Divergence-Free Shape Interpolation and Correspondence

Marvin Eisenberger  Zorah Lähner  Daniel Cremers

Abstract
We present a novel method to model and calculate deformation fields between shapes embedded in $\mathbb{R}^D$. Our framework combines naturally interpolating the two input shapes and calculating correspondences at the same time. The key idea is to compute a divergence-free deformation field represented in a coarse-to-fine basis using the Karhunen-Loève expansion. The advantages are that there is no need to discretize the embedding space and the deformation is volume preserving. Furthermore, the optimization is done on downsampled versions of the shapes but the morphing can be applied to any resolution without a heavy increase in complexity. We show results for shape correspondence, registration, inter- and extrapolation on the TOSCA and FAUST Scan data sets.

CCS Concepts
• Computing methodologies → Computer graphics; Shape analysis;

1. Challenges in Shape Analysis
Handling non-rigidly deformed versions of a 3D shape is at the heart of numerous problems in computer vision and graphics ranging from shape comparison, information and style transfer to the automatic generation of new but meaningful shapes. In contrast to rigid shape registration which has six degrees of freedom, the problem of finding non-rigid deformations is rather ambiguous and the complexity of the optimization grows quadratically with the number of vertices in the input shapes.

While many of these problems are intrinsically related, most existing methods address them independently and do not generalize to a wider range of tasks. For example, having the same texture on both input shapes can help to find correspondences between these, while knowing the correspondence makes it easy to transfer the texture. Doing joint optimization can help to improve the performance in both tasks. Our approach combines shape registration, correspondence and interpolation – bringing different viewpoints to the same question of how to model non-rigid shape deformations.

Shape registration aims at finding a transformation of a shape $X'$ that aligns its surface with another shape $Y'$ as closely as possible. This is trivial when the correspondence $\pi : X' \rightarrow Y'$ is already given, because then we can prescribe a deformation which aligns each point $x \in X'$ with its match $\pi(x) = y \in Y'$. Vice versa, when two surfaces are aligned a correspondence can easily be found by searching for the nearest neighbor of each point in the embedding space. The shape interpolation problem addresses the task of finding a sequence of intermediate shapes between $X'$ and $Y'$ such that $X'$ is transformed into $Y'$ in a natural and continuous way. Most methods tackling this task assume they are given a perfect corre-
spondence between the initial shapes but the existence of a smooth transition is actually a requirement for a good correspondence.

Many state-of-the-art correspondence methods can find fairly accurate matchings but they often rely on pointwise information like descriptor values or pairwise distances. While these are straightforward to compute, information about continuity is not encoded or increases the complexity of the optimization heavily. Consequently, outliers due to intrinsic symmetries or different sampling prevent a continuous morphing. In this work we will introduce a representation that can formulate both problems and makes a joint optimization possible.

2. Related Work

2.1. Deformation Fields

Deformation fields have a long history in image registration. Ashburner and colleagues made use of deformation fields for autonomous shape morphing [Ash07]. They consider temporally constant deformation fields offering limited flexibility to capture more complex deformations. Solving for a space and time dependent deformation field is a highly underdetermined problem. A remedy for this issue is provided by the geodesic shooting approach advocated by [MTY06] which only estimates the initial velocity field for each pixel and then how the velocity has to propagate in the image domain in order to preserve the kinetic energy and the momentum of the whole system. Further improvements of this framework were proposed in subsequent work, including a Gauss-Newton approach [AF11] and a particularly efficient adjoint calculation [VR12].

Closely related to our work is [vFTS06] in which the authors also model volume preserving shape deformations using divergence-free vector fields. Here, deformation fields are constructed from hand crafted templates which are meant to be used as interactive shape transformation tools.

It is often beneficial to have a probabilistic interpretation of deformation fields. This yields a comprehensive description with explicit schemata to impose uniformity on vector fields as well as a sound theoretical foundation. Such a model for image registration and 2D shape registration with a Gaussian process modeling of the correspondence mapping is proposed in [ALV08]. Further work [LG16, DGL*17] specified how one can extend this approach to Gaussian processes on the surface of a three dimensional shape. The authors in [BHB00], [THB08], [ALV08] and [PDBS*09] also model non-rigid transformations using a PCA type representation of permitted motions. Analogously, [MS10] and [MZT*14] pursue a reproducing kernel Hilbert space approach to model the vector field interpolation. However, for all these references the respective vector fields are not defined on the whole embedding space surrounding the shapes but rather only at the elements of the considered point clouds and they do not admit an interpretation as a deformation field.

Another classical approach to shape deformation is based on a rotation invariant representation of triangle meshes [LSLC05]. In [ZSCO08] it is then presented how this deformation model can be used to compute a sparse set of correspondences.

2.2. Shape Registration and Matching

Much work has been done in the direction of shape registration and matching and we would like to point the interested reader to in-depth surveys of these topics for an overview [vKZHCO11, SMFF07, TCL*13]. Here we will focus on work that is directly related to our approach.

A popular line of work in shape matching is based on spectral decomposition of the surface Laplace-Beltrami operator [DK10]. This is popular because it reduces the dimensionality of the problem from the number of vertices to the number of basis functions chosen [OBCS*12]. Nevertheless, extracting the correspondence from the low dimensional representation is still a complex problem and often retrieved solutions are noisy or hard to compute [RC15]. We also use a spectral approach but, instead of a basis for functions on the surface, we represent deformation fields in the embedding space using the eigenfunctions of the standard Laplacian.

Methods based on Multi-Dimensional Scaling find correspondences by reembedding and then aligning shapes in a (possibly) smaller embedding space where the complexity is reduced [BBK06, ADK16], [CK15] calculate a robust non-rigid registration based on Markov random fields but can not retrieve a continuous deformation. In [MS10] and [MZH*14] the authors address the non-rigid registration problem by modeling one point cloud as a Gaussian mixture model. Moreover, they also determine the correspondences and point mappings in an alternating manner using an expectation maximization algorithm. This work is strongly related to our framework. Like our approach they directly model the correspondence mapping of all points as a Gaussian process type mapping. There also exist extensions of this method which additionally include descriptor values [MZY16, MJL17].

2.3. Shape Interpolation

Although registration methods often compute a deformation between shapes, the focus is not on producing realistic intermediate shapes. Most methods realistically interpolating between shapes assume to be given a full correspondence in advance. [KMP07] interprets the solution to interpolation as geodesics in the pointwise shape space. [WBR11] and [HRS*16] model a space of shells with metrics induced by physical deformation energies in which geodesics or splines represent natural interpolations between shells. [VTSS15] make real-time interpolation on a set of preprocessed, given shapes with arbitrary resolution possible.

[AC00] only needs a handful of correct correspondences to find a volume preserving deformation between two shapes but they require both to be segmented in compatible simplicial complexes. In [XZB05] a dense deformation field similar to ours is calculated but the method depends on a consistent triangulation of the inputs. Other directions include taking user input to guide deformations in the right direction [VMW15] or rely on a known or learned model to generate new shapes [GCLX17].

3. Contribution

In the following, we will introduce a mathematical framework which allows to jointly tackle the problems of shape interpola-
tion/extrapolation, shape registration and correspondence estimation. Our method solely operates on two given 3D point clouds and in particular requires no connectivity information like a mesh. We propose to estimate a smooth and volume preserving 3D deformation field prescribing a plausible interpolation of these input shapes. More specifically, we solve an initial value problem for determining the shape deformation. This framework allows us to incorporate physical assumptions about the deformation field. We suggest to impose volume preservation by enforcing zero divergence. More specifically, we represent the deformation field as the curl of a potential function and propose a natural coarse-to-fine basis representation of these potential functions. The initial value problem is then integrated by a Runge-Kutta scheme. We use an expectation maximization approach to simultaneously determine a subset of the unknown point-to-point correspondences and the optimal deformation field parameters. The objective is aligning two shapes with a preferably uniform deformation field. We demonstrate that the proposed framework can be used to create plausible shape interpolations and extrapolations in numerous experiments. Moreover, it provides a shape correspondence which compares to state-of-the-art correspondence methods.

4. Problem Formulation

This section gives an introduction into the problem we want to solve and the mathematical background we use in later sections.

4.1. Deformation field shape correspondences

Consider two discrete sets of points \( X = \{x_1, \ldots, x_N\} \subset \Omega \) and \( Y = \{y_1, \ldots, y_M\} \subset \Omega \) contained in a compact domain \( \Omega \subset \mathbb{R}^D \). The points \( x_n \) and \( y_m \) are assumed to be uniformly sampled from the surface of two similar \( D \)-dimensional shapes. The shape registration problem now addresses the task of aligning the point clouds \( X \) and \( Y \) in a meaningful manner, such that similar regions of the two shapes are matched onto each other. In particular we are looking for a mapping \( f : \Omega \rightarrow \Omega \), such that the mapped points \( f(x_n) \) fit to the shape \( Y \).

Our approach chooses these mappings \( f \) in such a way that they imitate plausible transformations in the real world. For this purpose we make some natural assumptions about the trajectories of the transformed points \( x_n \). For once we require the points of our shape to move smoothly over time. We would also like points in a certain neighborhood to shift in a uniform manner. This assumption yields to some extent that the determined correspondences are continuous. Moreover, the volume of shifted objects should remain the same. We therefore assume that every point \( x_n \in \mathcal{X} \) moves according to the following initial value problem:

\[
\begin{align*}
\dot{x}(t) &= v(x(t)), \\
x(0) &= x_n.
\end{align*}
\]

In this context \( v : \Omega \rightarrow \mathbb{R}^D \) is some fixed deformation field moving the point \( x_n \) over time. In order to ensure a uniform movement of the samples the vector field \( v \) has to be sufficiently smooth. We will even assume that \( v \in C^\infty(\Omega, \mathbb{R}^D) \), which yields the following well-known result:

Proposition 1 For a deformation field \( v \in C^\infty(\Omega, \mathbb{R}^D) \), the initial value problem (1) admits a solution on the compact domain \( \Omega \) and this solution is unique. Furthermore it is infinitely many times continuously differentiable \( x \in C^\infty([0,1], \mathbb{R}^D) \).

Proof This follows directly from the theorem of Picard-Lindelöf [Tes12, Theorem 2.5] and [Tes12, Lemma 2.3].

As a consequence we can choose the correspondence mapping \( f \) to be the solution operator of (1) evaluated at an arbitrary time \( t \). For convenience we choose \( t = 1 \):

\[
f : \begin{cases} \\
\Omega \rightarrow \mathbb{R}^D, \\
x_n \mapsto x(1).
\end{cases}
\]

The advantage of this framework is that it provides an intuitive notion of deformation when analyzing shape correspondences. Therefore, one does not only get a matching of two shapes but also all intermediate states constituting the underlying transformation. Those are typically more meaningful than merely performing linear interpolation between the initial and the final position of each point \( x_n \). Especially when looking at nearly isometric shapes having a continuous correspondence and a natural deformation are inherently connected and solving for both simultaneously improves the results considerably.

4.2. Divergence free deformations

Another convenient quality of the presented framework is that it enables us to incorporate assumptions about the deformation field into our model. One reasonable restriction arises from the mathematical investigation of fluid dynamics [CM93], namely the restriction to divergence free velocity fields:

\[
\nabla \cdot v = 0.
\]

It is well known that this local property confines the vector field in such a way, that it yields volume conservation over time for any considered part \( U \subset \Omega \) of the shape:

Proposition 2 Consider an open set \( U \subset \Omega \). Let now \( U(t) \) be the set of solutions of (1), if each point in \( U \) is shifted individually. If we assume that the deformation field \( v \) is divergence-free (3), then the volume of \( U(t) \) is constant over time:

\[
\frac{d}{dt} \int_{U(t)} dx = 0.
\]

Proof This statement is a particular case of the result in [Tes12, Lemma 8.8].

Helmholtz’s theorem [Ari62] implies that any sufficiently smooth vector field on the compact domain \( \Omega \) can be decomposed into the sum of a curl free and a divergence free component. It furthermore provides us with an explicit construction of the divergence free component of any velocity field:

\[
v := \nabla \times \Phi.
\]

In this context \( \Phi : \Omega \rightarrow \mathbb{R}^D \) is a \( C^\infty \) potential field. This vector field \( \Phi \) arises from a construction in the Helmholtz decomposition and it typically depends on the considered vector field \( v \). Unfortunately constructing \( \Phi \) from \( v \) is not straightforward and it is in
Figure 2: Three examples of shapes that are morphed into one another according to the initial value problem of Eq. (1). The centaur (a) and the human (b) are from the TOSCA [BBK08] and FAUST [BRLB14] dataset respectively. The armadillo (c) is from the AIM@SHAPE shape repository [aim]. (b) is a scan of a real person and very high resolution (214k vertices). The source and target shape are shown in white and the interpolations at times $t = 0.25, 0.5, 0.75$ in blue. The translation is not part of our deformation and was only introduced for clarity in the figures.

In the case of $D = 3$ spatial dimensions the construction of $v$ in (5) admits the following form:

$$
\nabla \cdot (\nabla \times \Phi) = 0.
$$

(6)

In the case of $D = 3$ spatial dimensions the construction of $v$ in (5) admits the following form:

$$
v = \begin{pmatrix}
\partial_2 \Phi_3 - \partial_3 \Phi_2 \\
\partial_1 \Phi_3 - \partial_3 \Phi_2 \\
\partial_1 \Phi_2 - \partial_2 \Phi_1
\end{pmatrix}
= \begin{pmatrix}
0 & -\partial_1 \Phi_2 & -\partial_2 \Phi_3 \\
\partial_1 \Phi_1 & 0 & -\partial_2 \Phi_3 \\
-\partial_1 \Phi_1 & -\partial_2 \Phi_1 & 0
\end{pmatrix}.
$$

(7)

5. Method

Regarding relevant applications we will mainly restrict ourselves to the case of $D = 3$. However, extensions to higher dimensions or the 2D case are straightforward.

5.1. Spatial representation

We need to describe the velocity fields $v$ in a more tangible manner such that we obtain a computationally feasible method. The problem is that there are infinitely many choices for functions $\Phi \in C^\infty(\Omega, \mathbb{R}^D)$. The most straightforward approach is choosing a discretization of the embedding space, e.g. with a voxel grid. The potential and deformation fields can then be defined by assigning a three-dimensional vector to every voxel. The problem with this approach is that it has cubic complexity which becomes costly very fast. Furthermore, we do not get a spatially continuous deforma-
tion and loose volume conservation and other desirable properties as a consequence. In the following we will introduce a finite, linear basis \( \{ v_1, \ldots, v_K \} \) for any velocity field on \( \Omega \) and derive a formulation to restrict it to only span smooth, divergence-free fields. The number of basis function can be adjusted for either speed or expressiveness.

The eigenfunctions of the Laplace-Beltrami operator are often used in shape analysis because of their useful properties like invariance to non-rigid deformations, smoothness and natural ordering. We use a similar basis for smooth, divergence-free vector fields in \( \mathbb{R}^D \). Without loss of generality the considered domain is assumed to be a \( D \)-dimensional cube \( \Omega := [0,1]^D \) and we translate and scale any shape to generously fit inside. We start with the eigenfunctions \( \{ \phi_1, \phi_2, \ldots \} \) of the standard Laplacian \( \Delta \) on \( \Omega \):

\[
\Delta \phi_k = \lambda^A_k \phi_k. \tag{8}
\]

This basis of eigenfunctions \( \{ \phi_1, \phi_2, \ldots \} \) is ordered such that the eigenvalues \( 0 \geq \lambda_k^A \geq \lambda_2^A \geq \ldots \) are descending. Furthermore, we require the Laplacian to admit Dirichlet boundary conditions \( \phi|_{\partial \Omega} = 0 \). The \( \phi_k \) can be determined analytically and they are exactly the sin elements of the Fourier basis:

\[
\{ \phi_1, \phi_2, \ldots \} = \left\{ \prod_{d=1}^D \frac{1}{2} \sin(\pi j_d) \right\mid (j_1, \ldots, j_D) \in \mathbb{N}^D \}. \tag{9}
\]

These functions \( \phi_k \) now form an orthonormal basis wrt. the \( \| \cdot \|_{L^2(\Omega)} \) norm. The eigenvalue \( \lambda_k^A \) of \( \Delta \) corresponding to the eigenfunction \( \phi_k \) is the following:

\[
\lambda_k^A := -\pi^2 \sum_{d=1}^D \frac{j_d^2}{d^2}. \tag{10}
\]

We can now map the potential basis to the velocity field to obtain a feasible description \( v \). For this purpose we directly insert the basis elements \( \phi_k \) from (9) into the places of the components \( \Phi_1, \ldots, \Phi_D \) of \( \Phi \) in (5). Due to the linearity of the curl operator \( \nabla \times \), this can be done for every entry of \( \Phi \) at a time, see (7) for \( D = 3 \). Overall, we obtain a basis \( \{ v_1, v_2, \ldots \} \) of the velocity field \( v \). In the 3D case it is explicitly defined as:

\[
\{ v_1, v_2, \ldots \} = \bigcup_{k=1}^{\infty} \left\{ \begin{pmatrix} 0 & -\partial_3 \phi_k \\ \partial_3 \phi_k & 0 \end{pmatrix}, \begin{pmatrix} -\partial_2 \phi_k & 0 \\ \partial_2 \phi_k & -\partial_1 \phi_k \end{pmatrix} \right\}. \tag{11}
\]

In this context the basis elements \( v_k \) are again sorted according to the eigenvalues \( \lambda_k^A \) of the corresponding \( \phi_k \) in descending order. Note that there are in general multiple basis functions \( v_k \) for each eigenvalue \( \lambda_k^A \). This yields a feasible description of the velocity field by computing \( v \) as a linear combination of the first \( K \) basis elements:

\[
v(x) = \sum_{k=1}^K v_k(x) a_k. \tag{12}
\]

The coefficients \( a_k \) can be defined as random variables with a Gaussian prior distribution \( a_k \sim N(0, \lambda_k) \). This assumption yields a probability distribution over all admissible deformation fields \( v \) which we will use later to obtain the optimization energy function.

Figure 3: Cross section of some deformation field basis functions \( v_k : \Omega \to \mathbb{R}^3 \) at \( x_3 = 0.5 \). Notice the low frequency structures for low \( k \) and increasing frequencies with higher indices.

The weights \( \lambda_k \) are constructed from the eigenvalues \( \lambda_k^A \) in the following manner:

\[
\lambda_k := (-\lambda_k^A)^{- q} = \left( \pi^2 \sum_{d=1}^D \frac{j_d^2}{d^2} \right)^{- \frac{q}{2}}. \tag{13}
\]

Intuitively this kind of weighting promotes a damping of the high frequency components of \( v \) and therefore yields a uniform vector field, because the coefficients \( a_k \) are sampled with a smaller variance \( \lambda_k \). This also justifies the truncation in (12), because the weights corresponding to high frequency basis elements are insignificantly small anyway. The mathematical background of the sampling approach in (12) is provided by the Karhunen-Loève expansion [Sul15, Ch. 11] which is an extension of the principal component analysis (PCA) for more general vector spaces.

The Dirichlet boundary conditions automatically guarantee that there is no flow out of the domain \( \Omega \), in the sense that the components of \( v_k \) are orthogonal to the outer normals at the boundary \( \partial \Omega \). This can be easily verified for the case \( D = 3 \) by computing the basis elements (11) inserting (9), but we refrain from proving this property here. It also becomes obvious by looking at instances of the velocity basis functions in Figure 3.

Our representation of the velocity field has several beneficial properties. First of all, it can be evaluated at any point in the domain \( \Omega \) and we do not need to introduce a spatial discretization. We will use this in our experiments to make our method applicable to very
high resolution shapes. Secondly, the particular weighting of the summands and the truncation in (12) induces an inbuilt low-pass filtering which produces smooth deformations because it favors the low frequency basis elements.

Classical shape registration methods like the iterative closest point (ICP) algorithm [BM92] determine a rigid transformation between two shapes. Interestingly, our approach can be interpreted as a direct extension of these methods. The deformation fields corresponding to translations and rotations are divergence free and uniform, and therefore, contained in our framework by design. This becomes even clearer when the basis function \(v_1\) in Figure 3 is examined. It is fairly similar to a rotation around the \(x_3\) axis. This especially holds near the center of the domain \(\Omega\) and deteriorates at its boundary \(\partial\Omega\). Actually it is fairly straightforward to verify that this equivalence holds up to first order around the center. The basis elements \(v_2\) and \(v_3\) express rotations around the \(x_2\) and \(x_1\) axis in a similar way. The considerations in this subsection raise the question whether a cubic domain \(\Omega\) is the best choice for our purposes. Following the work in [ZB07], [ZB08] we could pursue our approach in a spherical domain. This would lead to more complex basis functions \(v_k\) but the first three eigenfunctions would span the space of rotations without undesirable artifacts at the boundaries of the domain. Although this would be a nice theoretical property we refrain from using these basis functions here due their complex structure, especially because this does not lead to a quantifiable improvement of our method.

5.2. Temporal discretization

In order to evaluate the correspondence mapping \(f\) in (2) we have to solve the initial value problem (1) by a numerical integration scheme. The simplest choice in this context is the explicit Euler method. However, we decided to use a second order Runge-Kutta method [GH10, Ch. 9], because it has a significantly higher accuracy and therefore enables us to choose a coarser discretization. We subdivide the time space in an equidistant grid with \(T \in \mathbb{N}\) intervals and set the step size \(h = \frac{1}{T}\). This yields the following explicit iteration scheme:

\[
\begin{align*}
   x_n^{(0)} &:= x_0, \\
   x_n^{(t+1)} &:= x_n^{(t)} + hv\left(x_n^{(t)} + \frac{h}{2}v(x_n^{(t)})\right), \\
   f_n &:= x_n^{(T)}.
\end{align*}
\]

We typically choose \(T \in \{1, \ldots, 100\}\) in our experiments. In general, we have to make a trade off between runtime and accuracy when selecting a proper number of steps \(T\). We commonly also get meaningful transformations if we choose \(T\) to be small but we might loose some key properties of our framework like the volume preservation. This effect is illustrated in Figure 4 for the 2D shape of a bat transformed by a 90 degree rotation around the center. Note that the deformation field corresponding to this transformation is actually not contained in our framework due to our choice of domain and boundary conditions, see discussion in the previous subsection. If for this setup we now choose too few time steps \(T\), the shape shifts outward and the area expands. On the other hand, this effect becomes insignificantly small if we choose \(T \geq 10\).

![Figure 4: Area expansion with different step sizes using the Runge-Kutta integration. Left: Rotation around 90 degrees on a bat shape of the MPEG-7 dataset [Rafal] (black). If executed in one step \((T = 1)\) the shape expands (red) whereas for ten steps \(T = 10\) the area of the interior stays nearly the same (green). Right: Relative area expansion when performing the same deformation with increasing amount of steps \(T\).](image)

5.3. Optimization

In the previous sections we derived a coherent description of shape morphing using volume preserving deformation fields. We can now use this framework to construct an algorithm that matches two given point clouds \(X\) and \(Y\) while at the same time computing plausible interpolated shapes. In order to do that we need to simultaneously optimize for the deformation fields and the unknown correspondences. We encode the latter using a soft correspondence probability for the point pair \((x_m, y_m)\). Moreover, we use the coefficients \(a \in \mathbb{R}^K\) to represent the deformation fields \(v(x)\) which according to (12) are completely determined by \(a\).

Similar to [MS10] and [MZT14] we address shape registration in a probabilistic manner. We interpret the point cloud \(X\) as a Gaussian mixture model with the means located at the shifted points \(f_m = x_m^{(T)}\) and the covariance \(Cf_D \in \mathbb{R}^{D \times D}\) for some \(\sigma > 0\). This enables us to simultaneously determine the deformation field coefficients \(a \in \mathbb{R}^K\) and the correspondences \(W \in [0,1]^{N \times M}\) by applying an expectation maximization approach.

Only using Euclidean distance as the measure of similarity between points suffices to capture rigid deformations but this fails if source or target undergo large non-rigid deformations and often leads to incorrect local optimums. Therefore, we incorporate point-wise feature descriptors in our model. They can account for large scale deformations but also encode information about fine scale structures which steers the optimization towards the right optimum. Specifically, we use the SHOT descriptor [TSS10] with the distance function \(d_{\text{SHOT}} = \|\text{SHOT}(x_m) - \text{SHOT}(y_m)\|_2\) on the sets of points \(X\) and \(Y\). The standard GMM formulation only uses the Euclidean distance \(d_{\text{Euclid}} = \|y_m - f_m\|_2\) of the shifted point \(f_m\) and \(y_m\). We want to use a combination of both, therefore we will define the distance of \(f_m\) and \(y_m\) to be:

\[
d := d_{\text{Euclid}} + \frac{d_{\text{Euclid}}}{d_{\text{SHOT}}} d_{\text{SHOT}}.
\]

Here, \(\bar{d} \geq 0\) is the mean distance of a metric \(d\) regarding all point...
pairs in $\mathcal{X}$ and $\mathcal{Y}$ and the factor at $d_{\text{SHOT}}$ is used to ensure that both metrics have a comparable scaling. Note that $d$ is a metric on the point clouds $\mathcal{X}$ and $\mathcal{Y}$ as a positive combination of metrics. Using this notion of distance, we can specify how to update the soft correspondences $W$ using the current estimate of the deformation field parameters $a$. This corresponds to the E step of the expectation maximization algorithm:

$$W_{nm} := \exp \left( -\frac{1}{2\sigma^2} d_{nm}^2 \right) \left( 2\pi\sigma^2 \right)^{-\frac{D}{2}} + \sum_{i=1}^{N} \exp \left( -\frac{1}{2\sigma^2} d_{im}^2 \right).$$  \hspace{1cm} (16)

In the context of (12) it mentioned that the prior of the coefficients $a$ is a Gaussian distribution $a \sim \mathcal{N}(0, L)$, where $L := \text{diag}(\lambda_1, \ldots, \lambda_K)$. Together with the GMM assumption for the shape $\mathcal{X}$ we can construct the following energy function for the M step:

$$E(a) := \frac{1}{2} d^T L^{-1} d + \frac{1}{2\sigma^2} \sum_{m=1}^{M} \sum_{n=1}^{N} W_{nm} \rho(||y_m - f_a||_2).$$  \hspace{1cm} (17)

The function $\rho: \mathbb{R} \to [0, \infty)$ is the Huber loss [Hub64]:

$$\rho(r) = \begin{cases} \frac{r^2}{2} & |r| \leq r_0, \\ \frac{1}{2}r_0^2 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (18)

In our evaluations, we choose the outer slope as $r_0 := 0.01$. Note that the Huber loss does not directly arise from the standard GMM formulation, but it admits another probabilistic interpretation as an additive mixture of Huber density functions. It also helps to account for outliers and makes the deformation field estimation more robust in general.

In our experiments we apply a Gauss-Newton type approach to minimize the energy in (17). This results in an iterative method similar to the Levenberg-Marquardt algorithm [Lev44]. The overall expectation maximization algorithm now alternates between computing the weights $W^{(i)}$ according to (16) and performing one Gauss-Newton update step to obtain $a^{(i)}$. To initialize the algorithm we usually set the deformation field to be zero $a^{(0)} := 0$.

6. Experiments

We perform experiments for several applications including shape matching and interpolation to show that our method is general and flexible. Although we handle shapes with up to 200k and more vertices, the computation of the deformation field is always done on a downsampled version with 3000 vertices and then applied to the full resolution. We use Euclidean farthest point sampling. When downsampling the subset should include all relevant fine scale structures in order for the deformation field to move these correctly but we found 3000 sufficient for all our applications. As a preprocessing step, we align the shapes using PCA and shift them such that the empirical mean of the point clouds corresponds to the middle of the domain. When averaging over all experiments presented here, our algorithm takes about 370 seconds to compute all correspondences for one pair of shapes. Due to our a priori down-sampling the runtime is pretty consistent and almost independent of the number of vertices. All experiments were performed with MATLAB on a system with an Intel Core i7-3770 CPU clocked at 3.40GHz, 32 GB RAM and a GeForce GTX TITAN X graphics card running a recent Linux distribution. In all our experiments, we solely operate on the raw shape data and in particular do not need any ground truth correspondences.

6.1. Matching

We verify our method using the TOSCA dataset [BBK08] which contains 76 triangular meshes. The dataset is divided into 8 classes of humans and animals with several poses each and known intra-class correspondences. We set the hyperparameters $\sigma^2 := 0.01$, $T := 20$ and choose $K = 3000$ basis functions for the deformation field. Because $W^{(i)}$ only contains 3000 correspondences, we perform a nearest-neighbor search with the metric (15) to obtain a dense mapping. The evaluation is done with the Princeton benchmark protocol [KLF11]. Given the ground-truth match $(x, x^*) \in \mathcal{X} \times \mathcal{Y}$, the error of the calculated match $(x, y)$ is given by the geodesic distance between $y$ and $y^*$ normalized by the diameter of $\mathcal{Y}$.

$$e(x) = \frac{d_{\text{Geo}}(y, y^*)}{\text{diam}(\mathcal{Y})}.$$  

We plot cumulative curves showing the percentages of matches that are below an increasing threshold. As zero is the value for ground-truth matches, the ideal curve would be constant at 100. See Figure 5 for our results and Figure 6 for an example matching.
Intermediate time of these shapes has approximately 200 vertices and some of them are severely affected by topological noise. We set $\sigma^2 := 0.01$, $K := 3000$ and we use a temporal discretization of $T = 20$ steps. Again we match the null shape of every person to all its other poses. In Figure 7 we display the surface distance of the morphed shapes to the goal shape for some examples.

### 6.2. Registration

We apply our framework to the FAUST dataset [BRLB14], which contains data from scans of real humans with different poses. Each of these shapes has approximately 200k vertices and some of them are severely affected by topological noise. We set $\sigma^2 := 0.01$, $K := 3000$ and we use a temporal discretization of $T = 20$ steps. Again we match the null shape of every person to all its other poses. In Figure 7 we display the surface distance of the morphed shapes to the goal shape for some examples.

### 6.3. Shape Interpolation

In comparison to other shape matching approaches, our setup models deformations in a comprehensive manner. It has a built-in description of the actual transformation shapes undergo over time to morph into one another. Therefore it produces interpolated shapes byproduct.

In some sense, our approach can be considered to be the extension of other shape registration methods based on a Gaussian mixture model representation of a point cloud [MS10], [MZT 14]. These particular methods correspond to our approach, if we choose to integrate the initial value problem (1) with the forward Euler scheme and $T = 1$ time step. In particular this leads to a linear dependence of the quantity $f_n$ on the unknowns $a$ and the mapping $f$ for this case admits the following form:

$$f_n := f(x_n) = x_n + \sum_{k=1}^{K} v_k(x_n)a_k.$$  \hspace{1cm} (19)

As a consequence the shift (19) of each point $x_n$ over time yields affine linear trajectories. Therefore the intermediate configurations $x_{\text{lin}}(t)$ are equivalent to a pointwise linear interpolation. In contrast to this our method produces point trajectories which correspond to solutions $x(t)$ of the ODE (1). We can now evaluate those at an intermediate time $t \in [0, 1]$. Three examples of this qualitative evaluation were already displayed in Figure 2, another one is provided in 8.

In comparison to our approach linear interpolation distorts the shapes considerably, see Figure 9. It also changes the size of certain parts of the shape which is inconsistent with transformations in the real world. In contrast to this our method is locally volume preserving. Nevertheless, it can still stretch and bend the shapes. Therefore, it is suitable for modeling shape morphing for rigid, as well as elastic objects. Linear interpolation performs especially bad if the considered transformations include large rotations.

### Extrapolation

According to (1) the deformation field is independent of the time $t$ which makes the ODE autonomous. Therefore, we can also use the computed vector field $v$ to determine the solutions of this initial value problem at times $t > 1$ which produces extrapolated shapes, see Figure 10. It is obviously an underdetermined task and it is hard to evaluate quantitatively. Nevertheless the extrapolation shapes our method produces are in many cases quite realistic for moderate time spans $t \in [1, 1.5]$. Another example is displayed in Figure 11. We observe that the speed of the extrapolated shapes seems to slow down after a certain timespan, especially when the shape is moving in previously unoccupied space. For the optimization there is simply no incentive to impose any particular movement on these parts of the domain $\Omega$. However, the volume preservation and uniformity assumption infer to the algorithm that an extension of the previous movement to some extend is desirable. Overall the resulting shapes are visually appealing and not too severely affected by distortions.

### 7. Conclusion

We presented a novel method solving the shape correspondence problem while simultaneously computing a smooth, volume preserving deformation field between the input shapes. Furthermore, this deformation can be used to efficiently calculate plausible interpolated shapes between the inputs at any intermediate point in time. The method consists of two parts, the first is the optimization of the deformation field using an expectation maximization approach and the second applies the deformation to the input shapes using a Runge-Kutta scheme. The big advantage is that a subsampling with around 3000 vertices is sufficient to obtain the deformation field defined in the continuous embedding space due to our choice of basis. Therefore, the result can be applied to any resolution mesh without slowing the optimization.

We show quantitative results for shape correspondence and registration that can compare to state-of-the-art methods for these specific tasks and examples of shape interpolation and extrapolation that arise naturally from our pipeline.

#### 7.1. Limitations

Due to our choice of basis the deformation field is forced to be volume preserving. This makes sense in applications with the same object but prevents inter-object matchings - for example between two humans with different body shapes.

For the same reason, our method has problems with topological changes. According to Proposition 2 the volume preservation property applies to every subregion of the domain $\Omega$, including the intermediate space between parts of the shape. Therefore, separating two touching parts (for example two hands) is in theory possible but requires many high frequency deformation basis elements which makes the optimization costly.

Since there is not one unique, volume preserving deformation
Figure 7: Example registrations from the FAUST scan data set. The surface color corresponds to the Euclidean surface distance between scan and registration. The scale is the same on all plots. All measures are in cm. We report the average and maximum error under each image. Many errors occur due to the SHOT descriptors being corrupted at holes and in noisy areas (e.g. the hands), the volume being exactly preserved although this is only an approximate property due to noise in real scans and topological changes (second to the right).

Figure 8: Example of an interpolation between two input shapes (white) from the Kids dataset [RBW*14]. The interpolated shapes (blue) are at times $t = \{0.25, 0.5, 0.75\}$.

7.2. Future Work

Right now, the method will always find a solution that is globally volume preserving. This allows to find good deformations fields in the case of severe non-rigid deformations but is not applicable to partial data. In the future, we want to extend this method to work on real scans, for example from the Kinect, which naturally only show partial shapes by making the deformation only locally volume preserving. This might also help with the separation of close parts and handling non volume preserving deformations like style or class changes. Furthermore, we only calculate one time independent field for the entire deformation which means mass at one spatial point always needs to move in the same direction, even at a later time step. This restricts the complexity of the deformations that our method can handle, especially for large-scale motions over a longer period of time. It also leads to problems when non-matching parts of the

between two shapes, our interpolation is not guaranteed to be as-rigid-as-possible which is plausible in many applications. If the displacement is spatially far, we might end up with squeezed intermediate states that are volume preserving but are affected by undesirable distortions.

The assumption of (1) being autonomous can also be problematic, if different parts of the shape move through the same region of the embedding space in a contradictory manner. One example for this is a hand closing to a fist. At first the index and middle finger occupy parts of the embedding space before the thumb moves in the same area but in a different direction. See Figure 12. A possible remedy for this problem is making the deformation fields time dependent.

Figure 9: Comparison of the interpolated shapes at time $t = 0.5$ produced by our method (left) and linear interpolation (right). This example is hard because the head is rotated by more than 90 degrees in the target. While linear interpolation squeezes the head in between our method finds a more realistic solution.
input shapes overlap in the initialization. Future versions should allow more flexible types of deformation fields to extend it to a broader range of applications. We could for example associate different parts of the shape with different deformations fields or let them vary over time to address more difficult tasks.

References

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (2000), SIGGRAPH ’00, pp. 157–164. 2

[ADK16] AFLALO Y., DUBROVINA A., KIMMEL R.: Spectral generalized multi-dimensional scaling. *IJCV* 118, 3 (2016), 380–392. 2, 7

[AF11] ASHBURNER J., FRISTON K. J.: Diffeomorphic registration using geodesic shooting and gauss–newton optimisation. *NeuroImage* 55, 3 (2011), 954–967. 2

[aим] AIM@SHAPE repository. http://visionair.ge.imati.cnr.it:8080/ontologies/shapes/viewgroup.jsp?id=657–Armadillo_1. 4

[ALV08] ABLRECHT T., LUTHI M., VETTER T.: A statistical deformation prior for non-rigid image and shape registration. 2008 *IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1–8. 2

[Ari62] ARIS R.: Vectors, tensors, and the basic equations of fluid mechanics. Englewood Cliffs, N.J., Prentice-Hall, 1962. 3

[Ash07] ASHBURNER J.: A fast diffeomorphic image registration algorithm. *Neuroimage* 38, 1 (2007), 95–113. 2

[BBK06] B. BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS* 103, 5 (2006), 1168–1172. 2

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Numerical geometry of non-rigid shapes. Springer, 2008. http://tosca.cs.technion.ac.il/book/resources_data.html. 4, 7

[BHH00] BREGLER C., HERTZMANN A., BIERMANN H.: Recovering non-rigid 3D shape from image streams. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* 2 (2000), 690–696. 2

[BM92] BESL P. J., MCKAY N. D.: Method for registration of 3d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures* (1992), vol. 1611, International Society for Optics and Photonics, pp. 586–607. 6

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE, 4, 8

Figure 10: Example of an extrapolated shape produced by our method for two shapes. It can be determined using the temporally fixed deformation field v for simulating the initial value problem (1) up to the time \( t = 1.3 \). Source and target shape are white, one interpolated shape is shown in blue and the extrapolation is pink.

Figure 11: Example of an extrapolated shape from the KIDS dataset [RBW*14] at the time \( t = 1.3 \). The deformation field is usually magnified in the area between the input shapes and fans out in several directions further away from the input shapes. Therefore, choosing a really high time does not lead to broken shapes but the movement slows down more and more until it basically stops.
morphable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99 (2016).

[LSC05] Lipman Y., Sorkine O., Levin D., Cohen-Or D.: Linear-rotation-invariant coordinates for meshes. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 479–487.

[ML17] Ma J., Jiang J., Liu C., Li Y.: Feature guided gaussian mixture model with semi-supervised em and local geometric constraint for retinal image registration. *Information Sciences* 417 (2017), 126–142.

[MS10] Myronenko A., Song X.: Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (2010), 2262–2275.

[MTY06] Miller M. I., Trouvé A., Younes L.: Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision* 24, 2 (2006), 209–228.

[MZT*14] Ma J., Zhao J., Tian J., Yuille A. L., Tu Z.: Robust point matching via vector field consensus. *IEEE Transactions on Image Processing* 23, 4 (2014), 1706–1721.

[MZY16] Ma J., Zhao J., Yuille A. L.: Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on image Processing* 25, 1 (2016), 53–64.

[OBCS*12] Ovsiankov M., Ben-Chen M., Solomon J., Butscher A., Guibas L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30:1–30:10.

[PDBS*09] Paladini M., Del Bue A., Stosic M., Dodg M., Xavier J., Agapito L.: Factorization for non-rigid and articulated structure using metric projections. 2898–2905.

[Ral] Ralph R.: MPEG-7 Core Experiment CE-Shape-1 Test Set. http://www.dabi.temple.edu/~shape/MPEG7/dataset.html.

[RBW*14] Rodola E., Bulò S. R., Windheuser T., Vemtn M., Cremers D.: Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), 9, 10.

[RMC15] Rodola E., Moeller M., Cremers D.: Point-wise Map Recovery and Refinement from Functional Correspondence. In *Vision, Modeling and Visualization* (2015), The Eurographics Association.

[SMLF07] Salvi J., Matabosch C., Fofi D., Forest J.: A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.* 25, 5 (2007), 578–596.

[Sul15] Sullivan T. J.: *Introduction to Uncertainty Quantification*. Springer, 2015.

[TCL*13] Tam G. K. L., Cheng Z.-Q., Lai Y.-K., Langbein F. C., Liu Y., Marshall D., Martin R. R., Sun X.-F., Rossin P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (2013), 1199–1217.

[Tes12] Tesch G.: *Ordinary Differential Equations and Dynamical Systems*. AMS, 2012.

[THB08] Torresani L., Hertzmann A., Bregler C.: Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 5 (2008), 878–892.

[TSS10] Tombari F., Salti S., Stefano L. D.: Unique signatures of histograms for local surface description. In *Proceedings ECCV 16*, 6 (2010), 356–369.

[VFTS06] Von Funck W., Theisel H., Seidel H.-P.: Vector field based shape deformations. In *ACM Transactions on Graphics (TOG)* (2006), vol. 25, ACM, pp. 1118–1125.

[kZHC11] Van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707.

[VLC*17] Vestner M., Lähner Z., Boyarski A., Litany O., Slossberg R., Remez T., Rodola E., Bronstein A. M., Bronstein M. M., Kimmel R., Cremers D.: Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision (3DV)* (October 2017), 7.

[VWM15] Vaxman A., Müller C., Weber O.: Conformal mesh deformations with mÖbius transformations. *ACM Trans. Graph.* 34, 4 (July 2015), 55:1–55:11.

[VRC12] Vialard F.-X., Risser L., Rueckert D., Cotter C. J.: Diffeomorphic 3d image registration via geodesic shooting using an efficient adjoint calculation. *International Journal of Computer Vision* 97, 2 (2012), 229–241.

[VTSHH15] Von-Tyocowicz C., Schulz C., Seidel H.-P., Hildebrandt K.: Real-time nonlinear shape interpolation. *ACM Trans. Graph.* 34, 3 (May 2015), 34:1–34:10.

[WBR11] Wirth B., Bar L., Rumpf M., Sapiro G.: A continuum mechanical approach to geodesics in shape space. *International Journal of Computer Vision* 93, 3 (Jul 2011), 293–318.

[XZWB05] Xu D., Zhang H., Wang Q., Bao H.: Poisson shape interpolation. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling* (2005), pp. 267–274.

[ZB07] Zhao C., Burge J. H.: Orthonormal vector polynomials in a unit circle, part i: basis set derived from gradients of zernike polynomials. *Opt. Express* 15, 26 (2007), 18014–18024.

[ZB08] Zhao C., Burge J. H.: Orthonormal vector polynomials in a unit circle, part ii: completing the basis set. *Opt. Express* 16, 9 (2008), 6586–6591.

[ZSC08] Zhang H., Sheffer A., Cohen-Or D., Zhou Q., Van Kaick O., Tagliasacchi A.: Deformation-driven shape correspondence. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1431–1439.