Journal of Mathematical Imaging and Vision (2019) 61:443–457
https://doi.org/10.1007/s10851-018-0854-5

Error-Controlled Model Approximation for Gaussian Process Morphable Models

Jürgen Dölz[1] · Thomas Gerig[1] · Marcel Lüthi[1] · Helmut Harbrecht[1] · Thomas Vetter[1]

Received: 12 March 2018 / Accepted: 9 October 2018 / Published online: 24 October 2018

Abstract
Gaussian Process Morphable Models (GPMMs) unify a variety of non-rigid deformation models for surface and image registration. Deformation models, such as B-splines, radial basis functions, and PCA models are defined as a probability distribution using a Gaussian process. The method depends heavily on the low-rank approximation of the Gaussian process, which is mandatory to obtain a parametric representation of the model. In this article, we propose the use of the pivoted Cholesky decomposition for this task, which has the following advantages: (1) Compared to the current state of the art used in GPMMs, it provides a fully controllable approximation error. The algorithm greedily computes new basis functions until the user-defined approximation accuracy is reached. (2) Unlike the currently used approach, this method can be used in a black-box-like scenario, whereas the method automatically chooses the amount of basis functions for a given model and accuracy. (3) We propose the Newton basis as an alternative basis for GPMMs. The proposed basis does not need an SVD computation and can be iteratively refined. We show that the proposed basis functions achieve competitive registration results while providing the mentioned advantages for its computation.

Keywords Non-rigid registration · Gaussian process · Image registration · Shape registration · Pivoted Cholesky · Gaussian Process Morphable Models · GPMM · Statistical shape modeling

1 Introduction

A common approach in medical image analysis and computer vision is analysis by synthesis: An image is analyzed by synthesizing it using a generative model [13,36]. The resulting model-parameters are then used to understand the content of the target image. Popular examples of analysis by synthesis in medical image analysis are atlas (or template) matching approaches [8,20,33], or statistical shape and appearance models [6,7,19]. The main idea behind all these methods is that any object $\Gamma_T \subset \mathbb{R}^d$ to be analyzed can be written with respect to a reference object $\Gamma_R \subset \mathbb{R}^d$ which is deformed by suitable deformation $\mathbf{u}^\star \colon \Gamma_R \to \mathbb{R}^d$. For better readability, we denote vector-valued quantities by bold-faced lower case letter, whereas we denote matrix-valued quantities by bold-faced capital letters.

For given image or surface representations Γ_R and Γ_T, we are interested in finding the corresponding deformation field $\mathbf{u}^\star$ that deforms Γ_R such that it matches Γ_T, i.e., it holds

$$\Gamma_T = \{\mathbf{x} + \mathbf{u}^\star(\mathbf{x}) \colon \mathbf{x} \in \Gamma_R\}.$$

Aiming at modeling non-rigid deformations, the crucial question for practical applications is how to model a family of possible deformations $\mathbf{u}$, which contains (a good approximation to) $\mathbf{u}^\star$.

Recently, Lüthi et al. proposed Gaussian Process Morphable Models (GPMM), which model the deformations as a Gaussian process $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$ with mean function $\boldsymbol{\mu} \colon \Omega \to \mathbb{R}^d$ and covariance (or kernel) function $\mathbf{K} \colon \Omega \times \Omega \to \mathbb{R}^{d\times d}$ [11,26,27], see Fig. 1 for a visual overview. In this view, all the above-mentioned models correspond to special choices of the covariance function and it becomes easy to combine characteristics of the individual models or to incorporate additional prior knowledge by, for example, enforcing mir-

Jürgen Dölz and Thomas Gerig share first authorship. Helmut Harbrecht and Thomas Vetter share last authorship. Author names in alphabetical order .

✉ Thomas Gerig
thomas.gerig@unibas.ch

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

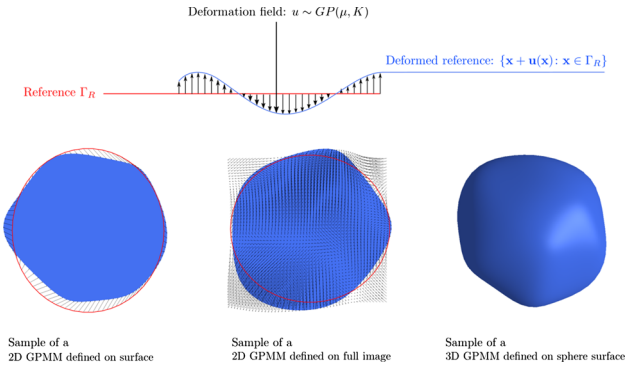

Fig. 1 In this figure, an overview about how the Gaussian Process Morphable Models are used on different domains. **Bottom left** A reference 2D mesh (colored in red) is deformed using a deformation field that is defined on the mesh itself. **Bottom middle** The GPMM deformation model is defined on the whole 2D image domain. **Bottom right** A random sample of a GPMM defined on a 3D mesh with a spherical shape (Color figure online)

ror symmetries [25], landmark constraints [24], or by making models spatially varying [11]. The key aspect of the method in [26] is that it allows modeling the expected deformations for individual registration tasks, which makes it easy to tailor the problem to a specific data set. This is generally done in three steps:

1. **Prior Model Building** Define a family of deformations $\mathbf{u}$ using a Gaussian process $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$ with a kernel function $\mathbf{K}$. The model can be customized by combining kernel building blocks with different properties to match the expected family of deformation functions for a given registration task.
2. **Model Approximation** The family of deformations $\mathbf{u}$ generated by a Gaussian process $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$ can often be well approximated using a truncated Karhunen–Loève-expansion [5], i.e., the family of deformations $\mathbf{u}$ can approximatively be described as

$$\mathbf{u} \approx \mathbf{u}_M := \boldsymbol{\mu} + \sum_{i=1}^{M} \alpha_i \sqrt{\lambda_i} \boldsymbol{\phi}_i, \quad \alpha_i \sim \mathcal{N}(0, 1).$$

   The corresponding pairs $(\lambda_i, \boldsymbol{\phi}_i)$ are given as eigenpairs corresponding to the M largest eigenvalues of an integral operator associated to the covariance function $\mathbf{K}$. Using this representation, any deformation $\mathbf{u}_M$ is given as a linear combination of the eigenfunctions and parametrized by finitely many parameters $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]$:

$$\mathbf{u}_M(\boldsymbol{\alpha}, \cdot) = \boldsymbol{\mu}(\cdot) + \sum_{i=1}^{M} \alpha_i \sqrt{\lambda_i} \boldsymbol{\phi}_i(\cdot).$$

3. **Model Fitting/Registration** The parameterized model $\mathbf{u}_M(\boldsymbol{\alpha}, \cdot)$ is used to find the parameters $\boldsymbol{\alpha}$ to match the deformed reference Γ_R to a given target Γ_T.

For the most optimal use of the prior knowledge incorporated in $\mathbf{K}$, the family of possible deformations $\mathbf{u}$ must be approximated as well as possible by the truncated Karhunen–Loève expansion $\mathbf{u}_M$. However, the approximation method proposed in [26] has three major disadvantages: (1) The Nyström method lacks a rigorous control of the approximation accuracy. (2) The method relies on parameters, such as the sampling of points on the reference. These parameters depend on the model to be approximated and need an experienced user to be chosen. (3) A refinement of the model (adding more basis functions) requires a re-computation of the whole model. The mentioned issues have a big impact on the practical usability of the registration framework, where a feedback about the approximation accuracy of the modeled prior is important to choose the model. Our contribution in this work is to introduce the use of the pivoted Cholesky decomposition for this task, see [9,18], which allows the computation of a Karhunen–Loève-expansion up to a prescribed accuracy with a rigorous error control and to show its properties and advantages compared to the Nyström method, which was originally proposed for this purpose by Lüthi et al. [26]:

1. We enable the Gaussian process registration framework to approximate models with a rigorous black-box error control. In particular, the parameter M is chosen adaptively for a user-defined tolerance. This is a major advantage over the Nyström method, which lacks feedback about the actual approximation accuracy and has several parameters to guess heuristically.
2. The pivoted Cholesky decomposition follows a greedy-type strategy, where the corresponding basis functions generate a subspace which is equal to the subspace spanned by the truncated Karhunen–Loève-expansion. We propose therefore to use the greedy-type basis from the pivoted Cholesky decomposition instead of the eigenfunctions for the representation of $\mathbf{u}_M$. Thanks to the greedy-type strategy, the subspace can easily be enlarged by adding additional basis functions, if required. We show that the proposed basis functions lead to competitive results. However, in contrast to the Nyström method, the basis is refinable and is computationally less intensive.

The paper is structured as follows: In Sect. 3, we recapitulate the fundamentals of Gaussian Process Morphable Models and the Nyström method which is currently used for its discretization. We particularly discuss the drawbacks of

the Nyström method and how this affects registration results. In Sect. 4, we introduce the pivoted Cholesky decomposition for the low-rank factorization of covariance matrices and discuss how it can be used to compute Karhunen–Loève expansions for GPMMs. Section 5 introduces the new greedy-type basis and contains also a discussion why the registration results with this basis should behave similar to the registration results with the eigenbasis from the Karhunen–Loève expansions. The numerical experiments in Sect. 6 are concerned with the approximation quality of the introduced method, whereas the numerical experiments in Sect. 7 demonstrate that the introduced greedy-type basis for GPMM is competitive to state-of-the-art registration methods. Finally, in Sect. 8, we draw our conclusions.

The core algorithms of the method proposed are implemented in the open-source project **Scalismo** [35].

2 Related Work

The Gaussian Process Morphable Model (GPMM) framework [26], on which our work is based on, can be seen as the unification of different concepts. On the one hand, statistical shape models (SSM) can be extended with additional flexibility using kernel functions. On the other hand, the models are used as statistical priors for surface and image registration. The work by Grenander et al. [13] contains similarities to the GPMM approach, as they propose to use a basis function representation to span the model space. However, in all these works, the basis functions have to be known analytically [3], or the initial model needs to be of finite rank [21]. In [22] and also in [28], the covariance function is not approximated, which is only feasible for compact kernels with small correlation lengths. In the context of Gaussian processes and the computation of low-rank approximations to covariance matrices, the pivoted Cholesky decomposition is an established algorithm, cf., e.g., [4,9,18,31]. Having the low-rank approximation at hand, it has been shown in [18] that the eigenpairs of the covariance matrix can be obtained approximately by solving an eigenvalue problem which has the dimension of the rank of the low-rank approximation. Whereas these works are restricted to the low-rank approximation of matrices, is has been analyzed in [17] how the continuous eigenvalue problem can be efficiently discretized and solved by the pivoted Cholesky decomposition by the use of finite elements. In [30] the authors employ the pivoted Cholesky decomposition to compute a low-rank factorization of kernel functions in terms of function skeletons. Since one can add another basis function to the low-rank factorization without recomputing the others, they call the obtained basis the "Newton" basis, in analogy to Newton interpolation. This kernel-based approach has been extended in [34] to compute a Karhunen–Loève expansion if radial basis functions are used for the spatial discretization.

3 Fundamentals of Gaussian Process Morphable Models

3.1 Modeling Deformation Priors

Gaussian Process Morphable Models (GPMM), which have been introduced in Lüthi et al. [26] allow to define prior models for registration analytically in advance using a matrix-valued Gaussian process. The vector fields, which are defined continuously on a domain $\Omega \subset \mathbb{R}^d$, act as the non-rigid transformation of the reference object $\Gamma_R \subset \Omega$, which could be any geometric object or grid defined in Ω. The GPMM are used on different domains, such as two- or three-dimensional surfaces and grid-like structures, which is also visualized in Fig. 1.

A Gaussian process $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$ is defined by its mean function $\boldsymbol{\mu}\colon \Omega \to \mathbb{R}^d$ and its covariance function $\mathbf{K}\colon \Omega \times \Omega \to \mathbb{R}^{d\times d}$, see [31]. Then, any deformation $\mathbf{u}$ sampled from $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$, gives rise to a new surface Γ by warping the reference surface Γ_R:

$$\Gamma = \{\mathbf{x} + \mathbf{u}(\mathbf{x})\colon \mathbf{x} \in \Gamma_R\}.$$

Similar to the PCA representation of a statistical shape model, a Gaussian process $\mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$ can be represented in terms of an orthonormal set of basis functions $\{\boldsymbol{\phi}_i\}_{i=1}^{\infty}$

$$\mathbf{u}(\mathbf{x}, \boldsymbol{\alpha}) \sim \boldsymbol{\mu}(\mathbf{x}) + \sum_{i=1}^{\infty} \alpha_i \sqrt{\lambda_i}\boldsymbol{\phi}_i(\mathbf{x}),\ \alpha_i \in \mathcal{N}(0,1), \tag{1}$$

where $(\lambda_i, \boldsymbol{\phi}_i)$ are the eigenpairs of the integral operator

$$\mathcal{T}_{\mathbf{K}}\mathbf{f}(\cdot) := \int_{\Omega} \mathbf{K}(\cdot, \mathbf{x})\mathbf{f}(\mathbf{x})\, \mathrm{d}\rho(\mathbf{x}) \tag{2}$$

with $\rho(\mathbf{x})$ denoting a measure. The representation (1) is known as the Karhunen–Loève expansion of the Gaussian process [5].

Since the random coefficients, α_i are uncorrelated, the variance of $\mathbf{u}$ is given by the sum of the variances of the individual components. Consequently, the eigenvalue λ_i corresponds to the variance explained by the i-th component. This suggests that, if the λ_i decay sufficiently quickly, we can, instead of (1), use the low-rank approximation

$$\mathbf{u}_M(\mathbf{x}, \boldsymbol{\alpha}) \sim \boldsymbol{\mu}(\mathbf{x}) + \sum_{i=1}^{M} \alpha_i \sqrt{\lambda_i}\boldsymbol{\phi}_i(\mathbf{x}). \tag{3}$$

The resulting model is a finite dimensional, parametric model with M components, similar to a standard statistical model. The expected error of this approximation is given by the tail sum

$$\sum_{i=M+1}^{\infty} \lambda_i. \tag{4}$$

Estimates for the decay of the λ_i show that the tail sum is reasonably small even for small M, provided that the covariance function $\mathbf{K}$ is sufficiently smooth, cf. [14]. In particular, any valid positive semi-definite covariance function can be used.

3.2 Nyström Method

To compute the truncated Karhunen–Loève-expansion (3) for an approximate GPMM model, the eigenpairs of the integral operator (2) have to be computed, i.e., the continuous eigenvalue problem

$$\left(\mathcal{T}_{\mathbf{K}}\boldsymbol{\phi}_m\right)(\mathbf{x}) = \lambda_m \boldsymbol{\phi}_m(\mathbf{x}) \tag{5}$$

has to be solved, where $\mathcal{T}_{\mathbf{K}}$ is the integral operator (2) given by the covariance function. In order to solve the eigenvalue problem numerically, it has to be discretized, i.e., it has to be transformed into a finite dimensional problem

$$\mathbf{C}\boldsymbol{\phi}_{m,N} = \lambda_{m,N}\boldsymbol{\phi}_{m,N} \tag{6}$$

with $\boldsymbol{\phi}_{m,N} \in \mathbb{R}^N$ and $\mathbf{C} \in \mathbb{R}^{N\times N}$.

The current state of the art proposed in [26] is to use the Nyström method, which performs the discretization by a sampling approach. Therefore, for some random samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ drawn according to ρ, one approximates

$$\int_{\Omega} \mathbf{K}(\cdot, \mathbf{x})\mathbf{f}(\mathbf{x})\, \mathrm{d}\rho(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{K}(\cdot, \mathbf{x}_i)\mathbf{f}(\mathbf{x}_i), \tag{7}$$

cf. [31] and the references therein. Although the estimator can be inaccurate, it comes with minimal assumptions on the measure ρ, which makes it highly attractive for problems with little information. If the reference domain is for example given by a set of vertices, appropriate samples can be drawn from this set.

Evaluating (7) at the sample points and multiplying with N yields, similar to (6), the finite dimensional eigenvalue problem

$$\mathbf{C}_{\text{Nystr}}\boldsymbol{\phi}_{m,N} = N\lambda_m\boldsymbol{\phi}_{m,N} \tag{8}$$

with the matrix

$$\mathbf{C}_{\text{Nystr}} = \left[\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^{N}$$

and the point values

$$\left(\boldsymbol{\phi}_{m,N}\right)_i \approx \boldsymbol{\phi}_m(\mathbf{x}_i), \quad i = 1, \ldots, N.$$

Combining (2), (5) and (7), the eigenfunctions can then be evaluated at any given point by

$$\boldsymbol{\phi}_m(\mathbf{x}) \approx \lambda_{m,N}^{-1}\sum_{i=1}^{N} \mathbf{K}(\mathbf{x}, \mathbf{x}_i)\left(\boldsymbol{\phi}_{m,N}\right)_i. \tag{9}$$

Probabilistic error bounds for the eigenpairs exist and show that the accuracy increases with the number of sample points, cf. [32].

3.3 Accuracy of the Nyström Method

The Nyström method has two major drawbacks which cause difficulties in building accurate GPMM approximations.

1. The number of required eigenfunctions is unknown: Even if it is known that the GPMM has a good low-rank approximation, the required number of eigenfunctions, i.e., the value of M, cannot be determined in advance or by the algorithm, but has to be chosen by the user.
2. Dependence on random samples: The common problem of randomized algorithms that there are no deterministic error bounds applies also for GPMM. In particular for GPMM, the sampling based approach might miss important features of the covariance function on small scales, which might be crucial for building accurate approximations.

These two points make it difficult to judge whether an insufficient registration result is caused by an inaccurate prior model. For a user, it is difficult to guess the right amount of basis functions and the right density of random sampling to make sure that the prior model accuracy is not the issue.

To overcome this problem, one may argue that a sufficiently large number of samples N and a sufficiently large number of eigenfunctions M will lead to a good approximation. However, whereas a large value of M may cause trouble in the optimization algorithm, the choice of $N \gg 1$ leads to a $N \times N$ dense eigenvalue problem, whose solution with deterministic algorithms has a complexity of $\mathcal{O}(N^3)$. In addition, for large N, the evaluation of the eigenfunctions (9) becomes computationally intensive for a large number of evaluation points. Using a randomized SVD, cf., e.g., [16] and the references therein, the complexity of the solution of the eigenvalue

problem can be lowered to $\mathcal{O}(MN^2)$. While this reduces the complexity of the solution of the eigenvalue problem, it introduces an additional probabilistic component to the algorithm and still requires assembling the full covariance matrix, which has a complexity in memory and computation time of $\mathcal{O}(N^2)$.

We thus require an algorithm satisfying the following requirements:

1. The algorithm should be completely deterministic, i.e., it should avoid random sampling and any other random input data.
2. Given a user-defined tolerance, the algorithm should automatically detect the number of required eigenfunctions, i.e., the value of M, such that the error is below that tolerance.
3. While the previous two requirements address the issues discussed at the beginning of this section, we require the algorithm to be computationally efficient without sacrificing accuracy.

While the first requirement can in principle be addressed by more advanced Nyström methods based on quadrature rules for the approximation of the integral in (7), see [15], these quadrature rules are either difficult to construct or lead to large system matrices, i.e., $N \gg 1$. Therefore, the following chapter shall discuss an algorithm fulfilling these requirements based on the pivoted Cholesky decomposition.

4 Pivoted Cholesky for GPMM Approximation

4.1 Low-Rank Approximation with the Pivoted Cholesky Decomposition

Although the system matrices of the discrete eigenvalue problems (6) and (8) are dense, they still have a low dimensional structure in the sense of low-rank approximations. In fact, the decay of the eigenvalues of the integral operator (2) has been well investigated in [14], where it has been proven that the eigenvalues satisfy the decay estimate

$$\lambda_m \leq Cm^{-2p/d}.$$

Here, p is some parameter which increases with the smoothness of the kernel. It is therefore evident that there exists a reasonably sized M such that the tail sum (4) is sufficiently small. Thus, assuming that the prescribed GPMM can be reasonably well approximated by a truncated Karhunen–Loève expansion implies that its correlation matrices can be well approximated by a truncated singular value decomposition and the matrices of the dense eigenvalue problems (6) and (8) have a low-rank structure.

A suitable tool to reveal the low-rank structure of a covariance matrix is the pivoted Cholesky decomposition. It only relies on the a priori knowledge that covariance matrices are positive semi-definite matrices, and does not require the precomputation of the matrix. Non-stationary covariance kernels are naturally included, since they yield positive semi-definite matrices.

Given some user-defined tolerance, the algorithm finds a low-rank factorization such that the approximation error is below that tolerance, measured in the trace-norm for positive semi-definite matrices. In particular, it automatically detects a rank M which is required to fulfill that tolerance. The pivoted Cholesky decomposition, cf. [9,18], is given in Algorithm 1.

Algorithm 1 The pivoted Cholesky decomposition

Input:

- Function $\mathbf{C}(i, j)$ computing entry (i, j) of matrix $\mathbf{C}$.
- Relative error tolerance $\varepsilon > 0$

Output:

- Required rank M for low-rank approximation
- Low-rank approximation $\mathbf{C}_M = \sum_{i=1}^{M} \boldsymbol{\ell}_i \boldsymbol{\ell}_i^\intercal$
- Approximation error $\operatorname{trace}(\mathbf{C} - \mathbf{C}_M) \leq \varepsilon \cdot \operatorname{trace}(\mathbf{C})$

Set $M = 1$
Set $\mathbf{d} = \left[\mathbf{C}(i, i)\right]_{i=1}^{N}$
Set $error = \|\mathbf{d}\|_{\ell^1}$
Set $\varepsilon = \varepsilon \cdot error$
Set $\boldsymbol{\pi} = [1, 2, \ldots, N]$
while $error > \varepsilon$ **do**
  Set $i = \arg\max\{d_{\pi_j} : j = M, M+1, \ldots, N\}$
  Swap π_M and π_i
  Set $\ell_{M,\pi_M} = \sqrt{d_{\pi_M}}$
  for $M + 1 \leq i \leq N$ **do**
    Compute
$$\ell_{M,\pi_i} = \left(\mathbf{C}(\pi_M, \pi_i) - \sum_{j=1}^{M-1} \ell_{j,\pi_M} \ell_{j,\pi_i}\right) / \ell_{M,\pi_M}$$
    Update $d_{\pi_i} = d_{\pi_i} - \ell_{M,\pi_M} \ell_{M,\pi_M}$
  end for
  Compute $error = \sum_{i=1}^{N} d_{\pi_i}$
  Set $M = M + 1$
end while

As can be seen from the algorithm, it requires a function computing the covariance matrix entries as input, rather than the covariance matrix itself. The computation of the full covariance matrix can thus be avoided since the algorithm determines automatically which entries of the matrix need to be computed, which are at most $\mathcal{O}(NM)$, rather than $\mathcal{O}(N^2)$. The complexity of the algorithm itself is $\mathcal{O}(NM^2)$.

The only input parameter the user has to control is the approximation accuracy. The error of the output low-rank approximation is guaranteed to be below that threshold. This gives the user a rigorous control over the approximation accu-

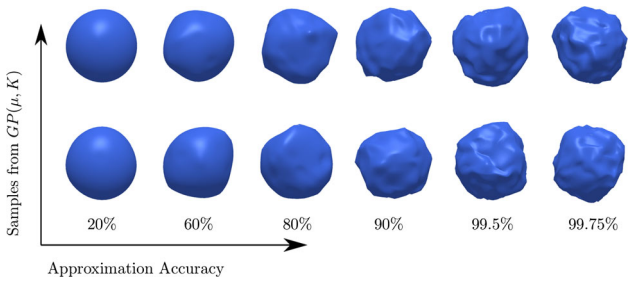

Fig. 2 Samples from a GPMM deforming a sphere reference mesh with a radius of 100 mm. The kernel has been chosen to be as a Gaussian kernel with a large length-scale, combined with a small local one (Gaussian kernel with $\sigma = 60$ and $scale = 30$ and a Gaussian kernel with $\sigma = 15$ and $scale = 10$). There is a visible increase in flexibility of the modeled deformations when increasing the approximation accuracy of the GPMM

racy of the GPMM as illustrated in Fig. 2. There, different random samples are shown from a GPMM approximated with iteratively refined accuracy.

Another interesting variant of the algorithm evolves when an upper limit to the rank M of the low-rank approximation is fixed. Since the output of the algorithm includes the error of the low-rank approximation, one can directly check whether the quality of the low-rank approximation is sufficient.

Having discussed a deterministic and efficient algorithm for the low-rank approximation of covariance matrices, we discuss next how this can be used for the computation of Karhunen–Loève expansions. Note, that this is relevant only for applications where the computation of the Karhunen–Loève expansion itself is an absolute necessity for further algorithms. We will show in Sect. 5 that for many registration algorithms a full KL-expansion is not necessary and the low-rank approximation from the pivoted Cholesky decomposition is sufficient. In this case, the post-processing step described in the remainder of this section can be omitted.

4.2 Computing Karhunen–Loève Expansions Using Low-Rank Approximations

Having a suitable algorithm for the computation of low-rank approximations at hand, one may replace the covariance matrices of the eigenvalue problems (6) and (8) by its low-rank factorizations $\mathbf{C} \approx \mathbf{L}_M \mathbf{L}_M^\intercal$. This yields an eigenvalue problem

$$\mathbf{L}_M \mathbf{L}_M^\intercal \mathbf{v}_{m,N} = \lambda_{m,N} \mathbf{v}_{m,N}.$$

Exploiting the fact that $\mathbf{L}_M \mathbf{L}_M^\intercal$ has the same eigenvalues as $\mathbf{L}_M^\intercal \mathbf{L}_M$, we obtain an equivalent eigenvalue problem

$$\mathbf{L}_M^\intercal \mathbf{L}_M \check{\mathbf{v}}_{m,N} = \lambda_{m,N} \check{\mathbf{v}}_{m,N}, \tag{10}$$

which has the reduced dimension $M \ll N$ and can thus be solved by standard eigensolvers for dense matrices. Approximations to the eigenvectors $\boldsymbol{\phi}_{m,N}$ are then given by

$$\boldsymbol{\phi}_{m,N} \approx \mathbf{v}_{m,N} = \mathbf{L}_M \check{\mathbf{v}}_{m,N}. \tag{11}$$

Thus, given a low-rank approximation $\mathbf{C} \approx \mathbf{L}_M \mathbf{L}_M^\intercal$, the solution of the dense small eigenvalue problem (10) and the computation of the eigenvectors by (11) can be accomplished in complexity $\mathcal{O}(M^3)$ and in $\mathcal{O}(NM^2)$, respectively. Since the computation of the small eigenvalue problem and the complexity of the pivoted Cholesky decomposition are both $\mathcal{O}(NM^2)$, the overall complexity for the computation of the Karhunen–Loève expansion is also $\mathcal{O}(NM^2)$.

Assuming that the GPMM can be reasonably well approximated by a truncated Karhunen–Loève expansion, M is reasonably small, and, thus, we can choose $N \gg 1$, for example in the range of millions. This allows for highly accurate spatial approximations. In the following, we discuss two suitable discretizations for surface and image registration. The discretizations are chosen illustratively and could be replaced by more advanced discretization schemes as discussed in the appendix.

4.2.1 Spatial Discretization for Surfaces

Because of its minimal assumptions, the Nyström approach is very popular in the machine learning community. It is also well suitable for surface to surface registration, where the shape is represented as a set of vertices and there is almost no structure available.

Thanks to using the pivoted Cholesky decomposition, we are no longer restricted by the size of the eigenvalue problem, and that the number of sampled points can be several million. Thus, we may choose to deterministically sample every vertex of the surface and can even include additional evaluation points in the deterministic sampling, such that the expensive interpolation (9) can be omitted. We can thus compute the eigenfunctions directly on all mesh and evaluation points and can thus completely avoid any random sampling on the surface and its corresponding uncertain error. Consequently, when using the pivoted Cholesky decomposition, we obtain a completely deterministic solution within the user-controlled error tolerance.

Instead of the quadrature formula (7) where each quadrature point has the same weight, one may also choose more sophisticated quadrature points, see, e.g., [15]. The pivoted Cholesky can also be employed in this case, as is discussed in Appendix "Advanced Nyström Schemes".

4.2.2 Spatial Discretization for Grid Approximations

Instead of the Nyström approach for the discretization of the eigenfunctions, one may also choose finite element schemes. Finite element schemes rely on finite dimensional function spaces which are defined on (possibly non-regular) grid structures. Thus, they are interesting for image registration, where the computational domain naturally provides such a structure.

A throughout analysis of a wide range of finite element schemes for the computation of Karhunen–Loève expansions using the pivoted Cholesky decomposition was given in [17]. For simplicity, we restrict the presentation in this section to piecewise trilinear finite elements, see Appendix "Finite Element Scheme on a Rectangular Grid", on a three-dimensional rectangular grid. The grid is assumed to consist of cells of size $h_1 \times h_2 \times h_3$, and we restrict ourselves to the case $\rho(\mathbf{x}) = 1$ and employ a well-known mass lumping scheme. This results in a function space spanned by N vector-valued basis functions $\boldsymbol{\varphi}_i$, $i = 1, \ldots, N$, in which the approximate eigenfunctions shall be represented.

Under these assumptions, the eigenvalue problem (5) discretized with said finite elements becomes

$$\mathbf{C}_{\text{FEM}} \boldsymbol{\phi}_{m,N} = h^2 \lambda_{m,N} \boldsymbol{\phi}_{m,N}. \tag{12}$$

Here, $h = \sqrt{h_1^2 + h_2^2 + h_3^2}$, with the system matrix

$$\mathbf{C}_{\text{FEM}} = \left[\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^{N}$$

and the approximate eigenfunctions

$$\boldsymbol{\phi}_m(\mathbf{x}) \approx \boldsymbol{\phi}_{m,N}(\mathbf{x}) = \sum_{i=1}^{N} \left(\boldsymbol{\phi}_{m,N}\right)_i \boldsymbol{\varphi}_i(\mathbf{x}).$$

Obviously, given the similarity of the eigenvalue problem (12) to the eigenvalue problems (6) and (8), the pivoted Cholesky decomposition can also be employed in the case of this finite element scheme to compute the Karhunen–Loève expansion. However, for more general finite element schemes, the structure of the eigenvalue problem (12) becomes more involved and we refer to Appendix "Advanced Finite Element Schemes" and [17] for more details.

A particular advantage of finite element schemes is that precise error estimates exist on how to choose the threshold of the pivoted Cholesky decomposition in dependence of the grid size h, see [17]. In the described case of piecewise trilinear finite elements, when the covariance function fulfills mild smoothness assumptions, the threshold should be chosen proportional to h^2. Then, the use of the pivoted Cholesky decomposition will not significantly change the approximation quality of the finite element scheme when the grid size is changed.

4.3 Image and Surface Registration

After the prior GPMM is approximated by a Karhunen–Loève expansion, it can be turned into a registration algorithm. Therefore, we have to define a reference image or 3D surface Γ_R and a target data set Γ_T. We also have to define a distance measure $\mathcal{D}$ between the objects. In the surface registration setting, the distance function is often defined as the distance between a point on the perturbed reference and its corresponding closest point on the target surface [26]. Together with the distance measure, we can formulate the registration problem as

$$\underset{\mathbf{u} \in \mathcal{F}_\mathbf{K}}{\arg\min}\, \mathcal{D}[\Gamma_R, \Gamma_T, \mathbf{u}] + \eta \|\mathbf{u}\|_\mathbf{K}^2, \tag{13}$$

where $\|\cdot\|_\mathbf{K}$ denotes the norm of the kernel functions's reproducing Hilbert space $\mathcal{F}_\mathbf{K}$ and η is a regularization parameter. Replacing $\mathbf{u}$ by its low-rank approximation $\mathbf{u}_M$ from (3), we can restate the problem in the parametric form

$$\underset{\alpha_1,\ldots,\alpha_M}{\arg\min}\, \mathcal{D}\left[\Gamma_R, \Gamma_T, \boldsymbol{\mu} + \sum_{i=1}^{M} \alpha_i \sqrt{\lambda_i} \boldsymbol{\phi}_i\right] + \eta \sum_{i=1}^{M} \alpha_i^2, \tag{14}$$

which can be optimized with common methods, such as gradient descent.

The next section shows that we can even omit the solution of the eigenvalue problem, if we choose an alternate basis for the subspace spanned by the Karhunen–Loève expansion.

5 Greedy GPMM Approximation with Newton Basis

The conversion of the optimization problem in the reproducing kernel Hilbert space from (13) to an optimization problem with finitely many parameters in (14) has strictly been done with a Karhunen–Loève expansion in previous work [26]. This automatically captures the most significant features of the problem in the reproducing kernel Hilbert space into the problem with finitely many parameters. However, the Karhunen–Loève expansion has more structure than actually needed for the optimization in (14). Therefore, we may choose a basis spanning the same subspace as the Karhunen–Loève expansion which is cheaper to compute and has the property of iterative refinement. Denoting the m-th eigenfunction obtained from the pivoted Cholesky decomposition by $\boldsymbol{\phi}_{m,N}$ and abbreviating

$$\boldsymbol{\Phi}_{N,M}(\mathbf{x}) = \left[\boldsymbol{\phi}_{1,N}(\mathbf{x}) \middle| \ldots \middle| \boldsymbol{\phi}_{M,N}(\mathbf{x})\right]$$

and $\boldsymbol{\Sigma}_{N,M} = \operatorname{diag}(\lambda_{1,N}, \ldots, \lambda_{M,N})$, the Karhunen–Loève expansion (3) can be written as

$$\mathbf{u}_M(\mathbf{x}, \boldsymbol{\alpha}) \sim \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\alpha}\boldsymbol{\Sigma}_{N,M}\boldsymbol{\Phi}_{N,M}(\mathbf{x})^\mathsf{T}.$$

By associating (11) with its corresponding functions, we deduce that

$$\boldsymbol{\Phi}_{N,M}(\mathbf{x}) = \mathbf{L}_M(\mathbf{x}) \underbrace{\left[\check{\mathbf{v}}_{1,N} \middle| \ldots \middle| \check{\mathbf{v}}_{M,N}\right]}_{=:\check{\boldsymbol{\Phi}}_{N,M}},$$

which yields

$$\mathbf{u}_M(\mathbf{x}, \boldsymbol{\alpha}) \sim \boldsymbol{\mu}(\mathbf{x}) + \underbrace{\boldsymbol{\alpha}\boldsymbol{\Sigma}_{N,M}\check{\boldsymbol{\Phi}}_{N,M}(\mathbf{x})^\mathsf{T}}_{=:\tilde{\boldsymbol{\alpha}}} \mathbf{L}_M(\mathbf{x})^\mathsf{T}.$$

We remark in particular that the coefficients $\tilde{\boldsymbol{\alpha}}$ are a linear combination of the coefficients $\boldsymbol{\alpha}$. Thus, instead of using the Karhunen–Loève expansion (3), and by denoting the function associated with the vector $\boldsymbol{\ell}_i$ by $\boldsymbol{\ell}_{i,N}$, we can use the expansion

$$\mathbf{u}_M(\mathbf{x}, \tilde{\boldsymbol{\alpha}}) \sim \boldsymbol{\mu}(\mathbf{x}) + \sum_{i=1}^{M} \tilde{\alpha}_i \boldsymbol{\ell}_{i,N}(\mathbf{x}),$$

see also [17], which spans the same subspace. The optimization (14) then turns into

$$\arg\min_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_M} \mathcal{D}\left[\Gamma_R, \Gamma_T, \boldsymbol{\mu} + \sum_{i=1}^{M} \tilde{\alpha}_i \boldsymbol{\ell}_{i,N}\right] + \eta \sum_{i=1}^{M} \tilde{\alpha}_i^2.$$

We can thus directly work with the Newton basis given by the column vectors of the low-rank approximation of the pivoted Cholesky decomposition, cf. [30], and omit the solution of any eigenvalue problems.

The Newton basis offers some new interesting possibilities since one can easily expand the basis if a higher accuracy is needed. Therefore, it is sufficient to proceed in the algorithm of the pivoted Cholesky decomposition. If the basis vectors need to be orthonormal, one can apply an orthonormalization method like the Gram-Schmidt algorithm, cf., e.g., [12]. As the pivoted Cholesky decomposition, the Gram-Schmidt algorithm can be continued when the basis has to be expanded.

We will show in the numerical experiments that this greedy-type basis is a competitive alternative for registration compared to the original Karhunen–Loève approach by [26]. It also gains a computational advantage, since the steps for the computation of the Karhunen–Loève expansion from Sect. 4.2 can be omitted.

The actual performance improvements very much depend on the numerical effort for the computation of a single matrix entry. The computation of the pivoted Cholesky decomposition is $\mathcal{O}(NM^2)$, with a constant which is dominated by that effort. The post-processing to obtain the Karhunen–Loève expansion as described in Sect. 4.2 consists of the computation of the small matrix for the dense eigenvalue problem (NM^2 operations), the solution of the small eigenvalue problem ($\mathcal{O}(M^3)$ operations, see [12]), and the computation of the eigenvectors for the large system (NM^2 operations). We can thus save $2NM^2 + \mathcal{O}(M^3)$ operations when using the Newton basis rather than the Karhunen–Loève expansion.

However, from our perspective, the possibility to refine the Newton basis greedily is much more valuable than the actual performance improvements gained from omitting the steps from Sect. 4.2. It is particularly attractive to increase the approximation accuracy as illustrated in Fig. 2. In Fig. 2, two random samples are visualized from models that are approximated with increasing accuracy. We emphasize again that, in contrast to the Karhunen–Loève approach, a refinement of the model only needs the calculation of the additional basis vectors instead of a full computation from scratch.

6 GPMM Approximation Experiments

6.1 Comparison of Generalizations for Surface GPMM

We use a data set consisting of 39 registered face scans to represent the ground-truth and as target surfaces for the registration. The data set is registered using the method proposed by [2] which includes additional constraints to cope with artifacts and noise of the raw 3D scans. To establish a fair comparison between the fitting accuracy of the methods, we create an experiment where only the generalization ability of the models is evaluated. The model for this experiment is defined with a scalar multi-scale B-spline kernel, which is introduced by Opfer [29]. Given a univariate third-order B-spline b_3 and the function $\psi(\mathbf{x}) = b_3(x_1)b_3(x_2)b_3(x_3)$, the kernel reads

$$k_{\mathrm{BSp}}(\mathbf{x}, \mathbf{y}) = \sum_{j=\underline{j}}^{\bar{j}} \sum_{k\in\mathbb{Z}^d} 2^{2-j}\psi\left(2^j\mathbf{x} - k\right)\psi\left(2^j\mathbf{y} - k\right).$$

According to [29], this results in a valid, positive definite kernel function on multiple scales. In our experiment, we define the levels from $\underline{l} = -5$ to $\bar{l} = -2$ and refer to [26] for details on matrix-valued B-spline kernels.

Taking the Nyström method proposed by [26] as a reference, we use the same kernel function to approximate three different parametric models. For the reference, we sample a uniform subset of 1000 points and approximate 1000 eigenfunctions, which, due to the expensive interpolation (9) to

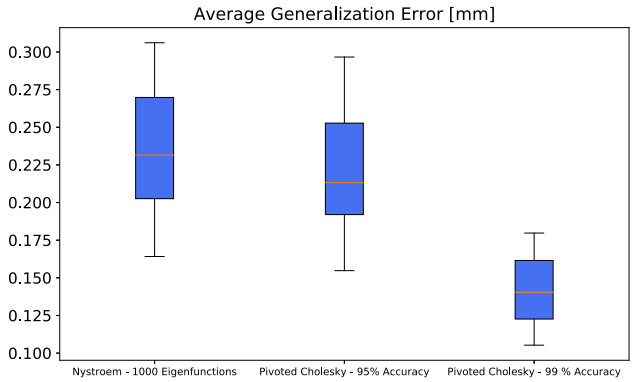

Fig. 3 Comparison of the generalization ability of three differently approximated GPMM face models to 36 example faces: The error is measured with the distance closest to the target surface in (mm). The GPMM approximated with the Nyström method on 1000 eigenfunctions is compared to the approximation of the same GPMM with 95% and 99% accuracy. We clearly see that an approximation of the model with 1000 eigenfunctions is only comparable to a model approximation with 95% accuracy. Approximating the model with an accuracy of 99% decreases the generalization error visibly

extend the eigenfunctions to all mesh points, amounts to the borderline of feasibility. With the pivoted Cholesky method, we create a model with a similar amount of basis functions with a tolerance of $\varepsilon = 0.05$ (1200 basis functions) and a more accurate model with a tolerance of $\varepsilon = 0.01$ (2200 basis functions). Especially, we are able to sample on all grid points and can avoid the expensive interpolation of the eigenfunctions. Since the face data set is registered and thus in correspondence with the reference, we can define a direct projection in the model space as

$$] \underset{\alpha_1,\ldots,\alpha_M}{\arg\min}\, \mathcal{D}_c\left[\Gamma_R, \Gamma_T, \boldsymbol{\mu} + \sum_{i=1}^{M} \alpha_i \sqrt{\lambda_i}\boldsymbol{\phi}_i\right], \tag{15}$$

where the distance function $\mathcal{D}_c$ consists of the squared Euclidean distance of every point $\mathbf{x}$ with its corresponding point on the registered target. This is a least squares problem and the optimal solution for this problem can be computed in closed form solution, which is shown in [1].

The 39 registered human faces are projected into all three models and the average point-to-point distance is measured and illustrated in Fig. 3. The Cholesky approximated model with $\varepsilon = 0.05$ performs similarly as the Nyström model. However, when the approximation accuracy of the Cholesky model is increased to $\varepsilon = 0.01$, the model generalizes better to the faces data set.

6.2 Spatially Varying Kernel Models

In this experiment, we compare the low-rank approximation methods on covariance functions, where the correlation length varies depending on predefined regions in the domain Ω. These type of covariance functions allow for the specification of different kind of smoothness depending on the region. In practice, this is especially useful for modeling different levels of detail depending on the region. To approximate a spatially varying model in practice, the Nyström approach is not optimal for two reasons: (1) Locally detailed regions can be approximated using a more compact representation than a global model with small details, which is, however, not controllable with the Nyström approach. This is illustrated in Fig. 4, where the spatially varying kernel is approximated by a few basis functions. Choosing the number of basis functions in advance, as it would be the case for the Nyström method, is difficult and unintuitive. (2) Since the Nyström approximation only computes the eigenfunctions on a uniformly sampled subset of points, it might well approximate coarse correlations, but it is likely to miss small deformation regions if the subset is not densely sampled, which is not the case when using the pivoted Cholesky algorithm. As an experimental setup, we define a coarse kernel function $k_c(\mathbf{x}, \mathbf{y})$ and also a fine kernel $k_f(\mathbf{x}, \mathbf{y})$, which are both defined as Gaussian kernels with $\sigma = 100$ for k_c and $\sigma = 15$ for k_f. Together with a function $t\colon \Omega \to (0, 1)$, which activates the fine kernel on a predefined region in Ω, we formulate a spatially varying kernel as

$$k_s(\mathbf{x}, \mathbf{y}) = k_c(\mathbf{x}, \mathbf{y}) + t(\mathbf{x})k_f(\mathbf{x}, \mathbf{y})t(\mathbf{y}). \tag{16}$$

In Fig. 4, the kernels and their individual amount of basis functions are visualized. By simply choosing the approximation error, we receive the right number of basis functions to approximate the model.

One practical example for spatially varying kernels is the construction of a registration prior for human faces or full heads. In [10], it has been shown how to model and apply such a prior for face registration. In this experiment, we compare a globally consistent kernel to a spatially varying kernel by the means of model approximation and registration accuracy. Similar to 6.1, we use a data set consisting of 48 registered full human head shapes as ground-truth target surfaces for the registration. The data set is registered using the method proposed by [2].

Except the spatial variance, we have built both deformation priors the same way with a multi-scale B-spline kernel as its basis. A detailed description has been done in [10]. In short, we have built kernel k_{SP}, where a function emphasizes more details for the face region than for the full head. A practical reason is that not much detailed information is expected from this region and it has to be more robust against occluders, such as hair. The kernel k_{NSP}, however, is built globally ignoring the regions of the head.

Using the method proposed in this paper, we can measure the effect of the spatially varying kernel using the approximation accuracy when we limit both models to a fixed amount of

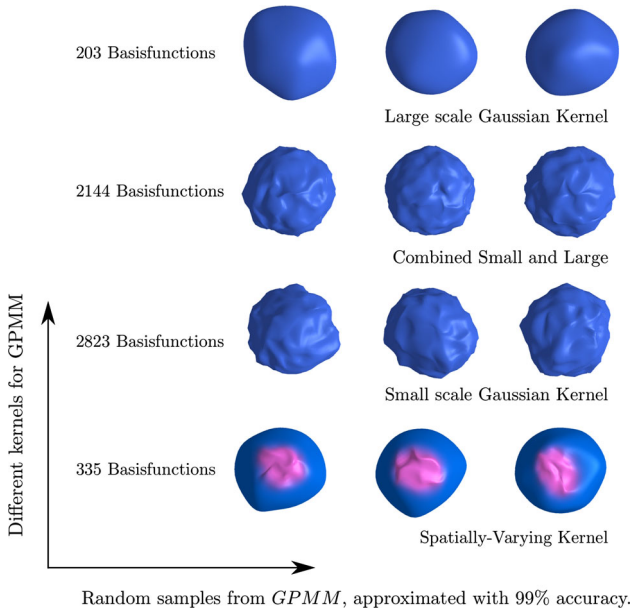

Fig. 4 In this figure, random samples of different kernel functions are visualized on a sphere with the radius of 100 mm. All of them are approximated with the proposed method using an accuracy of 99.0%. The method automatically selects the right amount of basis functions for the given model. The spatially varying model in the bottom row only contains local details at the specified region. This drastically decreases the number of basis functions to reach 99.0% accuracy

basis functions (1000). Since the spatially varying kernel only allows correlations in the face and ear region, the representation of this model can be more compact, which allows for more details at the face region. The spatially varying kernel can be approximated 6% better with the same amount of basis functions than used for the non-varying counterpart (Table 1). The better accuracy of the model and the additional modeling information has a direct effect on the face registration task itself. We have used both models to register 48 human ground-truth heads and measured the registration accuracy using the average corresponding point distance. Figure 5 shows an improvement of the full head as well as the isolated face region. The spatially varying kernel enables more details for the face region while keeping a smooth backhead. Visually, this is also shown in Fig. 6, where the error in facial details around the mouth and eye region has been decreased.

The models were computed on a *Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz* with 32 cores with a computation time of approximately 20 minutes. The head registrations take about 15 minutes per case.

7 Medical Image Registration

With the following medical image registration experiments, we measure the impact of approximation accuracy and the approximation basis on the registration accuracy of a human forearm CT data set. Also, we show a comparison of the GPMM registration with a state-of-the-art multi-scale B-spline image registration pipeline, which is implemented in Elastix [23]. The registrations are performed on a data set consisting of 27 CT images of the human forearm. The surface of Ulna and the Radius have been manually segmented by experts to provide a ground-truth measure of the accuracy. The data has been rigidly aligned to one arbitrary example of the data set using 4 landmarks and are provided in a resolution of 800 × 800 × 500. For the deformation model, we selected the same kernel function as in [26], which performed best in their experiment. To evaluate the accuracy of

Table 1 Model approximation accuracy of the head registration prior models k_{SP} and k_{NSP} limited to 1000 basisfunctions

	Global (k_{NSP})	Spatially Varying (k_{SP})
Accuracy	92.5%	**98.1%**

The spatially varying kernel has been approximated 6% more accurate than its global counterpart

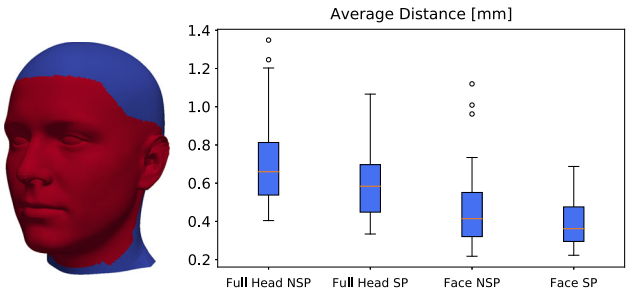

Fig. 5 The measured accuracy of the spatially varying model k_{SP} and its non-varying counterpart k_{NSP} is shown. **Left** The error is measured on the full head (Red and Blue) and only on the face region (Red). **Right** The average distance in (mm) of the GPMM registration compared against the ground-truth (Color figure online)

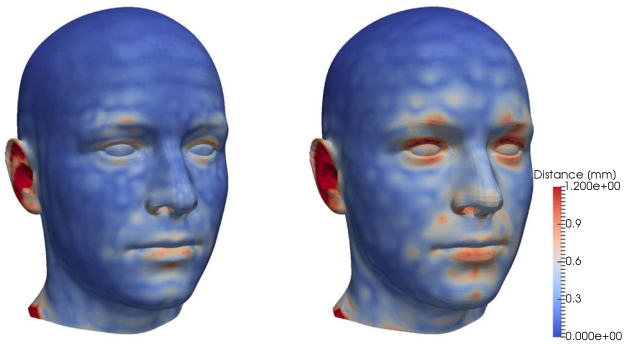

Fig. 6 A comparison between the registrations with and without a spatially varying kernel. The registrations using the spatially varying kernel (**left**) show less errors in the facial region than its non-varying counterpart (**right**). For the face registration application in [10] the variation at the ears were deliberately built smooth because to be robust against outliers. For this reason, the error is large for both models in this region

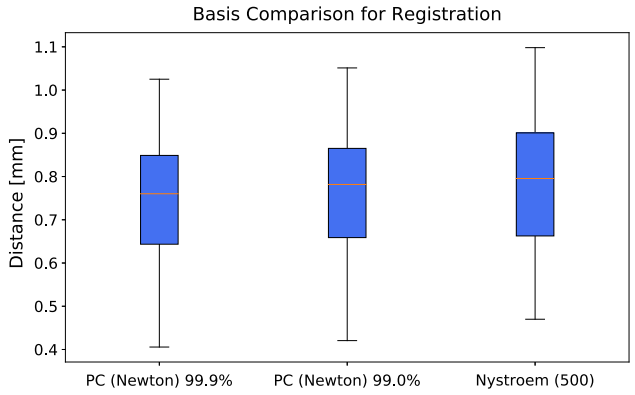

Fig. 7 A comparison between different model approximations using the registration accuracy of the Ulna registration. Our proposed Cholesky model with $\varepsilon = 0.01$ and $\varepsilon = 0.001$ error is comparable with the Nyström method with 500 eigenfunctions. However, the Pivoted Cholesky method provides additional information about the approximation accuracy of the given GPMM

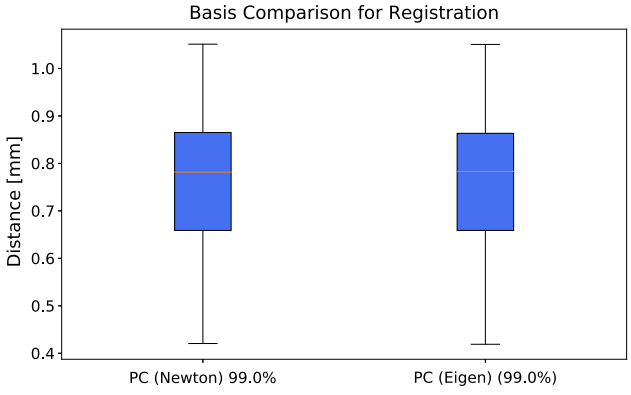

Fig. 8 A comparison of the registration accuracy of two different basis for the GPMM approximation. **Left** The proposed Newton basis is less computationally intensive and iteratively refinable. **Right** The originally used Eigenbasis, computed as in Sect. 4.2

our experiments, we computed the average squared distance error of the registered result to the ground-truth segmentations of the provided 27 CT images. For all the computations we used a *Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz* with 32 cores. The prior model approximations (Nystroem and PC) are computed offline once took less than an hour in average. The approximation of the Newton basis model is faster because it omits the last SVD step, as described in Sect. 5. The registrations itself are both in the range of 30 min per example, which counts for multi-scale B-spline of Elastix, but also for the GPMM registration.

7.1 Model Approximation Experiment

In this experiment, we compare the registration error between the two different approximation methods; the originally proposed Nyström method and the pivoted Cholesky approximation using the Newton basis and the finite element discretization. In Fig. 7, the registration accuracy of the two models is visualized. If we approximate the pivoted Cholesky model with an accuracy of 99% ($\varepsilon = 0.01$), the registration results are similar to the Nyström where we chose 500 eigenfunctions. When the Cholesky model is approximated with more accuracy (99.9%), a minor decrease of the error is still visible, which also due to the larger flexibility of the more accurate model, computed with pivoted Cholesky. The advantage of the proposed method and the new basis are not the approximation accuracy itself but the control over the approximation error and the black-box usage. Using the previously proposed Nyström example, the user will never be sure if a change of the number of basis functions still improves the GPMM approximation.

In Fig. 8, a direct comparison between the eigenbasis and the Newton basis is shown. For both models, we used the pivoted Cholesky method to approximate the model with 99% accuracy. For the first model, we kept the basis as the Newton basis and for the second model we used the method shown in 4.2 to compute the eigenbasis. Figure 8 shows that the Newton basis performs similarly at the given image registration task, while it needs less computational steps, is greedily computed and can be refined iteratively.

7.2 Image Registration Comparison

In this last experiment, we compare the proposed image deformation models to a state-of-the-art B-spline registration algorithm, which is implemented in Elastix [23]. In Fig. 9, the GPMM registration approach with the proposed basis functions is compared to the B-spline registration included in Elastix [23].The goal of this experiment is to evaluate the usage of the proposed method and basis functions in a practical setting against a state-of-the-art baseline. The Elastix registration framework also contains a multi-resolution strategy for the optimization, which is currently not present in the GPMM framework. To achieve a fair comparison between the deformation models, we compare the registrations in two settings: (1) A comparison with a single-scale Elastix with B-spline transformation, which omits the influence of the optimization strategy and enables a fair comparison of the models. (2) An experiment, where a landmark posterior model is compared to the multi-scale B-spline. Here, we show that even without an advanced optimization strategy the GPMM models can perform similarly to the GPMM models. When the B-spline method, provided in Elastix, is used with a single B-spline scale, our proposed method performs in a comparable range. To become more robust toward local optima, the GPMM method allows to build a prior of the deformations by defining a kernel that is more suited for a specific task. As Lüthi et al. [26] have shown, this can be

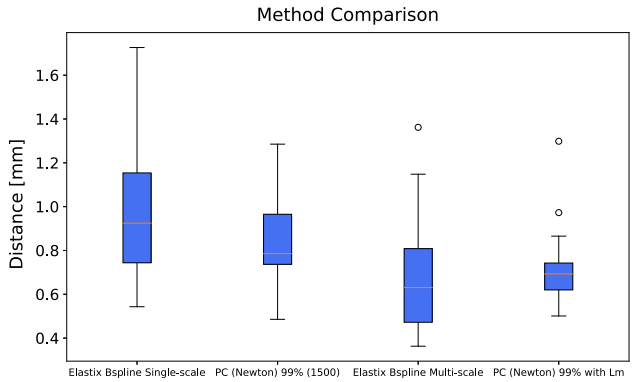

Fig. 9 A comparison of the built registration models with a state-of-the-art B-spline method, which is implemented in Elastix [23]. Our model is comparable with the single-scale approach. The inclusion of landmarks to compute a posterior GPMM guides the optimization procedure well and is comparable to the multi-scale version of Elastix [23]

done in multiple ways, as for example, making it spatially varying, symmetric or to include landmarks. We show exemplary how adding landmark constraints to restrict the prior to only deformations that match the landmarks, lead to a more accurate registration accuracy, and at the same time, much more robust results. For all the registration experiments we used the proposed Newton basis, as shown in Sect. 5, which show comparable performance to state-of-the-art registration methods, while having the desired properties for prior modeling and the ability to refine the model approximation iteratively.

8 Conclusion

We have presented a low-rank approximation method for the Gaussian Process Morphable Model framework (GPMM) with a controllable approximation error and a refinable and greedy basis. An error-controlled parameterization of the GPMM is a very important step to accurately specify and approximate deformation models for registration. In this paper, we show a GPMM approximation method with significant advantages over the originally proposed method: The method enables full control over the approximation error and the greedy algorithm stops at a predefined accuracy. In practice, this enables the user to treat the model parameterization as a black-box while still having guarantees about the approximation accuracy. As the second main contribution, we proposed a new basis for GPMM registration. We showed that the Newton basis contains enough structure for the registration problem optimization and has two advantages over the previously used eigenbasis: (1) The calculation of this basis is computationally more efficient because it omits the step of calculating an SVD and (2) the Newton basis are computed greedily and thus allow for iterative refinement of the approximation error without the re-computation of the preceding basis vectors. The contributions in this paper add some important missing pieces to the GPMM framework. In contrast to the initial framework, we provide a practical and theoretically sound way to control the approximation error of the GPMMs, which has a large impact to the usability in practice. We showed the registration accuracy of the models in the context of human face surface registration evaluated on ground-truth registrations. Also, we demonstrated the applicability of the method in the context of medical image registration, where the human forearm was registered, and showed that the method is competitive to state-of-the-art registration methods. The core algorithms proposed in this work are published open source in the **Scalismo** framework [35].

Acknowledgements This work has been funded as part of two Swiss National Science foundation projects in the context of the Projects SNF153297 and SNF156101. We thank Andreas Morel-Forster and Volker Roth for interesting and enlightening discussions. A special thanks goes to Ghazi Bouabene and Christoph Langguth for their work on the *Scalismo* software, in which all the methods are implemented.

A Appendix

A.1 Advanced Nyström Schemes

Nyström schemes are suitable if the eigenfunctions of the Karhunen–Loève expansion are only required in certain predetermined points $\mathbf{x}_1, \ldots, \mathbf{x}_N$. For this purpose, the integral operator (2) is approximated by a quadrature formula

$$\int_\Omega \mathbf{K}(\cdot, \mathbf{x})\mathbf{f}(\mathbf{x})\, \mathrm{d}\rho(\mathbf{x}) \approx \sum_{i=1}^{N} \omega_i \mathbf{K}(\cdot, \boldsymbol{\xi}_i)\mathbf{f}(\boldsymbol{\xi}_i)$$

with quadrature points $\boldsymbol{\xi}_i$ and weights ω_i. The discrete eigenvalue problem then reads

$$\mathbf{C}_{\mathrm{Nystr}}\hat{\boldsymbol{\phi}}_{m,N} = \lambda_{m,N}\hat{\boldsymbol{\phi}}_{m,N}$$

with the system matrix

$$\mathbf{C}_{\mathrm{Nystr}} = \left[\omega_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^{N}$$

and the point values

$$\hat{\boldsymbol{\phi}}_{m,N} \approx \left[\boldsymbol{\phi}_m(\mathbf{x}_i)\right]_i, \quad i = 1, \ldots, N.$$

Note that the system matrix $\mathbf{C}_{\mathrm{Nystr}}$ is not symmetric in general. Assuming positive quadrature weights, i.e. $\omega_i > 0$, defining

$$\mathbf{M}_{\mathrm{Nystr}} = \mathrm{diag}(\sqrt{\omega_1}, \ldots, \sqrt{\omega_N})$$

and setting $\boldsymbol{\phi}_{m,N} = \mathbf{M}_{\text{Nystr}}\hat{\boldsymbol{\phi}}_{m,N}$ yields a symmetric, generalized eigenvalue problem

$$\mathbf{M}_{\text{Nystr}}\mathbf{C}\mathbf{M}_{\text{Nystr}}^{\mathsf{T}}\boldsymbol{\phi}_{m,N} = \lambda_{m,N}\mathbf{M}_{\text{Nystr}}\boldsymbol{\phi}_{m,N}$$

with the matrix

$$\mathbf{C} = \left[\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^{N}, \tag{17}$$

see also [15]. As it turns out, the finite element scheme yields an eigenvalue problem with a similar structure.

A.2 Finite Element Scheme on a Rectangular Grid

Finite element schemes for functions with values in three dimensions rely on a finite dimensional subspace $\mathbf{V}_N \subset \left[L^2(\Omega)\right]^3$ with basis $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_N\}$ to represent the eigenfunctions of the Karhunen–Loève expansion. To construct such a finite dimensional space, we consider a uniform rectangular grid $\mathcal{Q}_h$ on Ω where each cell has a size of $h_1 \times h_2 \times h_3$. To each vertex $\mathbf{x}_1, \ldots, \mathbf{x}_N$ we assign a function φ_i with the property

$$\varphi_i(\mathbf{x}_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \quad i, j = 1, \ldots, N, \tag{18}$$

where on each cell $Q_h \in \mathcal{Q}_h$, the basis function φ_i is a trilinear polynomial, i.e.

$$\begin{aligned}\varphi_i(\mathbf{y})\big|_{Q_h} = a_1 &+ a_2 y_1 + a_3 y_2 + a_4 y_1 y_2 \\ &+ a_5 y_3 + a_6 y_1 y_3 + a_7 y_2 y_3 + a_8 y_1 y_2 y_3.\end{aligned}$$

Here, the coefficients are uniquely determined such that (18) holds. This means especially that the φ_i are only non-zero in the eight cells with vertex $\mathbf{x}_i$. Note especially that all φ_i are linearly independent, so we can define $V_h \subset L^2(\Omega)$ as the vector space spanned by the basis $\varphi_1, \ldots, \varphi_N$. A finite dimensional subspace of $\left[L^2(\Omega)\right]^3$ is then given by $\mathbf{V}_h = V_h \times V_h \times V_h$.

A.3 Advanced Finite Element Schemes

Having a finite dimensional subspace at hand yields, cf., e.g., [15], the generalized eigenvalue problem

$$\mathbf{C}_{\text{FEM}}\boldsymbol{\phi}_{m,N} = \lambda_{m,N}\mathbf{M}_{\text{FEM}}\boldsymbol{\phi}_{m,N} \tag{19}$$

with system matrices

$$\begin{aligned}\mathbf{C}_{\text{FEM}} &= \left[\left(\mathcal{T}_{\mathbf{K}}\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right)_{[L^2_\rho(D)]^3}\right]_{i,j=1}^{N}, \\ \mathbf{M}_{\text{FEM}} &= \left[\left(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right)_{[L^2_\rho(D)]^3}\right]_{i,j=1}^{N},\end{aligned}$$

$\mathcal{T}_{\mathbf{K}}$ denoting the integral operator from (2), and the approximate eigenfunctions

$$\boldsymbol{\phi}_m(\mathbf{x}) \approx \boldsymbol{\phi}_{m,N}(\mathbf{x}) = \sum_{i=1}^{N} \left(\boldsymbol{\phi}_{m,N}\right)_i \boldsymbol{\varphi}_i(\mathbf{x}).$$

It thus remains to explain how to assemble these matrices.

Since the basis functions $\boldsymbol{\varphi}_i$ are non-zero only on a few elements, the mass matrix $\mathbf{M}_{\text{FEM}}$ is sparse. Inserting the definition of $\mathcal{T}_{\mathbf{K}}$ into the definition of $\mathbf{C}_{\text{FEM}}$, we obtain

$$\mathbf{C}_{\text{FEM}} = \left[\int_D \int_D \mathbf{K}(\mathbf{x}, \mathbf{y})\boldsymbol{\varphi}_j(\mathbf{y})\boldsymbol{\varphi}_i^{\mathsf{T}}(\mathbf{x})\,\mathrm{d}\rho(\mathbf{y})\,\mathrm{d}\rho(\mathbf{x})\right]_{i,j=1}^{N}.$$

In order to compute this integral, it is very common in finite element methods to replace $\mathbf{K}$ by its interpolation $\mathbf{K}_h$ in the finite element space, i.e. we approximate

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) \approx \sum_{i,j=1}^{N} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)\boldsymbol{\varphi}_i(\mathbf{x})\boldsymbol{\varphi}_j^{\mathsf{T}}(\mathbf{y}).$$

Inserting this approximation into the definition of $\mathbf{C}_{\text{FEM}}$ yields

$$\mathbf{C}_{\text{FEM}} \approx \mathbf{M}_{\text{FEM}}\mathbf{C}\mathbf{M}_{\text{FEM}}^{\mathsf{T}},$$

with the matrix $\mathbf{C}$ defined as for the Nyström scheme in (17). The eigenvalue problem (19) thus turns into

$$\mathbf{M}_{\text{FEM}}\mathbf{C}\mathbf{M}_{\text{FEM}}^{\mathsf{T}}\boldsymbol{\phi}_{m,N} = \lambda_{m,N}\mathbf{M}_{\text{FEM}}\boldsymbol{\phi}_{m,N}. \tag{20}$$

A.4 Connection Between the Two Schemes

The two schemes can lead to the very same eigenvalue problem. In implementations of finite element schemes, there are almost always quadrature formulas involved. Using piecewise linear ansatz functions and replacing the integrals by a trapezoidal rule yields a diagonal matrix $\mathbf{M}_{\text{FEM}}$ (this is also referred to as "mass lumping"). The definition of $\mathbf{M}_{\text{Nystr}}$ then amounts to quadrature weights to a quadrature formula with the vertices of the finite element mesh as evaluation points. The two schemes are thus equivalent in this specific case.

A.5 Computing Karhunen–Loève Expansions using Low-rank Approximations

Again, having a low-rank factorization $\mathbf{C} \approx \mathbf{L}_M\mathbf{L}_M^{\mathsf{T}}$ of rank M at hand, one can reduce the dimension of the eigenvalue problems (20). For ease of notation, we do not distinguish between $\mathbf{M}_{\text{FEM}}$ and $\mathbf{M}_{\text{Nystr}}$ and consider the eigenvalue problem

$$\mathbf{M}\mathbf{C}\mathbf{M}^{\mathsf{T}}\boldsymbol{\phi}_{m,N} = \lambda_{m,N}\mathbf{M}\boldsymbol{\phi}_{m,N}. \tag{21}$$

By substituting the low-rank approximation $\mathbf{C} \approx \mathbf{L}_M \mathbf{L}_M^\intercal$ and $\mathbf{v}_{m,N} = \mathbf{M}^{1/2}\boldsymbol{\phi}_{m,N}$ into (21), the eigenvalue problem becomes

$$\mathbf{M}^{1/2}\mathbf{L}_M\mathbf{L}_M^\intercal(\mathbf{M}^{1/2})^\intercal\mathbf{v}_{m,N} = \lambda_{m,N}\mathbf{v}_{m,N}.$$

Exploiting the fact that $\mathbf{M}^{1/2}\mathbf{L}_M\mathbf{L}_M^\intercal(\mathbf{M}^{1/2})^\intercal$ has the same eigenvalues as $\mathbf{L}_M^\intercal(\mathbf{M}^{1/2})^\intercal\mathbf{M}^{1/2}\mathbf{L}_M = \mathbf{L}_M^\intercal\mathbf{M}\mathbf{L}_M$, we obtain an equivalent eigenvalue problem

$$\mathbf{L}_M^\intercal\mathbf{M}\mathbf{L}_M\tilde{\mathbf{v}}_{m,N} = \lambda_{m,N}\tilde{\mathbf{v}}_{m,N}.$$

This modified eigenvalue problem has again dimension $M \ll N$ and can thus be solved by standard eigensolvers for dense matrices.

References

1. Albrecht, T., Lüthi, M., Gerig, T., Vetter, T.: Posterior shape models. Med. Image Anal. **17**(8), 959–973 (2013)
2. Amberg, B., Romdhani, S., Vetter, T.: Optimal step nonrigid icp algorithms for surface registration. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
3. Amit, Y., Grenander, U., Piccioni, M.: Structural image restoration through deformable templates. J. Am. Stat. Assoc. **86**(414), 376–387 (1991)
4. Beebe, N.H.F., Linderberg, J.: Simplifications in the generation and transformation of two-electron integrals in molecular calculations. Int. J. Quantum Chem. **12**(4), 683–705 (1977)
5. Berlinet, A., Thomas-Agnan, C.: Reproducing Kernel Hilbert Spaces in Probability and Statistics, vol. 3. Springer, Berlin (2004)
6. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM Press (1999)
7. Cootes, T.F., Beeston, C., Edwards, G.J., Taylor, C.J .: A unified framework for atlas matching using active appearance models. In: Biennial International Conference on Information Processing in Medical Imaging, pp. 322–333. Springer (1999)
8. Cuadra, M.B., Duay, V., Thiran, J.-P.: Atlas-based segmentation. In: Handbook of Biomedical Imaging, pp. 221–244. Springer (2015)
9. Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M.J., Gazis, P., Srivastava, A.: Stable and efficient Gaussian process calculations. J. Mach. Learn. Res. **10**, 857–882 (2009)
10. Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schoenborn, S., Vetter, T.: Morphable face models—an open framework. In: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), pp. 75–82 (2018)
11. Gerig, T., Shahim, K., Reyes, M., Vetter, T., Lüthi, M.: Spatially varying registration using Gaussian processes. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014, pp. 413–420. Springer (2014)
12. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2012)
13. Grenander, U., Miller, M.I.: Computational anatomy: an emerging discipline. Q. Appl. Math. **56**(4), 617–694 (1998)
14. Griebel, M., Harbrecht, H.: Approximation of bi-variate functions: singular value decomposition versus sparse grids. IMA J. Numer. Anal. **34**(1), 28–54 (2014)
15. Hackbusch, W.: Integral Equations: Theory and Numerical Treatment, vol. 4. Birkhäuser, Basel (1995)
16. Halko, N., Martinsson, P.-G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
17. Harbrecht, H., Peters, M., Siebenmorgen, M.: Efficient approximation of random fields for numerical applications. Numer. Linear Algebra Appl. **22**(4), 596–617 (2015)
18. Harbrecht, H., Peters, M., Schneider, R.: On the low-rank approximation by the pivoted cholesky decomposition. Appl. Numer. Math. **62**(4), 428–440 (2012)
19. Heimann, T., Van Ginneken, B., Styner, M., Arzhaeva, Y., Aurich, V., Bauer, C., Beck, A., Becker, C., Beichel, R., Bekes, G.: Comparison and evaluation of methods for liver segmentation from ct datasets. IEEE Trans. Med. Imaging **28**(8), 1251–1265 (2009)
20. Iglesias, J.E., Sabuncu, M.R.: Multi-atlas segmentation of biomedical images: a survey. Med. Image Anal. **24**(1), 205–219 (2015)
21. Joshi, S.C., Banerjee, A., Christensen, G.E., Csernansky, J.G., Haller, J.W., Miller, M.I., Wang, L.: Gaussian random fields on sub-manifolds for characterizing brain surfaces. In: Information Processing in Medical Imaging, pp. 381–386. Springer (1997)
22. Jud, C., Mori, N., Cattin, P.C.: Sparse kernel machines for discontinuous registration and nonstationary regularization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–16 (2016)
23. Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P.W.: Elastix: a toolbox for intensity-based medical image registration. IEEE Trans. Med. Imaging **29**(1), 196–205 (2010)
24. Lüthi, M., Jud, C., Vetter, T.: Using landmarks as a deformation prior for hybrid image registration. In: Pattern Recognition, pp. 196–205 (2011)
25. Lüthi, M., Forster, A., Gerig, T., Vetter, T.: Gaussian process morphable models. In: Zheng, G., Li, S., Szekely, G. (eds.) Statistical Shape and Deformation Analysis—Methods, Implementation and Applications. Academic Press (2017)
26. Lüthi, M., Gerig, T., Jud, C., Vetter, T.: Gaussian process morphable models. IEEE Trans. Pattern Anal. Mach. Intell. **40**(8), 1860–1873 (2018)
27. Lüthi, M., Jud, C., Vetter, T.: A unified approach to shape model fitting and non-rigid registration. In: Machine Learning in Medical Imaging, pp. 66–73. Springer (2013)
28. Ma, J., Zhao, J., Tian, J., Tu, Z., Yuille, A.L.: Robust estimation of nonrigid transformation for point set registration. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June (2013)
29. Opfer, R.: Multiscale kernels. Adv. Comput. Math. **25**(4), 357–380 (2006)
30. Pazouki, M., Schaback, R.: Bases for kernel-based spaces. J. Comput. Appl. Math. **236**(4), 575–588 (2011)
31. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Springer, Berlin (2006)
32. Rosasco, L., Belkin, M., De Vito, E.: On learning with integral operators. J. Mach. Learn. Res. **11**, 905–934 (2010)
33. Rueckert, D., Frangi, A.F., Schnabel, J.A.: Automatic construction of 3d statistical deformation models using non-rigid registration. In: MICCAI'01: Medical Image Computing and Computer-Assisted Intervention, pp. 77–84 (2001)
34. Santin, G., Schaback, R.: Approximation of eigenfunctions in kernel-based spaces. Adv. Comput. Math. **42**(4), 973–993 (2016)
35. Scalismo-scalable image analysis and shape modelling. http://github.com/unibas-gravis/scalismo. Accessed 10 Oct 2018
36. Yuille, A., Kersten, D.: Vision as Bayesian inference: analysis by synthesis? Trends Cogn. Sci. **10**(7), 301–308 (2006)

Jürgen Dölz received the master's degree in Mathematics from the Department of Mathematics and Computer Science, University of Basel, Switzerland, in 2013. He completed his Ph.D. at the Department of Mathematics and Computer Science, University of Basel, Switzerland, in 2017. Currently, he is an Early Mobility PostDoc at the Technical University of Darmstadt, Germany, funded by the Swiss National Science Foundation. He is interested in the numerical treatment of partial differential equations and algorithms for low-rank structured matrices.

Thomas Gerig is a Ph.D. student at the Department of Mathematics and Computer Science at the University of Basel, Switzerland, from where he also received his Master's degree in Computer Science in 2013. He is interested in shape modeling, image analysis, algorithmic implementations and numerical methods.

Marcel Lüthi is a researcher at the University of Basel. Before completing his Ph.D. at the University of Basel in 2010, he received a Bachelor's degree in Computer Science from the University of Applied Science Bern, Switzerland and a Master's degree in Engineering Mathematics from Chalmers University of Technology, Sweden. His research interest are statistical shape modeling and its application to medical image analysis.

Helmut Harbrecht received his Ph.D. degree in Mathematics at the Technical University of Chemnitz in 2001 and his Habilitation degree in Mathematics at the University of Kiel in 2006. Currently, he is Full Professor for Computational Mathematics at the University of Basel. His research interests include efficient methods of nonlocal operators, shape optimization, and high-dimensional problems.

Thomas Vetter studied Mathematics and Physics and received the Ph.D. degree in Biophysics from the University of Ulm, Germany. As a Postdoctoral Researcher at the Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge, he started his research on computer vision. In 1993, he moved to the Max-Planck -Institute, Tübingen, Germany, and, in 1999, he became a Professor for computer graphics at the University of Freiburg, Germany. Since 2002, he has been a Professor of applied computer science at the University of Basel, Switzerland. His current research is on image understanding. He combines methods from machine learning, computer graphics and computer vision to implement analysis-by-synthesis systems for an automated image perception.