# Discrete Time Evolution Process Descriptor for Shape Analysis and Matching

SIMONE MELZI, University of Verona, Italy
MAKS OVSJANIKOV, École Polytechnique, France
GIORGIO ROFFO, University of Glasgow, United Kingdom
MARCO CRISTANI and UMBERTO CASTELLANI, University of Verona, Italy

In shape analysis and matching, it is often important to encode information about the relation between a given point and other points on a shape, namely, its *context*. To this aim, we propose a theoretically sound and efficient approach for the simulation of a discrete time evolution process that runs through all possible paths between pairs of points on a surface represented as a triangle mesh in the discrete setting. We demonstrate how this construction can be used to efficiently construct a multiscale point descriptor, called the *Discrete Time Evolution Process Descriptor*, which robustly encodes the structure of neighborhoods of a point across multiple scales. Our work is similar in spirit to the methods based on diffusion geometry, and derived signatures such as the HKS or the WKS, but provides information that is complementary to these descriptors and can be computed without solving an eigenvalue problem. We demonstrate through extensive experimental evaluation that our descriptor can be used to obtain accurate results in shape matching in different scenarios. Our approach outperforms similar methods and is especially robust in the presence of large nonisometric deformations, including missing parts.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; • **Theory of computation** → *Computational geometry*;

Additional Key Words and Phrases: Discrete time evolution process, geodesic distances, shape signature, point-to-point matching

**ACM Reference format:**
Simone Melzi, Maks Ovsjanikov, Giorgio Roffo, Marco Cristani, and Umberto Castellani. 2018. Discrete Time Evolution Process Descriptor for Shape Analysis and Matching. *ACM Trans. Graph.* 37, 1, Article 4 (January 2018), 18 pages.
https://doi.org/10.1145/3144454

## 1 INTRODUCTION

Accurate shape matching plays an important role in shape analysis and geometry processing with many applications, such as object animation (Sumner et al. 2005), object retrieval (Tangelder and Veltkamp 2004; Lian et al. 2013), model reconstruction from partial views (Bernardini and Rushmeier 2002), and shape exploration and co-segmentation (Huang et al. 2014), among many others. One prominent approach consists of defining a point *descriptor* or *signature* that captures the most notable characteristics of a given shape from the "point of view" of each point. Typically, such a descriptor is constructed by considering a (possibly arbitrary large) neighborhood of a point and encoding geometric properties of this neighborhood in a robust and easily comparable way (Johnson and Hebert 1999; Belongie et al. 2002; Frome et al. 2004). In this light, several strategies have been introduced to extract local information in an efficient and theoretically sound fashion. For instance, a popular approach is based on the concept of *diffusion* geometry (Coifman and Lafon 2006) for the description of 3D shapes (Sun et al. 2009; Gebal et al. 2009; Bronstein and Bronstein 2011; Aubry et al. 2011). The main idea consists of characterizing the neighborhood of a given point through an *evolution* process that measures how information propagates on the manifold. For example, it is well known that heat tends to diffuse slower at points with a positive curvature, and faster at points with a negative curvature (e.g., see Sun et al. (2009)). Although very informative, diffusion-based methods are by their nature global and potentially sensitive to nonisometric shape deformations. Moreover, for computational reasons, such methods are often based on an approximation of the spectral decomposition of the shape.

In this article, we propose to encode the relation between points by exploiting an alternative evolution paradigm. Rather than considering a *continuous* time evolution, we introduce a special operator that is applied iteratively on the surface. Our *process* operator is designed to explicitly integrate information across the shape by taking into account the relation of a given point to the rest of the surface. In particular, our operator is specified by a function that encodes the *direct* pairwise relations between surface points. Then, the iterative procedure allows our method to explore *indirect* (or second-order) relations. This leads to a new *discrete-time* evolution scheme to represent the gradual change of the "context" of the each point. As an example, when the relation function is defined by the geodesic distance, our process operator encodes the set of paths of gradually increasing lengths. Moreover, we effectively combine the contribution of each evolution state to obtain a final *score* that summarizes how a point is influenced by the rest of the shape after an infinite number of steps. Key to our approach is an observation that such multistep computation can be done *exactly* and efficiently in practice by solving a single linear system of equations without requiring an approximation via a reduced spectral basis.

Parts of this work were supported by Marie-Curie CIG-334283, a CNRS chaire d'excellence, chaire Jean Marjoulet from École Polytechnique, FUI project TANDEM 2, and a Google Focused Research Award.
Authors' addresses: S. Melzi, Computer Science Department, University of Verona, Strada le Grazie, Verona 15, 37134, Italy; email: simone.melzi@univr.it; M. Ovsjanikov, LIX, École Polytechnique, Paris, France; email: maks@lix.polytechnique.fr; G. Roffo, School of Computing Science, University of Glasgow, Glasgow, United Kingdom; email: giorgio.roffo@glasgow.ac.uk; M. Cristani and U. Castellani, Computer Science Department, University of Verona, Verona, Italy; emails: {marco.cristani, umberto.castellani}@univr.it.

https://doi.org/10.1145/3144454

We show the benefits of this new framework by using it to derive a novel point signature. We build our process operator by using the geodesic distances as relation functions. Therefore, rather than considering only the shortest paths, which are known to be susceptible to noise, the iterative use of our operator captures the information about all paths of arbitrary lengths between each pair of points. Finally, we introduce a *multiscale* strategy to capture information from both small and large neighborhoods by controlling the length of the distance allowed in a single step. We demonstrate that our descriptor, which we call the *Discrete time Evolution Process* (DEP) descriptor, is highly discriminative and is more robust than other methods to several kinds of shape transformations, such as nonisometric deformations and missing parts.

Our approach closely resembles methods based on diffusion geometry (Coifman and Lafon 2006; Sun et al. 2009; Gebal et al. 2009; Bronstein and Bronstein 2011; Aubry et al. 2011), especially in its use of an infinite number of paths to characterize points and their relations. Nevertheless, by basing our descriptor directly on discrete time evolution and geodesic distances rather than on the differential operator, such as the Laplace-Beltrami operator, we are able to provide complementary information with respect to existing diffusion-based signatures. Moreover, in contrast to these approaches, our descriptor can be computed *exactly*, without truncating an eigendecomposition.

We demonstrate the effectiveness of our method in matching shapes across a wide range of challenging scenarios. For this, we provide results on benchmarks with an increasing level of complexity and consider different matching strategies: (i) based on a direct comparison of point descriptors and (ii) based on a functional map (Ovsjanikov et al. 2012) framework. Our results outperform the state-of-the-art and show that the information contained in our DEP descriptor is an alternative to the one captured by existing techniques.

*Roadmap.* The rest of the article is organized as follows. We begin in Section 2 by describing related work and highlighting connections with our method. Section 3 provides the mathematical background of the proposed *discrete time evolution process* on surfaces. Connections to the continuous process are given in Section 4. In Section 5, we describe how to obtain a multiscale descriptor from the discrete time evolution process and provide further insights into relation to similar work in Section 6. Section 7 presents a wide range of experiments that explore the benefits of the obtained descriptor in several scenarios. Section 8 concludes the article with a discussion of limitations and future work.

## 2 RELATED WORK

In the past several decades, a wide variety of point or shape descriptors has been proposed in different research areas, such as computer graphics, computer vision, and pattern recognition (e.g., see Bronstein et al. (2012) for a summary of some approaches). In this section, we focus on the work that is most closely related to ours.

*Distance-Based Methods.* In their seminal paper, Osada et al. (2002) defined a signature as a probability distribution (i.e., *shape distribution*) sampled from a shape function that encodes the geometric properties of the given object. As an example, they proposed the distribution of the Euclidean distances between pairs of randomly selected points on the surface of a 3D model. Other approaches exploit intrinsic properties of the shape by collecting information from geodesic distances (Mitchell et al. 1987; Kimmel and Sethian 1998; Surazhsky et al. 2005). In Hilaga et al. (2001), a geodesic-based Multiresolutional Reeb Graph (MRG) is proposed to effectively capture the topological properties of 3D objects. Similarly, a pose-invariant shape descriptor is introduced in Gal et al. (2007) as a 2D histogram estimated from the *local-diameter* function (i.e., the measure of the diameter of the 3D shape in the neighborhood of each point on a surface, also known as Shape Diameter Function (Shapira et al. 2008)) and the *centricity* function (i.e., the average geodesic distance from one point to all other points on the shape). In Ion et al. (2011), the so-called *eccentricity* transform has been introduced as the distribution of the lengths of the longest geodesics on the 3D surface. Recently, Xin et al. (2016) introduced a new function called the *intrinsic girth function* (IGF), which captures the shortest nonzero geodesic path starting and ending at the same point, and have also described an efficient method to compute the IGF on a triangular mesh. Finally, Carrière et al. (2015) have introduced a multiscale signature based on topological structure of the distribution of geodesic distances centered at a given point.

*Local Descriptors.* Other nondistance-based methods have been employed to analyze *local* geometric properties of a shape and to characterize each point on a shape. Important examples of point-based signatures are the well-known *spin images* (Johnson and Hebert 1999) and *shape context* (Belongie et al. 2002; Frome et al. 2004). In particular, several methods have been proposed that consider a multiscale version of the local neighborhood of the given point (Yang et al. 2006). For instance, in Pottmann et al. (2009), a multiscale approach is employed for the computation of integral invariant features (Manay et al. 2004). In Zaharescu et al. (2012), a 3D version of the well-known SIFT descriptor (Lowe 2004) is proposed, while Tombari et al. (2010) present a signature called SHOT. These descriptors describe the local shape structure around a point starting from either the geodesic distances or the distribution of normal directions.

Finally, among the newer methods, there is Anisotropic Windowed Fourier Transform (AWFT) descriptors (Melzi et al. 2016). Starting from a collection of functions, the descriptors are obtained as a weighted linear combination of the coefficients of the AWFT.

*Diffusion-Based Methods.* Another trend in shape analysis consists of exploiting diffusion (e.g., *heat* diffusion) properties on geometric shapes (Coifman and Lafon 2006; Sun et al. 2009; Gebal et al. 2009; Bronstein and Bronstein 2011; Aubry et al. 2011). The general idea is to measure the propagation of information on 3D objects that, in some cases, can be interpreted as a random walk among surface points (Coifman and Lafon 2006; Bronstein and Bronstein 2011). This framework has led, in particular, to the well-known *Heat Kernel Signature* (HKS) (Sun et al. 2009; Gebal et al. 2009), which, roughly speaking, encodes the amount of heat remaining at a point after a certain amount of time. A similar approach is defined for the so-called *Wave Kernel Signature* (WKS) (Aubry et al. 2011), which captures the particle oscillations at different frequencies of a dynamic system defined on a shape. Interestingly, Bronstein and

Bronstein (2011) proposed a scheme that is able to generalize all the previously diffusion-based approaches. All of these methods are based on the spectral decomposition of the Laplace-Beltrami operator on manifold shapes. Recently, some work has been proposed to avoid this need by introducing an alternative method for heat kernel computation (Patané 2014) at the expense of additional computation time, when considering all points on the shape.

*Learning-Based Methods.* Generally speaking, methods based on spectral geometry are invariant to isometric transformations but are likely to fail when this hypothesis is violated. In order to address this problem, a class of methods has been introduced by exploiting a learning-by-example approach (Litman and Bronstein 2014; Corman et al. 2014; Litman et al. 2014; Boscaini et al. 2015). In Litman and Bronstein (2014), the so-called *optimal* point descriptor is proposed. In Corman et al. (2014), a learning procedure has been introduced within the functional map framework. Litman et al. (2014) proposed a learning strategy in the context of the bag-of-word paradigm. Finally, Boscaini et al. (2015) introduce a matching method that exploits convolutional neural networks in the spectral domain.

*Connections With Our Work.* Our work is related to diffusion-based methods in that it arises from an evolution process similar to the HKS and WKS, which are based on the evolution process of the heat and the motion of the particles on the surface, respectively. Our method exploits an alternative surface evolution paradigm and defines a new path-based, multiscale point descriptor by capturing the paths of multiple lengths without the need to compute any spectral decomposition. Rather than focusing on the infinitesimal or differential characteristics of the shape (such as those defined by the Laplace-Beltrami operator), we argue for encoding the "integral" properties of points and their neighborhoods by considering the relations across all other points on the shape and by simulating a discrete-time evolution process. Our method is highly discriminative and captures information that is alternative to the geometric attributes obtained by other traditional diffusion-based approaches. Even if our descriptor is built without employing any learning strategy, our results are stable and robust even when the hypothesis of isometric transformation is violated.

## 3 DISCRETE TIME EVOLUTION PROCESS

We define an iterative process that evolves on the manifold and that we observe at discrete and regular timestamps. Differently from the diffusion-based methods, such as HKS or WKS, our evolution process is not necessarily based on a differential equation that controls the process behavior. Therefore, we do not need to know the evolution law that explains this process, but the process itself is fully derived from a generic pairwise relation function as described below.

### 3.1 Evolution Process on Manifold

*Continuous Shape.* Let $S$ be a smooth surface and $\mathcal{F}(S,\mathbb{R})$ the set of real functions defined on $S$. We introduce a function $d$ that represents a generic relation between each pair of surface points.

$$d : S \times S \longrightarrow \mathbb{R}$$
$$d : (x,y) \longmapsto d(x,y) \in \mathbb{R} \text{ s.t. } d(x,y) \geq 0,$$

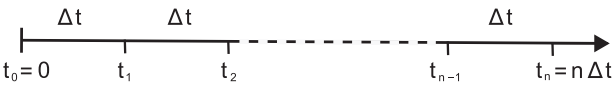


Fig. 1. Time discretization used for the discrete time evolution process.

Starting from the relation $d$, we then define a process that evolves in the discrete time setting and depends only on the underlying geometry of the surface. This process is governed by the relation $d$ that represents how each point $x$ is "*influenced*" by every other point $y$ on the surface.

We fix a finite time interval $\Delta t \in \mathbb{R}$, with $\Delta t > 0$, and divide the positive real line in a discrete collection of instants $\{t_0 = 0, t_1, t_2, \ldots, t_{l-1}, t_l, \ldots\}$, where $t_l = l\Delta t$, as shown in Figure 1. Given an initial state represented by a real function $f_0 \in \mathcal{F}(S,\mathbb{R})$, we define the desired process as follows:

$$f_1(x) := \int_S d(x,y) f_0(y) \mathfrak{d}\mu(y), \tag{1}$$

where $f_1$ is the state after one time interval $\Delta t$ and $\mathfrak{d}\mu$ is the infinitesimal area element. We assume that the process is homogeneous and the evolution is the same at every time step. Thus, we can iterate this operation by obtaining the state after a generic number $l$ of discrete intervals as

$$f_l(x) := \int_S d(x,y) f_{l-1}(y) \mathfrak{d}\mu(y). \tag{2}$$

Using this relation, we introduce the *process operator* $\mathbf{A}$ that gives us the state of the process after the discrete interval of time. Therefore, we write

$$\mathbf{A}(f_0) = f_1 = \int_S d(\cdot,y) f_0(y) \mathfrak{d}\mu(y) \tag{3}$$

and iterating on $l \in \mathbb{N}$, we obtain that

$$f_l = \mathbf{A}(f_{l-1}) = \mathbf{A}(\mathbf{A}(f_{l-2})) = \cdots = \underbrace{\mathbf{A}(\mathbf{A}(\cdots(\mathbf{A}(f_0))))}_{l\text{−times}}.$$

Finally, for every point $x$ on $S$, we define a scalar value that sums up the contributions of the process from every discrete time $t_l$ with $l \in \{0, \ldots, +\infty\}$. Therefore, we introduce an *evolution process* score $s$ at a point $x \in S$ as follows:

$$s(x) = \sum_{l=0}^{\infty} f_l(x) = f_0(x) + \sum_{l=1}^{\infty} \int_S d(x,y) f_{l-1}(y) \mathfrak{d}\mu(y). \tag{4}$$

*Discrete Shape.* In the discrete setting, we represent $S$ by a triangular mesh $\mathcal{M}$ with $N$ vertices $\mathcal{V} = \{v_i\}_{i=1}^N$. We divide the surface in barycells centered at every vertex $v_i$ of the mesh and denote by $\Omega_i$ their areas. Note that, in this case, the set $\mathcal{F}(S,\mathbb{R}) = \mathbb{R}^N$ and the function $d$ corresponds to a matrix $\mathbf{D} \in \mathbb{R}^N \times \mathbb{R}^N$, where

$$D_{i,j} = d(v_i, v_j) \in \mathbb{R} \text{ s.t. } d(v_i, v_j) \geq 0, \ \forall v_i, \ v_j \in \mathcal{V}.$$

Following the discussion above, we can redefine the same evolution process that we created in the continuous setting as follows. Given an initial state $f_0 \in \mathbb{R}^N$, our process is defined as

$$f_1(v_i) := \sum_{j=1}^{N} \Omega_j D_{i,j} f_0(v_j), \tag{5}$$

and $f_1$ is the state after one interval $\Delta t$. Here, the integral from Equation (1) is replaced by the weighted sum according to the local areas. Then, we can obtain the state after a generic number $l$ of discrete time intervals as

$$f_l(v_i) := \sum_{j=1}^{N} \Omega_j D_{i,j} f_{l-1}(v_j). \tag{6}$$

We build the diagonal matrix $\mathbf{\Omega} = diag(\Omega_i)$, with the area $\Omega_i$ of the barycell centered in $v_i$. Then, we can adopt a matrix notation to model the process. We denote by $\mathbf{A}$ the discrete *process operator* defined by the $N \times N$ real matrix $\mathbf{D\Omega}$; more explicitly, the element $(i, j)$ of the matrix $\mathbf{A}$ is $A(i, j) = \Omega_j d(v_i, v_j)$. Now, we can write the discrete analogue of Equation (3):

$$\mathbf{A}f_0 = f_1 = \mathbf{D\Omega}f_0 = \sum_{j=1}^{N} \Omega_j D_{\cdot,j} f_0(v_j)$$

and iterating on $l \in \mathbb{N}$ we obtain that

$$f_l = \mathbf{A}f_{l-1} = \mathbf{A}(\mathbf{A}(f_{l-2})) = \cdots = \underbrace{\mathbf{A}(\mathbf{A}(\cdots(\mathbf{A}f_0)))}_{l-\text{times}} = \mathbf{A}^l f_0,$$

where $\mathbf{A}^l = (\mathbf{D\Omega})^l$ is the process operator for $l$ steps. Finally, we obtain our evolution process score via the discrete version of Equation (4):

$$s(v_i) = \sum_{l=0}^{\infty} f_l(v_i) = f_0(v_i) + \sum_{l=1}^{\infty} \left( \sum_{j=1}^{N} A^l(i, j) f_0(v_j) \right). \tag{7}$$

### 3.2 Analysis of Higher Order Relations

Here, we discuss the meaning of the proposed evolution process in terms of higher-order relations between points that belong to the surface. Note that while the process evolves our process operator, $\mathbf{A}^l$ takes into account indirect links between the vertices. We consider these indirect links as higher-order relations.

More specifically, Equation (5) encodes the first-order relations of $v_i$ and can be rewritten as

$$f_1(v_i) := \sum_{j=1}^{N} a_{i,j} f_0(v_j), \tag{8}$$

where $a_{i,j} := A(i, j)$ represents how vertex $v_i$ is "*influenced*" by $v_j$ on the discretized shape. Now, we evaluate the behavior of the process in the higher-order relations. Let $\sigma$ denote a generic subset of $l + 1$ vertices $\{v_{\sigma(0)}, \ldots, v_{\sigma(l)}\} \subseteq \mathcal{V}$, with possible repetitions. We can define the contribution $\mathrm{w}_\sigma$ of $\sigma$ to the evolution process as

$$\mathrm{w}_\sigma = \prod_{k=0}^{l-1} a_{\sigma(k),\sigma(k+1)} = \prod_{k=0}^{l-1} \Omega_{\sigma(k+1)} d(v_{\sigma(k)}, v_{\sigma(k+1)}), \tag{9}$$

Let $\mathbb{P}^l_{i,j}$ denote the collection of all the subsets $l + 1$ vertices, starting with $i$ ($\sigma(0) = i$) and ending with $j$ ($\sigma(l) = j$). To account for all subsets in $\mathbb{P}^l_{i,j}$ and following standard linear algebra, we compute the following:

$$\mathbf{A}^l(i, j) = \sum_{\sigma \in \mathbb{P}^l_{i,j}} \mathrm{w}_\sigma \tag{10}$$

Therefore, our process operator for $l$ order $\mathbf{A}^l(i, j)$ represents how the $v_i$ is "*influenced*" by all $l$-order relations between $v_i$ and $v_j$.

Moreover, for every initial state $f_0$, we define the $l$-order evolution state at vertex $v_i$ as

$$f_l(v_i) = \sum_{j=1}^{N} \mathbf{A}^l(i, j) f_0(v_j). \tag{11}$$

Intuitively, $f_l(v_i)$ encodes the quantity of the state $f_0$ "*absorbed*" in $v_i$ from the $l$-order relations in the evolution process.

The *evolution process* score for each vertex of the mesh is defined summing over $l$ as $s(v_i) = \sum_{l=0}^{\infty} f_l(v_i)$; thus, it can be obtained as in Equation (7). Now, to generalize the computation of this score, we introduce the score operator $\mathbf{S}$ as the geometric series of matrix $\mathbf{A}$:

$$\mathbf{S} = \sum_{l=0}^{\infty} \mathbf{A}^l. \tag{12}$$

Note, however, that since $\mathbf{S}$ increases exponentially with $l$, the infinite sum may diverge; thus, $\mathbf{S}$ may not be well defined. To overcome this problem, we employ a simple generating function strategy. Generating function regularization (Graham et al. 1994) is used to assign a consistent value for the sum of a possibly divergent series. To this end, we define the *regularized* score operator as

$$\check{\mathbf{S}} = \sum_{l=0}^{\infty} r^l \mathbf{A}^l, \tag{13}$$

where $r$ is a scalar regularization parameter. In order to ensure the convergence, we choose $r$ so that $|r| < \frac{1}{\rho(\mathbf{A})}$, where $\rho(\mathbf{A})$ is the spectral radius of $\mathbf{A}$. See Appendix I for a more formal argument. From an algebraic view, $\check{\mathbf{S}}$ can be efficiently computed by using the convergence property of the geometric power series of a matrix (Hubbard and Hubbard 2001):

$$\check{\mathbf{S}} = (\mathbf{I} - r\mathbf{A})^{-1}. \tag{14}$$

Matrix $\check{\mathbf{S}}$ encodes the information about the geometry between our set of vertices and the chosen relation function $d$. Finally, we can obtain the evolution process scores for each vertex simply as

$$\check{s}(v_i) = [(\check{\mathbf{S}} f_0)](i), \tag{15}$$

which can equivalently be computed by solving the linear system: $(\mathbf{I} - r\mathbf{A})(\check{s}) = f_0$. This interpretation makes it clear that, for every vertex $v_i$, the computed score $\check{s}(i)$ is obtained by summing the contributions of all relations starting at $v_i$ and evolving along the surface under the conditions imposed by the process operator $\mathbf{A}$, for all time scales, going to infinity. Note that, from the pairwise relations encoded by $\mathbf{D}$, our evolution process facilitates obtaining and incorporating higher-order information. All components involved in our framework are highlighted. In Section 5, we will pre a deeper analysis of these components.

## 4 CONNECTION TO CONTINUOUS PROCESS

Our discrete time evolution process is defined by

$$f_{l+1} = \mathbf{A}f_l.$$

Let us now suppose that a matrix $\mathbf{B}$ exists such that:

$$\mathbf{A} = e^{\Delta t \mathbf{B}}.$$

This assumption is far from trivial and is possible only for a particular set of $\mathbf{A}$. If we satisfy this condition and focusing on the limit case of $\Delta t \to 0$, we can consider the continuous process associated to the following partial differential equation:

$$\frac{\partial f}{\partial t} = \mathbf{B}f.$$

It is well known that, for every initial state $f_0$, the state of this continuous process at time $t$ is defined as

$$f_t = e^{t\mathbf{B}} f_0.$$

Thus, we have that, for every discrete time interval $\Delta t$ and for every integer $l$,

$$f_{l\Delta t} = e^{l\Delta t\mathbf{B}} f_0 = \mathbf{A}^l f_0,$$

which corresponds to our discrete time evolution process. In the same spirit, we can obtain a connection between the integration over all times of the continuous process and our evolution process score. In continuous notation, we can compute the following equation:

$$\int_0^\infty f_\tau d\tau = \int_0^\infty e^{\tau\mathbf{B}} f_0 d\tau = -\mathbf{B}^{-1} f_0.$$

Now, if we fix a time interval $\Delta t$ as we did in Section 3, we can rewrite this integration at discrete times as

$$\sum_{l=0}^{\infty} \mathbf{A}^l f_0 \Delta t = (I - \mathbf{A})^{-1} f_0 \Delta t.$$

Assuming that $\mathbf{A} = e^{\Delta t\mathbf{B}}$ as above, for $\Delta t \to 0$, we have that

$$(I - \mathbf{A})^{-1} f_0 \Delta t = -(\Delta t\mathbf{B})^{-1} f_0 \Delta t = -\mathbf{B}^{-1} f_0.$$

Therefore, if we consider our evolution process score $\check{s} = (I - \mathbf{A})^{-1} f_0$ and forgetting $\Delta t$ (as multiplicative constant value), $\check{s}$ corresponds to the integration over all times of the continuous process.

This highlights how the continuous process is related to our discrete time evolution process and also how the score $\check{s}$ is related to the continuous process in this specific context.

Clearly, this connection depends on a strong assumption and holds only for very specific relation functions $d$. One of the main goals of our framework, however, is to enable the use of a generic pairwise relation without limiting its choice in order to meet some conditions. For this reason, we encode this relation via a discrete process, which allows us to obtain a score even when the evolution is not governed by a continuous-time, diffusion-like procedure.

## 5 PROPOSED DESCRIPTOR

We investigate how the proposed evolution process described in Section 3 can be exploited to define a new class of 3D point descriptors. The choice of the components of the evolution process is crucial to identify the encoded information. In particular, the main components are (i) the relation function, (ii) the regularization parameter, and (iii) the starting state.

*The Relation Function.* The most important parameter is the relation function $\mathbf{D}$ that, in principle, can be defined by any positive two-variable function. In practice, the characteristics of the chosen relation function determine the kind of information that is spread across the shape by the evolution process. Therefore, the effectiveness of the derived descriptor highly depends on this choice. In this context, the most natural options are distance functions, kernels, or generic (dis-)similarity measures. In our work, we build the process operator $\mathbf{A}$ using the geodesic distance $\mathcal{G}(v_i, v_j)$, that is, the length of the shortest path on $\mathcal{M}$ between vertices $v_i$ and $v_j$. We motivate this choice by the fact that geodesic distances are, by definition, invariant under isometric transformations and can be used to capture the geometry of the shape effectively. Moreover, as we demonstrate below, our discrete time evolution procedure that allows incorporation of information across an infinite set of paths helps to gain both informativeness and robustness against nonisometric shape changes. More specifically, we define the matrix of vertex relations $\mathbf{D}$ as

$$\mathbf{D}(i, j) = 1 - \hat{\mathcal{G}}(v_i, v_j),$$

where $\hat{\mathcal{G}}(v_i, v_j) = \frac{\mathcal{G}(v_i, v_j)}{diam(\mathcal{M})}$ is the *normalized* geodesic distance and $diam(\mathcal{M})$ is the diameter of $\mathcal{M}$, that is defined as the maximum of the geodesic distances between every pair of vertices on the surface. Therefore, $\mathbf{D}(i, j) \in [0, 1]$, $\mathbf{D}(i, i) = 1, \forall i$, and the process operator writes $\mathbf{A} = \mathbf{D}\Omega$. This choice implies that each vertex absorbs more information from its neighborhood, decreasing gradually the influence of vertices that are further away from it. Moreover, the integration of the geodesic distance in our process operator leads to higher-order relation $\mathbf{A}^l$ that is analogous to the $l$–order paths between pairs of points. Therefore, the interpretation of the evolution process is more intuitive as the encoding of paths at multiple lengths.

*The Regularization Parameter.* The second parameter that is important to fix is the regularization parameter. As mentioned in the previous section, $r$ must be smaller than $\frac{1}{\rho(\mathbf{A})}$ to ensure the convergence of the regularized score operator defined in Equation (13). We keep this choice as a free value $r = \frac{c}{\rho(\mathbf{A})}$ parameterized by $c \in (0, 1)$. It is worth noting that $c$ determines the speed of convergence: values close to 0 means fast convergence and vice versa[1]. In practice, when the convergence is fast (slow), the influence of the highest order relations is reduced (preserved). For instance, if $c$ is close to 1, the longest paths are just as relevant for the construction of the descriptor as the shorter ones.

*The Starting State.* A further important parameter to settle is the starting state $f_0$. The choice of this state also plays an important role in controlling the kind of information that is encoded. The options can be different and related to the specific application at hand. In our work, we are interested in evaluating how the evolution process itself is able to encode the geometric information. Therefore, we would like to keep the contribution of the starting state neutral. To this purpose, we choose a constant distribution on the surface as the initial state: $f_0 = \mathbf{e}$, that is, the constant function, encoded with a vector with all the entries equal to 1.

### 5.1 Multiscale Approach

Finally, in order to construct a multiscale descriptor, we can consider a *family* of weighted operators $\mathbf{A}_\delta$, parameterized by a scalar

[1] In Appendix II, we provide a discussion of the behavior with $r \to 0$.

**ALGORITHM 1:** Computation of DEP Descriptors

**Input**: $\mathcal{M}$ the mesh, $\left\{\delta_q\right\}_{q=1}^{Q}$ s.t $\delta_q \in [0, 1]$ .
**Output**: $DEP$ the matrix of descriptors.
**for** $i = 1 : n$ **do**
  **for** $j = 1 : n$ **do**
    Compute $\Omega$, diagonal matrix, with $\Omega(i, i) = \Omega_i$ area of the barycell centered in $v_i$;
    Compute $\hat{\mathcal{G}}(v_i, v_j)$, the normalized geodesic distance between $v_i$ and $v_j$;
    $\hat{\mathcal{G}}(v_i, v_j) = \frac{\mathcal{G}(v_i, v_j)}{diam(\mathcal{M})}$;
  **end**
**end**
**for** $q = 1 : Q$ **do**
  **if** $\hat{\mathcal{G}}(v_i, v_j) \leq \delta_q$ **then**
    $\hat{\mathcal{G}}_{\delta_q}(v_i, v_j) = \hat{\mathcal{G}}(v_i, v_j)$;
  **else**
    $\hat{\mathcal{G}}_{\delta_q}(v_i, v_j) = 1$;
  **end**
  $\mathbf{D}_{\delta_q}(i, j) = 1 - \hat{\mathcal{G}}_{\delta_q}(v_i, v_j), \forall i, j \in \{1, \ldots, n\}$;
  Compute $\mathbf{A}_{\delta_q} = \mathbf{D}_{\delta_q}\Omega$;
  Compute $r_{\delta_q} = \frac{c}{\rho(\mathbf{A}_{\delta_q})}$, with $c \in (0, 1)$;
  Solve the linear system $(\mathbf{I} - r_{\delta_q}\mathbf{A}_{\delta_q})\mathbf{v} = \mathbf{e}$;
  $\check{s}_q = \mathbf{v}$;
**end**
matrix $DEP = [\check{s}_1, \ldots, \check{s}_Q]$, s.t. the $i$th row, encodes the Discrete-Time Evolution Process descriptor for the vertex $i$.

$\delta \in [0, 1]$. We define the new matrix of vertex relations $\mathbf{D}_\delta$ as

$$\mathbf{D}_\delta(i, j) = 1 - \hat{\mathcal{G}}_\delta(v_i, v_j),$$

where $\hat{\mathcal{G}}_\delta(v_i, v_j) = \hat{\mathcal{G}}(v_i, v_j)$ if $\hat{\mathcal{G}}(v_i, v_j) \leq \delta$ and 1 otherwise. Thus, the new process operator becomes $\mathbf{A}_\delta = \mathbf{D}_\delta\Omega$ by only considering geodesic balls of radius $\delta$, which implies that in a single discrete-time step, the relation is limited to points at distance $\delta$. Intuitively, for small values of $\delta$, our matrix $\mathbf{A}_\delta$ makes a vertex dependent on a small neighborhood by capturing more local properties of the shape, while for larger values of $\delta$, $\mathbf{A}_\delta$ exploits more global structures of the shape. This way, the parameter $\delta$ can be interpreted as *the speed* at which information is propagated across the shape in our discrete-time evolution process. Note that this is somewhat in contrast with diffusion-based methods, in which all changes are completely global, since, for example, the classical heat equation implies that heat propagates at infinite speed, which is not compatible with the special theory of relativity (Eckert and Drake 1987).

## 5.2 Discrete time Evolution Process Descriptor

Once the main components of the evolution process are fixed and the multiscale paradigm is defined, we are ready to propose a new shape descriptor. We fix a set of $Q$ scale values $\left\{\delta_1, \ldots, \delta_Q\right\}$ such that $\delta \in [0, 1]$. For each choice of $\delta$, we construct an operator $\mathbf{A}_\delta$ and compute the score at scale $\delta$. For this, we solve the linear system:

$$(\mathbf{I} - r_\delta\mathbf{A}_\delta)\mathbf{v} = \mathbf{e}$$

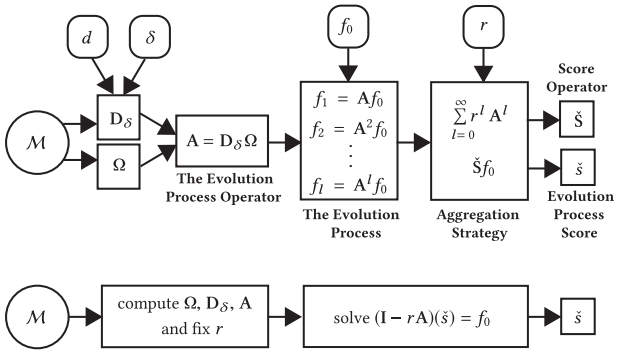


Fig. 2. A visualization of the entire discrete evolution process pipeline. The first row shows the pipeline that produces the discrete time evolution process. Starting with a triangle mesh $\mathcal{M}$, we introduce a relation function $d$ and a scale parameter $\delta$. The relation matrix $\mathbf{D}_\delta$ and the area elements matrix $\Omega$ are computed to obtain the evolution process operator $\mathbf{A}$. Then, fixing an initial state $f_0$, we run our discrete time evolution process by highlighting the involved discrete states. Finally, once a regularization parameter $r$ is selected, we employ our aggregation strategy to obtain the score operator $\check{\mathbf{S}}$ and the evolution process score $\check{s}$. The second row shows how the score $\check{s}$ can be computed in practice by solving a linear system.

and let the score vector at scale $\delta$ be $\check{s}_\delta = \mathbf{v}$. Here, as above, the score of vertex $i$ equals $\check{s}_\delta(i)$.

This way, we create our Discrete-Time Evolution Process (DEP) descriptor, by assembling a vector of $Q$ values to each vertex $i$:

$$DEP(i) = [\check{s}_1(i), \ldots, \check{s}_Q(i)].$$

In other words, for every vertex $i$, we obtain a vector that represents in each of its dimensions the sum of the discrete time evolution process at vertex $i$, where the process performs steps of fixed maximum length. In Figure 2, we represent the entire pipeline to compute our evolution process score $\check{s}$ starting from the pairwise relations encoded by $d$ and a set of parameters $\delta$, $f_0$, and $r$. The first row explains the theoretical interpretation of our framework, while the second row illustrates how it is computed efficiently in practice. For completeness, we summarize the construction of the Discrete-Time Evolution Process descriptor in Algorithm 1. Figure 3 shows the descriptors obtained using 16 different values of $\delta$ on a human shape (from the KIDS dataset (Rodola et al. 2014)).

## 5.3 Contribution of Higher Order Relations

In this section, we analyze the importance of the higher-order relations. Although the initial information (i.e., the geodesic distances) is already informative, it is not sufficient to compete with methods that are able to encode multiple paths between a pair of points, such as spectral-based methods. Therefore, to disambiguate between points having accidentally the same geodesic distance, we allow our evolution process to consider higher-order paths. We compare the performance of using just a 1-step score with the infinite path descriptor on the toy example in Figure 4. This shape is composed of the lateral surfaces of two tetrahedra having the same equilateral triangular bases. We can distinguish three types of points on this shape: the blue, the red, and the cyan. The three red points on the bases are considered the same because, thanks

Fig. 3. Scores for multiscale on a KIDS's shape. The scores at every vertex are plotted as a map on the shape; the values of $\delta$ are $[\frac{1}{100}, \frac{1}{50}, \frac{1}{40}, \frac{3}{100}, \frac{1}{25}, \frac{1}{20}, \frac{7}{100}, \frac{3}{40}, \frac{2}{25}, \frac{9}{100}, \frac{1}{10}, \frac{3}{20}, \frac{7}{20}, \frac{11}{20}, \frac{8}{10}, 1]$.

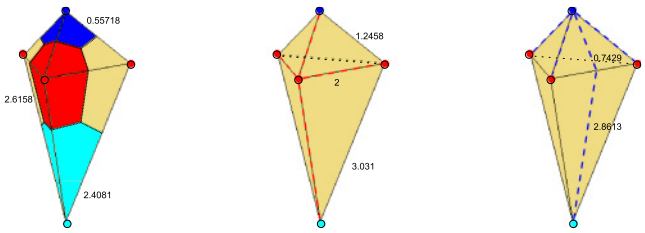


Fig. 4. Toy example. Three type of points are highlighted: blue, red, and cyan. On the left, the barycells associated to each vertex. Dashed lines are the geodesic paths: the four geodesic paths from one of the red points in the middle and the four geodesic paths from the blue point on the right.

to the intrinsic symmetry of this shape, they only differ by a rigid rotation. The barycells associated to each vertex are shown with their value in the figure on the left. The geodesic distances are plotted with dashed lines in the other two figures. Table 1 shows the scores computed for the 1-step and for **Še** on all points in the surface showed in Figure 4. We can see that the 1-step score confuses the blue point with the three red points. Conversely, when we let the evolution process exploit paths with multiple steps, our **Še** scores are able to correctly distinguishing between different types of point.

Table 1. Values of Descriptors at Points in the Toy Example

| POINT COLOR | 1-step | DEP |
|---|---|---|
| Blue | 0.5197 | 1.0643 |
| Red 1 | 0.5197 | 1.0483 |
| Red 2 | 0.5197 | 1.0483 |
| Red 3 | 0.5197 | 1.0483 |
| Cyan | 0.3373 | 0.5912 |

We can also obtain a theoretical bound on the robustness of our approach against shape perturbations. If we have a matrix of the linear system $M = (\mathbf{I} - r_\delta \mathbf{A}_\delta)$ and a perturbed matrix $M'$, we can give the following upper bound (Golub and Van Loan 2012):

$$||\check{s}'_\delta - \check{s}_\delta|| \leq O(\kappa(M)||M' - M||)||\check{s}_\delta||, \tag{16}$$

where $\check{s}_\delta$ and $\check{s}'_\delta$ are the solutions of the linear problem and the perturbed problem, respectively, and $\kappa(M)$ is the condition number of the matrix. This bound depends on the norm of the perturbation $||M' - M||$. In practice, we have observed that the condition number is well behaved. We performed an experimental evaluation on different perturbed shapes from the FAUST dataset. On average, we obtained $\kappa(M) = 1.12$, which supports our claim that the solutions resulting from our method are stable. Therefore, for small perturbations, as in the case of near-isometric shapes, our method can ensure a reliable solution.

In order to emphasize the robustness of our method, we analyze the matching experiment reported in Figure 5. The same Duck shape is shown on two meshes with different density.

It is clear that the geodesic distance information contained in the operator matrix **A** depends on the mesh structure that is used to discretize the mesh. But, thanks to the integration at all of the surface, the DEP descriptors are more informative even in the presence of changes of the mesh structure. In Figure 5, we illustrate this behavior. We show the 1-step descriptors and our descriptors for some selected points on the two Duck shapes. To emphasize the contribution of the evolution process, we evaluate two different relation functions. More precisely, other than using the geodesic distance, we also introduce the diffusion distance (Coifman and Lafon 2006) to build the process operator. Then, a descriptor is defined as described in Section 5.2 (in practice, the geodesic distance is simply substituted by the diffusion distance in the construction of the matrix **D**). Descriptors coming from diffusion distance are shown on the left, while those defined by the geodesic distance are shown on the right. In the 1-step case, we can see how all the descriptors are very similar if they come from the same mesh independently from the kind of point that they represent and that the light-blue and red one are very close despite representing different points. In contrast, when descriptors are defined with our new diffusion process, the matching is correct. In particular, with our approach, the performance is independent from the change of the mesh. In regions with similar density (ends of the wings) and in those with different density (belly), the proposed approach provided reliable results. Note that this behavior is observed for both chosen relation functions. This suggests that the contribution to

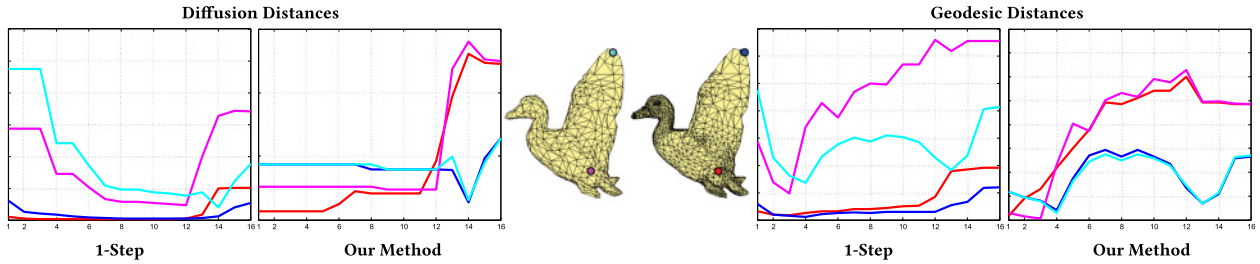


Fig. 5. Comparison between the 1-step approach and our method using diffusion (left) and geodesic distances (right) for two points on the Duck shape with two different meshes with 752 and 2497 vertices, respectively.

obtaining the correct matching is by the proposed evolution process when higher-order relations are considered.

## 6 RELATION TO HEAT KERNEL SIGNATURE

As mentioned in Section 1, our approach is related to existing techniques based on the diffusion process, in particular, with the HKS. Moreover, in addition to the relation between our discrete-time process and continuous diffusion described in Section 4, another informative connection can be obtained by considering the relation between the diffusion distance matrix (Coifman and Lafon 2006) and the HKS (Sun et al. 2009). As pointed out by Sun et al. (2009), for a fixed time parameter $t$:

$$\mathrm{HKS}_t(x) = \frac{1}{\Omega_{\mathcal{M}}} \int_{\mathcal{M}} d_t^2(x,y)dy - \frac{H_{\mathcal{M}}(t)}{\Omega_{\mathcal{M}}} + \frac{2}{\Omega_{\mathcal{M}}}.$$

Note that the last two terms do not depend on the vertex. Therefore, they will not influence distances between descriptors when comparing vertices on the same shape. In other words, the HKS is closely related to the eccentricity of the squared *diffusion* distance. This means that, for an appropriate discretization, we can obtain that

$$\mathrm{HKS}_t = \mathbf{D}_f \Omega \mathbf{e}$$

using a matrix $\mathbf{D}_f$ properly constructed in the following way:

$$\mathbf{D}_f(x,y) = \frac{1}{\Omega_{\mathcal{M}}} d_t^2(x,y) + \frac{1}{\Omega_y N}\left(\frac{2}{\Omega_{\mathcal{M}}} - \frac{H_{\mathcal{M}}(t)}{\Omega_{\mathcal{M}}}\right),$$

where $N$ is the number of vertices in the surface discretization and $\Omega_{\mathcal{M}} = \sum_{\forall j \in \mathcal{V}} \Omega_y$ is the total area of the surface. This shows how the HKS can be constructed marginalizing the rows of a matrix that represents specific relations between pairs of points, as proposed in our framework. The matrix $\mathbf{D}_f$ contains the pairwise relations between points on the surface encoded by the squared *diffusion* distance $d_t^2$. According to our framework, this can be considered as a special relation function of a 1-step evolution process defined by $\mathbf{Ae}$, with $\mathbf{A} = \mathbf{D}_f \Omega$, which demonstrates a direct link between our approach and the HKS. Moreover, even though the diffusion distance is itself obtained by considering an infinite set of paths, by incorporating it into our framework and considering a discrete evolution process, we can obtain more reliable and stable connections between surface points. This further highlights the utility of our framework.

Figure 6 shows the improvement of our framework to the HKS descriptors. We select 16 time scales and compute the respective 16 matrices $\mathbf{D}_f$. A 16-dimensional HKS descriptor is computed

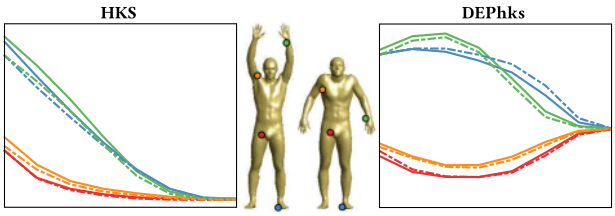


Fig. 6. Comparison between the 1-step approach (that is equivalent to HKS) and our discrete-time evolution version up to infinite (DEPhks). In the middle, two human shapes from the FAUST dataset with four vertices are highlighted in colored balls. On the sides, the 16-dimensional descriptors: HKS descriptors (left) and DEPhks descriptors (right). Continuous lines refer to the shape on the left, while dashed lines represent the shape on the right.

for every selected vertex as the 1-step on the matrix $\mathbf{D}_f \Omega$ (Figure 6, left). Then, on the same matrices, we compute the descriptors DEPhks related to our discrete time evolution process framework (controlled by the relation function $\mathbf{D}_f$ rather than the geodesic distances; Figure 6, right). As can be seen, the 1-step descriptors on different points are not very discriminative. In particular, the blue point is confused with the green point. Conversely, with our method, the behavior of DEPhks descriptors is more coherent and the matching is correct.

## 7 RESULTS

We have performed a wide range of experiments on several datasets to demonstrate the utility of our DEP descriptor in two application scenarios: (i) point-to-point matching using nearest neighbor search in descriptor space and (ii) incorporating our approach into the functional maps framework. In functional maps, we also evaluate how our score operator could improve the standard performances. Furthermore, from these results, we analyze qualitatively the performance of our descriptor in comparison with the widely used HKS and WKS. Finally, we explore the behavior of our descriptor in different settings and varying the choice of parameters.

### 7.1 Point-to-Point Matching

In order to evaluate the ability of our DEP descriptor to associate corresponding points of different shapes, we consider the following data sources:

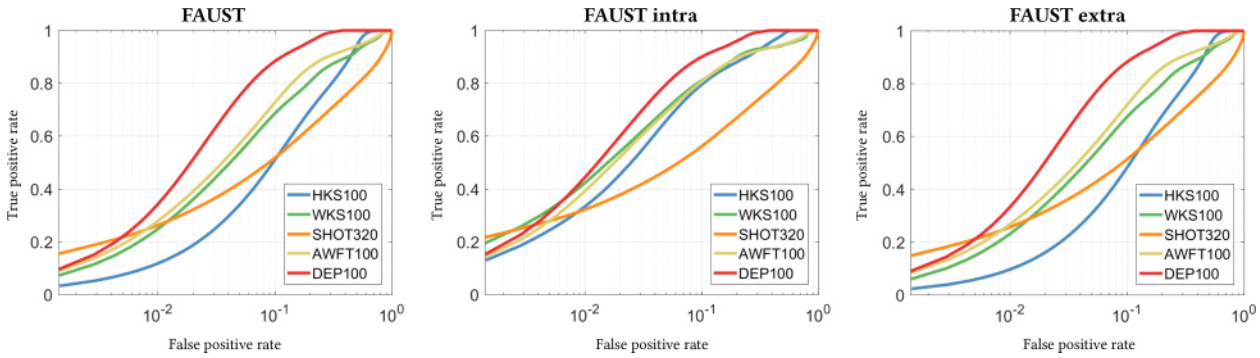


Fig. 7. Performance evaluation on the pairs FAUST dataset, using all 100 shapes (left), allowing only matching between shapes of the same subject (middle) and only between shapes from different subjects (right). Comparison with 4 different descriptors. Next to the descriptor name, we show its dimensionality.

FAUST (Bogo et al. 2014) is a recent dataset of scanned human shapes in different poses. The dataset is challenging due to the significant variability between different human subjects. Ground truth pointwise correspondence between the shapes is available for all points. All of these meshes have the same connectivity. We use the whole FAUST dataset consisting of 100 shapes, 10 poses of 10 different subjects, respectively, along with additional shapes that have been edited by adding different types of noise: Gaussian noise, heavy subsampling, voxelization noise, topological noise (glued fingers and missing parts). In addition, we perform a test using the real scans of humans, which are high-resolution, nonwatertight mesh. These meshes are more noisy, without registration and without ground truth correspondences available.

CAESAR is a human-shapes dataset recovered from MPII Human Shape (Pishchulin et al. 2015), a family of expressive 3D human body shape models learned from the CAESAR dataset (Robinette et al. 1999), the largest commercially available dataset that contains 3D scans of over 4500 subjects in a standard pose. We use a randomly selected subset of 21 shapes from the *CAESAR-fitted meshes* collection, in which a template is fitted. For every shape, we have around 6k vertices with 1:1 ground-truth correspondence.

KIDS (Rodola et al. 2014) consists of a collection of 3D shapes undergoing nearly isometric and intraclass deformations. In this dataset, we find two different shape classes (*kid* and *fat kid*) in 16 different poses. The same poses are applied to both classes. The authors provide all shapes with consistent triangulations using around 60k vertices consistently ordered, giving the ground-truth correspondence as the identity map. In our test, we uniformly downsample the given meshes with approximately 6k vertices maintaining the 1:1 ground-truth correspondence; we cannot guarantee maintaining the same connectivity for all meshes.

The MISC dataset is composed of pairs of highly nonisometric shapes, such as a horse and an elephant. Therefore, this dataset is particularly challenging since the usual hypothesis of isometric relations between shapes is totally violated. Manually generated ground-truth, pointwise correspondences are available for a dense subset of points in this dataset. See Figures 16 and 17 for examples of shapes from this dataset (Boyer et al. 2011). The class includes one full human shape (i.e., the *null* shape) and 5 versions of its simulated transformations of pose deformation with strong partiality.

These meshes were resampled to around 6K vertices. For every dataset, we take all possible pairs of shapes. Then, we randomly select 1K vertices on one shape and compute the closest vertex in the descriptor space among all the vertices from the second shape. For the MISC dataset, on the first shape, we use all the vertices for which the correct correspondence is given. We evaluated the performance of our descriptor using the *receiver operator characteristic* (ROC). The ROC represents the performance in the classification of positive and negative pairs depending on a discrimination threshold, related to their distance in the descriptor space, measuring the *true-positive rates* and *true -negative rates*. This kind of evaluation is particularly sensitive to instability in the descriptor space, resulting in large relative distances between correct matches.

We compared our method with the following descriptors:

- HKS (Sun et al. 2009) with 100 dimensions
- WKS (Aubry et al. 2011) with 100 dimensions
- SHOT (Tombari et al. 2010) with 320 dimensions
- AWFT (Melzi et al. 2016) with 100 dimensions

We use the code and settings available online. In order to be coherent with other methods, our proposed descriptor is estimated at 100 scales (DEP100). Note that HKS and WKS are the closest methods to ours, as described in Section 6. SHOT (Tombari et al. 2010) is a local descriptor that encodes very different information with respect to our method. AWFT (Melzi et al. 2016) also adopts a very different approach to encode local information and represents the state-of-the-art for point-to-point matching without the use of a learning procedure. In Figure 7, we evaluate the performance of our DEP descriptor in comparison with the other methods. We perform three different tests by varying the selected shapes: (i) FAUST with all pairs from the entire dataset of 100 shapes (Figure 7, left); (ii) FAUST-intra, with only the pairs belonging to the same subject (Figure 7, middle); and (iii)FAUST-extra, with only pairs from two different subjects (Figure 7, right). Our method outperforms the competitors in all of these settings. The greater improvement is achieved in the FAUST extra, in which strong nonisometric deformations are observed. Therefore, we claim that our descriptor is particularly effective in the case of change of subjects for which the isometric relation is clearly violated. Figure 8 shows the pointwise matching evaluation on other benchmarks: (i) CAESAR

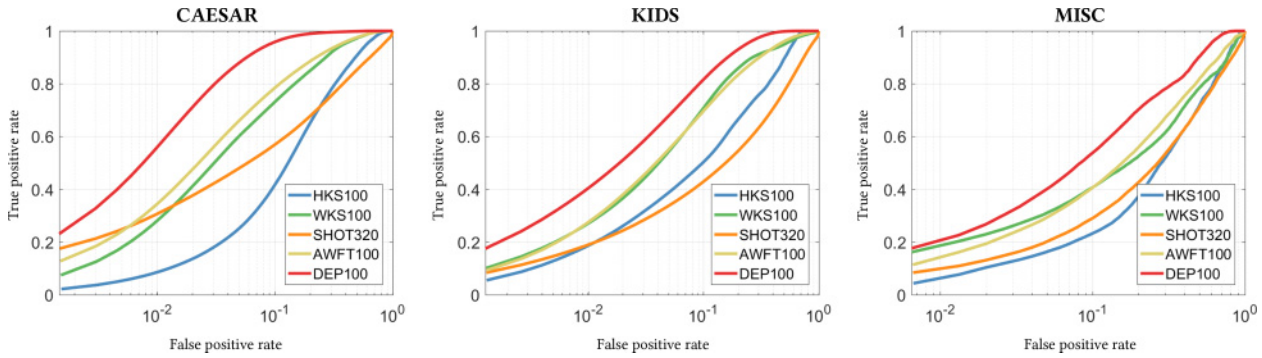


Fig. 8. Performance evaluation on the CAESAR dataset (left), KIDS dataset (middle), and on the pairs *Elephant and Elephant subsampled*, *Elephant and Horse subsampled*, *Alien and Robot*, *Homer and Alien*, *Boy and Baby* and *Man* (7 shapes), *Gorilla* (5 shapes), *and Woman* (12 shapes) from the MISC dataset (right). Comparison with 4 different descriptors. Next to the descriptor name, we show its dimensionality.

Table 2. AUC of ROC Curves

| dataset | HKS100 | WKS100 | SHOT320 | AWFT100 | DEP100 |
|---|---|---|---|---|---|
| **CAESAR** | 0.8069 | 0.9083 | 0.8034 | 0.9228 | **0.9762** |
| **KIDS** | 0.8210 | 0.8955 | 0.7279 | 0.8951 | **0.9449** |
| **MISC** | 0.6568 | 0.7227 | 0.6584 | 0.7513 | **0.8317** |
| **FAUST** | 0.8387 | 0.8736 | 0.7569 | 0.8946 | **0.9577** |
| **FAUST intra** | 0.9258 | 0.9176 | 0.7830 | 0.9163 | **0.9635** |
| **FAUST extra** | 0.8300 | 0.8692 | 0.7543 | 0.8924 | **0.9572** |
| **MEAN** | 0.8132 | 0.8645 | 0.7473 | 0.8788 | **0.9385** |

(Figure 8, left), (ii) KIDS (Figure 8, middle), and (iii) MISC (Figure 8, right). In the human datasets (i.e., CESAR and KIDS), the shapes are geometrically near isometric, although the natural articulated motion of humans and the change of the subject can lead to possibly significant geodesic distortions. These kinds of distortion are not as strong as in FAUST. On these datasets, DEP also clearly outperforms all competitors. In particular, we exceed the performance of AWFT, the most recent competitor in our analysis. The curves on the right are obtained on the MISC dataset. On this dataset, the point-to-point matching is very difficult since the shapes are not related by an isometric transformation. Therefore, methods that are very sensitive to nonisometric changes are likely to fail. Moreover, in general, the meshes have different resolution and different connectivity. We evaluate the average performance between all pairs of shapes on the vertices for which the correspondences are known. In this setting, our method also clearly outperforms all the other descriptors. As in the FAUST-extra in the presence of more nonisometric deformations that are characteristic to the MISC dataset, the DEP descriptor increases its positive gap with all competitors. In order to give a numerical comparison of the performance, we report the area under the curve (AUC) for every dataset in Table 2. Our DEP descriptor showed superior performance over all other methods across all datasets, with improvements of at least 5% and, on average, around 6%. In particular, our method is able to improve standard methods based on a diffusion geometry like HKS and WKS by confirming the benefit of our alternative evolution process. Moreover, our DEP descriptor is preferable to AWFT, which, to the best of our knowledge, represents the state-of-the-art of descriptors without learning.

## 7.2 Functional Maps

The experiments above suggest that our descriptor can identify related points across different shapes. Below, we show how our general approach, which includes the discrete evolution process and the derived DEP descriptor, can be used to obtain entire *maps* across shapes. For this, we use the so-called *functional map* framework (Ovsjanikov et al. 2012; Pokrass et al. 2013; Rodolà et al. 2016; Litany et al. 2016; Ovsjanikov et al. 2016). The key aspect of this approach is to phrase the estimation of correspondences in *the space of functions* rather than points, which can also be potentially enhanced using spectral analysis. To estimate this map, the authors suggested exploiting (i) functional and (ii) commutativity constraints. A functional constraint (i) is defined from a set of corresponding descriptors. The commutativity constraint (ii) is introduced by some commutative operator. In the original setting, (i) is introduced by standard descriptors such as HKS and WKS and (ii) is imposed by the Laplace-Beltrami Operator (LBO). A detailed description can be found in Chapter 2 of Ovsjanikov et al. (2016).

The discrete-time evolution process provides a new set of continuous functions $\check{s}$, the score of our process for different selections of parameters and a new shape operator $\check{\mathbf{S}}$. In the following experiments, we propose injecting our evolution process scheme into the functional map framework by introducing our DEP descriptor for the functional constraint and by exploiting our regularized score operator $\check{\mathbf{S}}$ as a commutative operator.

*Dataset and Evaluation.* We choose to perform our tests on FAUST because it is the largest dataset and it contains more isometric and nonisometric variations. Moreover, as we have done in the point-to-point matching evaluation, we perform three different tests on FAUST considering the entire dataset, only the intrasubject pairs, and, finally, restricting to pairs belonging to different subjects.

All of the following evaluations are obtained by randomly selecting 10 pairs of shapes from the 100 available in the dataset and we plot the average performances. The test on the entire dataset is

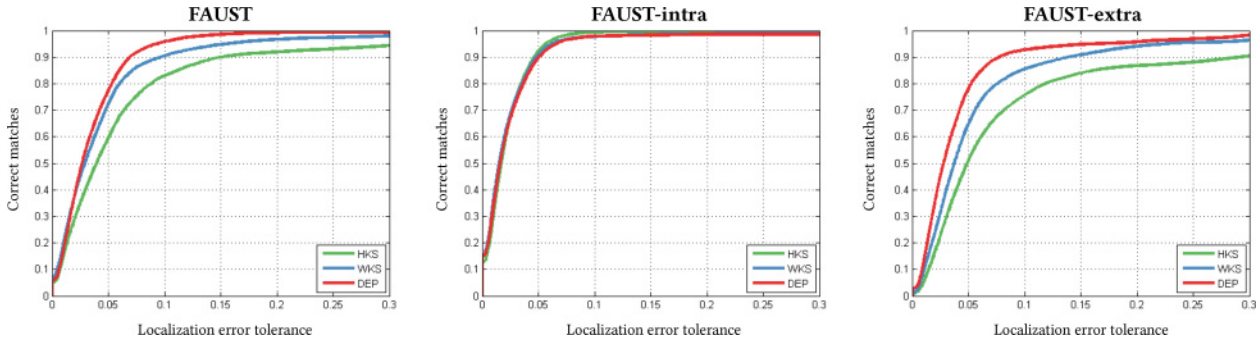


Fig. 9. Test varying the functional constraints functional maps framework. Visualization of the error rates given an unnormalized radius $r$, the percentage $y$ of the points that are mapped by the correspondence at a distance at most $r$ from their ground-truth image.

obtained with 10% of intrasubject shapes and 90% of extrasubject shapes, as happens in the whole dataset. As in the original version, we use a postprocessing step on the obtained functional map based on a high-dimensional ICP, which also results in a point-to-point map between shapes (see Ovsjanikov et al. (2012) for details). We evaluate the performance using the *correspondence quality characteristic* (Kim et al. 2011), which is the standard evaluation used for functional maps. These curves show the percentage of nearest-neighbor matches that are at most $r$-geodesically distant from the ground-truth correspondence. Here, we accept the symmetric images of ground-truth correspondences and use the minimum between the distance from the matched point to the ground truth and its symmetric image, as done in prior work (Ovsjanikov et al. 2012).

*Functional Constraints.* Starting with Ovsjanikov et al. (2012), many works have tried to explore the quality and best method for the selection of the descriptor constraints to be used in this framework (see Ovsjanikov et al. (2016) Chapter 2 for an overview). Corman et al. (2014), for example, compute optimal descriptor weights by learning the contribution of each descriptor for the estimation of overall matching. In this experiment, we adopt a new approach proposed recently in Nogneng and Ovsjanikov (2017) that is based on the enhancement of the descriptor preservation constraints. Nogneng and Ovsjanikov (2017) noticed that the original functional constraints (Ovsjanikov et al. 2012) do not capture all of the information contained on a given descriptor. Rather than only preserving the descriptor *values*, as done before, they have demonstrated that preserving *function products* with descriptors leads to a significant improvement in map quality, even in the presence of a few descriptor functions. Moreover, the authors show how the resulting new constraints can be efficiently encoded via commutativity of the unknown map with linear operators defined by the descriptors, which retains the overall efficiency of the framework.

Note that, in addition to the descriptors, Ovsjanikov et al. (2012) and Nogneng and Ovsjanikov (2017) proposed adding some consistent segmentation of the shapes as functions to be preserved. Here, we prefer to use only descriptors in order to evaluate strictly their contribution to the framework.

We consider HKS100, WKS100, and our DEP100. From each of these, we select 6 equally spaced scales as input descriptors. Explicitly, we use the first, the last, and all scales that are multiples

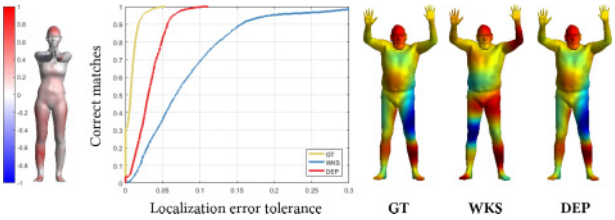


Fig. 10. Evaluation of a functional map on a pair of shapes from two different subjects in the FAUST dataset. In the middle, the evaluation curves for ground truth (GT) and the comparison between WKS and DEP are shown. The colors on the left show points that are matched better (red) or worse (blue) with DEP compared to WKS. On the right, we show on the second shape the transportations of a smooth function defined on the first shape using GT, WKS, or DEP.

of 20 in order to represent all features contained in the whole collection. In Figure 9, we show the performance of the functional maps framework using the three different descriptors in three different tests. As can be seen, the best performance on FAUST is obtained with DEP descriptors. In particular, on shapes for undergoing strong nonisometric deformations, our DEP descriptor (red line) clearly outperforms the competitors. In the FAUST intratest, in which isometry is preserved, we note that the three descriptors achieve the same results. This experiment confirms the utility of our descriptor for the computation of shape matching within the functional map framework. In Figure 10, we show some details for particular pairs of nonisometric shapes. We compute functional maps from the female shape on the left to the male shape on the right of this figure. In the middle, we plot the evaluation curves. DEP clearly outperforms WKS and is also very close to the ground truth (GT) performance. On the shape on the left, we plot the differences between the error of the functional map correspondences computed with WKS or DEP in the following way. If $F_{\text{WKS}}$ and $F_{\text{DEP}}$ are the functional maps computed using WKS and DEP, respectively, and $GT$ is the ground truth map, then we can define the function $g$ at every vertex $v_i$ as

$$g(v_i) = \|(GT - F_{\text{WKS}})\delta_{v_i}\| - \|(GT - F_{\text{DEP}})\delta_{v_i}\|,$$

where $\delta_{v_i}$ is the function equal to 0 everywhere but equal to 1 in $v_i$. The norm $\|(GT - F)\delta_{v_i}\|$ is a measure of the local error of transportation for the functional map $F$ in $v_i$. Plotting this function

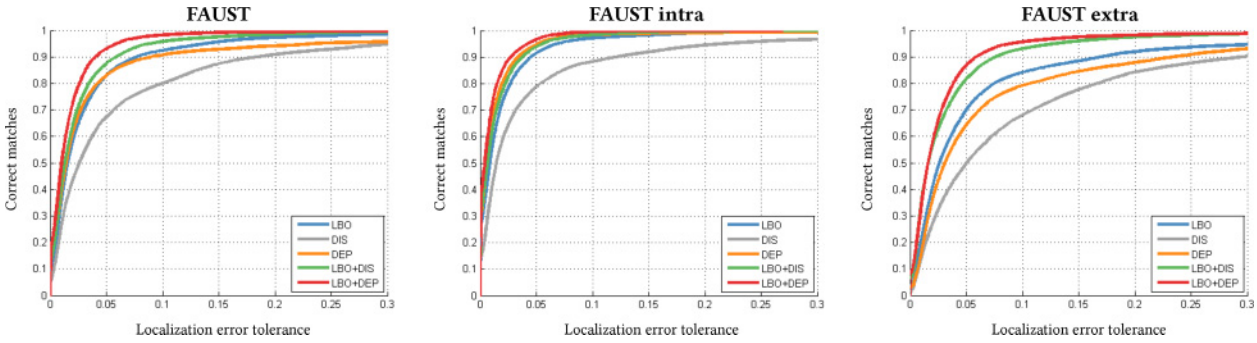


Fig. 11. Test varying the commutativity constraints functional maps framework. Visualization of the error rates given an unnormalized radius $r$, the percentage $y$ of the points that are mapped by the correspondence at a distance at most $r$ from their ground-truth image.

$g$ on the shape, we have that the red areas are the ones for which $F_{\text{DEP}}$ is more precise while in blue areas, we can see where $F_{\text{WKS}}$ is better. More intense color corresponds to greatest difference in performances. On the right, we can see on the second shape the transportations of a function defined on the first shape using GT, WKS, or DEP. Here, GT corresponds to the original function defined on the first shape. In comparison, the transported function using DEP is similar to GT, unlike the one obtained using WKS.

*Commutativity Constraint.* In Figure 11, we evaluate different commutativity constraints. In order to give more emphasis to the contribution of the operators, here we use the standard functional constraint and the standard set of preserved functions (HKS and WKS), Ovsjanikov et al. (2012), without any precomputed segment correspondences. We use (i) our regularized score operator Š (DEP) defined in Equation (13), (ii) the LBO as in the original version of functional map, and (iii) the matrix of geodesic distances (DIS) as a basic approach to introduce the geodesic information in this framework, as proposed in Aflalo et al. (2016).

We also compare the performance of each operator with the pairs (LBO+DIS) and (LBO+DEP). This way, we can see which operator based on geodesic distances adds more information to the original LBO. As can be seen while in the FAUST-intratest (Figure 11, middle), the improvement given by adding DEP to LBO is not significant; the gap between LBO and LBO+DEP clearly grows in the FAUST-extra test (Figure 11, right). This result confirms that in the discrete-time evolution process framework also, the operator Š is more stable with respect to nonisometric deformation.

As can be seen in Figure 11, the performance using DEP alone is very close to the standard performance using only LBO. When DEP and LBO are integrated, the performance is definitely improved. Note also that the quality of the maps obtained with (LBO+DEP) significantly exceeds (LBO+DIS), confirming that our evolution process provides an effective approach to further improving the contribution of geodesic distances in this framework. In Figure 12, we show some details for a particular pair of nonisometric shapes. This figure contains the same analysis of Figure 10, but we compare LBO and LBO+DEP in the Commutativity constraint. In the middle, we plot the evaluation curves. DEP clearly improves the results of LBO and LBO+DEP is closer to the ground truth (GT) performance. On the right we can see on the second shape, the transportations of a function defined on the the first shape, using

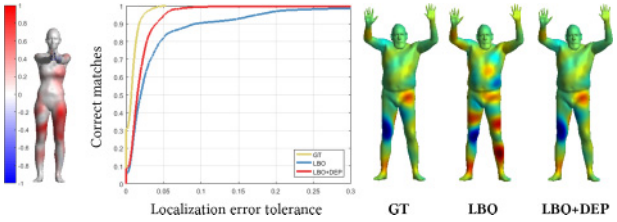


Fig. 12. Evaluation of a functional map on a pair of shapes from two different subjects in the FAUST dataset. In the middle, the evaluation curves for ground truth (GT) and the comparison between LBO and LBO+DEP are shown. The colors on the left show points that are matched better (red) or worse (blue) with LBO+DEP. On the right, we show on the second shape the transportations of a smooth function defined on the first shape, using GT, LBO, or LBO+DEP.

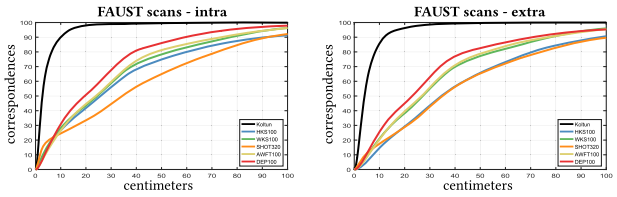


Fig. 13. Cumulative error distribution on the FAUST real scan training dataset. The results are an average on 50 pairs provided by Chen and Koltun (2015), 50 for intrasubject (left) and 50 for extrasubject (right). The matching is computed via distances in the descriptor space. In black, we show the performance of Chen and Koltun (2015) for reference.

GT, LBO or LBO+DEP respectively. These results show that adding DEP to LBO results in maps that are significantly better than those obtained using LBO alone and that are very close to GT.

## 7.3 Evaluation on Real Scans Dataset

We evaluate the proposed method on the full real scans dataset (Bogo et al. 2014) following the settings reported on Chen and Koltun (2015). For this experiment, to make a direct comparison with Chen and Koltun (2015), we evaluate our results using the correspondence quality characteristic (Kim et al. 2011) with error measured in centimeters. Figure 13 shows point-to-point matching results on 50 intrasubject and 50 extrasubject pairs. We compare

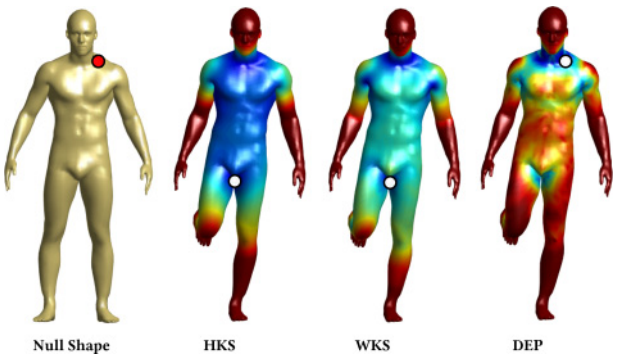


Fig. 14. Visualization of dissimilarity maps. Point selected on the Man shape in the first pose (left) and the dissimilarity maps on the Man shape in a different pose for HKS, WKS, and our method from left to right, respectively.

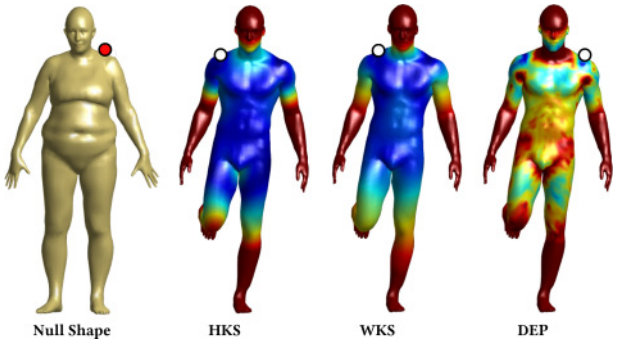


Fig. 15. Visualization of dissimilarity maps. The selected point on the Woman shape (left), and the dissimilarity maps on the Man shape for HKS, WKS and our method respectively from left to right.

our DEP100 descriptor with Chen and Koltun (2015) and the 4 descriptors used in the point-to-point matching section: (i) HKS100, (ii) WKS100, (iii) SHOT320 and (iv) AWFT100. Note that as expected the method from Chen and Koltun (2015) shows the best performance. These excellent results were indeed obtained by using an extrinsic alignment of the shapes, which is not used by the other methods. On the other hand, among the pure descriptor-based methods, our DEP descriptor outperforms all other alternatives and is comparable with most of the methods evaluated in Chen and Koltun (2015).

## 7.4 Qualitative Evaluation

In order to obtain a visual evaluation of the proposed approach, we show the dissimilarity maps of some pairs of shapes for some fixed points. For instance, in Figure 14, we fix a point on the *Male* shape in a first pose (the red ball), and show its dissimilarity (i.e., the Euclidean distance on the descriptor space) with all the points on the second pose of the *Male* shape. The minimum of the dissimilarity for every descriptor is highlighted with a white ball. The distances grow under varying from cold colors (similarity) to warm colors (no similarity). It is clear that both HKS and WKS methods are not able to localize the selected point, and the white ball is far from the target point. Conversely, our method gives more local result and identifies the correct corresponding point.

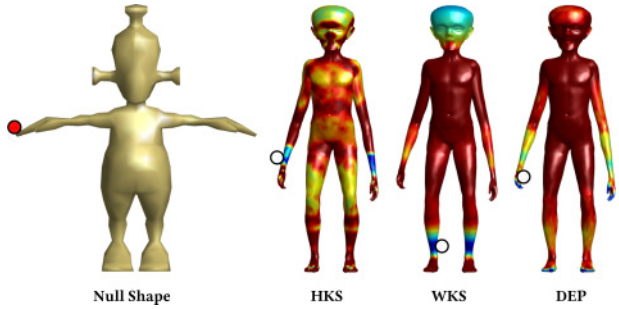


Fig. 16. Visualization of dissimilarity maps. The selected point on the Robot shape (left), and the dissimilarity maps on the Alien shape for HKS, WKS and our method respectively from left to right.

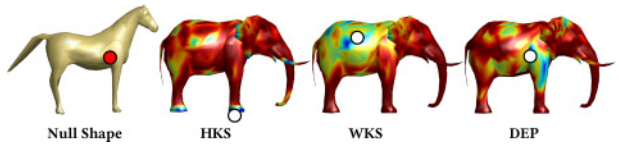


Fig. 17. Visualization of dissimilarity maps. The selected point on the Horse shape (left), and the dissimilarity maps on the Elephant shape for HKS, WKS, and our method from left to right, respectively.

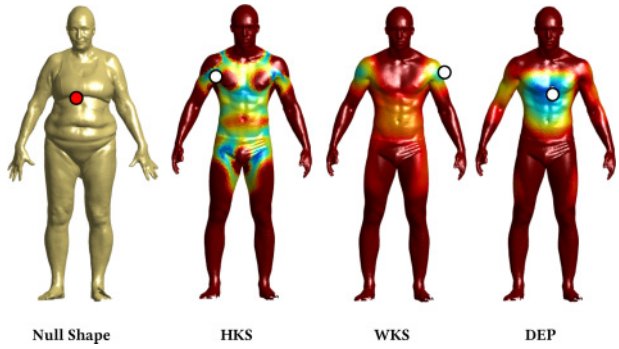


Fig. 18. Visualization of dissimilarity maps. The selected point on the scanned Woman shape (left) and the dissimilarity maps on the scanned Man shape for HKS, WKS, and our method from left to right, respectively.

Gradually losing the isometry property between the shapes, the robustness of our method is highlighted. This is shown in Figure 15 for Female and Male subjects from the FAUST dataset, in Figure 16 for Robot and Alien from MISC, and in Figure 17 for Horse and Elephant from MISC. In particular, in the last pair, the two shapes are clearly nonisometric and their meshes are totally different in the number of vertices and connections.

We perform the same test using a pair of real from scans the FAUST dataset to evaluate the robustness of the proposed method against noise and missing parts. Real scans have more than 160K vertices and the number of vertices is different for different scans. These high-resolution, triangulated, nonwatertight meshes present a lot of challenging features. Also, in this case, our descriptor correctly identifies a local region of points on the shape

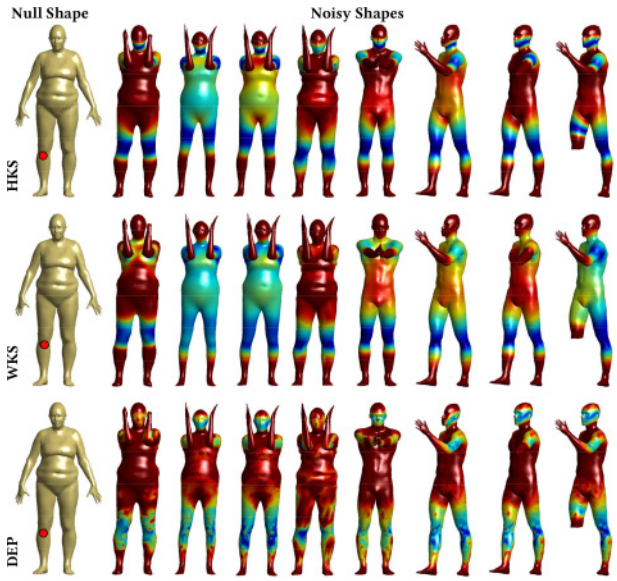


Fig. 19. Visualization of the dissimilarity maps for shapes with different kinds of noise. The vertex on the right knee of the clean shape on the left is selected and the dissimilarity maps on shapes with different kinds of noise are shown in the first row for HKS, in the second row for WKS, and finally for our method. Respectively, from left to right, the original shape, an isometric remeshing, two smoothed versions of the surface, a subsampled mesh with 1000 vertices, topological noise (glued fingers), and the last three are partial views or surfaces with missing parts.

without spurious areas, as in the HKS and WKS results. A similar evaluation analysis is shown in Figure 19 for testing the robustness of proposed descriptors against strong subsampling and topological noise (i.e., glued fingers and missing parts). This experiment is carried out on a small collection of shapes from the FAUST dataset with different poses of the same subject and different subjects with several kinds of noise (see the caption of Figure 19 for an exhaustive list). We observe that our DEP descriptor is robust to smoothing, subsampling and topological noise and shows better localization and accuracy than other methods. We also evaluate the robustness of our DEP for partial and broken parts from the SHREC'11 benchmark. Note that spectral methods, such as HKS and WKS, are known to be sensitive to this kind of failure, since the Laplace operator changes its spectral representation.

Figure 20 shows the dissimilarity maps for some selected points for different partial models. We note that HKS is very sensitive to this kind of shape alteration and highlighted wrong areas generally. WKS performed clearly better but it resulted in several ambiguous parts. In contrast, our method is robust in all selected experiments.

### 7.5 Comparison of Parameter Choices in Different Settings

Finally, to highlight the flexibility of the proposed framework, we evaluate the performance of point-to-point matching across a range of parameter choices: (i) different approaches to encode the geodesic distance, (ii) different relation functions, and (iii) different

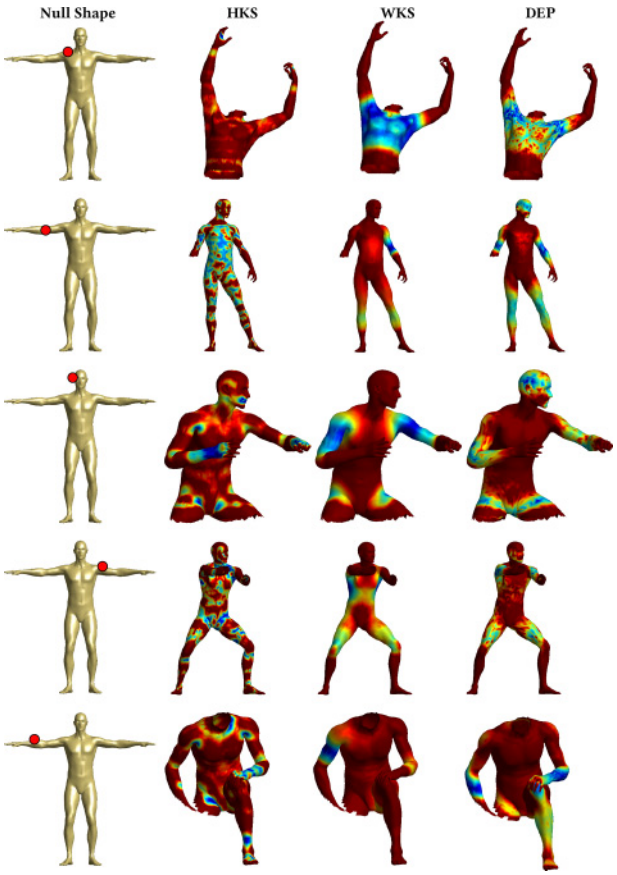


Fig. 20. Visualization of the dissimilarity maps for partial matching. The selected points on the null shape and the dissimilarity maps on different partial models for HKS, WKS, and our method from left to right, respectively.

choice of regularization values. In Figure 21, we show a set of such evaluations.

In order to access the role of geodesic distance and the importance of higher-order relations, we consider five methods to encode the information contained in the matrix $\mathbf{D}$:

- Basic geodesic distance (D16). We use the matrix $\mathbf{D}$ introduced in Section 5.2 as process operator and perform 1 step of our evolution process with $f_0 = \mathbf{e}$. Note that, in this case, the contribution of the areas $\mathbf{\Omega}$ is not considered.
- 1-step (1-step16). As described in Section 5.3, we consider only the first-order relation (i.e., 1 step of the evolution process with $f_0 = \mathbf{e}$).
- Average Geodesic Distance (AGD16). According to the method introduced in Hilaga et al. (2001), we compute the average geodesic distances for every point at each scale as descriptor. Note that in AGD, the areas are considered; in contrast to 1-step16, a more effective normalization method is introduced (see Hilaga et al. (2001) for more details).
- The 1−ring version of our method (DEP1ring). For every vertex, we compute the geodesic distances to each point in the 1-ring (i.e., the length of the edge). Different scales of this

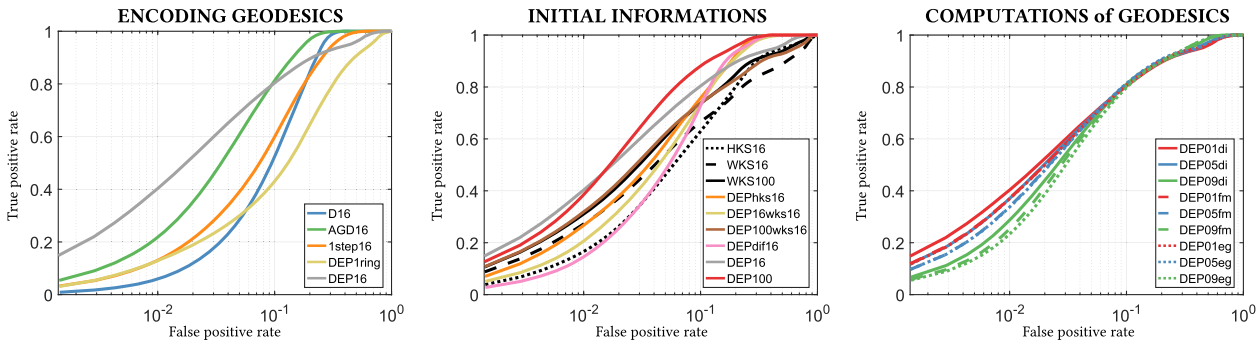


Fig. 21. Performance evaluation on the FAUST dataset (test for 10 different poses of 2 different subjects) in different settings. From left to right: Comparison with different methods to encode geodesic distances (left), comparison starting from different initial information matrices (middle), comparison starting from different computations of geodesic distances and regularizations (right).

descriptor are then obtained using 16 different values of the regularization parameter for $r \in (0,\ 1)$.

- Our descriptor (DEP16). We evaluate our descriptor as described in Section 5.2 with $f_0 = \mathbf{e}$. In this case, the higher-order relations are considered by evaluating the evolution process up to infinite.

We fix 16 different scales $\delta \in [0, 1]$ in order to obtain for every method a 16-dimensional descriptor as defined in Section 5.1, with the exception of the DEP1ring method. Figure 21 on the left shows the performance of the considered approaches. As expected, when the geodesic distance is not combined with the areas (D16), the performance is the worst. In the same way, starting only from the 1-ring information, the obtained descriptor does not perform well. This construction is indeed strongly related to the mesh and is not stable under near-isometric or nonisometric deformations. Moreover, we confirm with a more exhaustive evaluation the benefit of higher-order relations discussed in Section 5.3. In particular, our method (DEP16) clearly outperforms both 1-step16 and AGD16.

In a second setting, we evaluate the evolution process by defining different relation functions:

- Diffusion distance (DEPdiff16). The relation function is defined as in Section 5, in which the normalized diffusion distance is used instead of the geodesic distance.
- HKS from diffusion distance (DEPhks16). The relation function is defined by the matrix $\mathbf{D}_f$ as proposed in Section 6, in which the HKS is derived from the diffusion distance.
- Relation from other descriptor 1 (DEP16wks16). In this case, a different approach is exploited. We compute a generic descriptor for each vertex, namely, the WKS. We define as relation function the Euclidean distance on the descriptor space.
- Relation from other descriptor 2 (DEP100wks16). The same as DEP16wks16 but the WKS descriptor is 100-dimensional rather than 16-dimensional.
- Our descriptor (DEP16). The relation function is defined starting from the geodesic distance, as described in Section 5.

We fix 16 different scales $\delta \in [0, 1]$ in order to obtain for every method a multiscale descriptor with the same dimensionality. In more detail, in DEPdiff16, we fix a time value for each scale and compute the descriptor using the normalization procedure described in Section 5. For DEPhks16, we use the 16 matrices $\mathbf{D}_f$ (using 16 different time scales). Finally, DEP16wks16 and DEP100wks16 are obtained as described in Section 5, in which the normalized Euclidean distance among descriptors is used instead of $\hat{\mathcal{G}}$. For all the considered relation functions, we compute the evolution process and the evolution score by obtaining different versions of the DEP descriptor. To complete the evaluation, we add in the comparison DEP100, HKS16, WKS16, and WKS100. Figure 21 middle, shows the results. As above, this test also highlights the importance of the higher-order relations captured using our discrete evolution process. In particular, it is interesting to observe how DEPhks16 (which employs our evolution process on the same relation function that generates the HKS at 1-step, as explained in Section 6) clearly outperforms HKS16. Moreover, even if WKS16 performs better than DEP16wks16, we observe that DEP100wks16 is comparable with WKS100, showing that our DEP approach is able to obtain the same performance but with a much lower dimensional descriptor (from 100- to 16-dimensional). Overall, our DEP descriptor based on geodesic distance (DEP16) achieves the best performance. Conversely, the use of diffusion distance (DEPdiff16) seems not convincing. From this test, we conclude that the best choice for the relation function is derived from the geodesic distance, as in our DEP descriptor, even if the other choices have confirmed the effectiveness of our evolution process scheme.

Finally, we evaluate the dependence of the proposed method on the regularization parameter. In Figure 21, on the right, we compute DEP descriptors with three different values of $c \in (0, 1)$ in the computation of the parameter $r$ as described in Section 5. The choice of $c$ gives different weights to the paths with a large number of steps (and vice versa). We evaluate $c = 0.1$, $c = 0.5$, and $c = 0.9$. Note that the performances are very similar, suggesting that the best choice of $c$ can be estimated with respect to the task at hand. In general, the choice $c = 0.1$ consistently gives good and stable performance; therefore, we fix this value throughout our experiments. Finally, in Figure 21, on the right, we also evaluate the performance of our descriptor by computing the geodesic distance on the surface using different methods. Namely, we compare the Dijkstra algorithm (*di*) (Mitchell et al. 1987), fast marching (*fm*)

(Kimmel and Sethian 1998), and exact geodesic (*eg*) (Surazhsky et al. 2005). As can be seen on the right of Figure 21, the performance is similar across different choices, and for computational efficiency we use Dijkstra's algorithm in the following experiments.

### 7.6 Complexity

The complexity of our method is dominated by computing and storing the pairwise geodesic distance matrix $\mathcal{G}$. In practice, we use Dijkstra's algorithm to approximate the geodesic distances on a triangle mesh. By using a straightforward nonoptimized implementation, our method required, on average, on a triangle mesh with around 7000 vertices, just over 10 minutes to compute the descriptors of all points across a range of 100 scales and just over 1 minute across a range of 16 scales, on a machine with 8GB of RAM using an Intel 2.6 GHz Core i7 processor. HKS and WKS take a few seconds on the same machine.

## 8 CONCLUSIONS

In this article, we propose a novel multiscale signature, the *Discrete-Time Evolution Process Descriptor* (DEP), which is able to effectively encode the structure of geodesic neighborhoods of a point across multiple scales. This point descriptor is derived from a novel paradigm for the simulation of a discrete-time evolution process that runs through all possible geodesic paths between pairs of points on the surface. We have shown that our DEP descriptor outperformed the state-of-the-art in point-to-point matching on different scenarios. We demonstrated that our work is similar in principle to methods and signatures inspired by the concept of diffusion geometry, such as the HKS or the WKS, but provides information that is an alternative to these descriptors.

In our framework, the simulation of the discrete evolution process is encoded by a new process operator for functions defined on the surface. This operator is strongly dependent on the chosen relation function between pairs of points. Our choice of geodesic distance has been shown to be very effective in terms of matching performance, but it is still computationally expensive. In Crane et al. (2013), an efficient alternative computation of geodesic distances was proposed starting from the heat kernel. Adopting this method, the computation of our descriptor can be improved. We leave this as future work. Moreover, the construction of the process operator can be further investigated by exploiting different relation functions that can be used as an alternative to the geodesic distance. Finally, it will be of interest to analyze the use of our process operator for a nonconstant initial state to encode additional information, such as texture or color.

## APPENDIX I

In this section, we want to justify the correctness of the method in terms of convergence. The value of $r$ (used in the generating function) can be determined by relying on linear algebra (Hubbard and Hubbard 2001). Let us consider $\{\lambda_0, \ldots, \lambda_{n-1}\}$ eigenvalues of the matrix $\mathbf{A}$; drawing from linear algebra, we can define the spectral radius $\rho(\mathbf{A})$ as

$$\rho(\mathbf{A}) = \max_{\lambda_i \in \{\lambda_0, \ldots, \lambda_{n-1}\}} (|\lambda_i|).$$

For the theory of convergence of the geometric series of matrices, we have also that

$$\lim_{l \to \infty} \mathbf{A}^l = 0 \iff \rho(\mathbf{A}) < 1 \iff \sum_{l=0}^{\infty} \mathbf{A}^l = (\mathbf{I} - \mathbf{A})^{-1}.$$

Furthermore, another theoretical result, indicated as Gelfand's formula, states that for every matrix norm we have that

$$\rho(\mathbf{A}) = \lim_{k \longrightarrow \infty} ||\mathbf{A}^k||^{\frac{1}{k}}.$$

This formula leads directly to an upper bound for the spectral radius of the product of two matrices, which commutes, given by the product of the spectral radii of the two matrices, that is, for each pair of matrices $\mathbf{A}$ and $\mathbf{B}$:

$$\rho(\mathbf{AB}) \leq \rho(\mathbf{A})\rho(\mathbf{B}).$$

Starting from the definition of $\check{s}(i)$ and from the following trivial consideration

$$r^l \mathbf{A}^l = \left(r^l \mathbf{I}\right) \mathbf{A}^l = \left[(r\mathbf{I})\,\mathbf{A}\right]^l,$$

we can use Gelfand's formula on $r\mathbf{I}$ and $\mathbf{A}$ and thus obtain that

$$\rho((r\mathbf{I})\mathbf{A}) \leq \rho(r\mathbf{I})\rho(\mathbf{A}) = r\rho(\mathbf{A}). \tag{17}$$

For the property of the spectral radius: $\lim_{l \to \infty} (r\mathbf{A})^l = 0 \iff \rho(r\mathbf{A}) < 1$. Thus, if we choose $r$ such as $0 < r < \frac{1}{\rho(\mathbf{A})}$, then we have that

$$0 < \rho(r\mathbf{A}) = \rho((r\mathbf{I})\mathbf{A}) \leq \rho(r\mathbf{I})\rho(\mathbf{A}) = r\rho(\mathbf{A}) < \frac{1}{\rho(\mathbf{A})}\rho(\mathbf{A}) = 1\,,$$

which implies that $\rho(r\mathbf{A}) < 1$, and so that

$$\check{\mathbf{S}} = \sum_{l=0}^{\infty} (rA)^l = (I - r\mathbf{A})^{-1}.$$

This choice of $r$ allows us to have convergence in the sum that defines $\check{s}(i)$.

## APPENDIX II

This section is dedicated to the limit case of $r \to 0$. In our method, we compute our score $v$ solving the following linear system:

$$(I - r\mathbf{A})v = e.$$

If we consider the exponential of the matrix $-r\mathbf{A}$, we have that

$$e^{-r\mathbf{A}} = \sum_{k=0}^{\infty} \frac{(-r)^k (\mathbf{A})^k}{k!} = I - r\mathbf{A} + \frac{r^2\mathbf{A}^2}{2} + \cdots$$

Now, in the limit case of $r \to 0$, we can approximate $e^{-r\mathbf{A}}$ as

$$e^{-r\mathbf{A}} \approx I - r\mathbf{A}.$$

Thus, we can write:

$$(I - r\mathbf{A})v \approx e^{-r\mathbf{A}}v = e$$

and multiplying left and right sides for $e^{r\mathbf{A}}$ we obtain that

$$v = e^{r\mathbf{A}}e.$$

From this point of view, $v$ is the state after a time $r$ of a process governed by $e^{r\mathbf{A}}$. Generalizing this process for a initial state $f$ at time 0, we obtain the state at time $r$ as

$$f_r = e^{r\mathbf{A}}.f.$$

Using the previous approximation $(I - r\mathbf{A})v \approx e^{-r\mathbf{A}}v = e$, we can recover the partial differential equation that defines this process:

$$f_r = e^{r\mathbf{A}}f \approx (I + r\mathbf{A})f = f + r\mathbf{A}f,$$

from which we have that

$$\frac{f_r - f}{r} = \mathbf{A}f, \text{ and for } r \to 0, \text{ we get that } \frac{\partial f}{\partial t} = \mathbf{A}f.$$

Thus, we can conclude that, in the limit of $r \to 0$, our score can be computed as the first step of a different process for a time equal to $r$. This is not very useful for our framework because it is only a limit case and the connection with our method is given only for an infinitesimal time interval.

## ACKNOWLEDGMENTS

The authors thank Emanuele Rodolà for his suggestions and help given in the evaluation.

## REFERENCES


Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. 2016. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision* 118, 3, 380–392.

M. Aubry, U. Schlickewei, and D. Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *International Conference on Computer Vision (ICCV'11)*. 1626–1633.

S. Belongie, J. Malik, and J. Puzicha. 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 4, 509–522.

F. Bernardini and H. Rushmeier. 2002. The 3D model acquisition pipeline. *Computer Graphics Forum* 21, 2, 149–172.

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*. 3794–3801.

D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum* 35, 5, 13–23.

E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, and others. 2011. SHREC 2011: Robust feature detection and description benchmark. In *Proceedings of 3DOR*. Eurographics Association, 71–78. Retrieved October 17, 2017 from http://tosca.cs.technion.ac.il/book/shrec_correspondence.html.

A. M. Bronstein and M. M. Bronstein. 2011. Shape recognition with spectral distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5, 1065–1071.

Alexander M. Bronstein, Michael M. Bronstein, and Maks Ovsjanikov. 2012. Feature-based methods in 3D shape analysis. In *3D Imaging, Analysis and Applications*. Springer, 185–219.

Mathieu Carrière, Steve Oudot, and Maks Ovsjanikov. 2015. Stable topological signatures for points on 3D shapes. *Computer Graphics Forum* 34, 5, 1–12.

Qifeng Chen and Vladlen Koltun. 2015. Robust nonrigid registration by convex optimization. In *International Conference on Computer Vision (ICCV'15)*.

R. R. Coifman and S. Lafon. 2006. Diffusion maps. *Applied and Computational Harmonic Analysis* 21, 1, 5–30.

Étienne Corman, Maks Ovsjanikov, and Antonin Chambolle. 2014. Supervised descriptor learning for non-rigid shape matching. In *European Conference on Computer Vision (ECCV'14) Workshops*. Springer, 283–298.

Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2013. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transaction on Graphics* 32, 5, 152:1–152:11.

Ernst Rudolf, Georg Eckert, and Robert M. Drake. 1987. Analysis of heat and mass transfer.

Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. 2004. Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision (ECCV'04)*. Springer, Berlin, 224–237.

R. Gal, A. Shamir, and D. Cohen-Or. 2007. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics* 13, 261–271.

K. Gebal, J. A. Bærentzen, H. Anæs, and R. Larsen. 2009. Shape analysis using the auto diffusion function. *Computer Graphics Forum* 28, 5, 1405–1413.

Gene H. Golub and Charles F. Van Loan. 2012. *Matrix Computations*. Vol. 3. Johns Hopkins University Press, Baltimore, MD.

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. 1994. *Concrete Mathematics: A Foundation for Computer Science* (2nd ed.). Addison-Wesley Longman Publishing Co., Inc., New York, NY.

Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of SIGGRAPH*. 203–212.

Qixing Huang, Fan Wang, and Leonidas Guibas. 2014. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics* 33, 4, 36.

J. H. Hubbard and B. B. Hubbard (Eds.). 2001. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach (2nd ed.)*. Pearson, New York, NY.

Adrian Ion, Nicole M. Artner, Gabriel Peyr, Walter G. Kropatsch, and Laurent D. Cohen. 2011. Matching 2D and 3D articulated shapes using the eccentricity transform. *Computer Vision and Image Understanding* 115, 6, 817–834.

Andrew E. Johnson and Martial Hebert. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 5, 433–449.

V. G. Kim, Yaron Lipman, and T. Funkhouser. 2011. Blended intrinsic maps. *ACM Transaction on Graphics* 30, 4, 79:1–79:12.

Ron Kimmel and James A. Sethian. 1998. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences* 95, 15, 8431–8435.

Zhouhui Lian, Afzal Godil, Benjamin Bustos, et al. 2013. A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition* 46, 1, 449–461.

O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, and D. Cremers. 2016. Non-rigid puzzles. *Computer Graphics Forum* 35, 5, 135–143.

Roee Litman, Alex Bronstein, Michael Bronstein, and Umberto Castellani. 2014. Supervised learning of bag-of-features shape descriptors using sparse coding. *Computer Graphics Forum* 33, 5, 127–136.

Roee Litman and Alexander M. Bronstein. 2014. Learning spectral descriptors for deformable shape correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 1, 170–180.

David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 2, 91–110.

Siddharth Manay, Byung-Woo Hong, Anthony J. Yezzi, and Stefano Soatto. 2004. Integral invariant signatures. In *European Conference on Computer Vision (ECCV'04)*. 87–99.

Simone Melzi, Emanuele Rodola, Umberto Castellani, and Michael Bronstein. 2016. Shape analysis with anisotropic windowed Fourier transform. In *International Conference on 3D Vision (3DV'16)*.

Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. 1987. The discrete geodesic problem. *SIAM Journal of Computing* 16, 4, 647–668.

Dorian Nogneng and Maks Ovsjanikov. 2017. Informative descriptor preservation via commutativity for shape matching. In *Annual Conference of the European Association for Computer Graphics (Eurographics'17)*.

Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2002. Shape distributions. *ACM Transaction on Graphics* 21, 4, 807–832.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics* 31, 4, 30:1–30:11.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. 2016. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*. Article 9, 60 pages.

Giuseppe Patané. 2014. Laplacian spectral distances and kernels on 3D shapes. *Pattern Recognition Letters* 47 102–110.

Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. 2015. Building statistical shape spaces for 3D human modeling. In *ArXiv*.

Jonathan Pokrass, Alexander M. Bronstein, Michael M. Bronstein, Pablo Sprechmann, and Guillermo Sapiro. 2013. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, Vol. 32. 459–468.

Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. 2009. Integral invariants for robust geometry processing. *Computer Aided Geometric Design* 26, 1, 37–60.

Kathleen M. Robinette, Hans Daanen, and Eric Paquet. 1999. The CAESAR project: A 3-D surface anthropometry survey. In *Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*. IEEE, 380–386.

Emanuele Rodolà, Luca Cosmo, Michael M. Bronstein, Andrea Torsello, and Daniel Cremers. 2016. Partial functional correspondence. In *Computer Graphics Forum*.

Emanuele Rodola, S. Rota Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. 2014. Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*. 4177–4184.

Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* 24, 4, 249–259.

Robert W. Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. 2005. Mesh-based inverse kinematics. *ACM Transaction on Graphics* 24, 3, 488–495.

Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum* 28, 5, 1383–1392.

Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. 2005. Fast exact and approximate geodesics on meshes. *ACM Transaction on Graphics* 24, 3, 553–560.

J. W. Tangelder and R. C. Veltkamp. 2004. A survey of content based 3D shape retrieval methods. In *IEEE International Conference on Shape Modeling and Applications (SMI'04)*. 145–156.

Federico Tombari, Samuele Salti, and Luigi Di Stefano. 2010. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV'10)*. 356–369.

Shi-Qing Xin, Wenping Wang, Shuangmin Chen, Jieyu Zhao, and Zhenyu Shu. 2016. Intrinsic girth function for shape processing. *ACM Transaction on Graphics* 35, 3, 25:1–25:14.

Yong-Liang Yang, Yu-Kun Lai, Shi-Min Hu, and Helmut Pottmann. 2006. Robust principal curvatures on multiple scales. In *Symposium on Geometry Processing (SGP'06)*. 223–226.

Andrei Zaharescu, Edmond Boyer, and Radu Horaud. 2012. Keypoints and local descriptors of scalar functions on 2D manifolds. *International Journal of Computer Vision* 100, 1, 78–98.

Received June 2016; revised July 2017; accepted August 2017