Dense Non-Rigid Shape Correspondence using Random Forests

Emanuele Rodolà
TU München
rodola@in.tum.de

Samuel Rota Bulò
Fondazione Bruno Kessler
rotabulo@fbk.eu

Thomas Windheuser
TU München
windheus@in.tum.de

Matthias Vestner
TU München
vestner@in.tum.de

Daniel Cremers
TU München
cremers@tum.de

Abstract

We propose a shape matching method that produces dense correspondences tuned to a specific class of shapes and deformations. In a scenario where this class is represented by a small set of example shapes, the proposed method learns a shape descriptor capturing the variability of the deformations in the given class. The approach enables the wave kernel signature *to extend the class of recognized deformations from near isometries to the deformations appearing in the example set by means of a* random forest *classifier. With the help of the introduced spatial regularization, the proposed method achieves significant improvements over the baseline approach and obtains state-of-the-art results while keeping short computation times.*

1. Introduction

In the last decade there has been an increasing influx of work on finding and describing correspondences among 3-dimensional shapes (*i.e.*, 2-dimensional surfaces embedded into $\mathbb{R}^3$). Nevertheless, while the advances made by works such as [12, 3, 18, 21, 1, 8] have been dramatic, the problem is far from being solved.

Many methods in shape matching use a notion of similarity that is defined on a very general set of possible shapes. Due to the highly ill-posed nature of the shape matching problem, it is very unlikely that a general method will reliably find good matchings between arbitrary shapes. In fact, while many matching methods (such as methods based on metric distortion [17, 3, 16] and eigen-decomposition of the Laplacian [18, 21, 1]) mostly capture near-isometric deformations, others might consider too general deformations which are not consistent with the human intuition of correspondence. In applications where the class of encountered shapes is in-between, adapting the matching methods at hand is often very tedious.

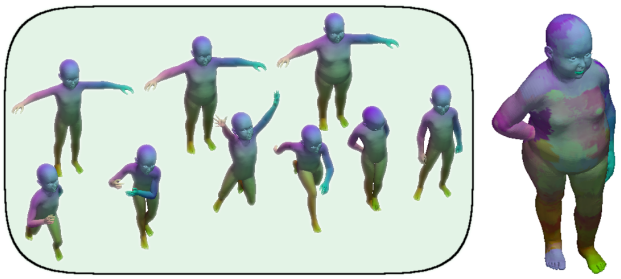

Figure 1. Example of dense shape matching using random forests under non-isometric deformations. Shapes in the shaded area are a subset of the training set. The forest is trained with wave kernel descriptors and consists of 80K training classes with 19 samples per class. Matches are encoded by color.

In this paper we try to bridge the gap between general shape matching methods and application-specific algorithms by taking a learning-by-examples approach. In our scenario, we assume to be given a set of training shapes which are equivalent up to some class of non-isometric deformations. Our goal is to learn from these examples how to match two shapes falling into the equivalence class represented by the training set. Recently, Litman *et al.* [10] took a signal processing approach to formulate a generic family of parametric spectral descriptors for deformable shapes. Differently, we treat the shape matching problem as a *classification* problem, where input samples are points on the shape manifold and the output class is an element of a canonical label set, which might *e.g.* coincide with the manifold of one of the shapes in the training set.

A first contribution of this paper consists in a new random forest classifier, which can tackle this unconventional classification problem in an efficient and effective way, starting from a general parametrizable shape descriptor. Our classifier is designed in a way to randomly explore the descriptor's parametrization space, and find the most discriminative features that properly recover the transforma-

tion map characterizing the shape category at hand. In this work, we consider the *wave kernel signature* (WKS) [1] as the shape descriptor. This descriptor is known to be invariant to isometric transformations, but the forest can effectively exploit it to match shapes undergoing deformations that are far from isometric. In a broad sense, the output of the random forest can be seen as a new descriptor by itself that is tuned to the shapes and deformations appearing in the training set. In this respect, the proposed method is complementary to existing shape descriptors as it can improve the performance of a given descriptor.

One of the main benefits of our approach is the fact that the random forest classifier gives for each point on the shape an ordered set of matching candidates, hence delivering a dense point-to-point matching. Since such a descriptor does not include any spatial regularity, we propose a regularization technique based on the *functional maps* representation [14]. We show experimentally that the proposed learning approach improves significantly the underlying general descriptor, being competitive with respect to other state-of-the-art matching pipelines on equivalent benchmarks.

It is worth mentioning that an approach related to ours was taken by Taylor *et al*. [22] for the task of human pose estimation from RGB-D data. To this end, they trained a random forest with large amounts of data covering all possible pose variations of a human shape. Their regression model relies on a parametrized, skinned, articulated reference 3D model in a canonical pose; the regression process then tries to infer a model parametrization that best explains the image data. Differently from this approach we do not need parametrized models, as we work directly with manifolds and their intrinsic quantities, thus allowing for more general matching scenarios. In addition, our method only requires a few exemplary training models encompassing the range of deformations the shape is expected to undergo, which drastically reduces the required computational effort.

2. Learning a Canonical Transformation with Random Forests

In order to employ a random forest classifier to address non-rigid shape matching, we learn from examples a *canonical transformation*, *i.e*. a transformation from the points of a shape $\mathcal{M}$ represented as a triangular mesh defined over a vertex set $V_{\mathcal{M}}$, to a canonical label set L. In Section 3 we will show how this can be used to obtain dense correspondences between non-rigid shapes.

Random forests [2] are ensembles of decision trees that have become very popular in the computer vision community to solve both classification and regression problems. Applications range from object detection, tracking and action recognition [7] to semantic image segmentation and categorization [20], and 3D pose estimation [22] to name just a few. The forest classifier is particularly appealing because its trees can be trained efficiently and techniques like bagging and randomized feature selection allow to limit the correlation among trees and thus ensure good generalization. We refer to [5] for a detailed review.

Inference. In the context of shape matching, a decision tree routes a point $\boldsymbol{m}$ of a test shape $\mathcal{M}$ from the root of the tree to a leaf node, where a probability distribution defined on a discrete label set L is assigned to the point. The path from the root to a leaf node is determined by means of binary decision functions called *split functions* located at the internal nodes, which given a shape point return L or R depending on whether the point should be forwarded to the left or to the right with respect to the current node. According to this inference procedure, each tree $t \in \mathcal{F}$ of a forest $\mathcal{F}$ provides a posterior probability $\mathrm{P}\left(\ell|\langle\boldsymbol{m}\rangle_{\mathcal{M}}, t\right)$ of label $\ell \in L$, given a point $\langle\boldsymbol{m}\rangle_{\mathcal{M}}$ in a test shape $\mathcal{M}$. This probability measure is the one associated with the leaf of tree $t \in \mathcal{F}$ that the shape point would reach. The prediction of the whole forest $\mathcal{F}$ is finally obtained by averaging the predictions of the single trees as follows:

$$\mathrm{P}\left(\ell|\langle\boldsymbol{m}\rangle_{\mathcal{M}}, \mathcal{F}\right) = \frac{1}{|\mathcal{F}|}\sum_{t\in\mathcal{F}} \mathrm{P}\left(\ell|\langle\boldsymbol{m}\rangle_{\mathcal{M}}, t\right) . \qquad (1)$$

The outcome of the prediction over a shape $\mathcal{M}$ can be represented as a left-stochastic matrix $\mathtt{X}_{\mathcal{M}}$ encoding the probabilistic canonical transformation, where

$$(\mathtt{X}_{\mathcal{M}})_{\ell\boldsymbol{m}} = \mathrm{P}\left(\ell|\langle\boldsymbol{m}\rangle_{\mathcal{M}}, \mathcal{F}\right) . \qquad (2)$$

for each $\ell \in L$ and $\boldsymbol{m} \in \mathcal{M}$.

Learning. During the learning phase, the structure of the trees, the split functions and the leaf posteriors are determined from a training set. Let $\{(\mathcal{R}_i, T_i)\}_{i=1}^{\mathsf{m}}$ be a set of m reference shapes $\mathcal{R}_i$ each equipped with a canonical transformation, *i.e*. a bijection $T_i : V_{\mathcal{R}_i} \to L$ between the vertex set of the reference shape and the label set L. A training set $\mathbb{T}$ for the random forest is given by the union of the graphs of the mappings T_i, *i.e*.

$$\mathbb{T} = \{(\boldsymbol{r}, T_i(\boldsymbol{r})) \,:\, \boldsymbol{r} \in V_{\mathcal{R}_i},\, 1 \le i \le \mathsf{m}\} .$$

The learning phase that creates each tree forming the forest consists in a recursive procedure that starting from the root iteratively splits the current terminal nodes. During this process, each shape point of the training set is routed through the tree in a way to partition the whole training set across the terminal nodes. The decision whether a terminal node has to be further split and how the splitting will take place is purely local, as it involves exclusively the shape points that have reached that node. A terminal node typically becomes a leaf of the tree if the depth of the node exceeds a given

limit, if the size of the subset of training samples reaching the node is small enough, or if the entropy of the label distribution for the sample is low enough. If this is the case, then the leaf node is assigned the label distribution of subset $\mathbb{S}$ of training samples that have reached the leaf, *i.e.*

$$\mathrm{P}(\ell|\mathbb{S}) = \frac{|\{(\boldsymbol{r},\ell) \in \mathbb{S}\}|}{|\mathbb{S}|} . \tag{3}$$

The probability distribution $\mathrm{P}(\cdot|\mathbb{S})$ will become the posterior probability during inference for every shape point reaching the leaf. Consider now the case where the terminal node is split. In this case, we have to select a proper split function $\psi(\langle \boldsymbol{r} \rangle_{\mathcal{R}_i}) \in \{\mathsf{L}, \mathsf{R}\}$ that will route a point $\boldsymbol{r}$ of shape $\mathcal{R}_i$ reaching the node to the left or right branch. An easy and effective strategy for guiding this selection consists in generating a finite pool Ψ of random split functions and retaining the one maximizing the information gain with respect to the label space L. The information gain $\mathrm{IG}(\psi)$ due to split function $\psi \in \Psi$ is given by the difference between the entropy of the node posterior probability defined as in (3) before and after having performed the split. In detail, if $\mathbb{S} \subseteq \mathbb{T}$ is the subset of the training set that has reached the node to be split and $\mathbb{S}^{\mathsf{L}}$, $\mathbb{S}^{\mathsf{R}}$ is the partition of $\mathbb{S}$ induced by the split function ψ, then $\mathrm{IG}(\psi)$ is given by

$$\mathrm{IG}(\psi) = \mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})) - \mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})\,|\psi) ,$$

where $\mathrm{H}(\cdot)$ denotes the entropy and

$$\mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})\,|\psi) = \frac{|\mathbb{S}^{\mathsf{L}}|}{|\mathbb{S}|}\mathrm{H}\left(\mathrm{P}\left(\cdot|\mathbb{S}^{\mathsf{L}}\right)\right) + \frac{|\mathbb{S}^{\mathsf{R}}|}{|\mathbb{S}|}\mathrm{H}\left(\mathrm{P}\left(\cdot|\mathbb{S}^{\mathsf{R}}\right)\right) .$$

2.1. Split functions for shape matching

During the build up of the forest the randomized training approach allows us to vary the parametrization of the shape descriptor for each point of the shape. In fact, we can in principle let the forest automatically determine the optimal discriminative features of the chosen descriptor for the matching problem at hand. Among the wide range of available choices [18, 21] we have opted for the recently introduced Wave Kernel Signature (WKS) [1]. The WKS is invariant to quasi-isometric deformations of the shape manifold, and is demonstrably robust to various other transformations that can happen in practice.

The WKS evaluates the probability of a quantum particle to be located at a point $\boldsymbol{m}$ of shape $\mathcal{M}$ under a certain energy distribution. Given an energy level e and by considering the following family of log-normal energy distributions

$$f_e(\nu) \propto \exp\left(-\frac{(\log e - \log \nu)^2}{2\sigma^2}\right) ,$$

with mean $\log e$ and variance σ^2, the expected probability of measuring the particle in $\boldsymbol{m}$ at any time is given by

$$p(\langle \boldsymbol{m} \rangle_{\mathcal{M}}; e) = \sum_{k=1}^{\infty} f_e^2(\nu_k)\phi_k^2(\langle \boldsymbol{m} \rangle_{\mathcal{M}}) , \tag{4}$$

where ν_k are the nonnegative eigenvalues of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ and ϕ_k are the corresponding orthonormal eigenfunctions. From a practical perspective, it can be shown [1] that the sum in (4) can be restricted to the first $\overline{k} < \infty$ components. We make explicit in (4) the dependency on $\overline{k}$ by writing:

$$p(\langle \boldsymbol{m} \rangle_{\mathcal{M}}; e, \overline{k}) = \sum_{k=1}^{\overline{k}} f_e^2(\nu_k)\phi_k^2(\langle \boldsymbol{m} \rangle_{\mathcal{M}}) . \tag{5}$$

We are now in the position of generating at each node of a tree during the training phase a pool of randomized split functions by sampling an energy level e, a number of eigenpairs $\overline{k}$ and a threshold τ. Accordingly, the split functions will take the form:

$$\psi(\langle \boldsymbol{m} \rangle_{\mathcal{M}}; e, \overline{k}, \tau) = \begin{cases} \mathsf{L} & \text{if } p(\langle \boldsymbol{m} \rangle_{\mathcal{M}}; e, \overline{k}) > \tau \\ \mathsf{R} & \text{otherwise} . \end{cases}$$

By doing so, we retain the full power of the WKS descriptor without resorting to a pre-defined parametrization, which might not be optimal over the whole shape.

3. Shape Matching and Regularization

In the following sections, for the sake of fluency we will simplify the notation and write $\boldsymbol{m}$ in place of $\langle \boldsymbol{m} \rangle_{\mathcal{M}}$, with the understanding that $\boldsymbol{m} \in V_{\mathcal{M}}$.

The simplest way to infer a correspondence from a forest prediction consists in assigning each point $\boldsymbol{m} \in V_{\mathcal{M}}$ to the most likely label according to its final distribution, *i.e.*, the label maximizing $\mathrm{P}(\ell|\boldsymbol{m}, \mathcal{F})$. If we are also given a reference shape $\mathcal{R}$ from the training set, the maximum a posteriori estimate of ℓ can be transformed into a point-to-point correspondence from $\mathcal{M}$ to $\mathcal{R}$ via the known bijection $T : V_{\mathcal{R}} \to L$. Figures 2(a)(b) show an example of this approach. The resulting correspondence is exact for about 50% of the points, whereas it induces a large metric distortion on the rest of the shape. However, this is not a consequence of the particular criterion we adopted when applying the prediction. Indeed, the training process is oblivious to the underlying manifolds as it is only based on pointwise information: the correspondence estimates are taken *independently* for each point and thus the metric structure of the test shape is not taken into account during the regression. Nevertheless, as we shall see, the predicted distributions carry enough information that can be exploited to obtain a consistent matching.

3.1. Regularization

In the following we show how a given forest prediction can be regularized in a way to produce a meaningful correspondence. However, instead of acting directly on the matching, we will operate in the space of *functions* defined on shapes.

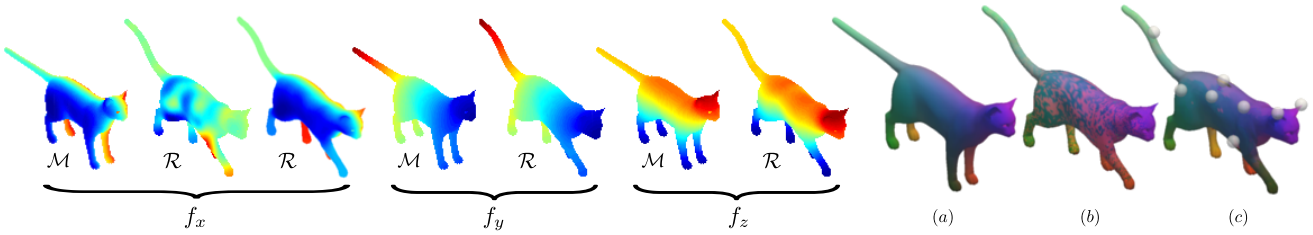

Figure 2. The coordinate functions from a test shape $\mathcal{M}$ (standing cat) are transferred to a reference shape $\mathcal{R}$ (walking cat) via the functional map $C_{\mathcal{M},\mathcal{R}}$ induced by the forest prediction. Most of the ambiguities arise in f_x, and are due to the global intrinsic symmetry of the cat. The first column shows the map f_x on the test cat, while the second and third columns are obtained by mapping f_x without and with regularization respectively. The remaining four columns show the mappings of f_y and f_z *without* regularization. The symmetric ambiguities disappear as a result of the regularization process (columns (a)-(c), matches encoded by color).

Functional maps. We make use of the functional map framework introduced in [14]. A (probabilistic) correspondence between two shapes $\mathcal{M}$ and $\mathcal{R}$, given in terms of a left-stochastic matrix $\mathtt{X}_{\mathcal{M},\mathcal{R}}$, can be related to a linear map $C : \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{R})$ between the sets of square-integrable scalar valued functions on $\mathcal{M}$ and $\mathcal{R}$ via

$$\mathtt{C}_{\mathcal{M},\mathcal{R}} = \Phi_{\mathcal{R}}^{\top} \mathtt{X}_{\mathcal{M},\mathcal{R}} \Phi_{\mathcal{M}} \,, \tag{6}$$

where $\mathtt{C}_{\mathcal{M},\mathcal{R}}$ denotes the matrix-form of the linear map C and matrices $\Phi_{\mathcal{M}}, \Phi_{\mathcal{R}} \in \mathbb{R}^{\mathtt{n} \times k}$ contain the first k eigenfunctions of the discrete Laplace-Beltrami operators $\Delta_{\mathcal{M}}$ and $\Delta_{\mathcal{R}}$, respectively. The correspondence $\mathtt{X}_{\mathcal{M},\mathcal{R}}$ can be expressed in terms of the canonical transformation $T_{\mathcal{R}}$ and the forest prediction $\mathtt{X}_{\mathcal{M}}$ as

$$(\mathtt{X}_{\mathcal{M},\mathcal{R}})_{\boldsymbol{rm}} = (\mathtt{X}_{\mathcal{M}})_{T_{\mathcal{R}}(\boldsymbol{r})\boldsymbol{m}} = \mathtt{X}_{\mathcal{R}}^{-1} \mathtt{X}_{\mathcal{M}} \,, \tag{7}$$

where $\mathtt{X}_{\mathcal{R}}$ is the matrix-form of transformation $T_{\mathcal{R}}$, which is invertible because $T_{\mathcal{R}}$ is a bijection. By combining (6) and (7) we finally get

$$\mathtt{C}_{\mathcal{M},\mathcal{R}} = \Phi_{\mathcal{R}}^{\top} \mathtt{X}_{\mathcal{R}}^{-1} \mathtt{X}_{\mathcal{M}} \Phi_{\mathcal{M}} \,, \tag{8}$$

which maps scalar functions between test and reference shape.

In Figure 2 (first 7 columns) we use such a construction to map the coordinate functions $f_i : \mathcal{M} \to \mathbb{R}$ (where $i \in \{x, y, z\}$) to scalar functions on $\mathcal{R}$. Specifically, we plot $\boldsymbol{f}_i$ and their reconstructions $\boldsymbol{g}_i = \Phi_{\mathcal{R}} \mathtt{C}_{\mathcal{M},\mathcal{R}} \Phi_{\mathcal{M}}^{\top} \boldsymbol{f}_i$. Note that the reference shape is axis-aligned, so that the x coordinates of its points grow from the right side (blue) to the left side of the model (red).

Metric distortion using functional maps. The plots we show in Figure 2 tell us that most of the error in the correspondence arises from the (global) intrinsic symmetries of the shape. As mentioned previously, this is to be expected since the training process does not exploit any kind of structural information about the manifolds. This suggests the possibility to regularize the prediction by introducing metric constraints on the correspondence. Specifically, we consider an objective of the form

$$E(\mathtt{X}) = c(\mathtt{X}_{\mathcal{M},\mathcal{R}}, \mathtt{X}) + \rho(\mathtt{X}) \,, \tag{9}$$

where $\mathtt{X}$ is a correspondence between shapes $\mathcal{M}$ and $\mathcal{R}$. The first term (or *cost*) ensures closeness to the prediction given by the forest, while the second term is a regularizer giving preference to geometrically consistent solutions. A natural choice for such *regularity* term is the L_p-relaxed Gromov-Hausdorff metric distortion [11, 16]

$$\rho(\mathtt{X}) = \frac{1}{2} \sum_{\substack{\boldsymbol{r},\boldsymbol{r}' \in V_{\mathcal{R}} \\ \boldsymbol{m},\boldsymbol{m}' \in V_{\mathcal{M}}}} \epsilon(\boldsymbol{m},\boldsymbol{r},\boldsymbol{m}',\boldsymbol{r}') \mathtt{X}_{\boldsymbol{rm}} \mathtt{X}_{\boldsymbol{r}'\boldsymbol{m}'} \,, \tag{10}$$

where function ϵ is the absolute distortion of the metric functions $d_{\mathcal{M}}$, $d_{\mathcal{R}}$ on the two manifolds, namely

$$\epsilon(\boldsymbol{m},\boldsymbol{r},\boldsymbol{m}',\boldsymbol{r}') = |d_{\mathcal{M}}(\boldsymbol{m},\boldsymbol{m}') - d_{\mathcal{R}}(\boldsymbol{r},\boldsymbol{r}')| \,. \tag{11}$$

Other choices for ϵ are also possible [16]. With these definitions, $\rho(\mathtt{X})$ directly quantifies to what extent the given mapping $\mathtt{X}$ deviates from isometry. In other words, a minimizer of (9) is expected to be close to the predicted matching $\mathtt{X}_{\mathcal{M},\mathcal{R}}$ while at the same time preserving pairwise distances on the two shapes as much as possible.

Since finding a solution to (9) involves taking all possible pairs of matches on the two shapes, the problem is of combinatorial nature and thus in general very difficult to solve. Fortunately, a more convenient formulation can be obtained if we use the language of functional maps. Let $\mathtt{C}_{\mathcal{M},\mathcal{R}}$ be the functional map induced by the correspondence $\mathtt{X}_{\mathcal{M},\mathcal{R}}$ according to Eq. (8). The functional (9) can be rewritten as

$$E(\mathtt{C}) = \|\mathtt{C}_{\mathcal{M},\mathcal{R}} - \mathtt{C}\|_F^2 + \rho(\mathtt{C}) \,, \tag{12}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. Suppose we are given a (possibly sparse) collection of matches $O \subset V_{\mathcal{M}} \times V_{\mathcal{R}}$. Then for each $(\boldsymbol{p},\boldsymbol{q}) \in O$ we can define two distance maps $d_{\boldsymbol{p}} : \mathcal{M} \to \mathbb{R}$ and $d_{\boldsymbol{q}} : \mathcal{R} \to \mathbb{R}$ as

$$d_{\boldsymbol{p}}(\boldsymbol{x}) = d_{\mathcal{M}}(\boldsymbol{p},\boldsymbol{x}) \,, \qquad d_{\boldsymbol{q}}(\boldsymbol{y}) = d_{\mathcal{R}}(\boldsymbol{q},\boldsymbol{y}) \,. \tag{13}$$

shape $\mathcal{M}$ | shape $\mathcal{N}$ colors from $\mathcal{M}$ mapped via $\mathtt{C}$ | shape $\mathcal{N}$ colors from $\mathcal{M}$ mapped via $\mathtt{C}_{\text{reg}}$

Figure 3. Example of a dense matching among two shapes $\mathcal{M}$ and $\mathcal{N}$ without making use of an intermediate reference. Both shapes are taken from the 'jumping' set, while the shapes of the training set are taken from the 'squat' sequence by Vlasic *et al.*[1]

With these definitions, we can express the regularity term $\rho(\mathtt{C})$ in the embedding functional space as

$$\rho(\mathtt{C}) = \sum_{(\boldsymbol{p},\boldsymbol{q})\in O} \|\mathtt{C}\hat{\boldsymbol{d}}_{\boldsymbol{p}} - \hat{\boldsymbol{d}}_{\boldsymbol{q}}\|_2^2 \,, \tag{14}$$

where $\hat{\boldsymbol{d}}_{\boldsymbol{p}} = \Phi_{\mathcal{M}}^\top \boldsymbol{d}_{\boldsymbol{p}}$ and $\hat{\boldsymbol{d}}_{\boldsymbol{q}} = \Phi_{\mathcal{R}}^\top \boldsymbol{d}_{\boldsymbol{q}}$ are the distance map representations in the respective bases. Note that Eqs. (13) are now encoding the pairwise distances appearing in (11). In order for the regularization to work as expected, the provided collection of matches should constrain well the solution, in the sense that it should help to disambiguate the intrinsic symmetries of the shape. For example, matches along the tail of the cat would bring little to no information on what solution to prefer. In practice, we can seek for a few matches that cover the whole shape and be as accurate as possible. To this end, we generate evenly distributed samples $V_{\text{fps}} \subset V_{\mathcal{M}}$ on the test shape via farthest point sampling [11] by using the extrinsic Euclidean metric. Then, we construct a matching problem that attempts to minimize an objective of the form given in Eq. (10), but restricted to the set of *predicted* matches

$$O = \{(\boldsymbol{m},\boldsymbol{r}) \in V_{\text{fps}} \times V_{\mathcal{R}} \,|\, (\mathtt{X}_{\mathcal{M},\mathcal{R}})_{\boldsymbol{r}\boldsymbol{m}} > 0\} \,. \tag{15}$$

In practice this set is expected to be small, since the prediction given by the forest is very sparse and we select around 50 farthest samples per test shape (≈0.2% of the total number of points on the adopted datasets). This results in a small matching problem that we solve via game-theoretic matching [16], a ℓ_1-regularized technique that allows to obtain sparse, yet very accurate solutions in an efficient manner. Once a sparse set of matches is obtained, we solve (12) as the weighted least-squares problem

$$\min_{\mathtt{C}} \|\mathtt{C}_{\mathcal{M},\mathcal{R}} - \mathtt{C}\|_F^2 + \sum_{(\boldsymbol{p},\boldsymbol{q})\in O} \omega_{\boldsymbol{p}\boldsymbol{q}} \|\mathtt{C}\hat{\boldsymbol{d}}_{\boldsymbol{p}} - \hat{\boldsymbol{d}}_{\boldsymbol{q}}\|_2^2 \,, \tag{16}$$

where $\omega_{\boldsymbol{p}\boldsymbol{q}} \in [0,1]$ are weights (provided by the game-theoretic matcher) giving a measure of confidence for each match $(\boldsymbol{p},\boldsymbol{q}) \in O$. Figure 2(c) shows the result of the regularization performed using 25 sparse matches (indicated by small spheres).

3.2. Matching without a reference

In this section we consider a scenario in which a reference shape $\mathcal{R}$ is *not* available for the matching process, but one is instead interested in a correspondence between two new shapes, both unknown to the forest.

Let $\mathcal{M}$ and $\mathcal{N}$ be two test shapes, and let $\mathtt{X}_{\mathcal{M}}$, $\mathtt{X}_{\mathcal{N}}$ denote the corresponding label predictions as defined in (2), *i.e.* for each point $\boldsymbol{m} \in V_{\mathcal{M}}$ and each label $\ell \in L$ the probability $\mathrm{P}(\ell|\boldsymbol{m})$ is given by $(\mathtt{X}_{\mathcal{M}})_{\ell\boldsymbol{m}}$, and accordingly for $\mathcal{N}$. We are now interested in obtaining a probabilistic correspondence matrix $\mathtt{X}_{\mathcal{M},\mathcal{R}}$ between $\mathcal{M}$ and $\mathcal{R}$. To this end, we interpret each element of $\mathtt{X}_{\mathcal{M},\mathcal{N}}$ as the probability that a given point from $\mathcal{M}$ corresponds to a point in $\mathcal{N}$, *i.e.* $(\mathtt{X}_{\mathcal{M},\mathcal{N}})_{\boldsymbol{n}\boldsymbol{m}} = \mathrm{P}(\boldsymbol{n}|\boldsymbol{m})$ for any $\boldsymbol{m} \in V_{\mathcal{M}}$ and $\boldsymbol{n} \in V_{\mathcal{N}}$. By using Bayes' theorem and by taking a uniform prior over the shapes' points, we obtain

$$\begin{aligned}(\mathtt{X}_{\mathcal{M},\mathcal{N}})_{\boldsymbol{n}\boldsymbol{m}} = \mathrm{P}(\boldsymbol{n}|\boldsymbol{m}) &= \sum_{\ell\in L} \mathrm{P}(\boldsymbol{n}|\ell)\,\mathrm{P}(\ell|\boldsymbol{m}) \\ &= \sum_{\ell\in L} (\tilde{\mathtt{X}}_{\mathcal{N}})_{\ell\boldsymbol{n}} (\mathtt{X}_{\mathcal{M}})_{\ell\boldsymbol{m}} = \tilde{\mathtt{X}}_{\mathcal{N}}^\top \mathtt{X}_{\mathcal{M}} \,,\end{aligned} \tag{17}$$

where

$$(\tilde{\mathtt{X}}_{\mathcal{N}})_{\ell\boldsymbol{n}} = \mathrm{P}(\boldsymbol{n}|\ell) = \frac{\mathrm{P}(\ell|\boldsymbol{n})}{\sum_{\boldsymbol{n}'\in V_{\mathcal{N}}} \mathrm{P}(\ell|\boldsymbol{n}')} = \frac{(\mathtt{X}_{\mathcal{N}})_{\ell\boldsymbol{n}}}{\sum_{\boldsymbol{n}'\in V_{\mathcal{N}}} (\mathtt{X}_{\mathcal{N}})_{\ell\boldsymbol{n}'}} \,.$$

As in the case of matching to a reference shape, there is the need to regularize the obtained correspondence $\mathtt{X}_{\mathcal{M},\mathcal{N}}$ with the techniques introduced in Section 3.1. However, in this case, the correspondence matrix is not necessarily sparse and, hence, the set of candidates given in (15) is in general not small. In addition, we would like to avoid calculating (17) explicitly as this is a product of two big matrices. Again, we overcome these issues by shifting to a functional map representation:

$$\mathtt{X}_{\mathcal{M},\mathcal{N}} \approx \Phi_N \underbrace{(\Phi_N^\top \tilde{\mathtt{X}}_{\mathcal{N}}^\top)(\mathtt{X}_{\mathcal{M}} \Phi_M)}_{\mathtt{C}} \Phi_M^\top \,. \tag{18}$$

Note that the brackets are crucial to simplify and significantly speed up computation. Also, columns of $\mathtt{X}_{\mathcal{M},\mathcal{N}}$ can be calculated on-the-fly without the need of storing the whole correspondence matrix. It is indeed enough to store $\Phi_N \mathtt{C}$ and Φ_M. This is useful to determine the candidate points for the game-theoretic matching step, which can be

[1] http://people.csail.mit.edu/drdaniel/mesh_animation/

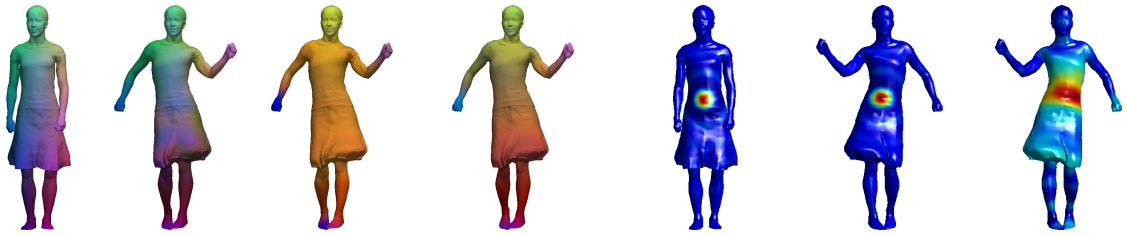

Figure 4. Comparison between our method and an approach based on WKS affinity using shapes from the dataset of Vlasic *et al*. Columns one to four show the predicted and regularized solutions for both approaches. The last three columns show how the indicator function at one point gets functionally mapped to a second shape, by using the (non-regularized) $\mathtt{C}$ obtained from the forest, and by $\mathtt{C}_{\mathrm{WKS}}$.

obtained iteratively by following a sampling strategy on the support of each column of $\mathtt{X}_{\mathcal{M},\mathcal{N}}$. Even the most simple strategy, such as choosing the 20 most likely points on $\mathcal{N}$ for each of the farthest samples on $\mathcal{M}$ leads to very accurate results (see Figure 3).

4. Experimental results

In all our experiments we used the WKS as pointwise descriptor for the training process. As in [14], we limited the size of the bases on the shapes to the first 100 eigenfunctions of the Laplace-Beltrami operator, computed using the cotangent scheme [13].

4.1. Comparison with dense methods

In this set of experiments we compare with the state of the art techniques in (dense) non-rigid shape matching, namely the functional maps pipeline [14], blended intrinsic maps (BIM) [8], and the coarse-to-fine combinatorial approach of [19]. We perform these comparisons on the TOSCA high-resolution dataset [4]. The dataset consists of 80 shapes belonging to different classes, with resolutions ranging in 4K-52K points. Shapes within the same class have the same connectivity and undergo nearly-isometric deformations. Ground-truth point mapping among shapes from the same class is available. In particular, given a predicted map $f : \mathcal{M} \to \mathcal{N}$ and the corresponding ground-truth $g : \mathcal{M} \to \mathcal{N}$, we define the *error* of f as

$$\varepsilon(f, g) = \sum_{\boldsymbol{m} \in V_{\mathcal{M}}} d_{\mathcal{N}}(f(\boldsymbol{m}), g(\boldsymbol{m}))\,, \qquad (19)$$

where $d_{\mathcal{N}}$ is the geodesic metric on $\mathcal{N}$, normalized by $\sqrt{Area(\mathcal{N})}$ to allow inter-class comparisons. Similarly, we define the average (pointwise) geodesic error as $\frac{\varepsilon(f,g)}{|V_{\mathcal{M}}|}$.

Although the methods considered in these experiments do not rely on any prior learning, the comparison is still meaningful as it gives an indication of the level of accuracy that our approach can attain in this class of problems. The experiments were designed on the same benchmark and following a procedure similar to the one reported in [8, 14]. Specifically, for each model $\mathcal{M}$ of a class (*e.g.*, the class of dogs), we randomly picked other 6 models from the same class (not including $\mathcal{M}$), and trained a random forest with them (thus, we only considered classes with at least 6 shapes). Then we predicted a dense correspondence for $\mathcal{M}$ according to the technique described in Section 2.

We show the results of this experiment in Fig. 5 (right). Each curve depicts the percentage of matches that attain an error below the threshold given on the x-axis. Our method (red line) detects 90% correct correspondences within a geodesic error of 0.05. Almost all correct matches are detected within an error of 0.1. This is compatible with and even improves the results given by the other methods on the same data. Note that our training process only makes use of pointwise information (namely, the WKS); in contrast, the functional maps pipeline (blue line) adopts several heuristics (WKS preservation constraints in addition to orthogonality of $\mathtt{C}$, region-wise features, etc.) in order to constrain the solution optimally [14]. Upon visual inspection, we observed that most of the errors in our method were due to the poor choice of points made in the regularization step. This is analogous to what is reported for the BIM method [8]. Typically, we observed that around 20 well-distributed points are sufficient to obtain accurate results.

4.2. Sensitivity to training parameters

We performed a sensitivity analysis of our method with respect to the parameters used in the training process, namely the size of the training set and the number of trees in the forest. In these experiments we employed the cat models from the TOSCA dataset (28K vertices) with the corresponding ground-truth.

In Fig. 5 (middle) we plot the average geodesic error obtained by a test shape (depicted along the curve) as we varied the number of shapes in the training set. The geodesic error of the correspondence stabilizes when at least 6 shapes are used for training. This means that only 6 samples per label are sufficient in order to determine an optimal parametrization of the nearly-isometric deformations occur-

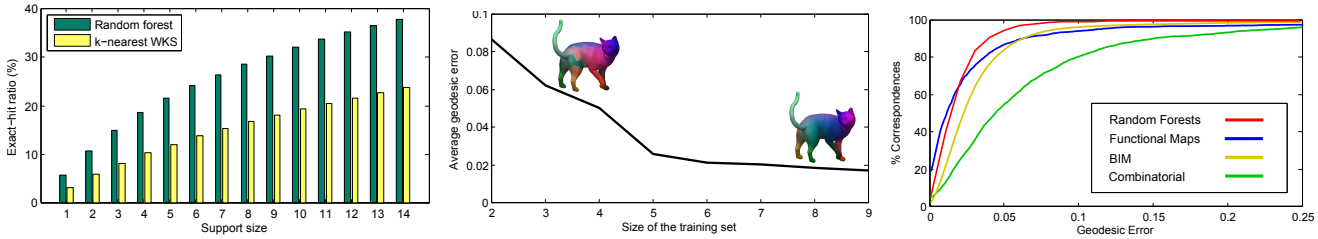

Figure 5. Left: Fraction of exact matches predicted by a random forest vs. maximum support size of the probability distributions on a test shape. The forest was trained with 9 shapes. Middle: Sensitivity to number of shapes used in the training set. Note how the correspondence predicted using little training data (top-left model) is only partially regularized. Right: Comparison with the state-of-the-art methods on nearly-isometric shapes (TOSCA). Symmetric correspondences are considered correct solutions for all methods.

ring on the shape. This result contrasts the common setting in which random forests are trained with copious amounts of data [22, 6], making the approach rather practical when only limited training data is available.

Figure 5 (left) shows the change in accuracy as we increase the number of trees in the forest. Note that increasing the number of trees directly induces a larger support of the probability distributions over L. In other words, each point of the test shape receives more candidate matches if the forest is trained with more trees (see Eq. (1)). The hit ratio in the bar plot is defined as the fraction of *exact* predictions given by the forest over the entire test shape. We compare the results with the hit ratio obtained by looking for k-nearest neighbors in WKS descriptor space, with k equal to the maximum support size employed by the forest at each level. From this plot we see that the forest predictions are twice as accurate as WKS predictions for equal support sizes. In particular, random forest predicts the *exact* match for almost half (around 14K points) of the shape when trained with 15 trees.

Finally, in Fig. 4 we show a qualitative comparison between our method and an approach based on WKS. The rationale of this experiment is to show that the prediction given by the forest gives better results than what can be obtained without prior learning within the same pipeline (*i.e.*, prediction followed by regularization). Specifically, for each point in one shape we construct a probability distribution on the other shape based on a measure of descriptor affinity in WKS space. We then estimated a functional map $\mathsf{C}_{\mathrm{WKS}}$ from the resulting set of constraints, and plotted a final correspondence before and after regularization.

4.3. Learning non-isometric deformations

In this section we consider a scenario in which the shapes to be matched may undergo more general (*i.e.*, far from isometric) deformations. Examples of such deformations include local and global changes in scale, topological changes, resampling, partiality, and so forth. Until now, few methods have attempted to tackle this class of problems. Most dense approaches [8, 14, 19, 15] are well-defined in the quasi-isometric and conformal cases only; instances of inter-class matching were considered in [8], but the success of the method depends on the specific choice of (usually hand-picked) feature points used in the subsequent optimization. Sparse methods considering the general setting from a metric perspective [16, 3, 17] attempt to formalize the problem by using the language of quadratic optimization, leading to difficult and highly non-convex formulations. An exception to the general trend was given in [24], where the matching is formulated as a linear program in the product space of manifolds. The method allows to obtain dense correspondences for more general deformations, but it assumes consistent topologies and is computationally expensive (∼2 hours to match around 10K vertices). Another recent approach [9] attempts to model deviation from isometry in the framework of functional maps, by seeking compatible harmonic bases among two shapes. However, it relies on a (sparse) set of matches being given as input and it shares with [24] the high computational cost.

As described in Section 2, the forest does not contain any explicit knowledge of the type of deformations it is asked to parametrize. This means that, in principle, one could feed the learning process with training data coming from any collection of shapes, with virtually no restrictions on the transformations that the shapes are allowed to undergo. Clearly, an appropriate choice of the pointwise descriptor should be made in order for the forest to provide a concise and discriminative model. To test this scenario, we constructed a synthetic dataset consisting of 8 high-resolution (80K vertices) models of a kid under different poses (quasi-isometries), and 11 additional models of increasingly corpulent variants of the same kid (local scale deformations) with a fixed pose (see Fig. 1). The shapes have equal number of points and point-to-point ground-truth is available.

We test the trained random forest with a plump kid having a previously unseen pose. Note that the result is reasonably accurate if we keep in mind the noisy setting: the forest was trained with WKS descriptors, which are originally designed for quasi-isometric deformations, and thus not expected to work well in the more general setting [10]. Despite being just a qualitative evaluation, this experiment

demonstrates the generality of our approach. The matching process we described can still be employed in general non-rigid scenarios if provided with limited, yet sufficiently discriminative training data.

4.4. Performance

The proposed approach was implemented in C++ and tested on an Intel Core i7 with 8GB memory. In order to assess performance of the method, we built a training set from 20 nearly-isometric shapes of 80K points each. The learning process on this dataset took ∼35 min to train one tree. We trained 15 trees and subsequently employed the resulting forest to produce dense matches for a collection of 10 shapes of 80K points each. The average matching time was 4.30 sec per shape without regularization. Regularization took 22 sec on average including farthest point sampling of 50 points (5%, Eq. (15)), computation of exact geodesics [23] (85%, Eq. (13)), minimization of the metric distortion (5%, Eq. (10)), and solving the resulting least squares problem (5%, Eq. (16)).

5. Conclusions

In this paper we proposed the adoption of the random forest training paradigm for dense correspondence problems among deformable shapes. To our knowledge, this is the first attempt at introducing a statistical learning view on this family of problems. We demonstrate the effectiveness of our approach on a standard benchmark, where we obtain state-of-the-art results and very low prediction times for shapes with tens of thousands of vertices. The approach is flexible in that it provides a means to model deformations which are far from isometric, and it consistently obtains high predictive performance on all tested scenarios.

Acknowledgments

E. R. was supported through an Alexander von Humboldt Fellowship. In addition, we acknowledge support through the ERC Starting Grant "ConvexVision".

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV Workshops*, 2011. 1, 2, 3

[2] L. Breiman. Random forests. In *Machine Learning*, volume 45, 2001. 2

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *PNAS*, 2006. 1, 7

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008. 6

[5] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. In *Found. and Trends in Comput. Graph. Vis.*, 2012. 2

[6] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *Proc. CVPR*, June 2011. 7

[7] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 33(11), 2011. 2

[8] V. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *SIGGRAPH 2011*, 2011. 1, 6, 7

[9] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2pt4), 2013. 7

[10] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *TPAMI*, 36:171–180, aug 2013. 1, 7

[11] F. Mémoli. Gromov-Wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11, 2011. 4, 5

[12] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. of Comput. Math.*, 5(3), 2005. 1

[13] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. VisMath*, 2002. 6

[14] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–30:11, July 2012. 2, 4, 6, 7

[15] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. *Comput. Graph. Forum*, 29(5):1555–1564, 2010. 7

[16] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. CVPR*, 2012. 1, 4, 5, 7

[17] E. Rodolà, A. Torsello, T. Harada, T. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. In *Proc. ICCV*, 2013. 1, 7

[18] R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*. Eurographics Association, 2007. 1, 3

[19] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum*, 30(5), 2011. 6, 7

[20] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. CVPR*, 2008. 2

[21] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP*. Eurographics Association, 2009. 1, 3

[22] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. CVPR*, 2012. 2, 7

[23] O. Weber, Y. S. Devir, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Trans. Graph.*, 27(4), Nov. 2008. 8

[24] T. Windheuser, U. Schlickewei, F. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *Proc. ICCV*, 2011. 7