registration of medical images. In: Proceedings of the 4th International Conference on Visualization in Biomedical Computing, pp. 267–276. Springer-Verlag (1996)
6. Cahill, N.D., Noble, J.A., Hawkes, D.J.: A demons algorithm for image registration with locally adaptive regularization. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009, pp. 574–581. Springer (2009)
7. Chen, M., Lu, W., Chen, Q., Ruchala, K.J., Olivera, G.H.: A simple fixed-point approach to invert a deformation field. Medical physics **35**, 81 (2008)
8. Christensen, G.E., Johnson, H.J.: Consistent image registration. Medical Imaging, IEEE Transactions on **20**(7), 568–582 (2001)
9. Christensen, G.E., Yin, P., Vannier, M.W., Chao, K., Dempsey, J., Williamson, J.F.: Large-deformation image registration using fluid landmarks. In: Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium, pp. 269–273. IEEE (2000)
10. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications **21**(4), 1253–1278 (2000)
11. Evans, L.: Partial differential equations. Graduate studies in mathematics. American Mathematical Society (1998)
12. Girdziušas, R., Laaksonen, J.: Gaussian process regression with fluid hyperpriors. In: Neural Information Processing, vol. 3316, pp. 567–572. Springer (2004)
13. Guo, H., Rangarajan, A., Joshi, S.: Diffeomorphic point matching. In: Handbook of Mathematical Models in Computer Vision, pp. 205–219. Springer (2006)
14. Haber, E., Heldmann, S., Modersitzki, J.: A scale-space approach to landmark constrained image registration. In: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision, pp. 612–623. Springer-Verlag (2009)
15. Harshman, R.: Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis (1970)
16. Johnson, H., Christensen, G.: Consistent landmark and intensity-based image registration. IEEE Transactions on Medical Imaging **21**(5), 450–461 (2002)
17. Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P., et al.: Elastix: a toolbox for intensity-based medical image registration. IEEE transactions on medical imaging **29**(1), 196–205 (2010)
18. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM review **51**(3), 455–500 (2009)
19. Long, Z., Yao, L., Peng, D.: Fast non-linear elastic registration in 2d medical image. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004 pp. 647–654 (2004)
20. Lu, H., Cattin, P., Reyes, M.: A hybrid multimodal non-rigid registration of mr images based on diffeomorphic demons. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society–EMBC 2010, pp. 5951–5954. IEEE (2010)
21. Lüthi, M., Jud, C., Vetter, T.: Using landmarks as a deformation prior for hybrid image registration. In: Proceedings of the 33rd international conference on Pattern recognition, pp. 196–205. Springer-Verlag (2011)
22. McOwen, R.: Partial differential equations: methods and applications. Tsinghua University Press (1996)
23. Modersitzki, J.: Numerical Methods for Image Registration. Oxford university press (2004)
24. Opfer, R.: Multiscale kernels. Advances in computational mathematics **25**(4), 357–380 (2006)
25. Papademetris, X., Jackowski, A., Schultz, R., Staib, L., Duncan, J.: Integrated intensity and point-feature nonrigid registration. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004 pp. 763–770 (2004)
26. Papenberg, N., Olesch, J., Lange, T., Schlag, P., Fischer, B.: Landmark constrained non-parametric image registration with isotropic tolerances. Bildverarbeitung für die Medizin 2009 pp. 122–126 (2009)
27. Rasmussen, C.: Gaussian processes in machine learning. Advanced Lectures on Machine Learning pp. 63–71 (2004)
28. Schmidt-Richberg, A., Ehrhardt, J., Werner, R., Handels, H.: Diffeomorphic diffusion registration of lung ct images. Medical Image Analysis for the Clinic-A Grand Challenge–MICCAI 2010 (2010)
29. Schölkopf, B., Steinke, F., Blanz, V.: Object correspondence as a machine learning problem. In: Proceedings of the 22nd international conference on Machine learning, pp. 776–783. ACM (2005)
30. Sotiras, A., Paragios, N., et al.: Deformable image registration: A survey (2012)

31. Stefanescu, R., Pennec, X., Ayache, N.: Grid powered nonlinear image registration with locally adaptive regularization. Medical image analysis **8**(3), 325–342 (2004)
32. Steinke, F., Schölkopf, B.: Kernels, regularization and differential equations. Pattern Recognition **41**(11), 3271–3286 (2008)
33. Thirion, J.: Image matching as a diffusion process: an analogy with maxwell's demons. Medical image analysis **2**(3), 243–260 (1998)
34. Tucker, L.: Implications of factor analysis of three-way matrices for measurement of change. Problems in measuring change pp. 122–137 (1963)
35. Twining, C.J., Marsland, S.: Constructing diffeomorphic representations of non-rigid registrations of medical images. In: Information Processing in Medical Imaging, pp. 413–425. Springer (2003)
36. Vandemeulebroucke, J., Sarrut, D., Clarysse, P.: The popi-model, a point-validated pixel-based breathing thorax model. In: Proceedings of the 15th International Conference on the Use of Computers in Radiation Therapy–ICCR 2007 (2007)
37. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Non-parametric diffeomorphic image registration with the demons algorithm. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007 pp. 319–326 (2007)
38. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Symmetric log-domain diffeomorphic registration: A demons-based approach. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008 pp. 754–761 (2008)
39. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N., et al.: Diffeomorphic demons using itk's finite difference solver hierarchy. In: Insight Journal–ISC/NA-MIC Workshop on Open Science at MICCAI (2007)
40. Wang, H., Dong, L., O'Daniel, J., Mohan, R., Garden, A., Ang, K., Kuban, D., Bonnen, M., Chang, J., Cheung, R.: Validation of an accelerated'demons' algorithm for deformable image registration in radiation therapy. Physics in Medicine and Biology **50**(12), 2887–2905 (2005)