Efficient Globally Optimal 2D-to-3D Deformable Shape Matching

Zorah Lähner
TU München
laehner@in.tum.de

Emanuele Rodolà
TU München
Università della Svizzera Italiana
emanuele.rodola@usi.ch

Frank R. Schmidt
TU München
f.schmidt@cs.tum.edu

Michael M. Bronstein
Università della Svizzera Italiana
michael.bronstein@usi.ch

Daniel Cremers
TU München
cremers@tum.de

Abstract

We propose the first algorithm for non-rigid 2D-to-3D shape matching, where the input is a 2D query shape as well as a 3D target shape and the output is a continuous matching curve represented as a closed contour on the 3D shape. We cast the problem as finding the shortest circular path on the product 3-manifold of the two shapes. We prove that the optimal matching can be computed in polynomial time with a (worst-case) complexity of $\mathcal{O}(mn^2 \log(n))$, where m and n denote the number of vertices on the 2D and the 3D shape respectively. Quantitative evaluation confirms that the method provides excellent results for sketch-based deformable 3D shape retrieval.

1. Introduction

The last decade has witnessed a tremendous growth in the quantity and quality of geometric data available in the public domain. One of the driving forces of this growth has been the development in 3D sensing and printing technology, bringing affordable sensors such as Microsoft Kinect or Intel RealSense and 3D printers such as MakerBot to the mass market. The availability of large geometric datasets brings forth the need to explore, organize, and search in 3D shape collections, ideally in the same easy and efficient way as modern search engines allow to process text documents.

Numerous works on *content-based shape retrieval* [8, 32, 5] try to extend popular search paradigms to a setting where the3D query shape is matched to shapes in the database using geometric similarities. Typically, a 3D shape is represented as a descriptor vector aggregating some local geometric features, and retrieval is done efficiently by comparing such vectors [5]. However, the need for the query to be a 3D shape significantly limits the practical usefulness of such search engines: non-expert human users are typically not very skilled with 3D modeling, and thus providing a good query example can be challenging.

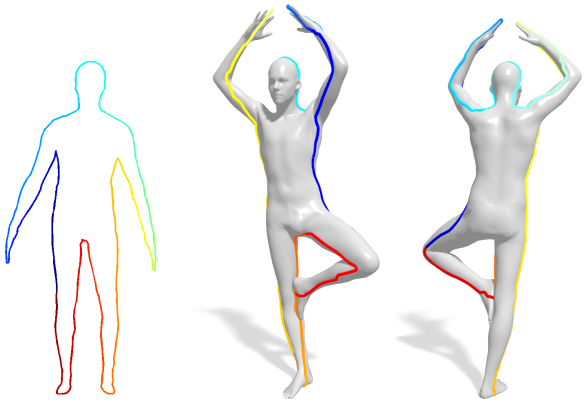


Figure 1. We propose the first shape matching method between a 2D query shape (left) and a 3D target shape (right), both of which are allowed to deform non-rigidly. The globally optimal matching (shown on top of the 3D target) is guaranteed to be continuous.

As an alternative to 3D-to-3D shape retrieval, several recent works proposed 2D-to-3D or *sketch-based* shape retrieval, where the query is a 2D image representing the projection or the silhouette of a 3D shape as seen from some viewpoint [7, 9, 14, 30, 11]. This setting is much more natural to human users who in most cases are capable of sketching a 2D drawing of the query shape; however, the underlying problem of 'multi-modal' similarity between a 3D object and its 2D representation is a very challenging one, especially if one desires to deal with non-rigid shapes such as human body poses. In fact, so far all methods for 2D-to-3D matching have limited the attention to rigid shapes such as chairs, cars, etc.

In this paper, we propose a method for automatically finding correspondence between 2D and 3D deformable ob-

jects. To the best of our knowledge, this is the first method to address the problem in the deformable setting. The input to our algorithm is a 2D query curve and a 3D target surface, and the output is the corresponding continuous curve on the surface (see Fig. 1). Our method guarantees a globally optimal solution and has polynomial time complexity. As a byproduct of the correspondence we also get a 2D-to-3D similarity criterion, allowing to perform efficient sketch-based shape retrieval.

The rest of this paper is organized as follows. In the remaining part of this section, we review the previous works and summarize our contributions. In Section 2, we formulate the 2D-to-3D matching problem as an energy minimization problem and discuss its discretization and optimization. Section 3 shows experimental results. We consider a challenging application, namely deformable sketch-based retrieval. Finally, Section 4 concludes the paper discussing the limitations and potential extensions of the approach.

1.1. Related work

2D and 3D shape correspondence. The classical 2D-to-2D and 3D-to-3D settings of the shape matching problem have been thoroughly researched in the computer vision and graphics communities (see [33] for a survey). In the domain of 3D-to-3D shape matching, a major challenge is to have theoretical guarantees about the optimality and quality of the correspondence. Several popular methods try to find a correspondence that minimally distorts intrinsic distances between pairs of corresponding points by approximate solution to the quadratic assignment problem [13, 16, 6, 21, 25]. A recent line of works builds upon the functional representation [18, 19, 22, 12, 23], where a point-wise map is replaced by a linear map between function spaces. This way, finding correspondence boils down to solving a linear system of equations. The drawback of this framework is its inability to guarantee basic properties of the map such as continuity. This is because not every linear operator is automatically continuous. A method for finding guaranteed continuous correspondence was proposed by Windheuser *et al*. [34]. Similarly to [34], our method comes with the theoretical guarantee of a continuous solution, however at a significantly lower computational cost (about half a minute instead of several hours).

In our 2D-to-3D correspondence problem, the 2D shape is modeled as a closed planar curve, and the 3D shape as a surface in $\mathbb{R}^3$. To find the correspondence, we look for a closed curve on the surface. From this perspective, our method can be seen as an extension of an image segmentation task that looks for a closed curve within a 2D image domain. It was shown in [28] that this segmentation problem can be formulated as finding a shortest path in the product graph of the 2D image domain and the 1D curve domain, where the size of the graph depends on the Lipschitz constant of the mapping. The drawback is that this constant is typically unknown in advance. Differently from [28], the size of the constructed graph with our method is independent of the Lipschitz constant.

Furthermore, one of the main challenges in our method is to find an initial match on the product graph. In [28] this problem was solved by parallelization. As a result, the overall computation time is not reduced, but just distributed intelligently among several computational cores. Instead, we use a branch-and-bound approach that only computes shortest paths in those regions that are 'most promising'. This strategy reduces the running time substantially (especially with well-chosen shape descriptors), while still converging to a global optimum.

Even in the simpler 2D-to-2D setting, the computation of a globally-optimal correspondence can be very slow if we do not know an initial match. For example, the running time of Dynamic Time Warping methods is $\mathcal{O}(n^2)$ if an initial match is given, and $\mathcal{O}(n^3)$ if every possible initial match is tested independently, where n is the number of shape samples. It was shown that by exploiting the planarity of the involved graph, the running time of the whole matching including an initial match can be reduced to $\mathcal{O}(n^2 \log(n))$ by using shortest circular path or graph cut approaches [15, 26, 27]. A competitive approach is the branch-and-bound approach of [1]. While it does not reduce the worst case time complexity of $\mathcal{O}(n^3)$, it is rather fast in practice. Since it does not use the planarity of the involved graph, we can adapt it to our scenario in order to reduce the practical running time substantially.

Sketch-based retrieval. One of the important applications of 2D-to-3D matching is shape-from-sketch retrieval. This problem has recently drawn the attention of the machine learning community as a fertile playground for cross-modal feature learning [7, 9, 14, 30, 11]. Herzog *et al*. [10] recently proposed to learn a shared semantic space from multiple annotated databases, on which a metric that links semantically similar objects represented in different modalities (namely 2D drawings and 3D targets) is learned. Although this approach yields promising results in the *rigid* setting and can address some variability of the shapes, its applicability to the *non-rigid* setting is an open question. In contrast, our method targets explicitly the setting when both the 3D target and the 2D query are allowed to deform in a non-rigid fashion. Furthermore, the method of [10] as well as other existing approaches mostly focus on finding *similarity* between a 2D sketch and a 3D shape, while we solve the more difficult problem of finding *correspondence* (from which a criterion of similarity is obtained as a byproduct).

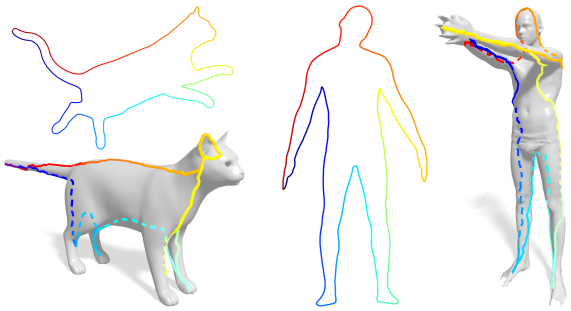


Figure 2. Correspondences on a cat (left) and a human. The human shapes exhibit a topological change along the hands (compare "middle" and "right"), which is handled well by our method.

1.2. Contribution

The main contribution of our paper is to present the first automatic method for guaranteed continuous matching of 2D and 3D non-rigid shapes. Prior approaches to the 2D-to-3D matching problem were limited to the less challenging rigid setting. Our method finds a globally optimal solution in polynomial time. In our experiments we observed running times in the range of 5-30 seconds. Secondly, our approach allows using different local feature descriptors for 2D and 3D data. In particular, we show how spectral 2D and 3D features can be compared such that we obtain a semantically driven matching between 2D and 3D shapes. Thirdly, we show the application of our 2D-to-3D matching for deformable shape retrieval from sketches, a task that traditionally requires large sets of human-labeled data.

2. 2D-to-3D matching

In this section, we formulate, discretize and optimize the shape matching problem between a 2D *query shape* and a 3D *target shape*. By a *shape* we refer to the outer shell of an object. The object itself will be referred to as the shape's *solid*. The 3D ball $B = \{x \in \mathbb{R}^3 | \, \|x\| \leq 1\}$ is for example the solid of the sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 | \, \|x\| = 1\}$. We summarize this convention in the following definition:

Definition 1. *A compact set $S \subset \mathbb{R}^d$ is called a* shape *of dimension d if it is a connected, smooth manifold and if it can be represented as the boundary $S = \partial U$ of an open subset $U \subset \mathbb{R}^d$. In this case, we call U the* solid *of S.*

Note that this definition implies that a 3D shape is a 2-manifold and a 2D shape is a 1-manifold.

This section is organized as follows. In Section 2.1 we will cast the 2D-to-3D shape matching problem as an energy minimization problem, which we will globally optimize in Section 2.2. To this end, we assume that descriptive features for both shapes are given and that it is possible to measure the dissimilarity between 2D and 3D features. The specific choice of such features depends on the application.

For the application of *shape retrieval* that we discuss in Section 3.3 we use purely spectral features.

2.1. Energy formulation

Given the 2D query shape $M \subset \mathbb{R}^2$ and the 3D target shape $N \subset \mathbb{R}^3$, we call a mapping $\varphi \colon M \to N$ a *2D-to-3D matching* if it is an immersion, *i.e.*, if $d\varphi$ is of maximal rank. The goal of this paper is it to find a 2D-to-3D matching that sets points that look alike into correspondence. To this end let $f_M \colon M \to \mathbb{R}^{k_M}$ and $f_N \colon N \to \mathbb{R}^{k_N}$ be two different feature maps. We want to stress that the dimensions k_M and k_N do not need to agree. In order to measure the dissimilarity between the 2D feature $f_M(x)$ of $x \in M$ and the 3D feature $f_N(y)$ of $y \in N$, we assume a positive distance function $\mathrm{dist} \colon \mathbb{R}^{k_M} \times \mathbb{R}^{k_N} \to \mathbb{R}_0^+$. This distance takes care of the difficult task of comparing 2D features with 3D features.

Given the two feature maps f_M and f_N as well as the distance function dist, we call a 2D-to-3D matching φ optimal if it minimizes the energy

$$E(\varphi) \colon = \int_{\Gamma_\varphi} \mathrm{dist}(f_M(s_1), f_N(s_2)) \mathrm{ds}, \tag{1}$$

where $\Gamma_\varphi = \{(s_1, s_2) \in M \times N | s_2 = \varphi(s_1)\}$ denotes the graph of φ. Note that Γ_φ is a simplicial complex due to the immersion property of φ. E is therefore defined as a line integral and takes the differential $d\varphi$ into account.

Since M is a one-dimensional manifold, the endomorphism $d\varphi_x^\top d\varphi_x$ of the tangent space $T_x M$ is uniquely represented by a scalar, to which we refer to as $J_\varphi(x)$. Using the substitution rule we can rewrite E as

$$E(\varphi) = \int_M \mathrm{dist}(f_M(x), f_N \circ \varphi(x)) \sqrt{1 + J_\varphi(x)} \mathrm{dx}. \tag{2}$$

Hence, the energy E can be broken into the *data term* $\mathrm{dist}(f_M(\cdot), f_N \circ \varphi(\cdot))$ and the *regularizer* $\sqrt{1 + J_\varphi(\cdot)}$.

Regularization. If we ignore the data term, the global minimum of E would result in a constant φ. This φ is continuous, but matches every point on M to the same point on N. It therefore ignores the similarity information stored in the data term.

Data term. If we ignore the regularizer, the global minimum of E can be computed by selecting for each $x \in M$ a $y \in N$ that minimizes the feature distance $\mathrm{dist}(f_M(x), f_N(y))$. In this case, the minimizer of E might be neither injective nor continuous. Combining the data term with the regularization results in a smooth matching function φ that also takes the data term into account.

Alternatively to the energy described here, one could also choose to enforce injectivity of $\varphi \colon M \to N$. This would lead to a linear assignment problem (LAP), which allows for non-continuous matchings and is rather slow. If

the shapes M and N are discretized at m and n points, respectively, the overall running time of the Hungarian method [17] to solve this problem is $\mathcal{O}(n^3)$. The method that we propose does not only provide for a smoother solution, but also has a better worst case runtime complexity than the LAP (cf. Theorem 1). Exploring the running time of the LAP approach for one matching instance resulted in a running time of 11 hours instead of just a few seconds for our method.

2.2. Optimization

So far we defined the energy that we like to minimize. In the following we address the discretization of this minimization problem and we show that a globally optimal solution can be computed efficiently.

To this end we assume that the 2D query shape M and the 3D target shape N are discretized. Thus, M is given as a simple circular graph, *i.e.*, $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$ with

$$\begin{aligned}\mathcal{V}_M =& \{x_0, \dots, x_{m-1}\} \subset \mathbb{R}^2 \\ \mathcal{E}_M =& \{(x_i, x_j) \in \mathcal{V}_M^2 \mid j \equiv i+1 \mod m\}.\end{aligned}$$

The target shape N on the other hand is given as a 3D mesh $\mathcal{G}_N = (\mathcal{V}_N, \mathcal{E}_N, \mathcal{F}_N)$, where $\mathcal{V}_N = \{y_0, \dots, y_{n-1}\}$ denotes the set of vertices, $\mathcal{E}_N$ the set of unoriented edges, and $\mathcal{F}_N$ the set of faces. Further, we assume that the feature maps $f_M \colon M \to \mathbb{R}^{k_M}$ and $f_N \colon N \to \mathbb{R}^{k_N}$ are given as information on the vertices. Thus, we can write $\operatorname{dist}(\cdot,\cdot)$ as a matrix $D \in \mathbb{R}^{m \times n}$ with $D_{ij} = \operatorname{dist}(f_M(x_i), f_N(y_j))$.

Given this discretization, we define the product graph $\mathcal{G}_{M\times N} = (\mathcal{V}_{M\times N}, \mathcal{E}_{M\times N}, \mathcal{C}_{M\times N})$ via

$$\begin{aligned}\mathcal{V}_{M\times N} =& \{0,\dots,m\} \times \{0,\dots,n-1\} \\ \mathcal{E}_{M\times N} =& \big\{[(i_0,j_0),(i_1,j_1)] \in \mathcal{V}_{M\times N}^2 \big| \\ & (i_1 = i_0) \wedge (y_{j_1}, y_{j_0}) \in \mathcal{E}_N \text{ or} \\ & (i_1 = i_0 + 1) \wedge (j_1 = j_0) \text{ or} \\ & (i_1 = i_0 + 1) \wedge (y_{j_1}, y_{j_0}) \in \mathcal{E}_N\big\} \\ \mathcal{C}_{M\times N}[(i_0,j_0),(i_1,j_1)] =& \\ & \frac{D_{\bar{i}_0,j_0} + D_{\bar{i}_1,j_1}}{2} \cdot \left\| (x_{\bar{i}_0}, y_{j_0}) - (x_{\bar{i}_1}, y_{j_1}) \right\|,\end{aligned}$$

where $\bar{i} := i \mod m$. The product graph takes the Cartesian product of the vertices in $\mathcal{G}_M$ and $\mathcal{G}_N$ and connects them iff their projections on both shapes are connected (or identical). See Fig. 3 for an illustration.

In addition, the representation of the 2D shape M is extended by having two copies of x_0, namely at position $i = 0$ and at position $i = m$. As a result, any matching can be represented by a path from $(0, j)$ to (m, j). Hence, an optimal matching can be cast as finding a shortest path in a graph if an initial match $(x_0, y_j) \in \Gamma_\varphi$ is given. Such a computation can easily be done by Dijkstra's algorithm. Using a priority

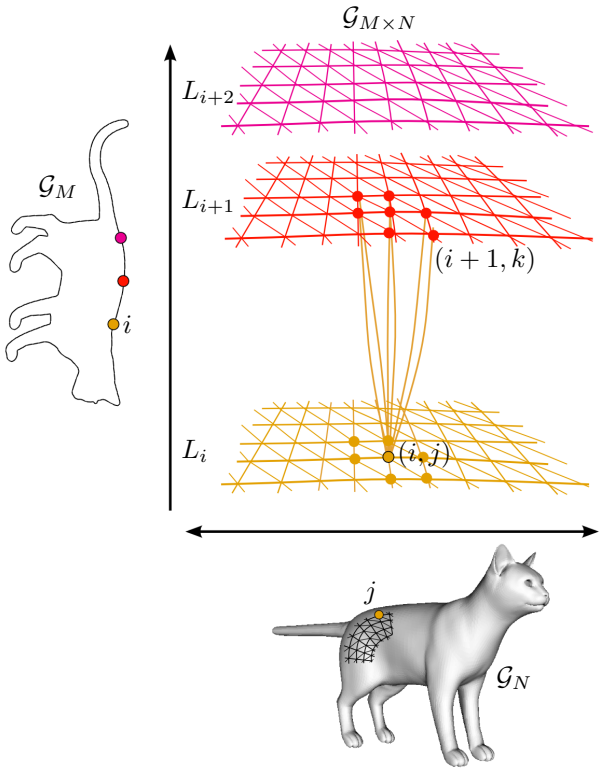


Figure 3. A node (i, j) in the product graph $\mathcal{G}_{M\times N}$ represents a match between the vertex $i \in \mathcal{V}_M$ of the contour M and the vertex $j \in \mathcal{V}_N$ of the surface N. All feasible matches with respect to vertex i form the layer $L_i = \{i\} \times \mathcal{V}_N$. Edges are defined between a node (i, j) and $(i+1, j)$ as well as (i, k) and $(i+1, k)$ for all surface vertices $k \in \mathcal{V}_N$ that are adjacent to j. All these edges are directed and enforce a continuous matching.

heap the computation takes $\mathcal{O}(mn \cdot \log(mn))$ time. Since there is no path from (i_1, j_1) to (i_0, j_0) if $i_1 > i_0$, we associate to each *layer* $\{i\} \times \{0, \dots, n-1\}$ a different priority heap and reduce the running time to $\mathcal{O}(mn \log(n))$. These observations lead to the following theorem.

Theorem 1. *Given a 2D query shape M and a 3D target shape N, discretized by m and n vertices, respectively, we can find a minimizer of (1) in $\mathcal{O}(mn^2 \log(n))$ time. If $n = \mathcal{O}(m^2)$, this leads to the subcubic runtime of $\mathcal{O}(n^{2.5} \log(n))$.*

Proof. The runtime analysis follows from the above mentioned observations and the fact that we have to test n different initial matches. The only thing that remains to be shown is that the edge costs $\mathcal{C}_{M\times N}[(i_0, j_0), (i_1, j_1)]$ discretize the line integral of (1). Since we have

$$\begin{aligned}D_{\bar{i}_0,j_0} &= \operatorname{dist}\left(f_M(x_{\bar{i}_0}), f_N(y_{j_0})\right) \\ D_{\bar{i}_1,j_1} &= \operatorname{dist}\left(f_M(x_{\bar{i}_1}), f_N(y_{j_1})\right)\end{aligned}$$

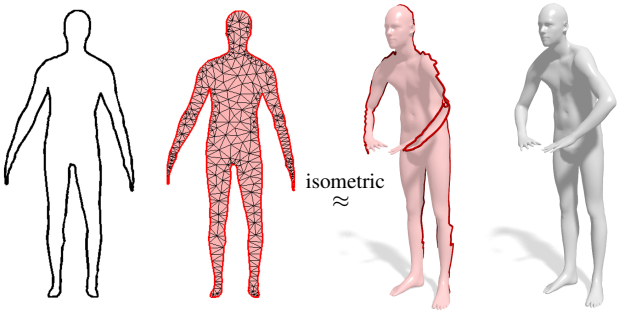
the edge cost is a linear approximation of the line integral associated with the edge $[(i_0, j_0), (i_1, j_1)]$. □

This theorem shows that we can find a globally optimal matching in polynomial time. Nonetheless, this may still lead to a high running time since we have to find for each vertex $y \in \mathcal{V}_N$ a shortest path in $\mathcal{G}_{M\times N}$. In order to circumvent this problem we follow a branch-and-bound strategy that is inspired by the method of [1].

Data: $\mathcal{G}_{M\times N} = (\mathcal{V}_{M\times N}, \mathcal{E}_{M\times N}, \mathcal{C}_{M\times N})$
Result: Matching path Γ_φ
Let $R := \{0, \dots, n-1\}$;
Define $\mathcal{R} = \{R\}$ and $b\colon \mathcal{R} \to \mathbb{R}$ via $b(R) = 0$;
Define `isFound=false`;
Define `bound`$=\infty$;
while *(*`isFound=false`*) and (*$\min b <$`bound`*)* **do**
  Let $R \in \arg\min b$;
  $\mathcal{R} = \mathcal{R} \setminus \{R\}$;
  Find shortest path Γ in $\mathcal{G}_{M\times N}$
    from $\{0\} \times R$ to $\{m\} \times R$;
  Γ is a path from $(0, i)$ to (m, j);
  if $i = j$ **then**
    `isFound=true`;
    if length(Γ) $<$`bound` **then**
      `bound`=length(Γ);
      $\Gamma_\varphi = \Gamma$
    end
  else
    Divide R into $R = R_1 \cup R_2$ such that
      $x \in R_1 :\Leftrightarrow \mathrm{dist}_N(x, i) < \mathrm{dist}_N(x, j)$;
    Set $\mathcal{R} := \mathcal{R} \cup \{R_1, R_2\}$;
    Set $b(R_1) = b(R_2) = \mathrm{length}(\Gamma)$;
  end
end

Algorithm 1: 2D-to-3D matching via branch-and-bound.

We observe that in practice we only have to compute a few shortest paths if we employ Algorithm 1. The main idea is to follow a coarse-to-fine strategy. First we compute the shortest path between $\{0\}\times R$ and $\{m\}\times R$, which connects $(0, i)$ with (m, j). If this path connects the corresponding points, *i.e.*, $i = j$, we found a matching path. Otherwise, we separate the region R into two sub-regions, R_1 containing i and R_2 containing j, and recompute shortest paths in those smaller sub-regions. Since the shortest path of corresponding sub-regions will not include the previously computed path, we can use its length as lower bound for both R_1 and R_2. Thus, we can control the order in which these subdomains are processed. We propose to separate R with respect to the geodesic distance dist_N on the target shape N. After we found our first matching path, we still have to process those subdomains whose lower bound is smaller than the computed matching path. Afterwards, we are sure that we found the globally optimal matching path Γ_φ.



(a) input 2D (b) (c) (d) input 3D

Figure 4. Spectral features are constructed by considering the query shape (a) as the boundary of a 2D region (b), which is assumed to be a near-isometric deformation of a sub-region (c) of the 3D target shape (d). The 2D tessellation (b) is obtained via [29].

3. Applications and results

In this section, we apply the proposed method to the problem of sketch-based deformable shape retrieval. We emphasize that our method is parameter-free, and the only choice is with respect to the 2D and 3D features $f_M(\cdot), f_N(\cdot)$ as well as the distance function $\mathrm{dist}(\cdot,\cdot)$ between them.

Datasets. Due to the novelty of the application, to date there is no benchmark available for evaluating deformable 2D-to-3D shape retrieval methods. We therefore construct such a benchmark using the FAUST [3] and TOSCA [4] datasets. The former dataset consists of 100 human shapes, subdivided into 10 classes (different individuals), each in 10 different poses. The latter consists of 80 shapes, subdivided into 9 classes (humans and animals in different poses). Ground-truth correspondences within each class are available. Shape sizes are fixed to 7K (FAUST) and 10K (TOSCA) vertices.

In both datasets, each class comes with a 'null' shape in a "neutral pose", which we use to define the 2D queries. To this end, we cut each null shape across a plane of symmetry, and project the resulting boundary onto a plane. This gives rise to 2D queries of 200–400 points on average. Note that by doing so we retain the ground-truth point-to-point mapping between the resulting 2D silhouette and the originating 3D target. This allows us to define a quantitative measure on the quality of the 2D-to-3D matching between objects of the same class[1].

Error measure. Let M be a 2D shape represented as a planar curve, and N a 3D shape represented as a surface. Let $\varphi\colon M \to N$ be a matching between a 2D query shape

[1]The dataset containing shapes and matching is publicly available at https://vision.in.tum.de/~laehner/Elastic2D3D/

and a 3D target shape, and let φ_0 be the ground-truth matching. The matching error of φ at point $x \in M$ is given by

$$\varepsilon_\varphi(x) = \frac{\mathrm{dist}_N(\varphi(x), \varphi_0(x))}{\mathrm{diam}(N)}, \tag{3}$$

where $\mathrm{dist}_N : N \times N \to \mathbb{R}_0^+$ denotes the geodesic distance on N and $\mathrm{diam}(N) = \max_{x,y \in N} \mathrm{dist}_N(x, y)$ the geodesic diameter of N. Note that due to the normalization, the values of the error ε are within $[0, 1]$.

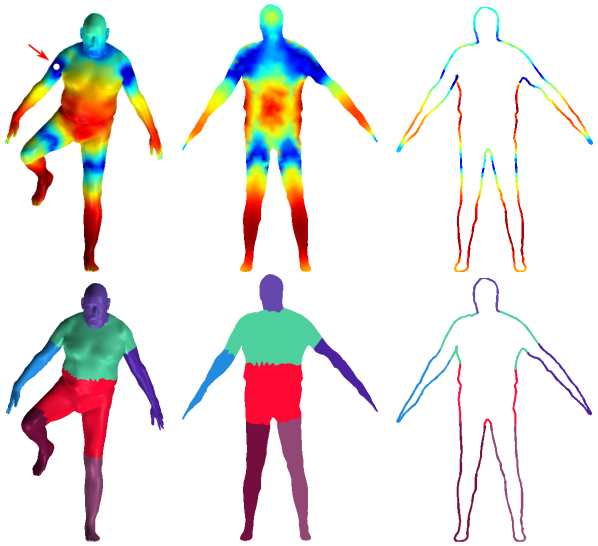
Spectral features. In this paper we advocate the adoption of spectral quantities to define compatible features between 2D and 3D shapes. Note that differently from existing methods for 2D-to-3D matching, we compute local features independently for each given pair of shapes, *i.e.*, no cross-modal metric learning is carried out.

Let Δ_N be the symmetric Laplace-Beltrami operator on the 3D shape N. Δ_N admits an eigendecomposition with non-negative eigenvalues $0 = \lambda_0 < \lambda_1 \leq \ldots$ and corresponding orthogonal eigenfunctions $\psi_j \colon N \to \mathbb{R}$ such that $\Delta_N \psi_j = \lambda_j \psi_j$. For the definition of our feature maps, we consider multi-dimensional spectral descriptors, *i.e.*, features constructed as functions of λ_j and ψ_j. Note that, since Δ_N is invariant to isometric transformations of N, the derived spectral descriptors also inherit this invariance.

Popular spectral features are the scaled Heat Kernel Signature (HKS) [31] and the Wave Kernel Signature (WKS) [2], which we denote as $f_N^{\mathrm{HKS}} \colon N \to \mathbb{R}^d$ and $f_N^{\mathrm{WKS}} \colon N \to \mathbb{R}^d$ respectively. In our experiments we used $d = 100$ and default values as described in the respective papers. All descriptors were normalized to have maximum value 1 in order to improve their robustness.

For the 2D case, designing features that can be compared to their 3D counterparts can be a challenging task. The difficulty is exacerbated here because the shapes are allowed to deform. To this end, we consider the solid U of M, *i.e.*, $\partial U = M$ (cf. Definition 1). In other words, we model the 2D query as a flat 2-manifold *with* boundary. This new manifold can be regarded as a nearly isometric transformation (due to flatness and possibly a change in pose) of a *portion* of the full 3D target (see Fig. 4). Taking this perspective allows us to leverage some recent advances in partial 3D matching [23], namely that partiality transformations of a surface preserve the Laplacian eigenvalues and eigenfunctions, up to some bounded perturbation.

An implication of this is that we can still compute spectral descriptors on the flat solid U and expect them to be comparable with those on the full 3D target. By doing so, we make the assumption that U can be approximated as a part of nearly-isometrically deformed M. We then define the feature maps $f_M^{\mathrm{HKS}}, f_M^{\mathrm{WKS}} \colon M \to \mathbb{R}^d$ on M by restricting the descriptors computed on U to its boundary curve $\partial U = M$ (see Fig. 5, top row). In other words, we have the spectral features $f_M = f_U|_M$ for the query shape M.



input 3D input 2D, tessellated input 2D

Figure 5. Computation of local features for elastic matching. *Top*: L_1-distance (blue to red) between spectral 3D descriptors of a reference point (white dot) and 3D descriptors computed on the remaining shape (left) as well as 2D descriptors on the tessellated query solid (middle). *Bottom*: Consensus regions detected in 3D and 2D using [24]. The restricted features (right) are used in the energy (1) to drive the matching process.

Segment features. As an additional 'coarse' feature we use corresponding regions on the 2D query and the 3D shape. Based on the previous observations, we are able to automatically extract compatible regions on the two objects (namely on U and N) by consensus segmentation [24], a deformation-invariant region detection technique which directly operates with the Laplace-Beltrami eigenfunctions of a given shape. The region detection step on the two shapes is performed independently; we then obtain the 2D-to-3D region mapping by solving a simple linear assignment problem via the Hungarian algorithm [17]. Note that this assignment problem is typically very small. Assuming we have r regions per shape (usually in the range of 5 to 10), the final result of this procedure is a pair of corresponding labelings $f_M^{\mathrm{SEG}} : M \to \mathbb{N}^r$ and $f_N^{\mathrm{SEG}} : N \to \mathbb{N}^r$ (see Fig. 5, bottom row).

Distance function. The final feature maps are obtained by simple concatenation, namely $f := (f^{\mathrm{HKS}}, f^{\mathrm{WKS}}, f^{\mathrm{SEG}})$. In order to compare the feature maps on M and N, we define the distance function:

$$\begin{aligned} \mathrm{dist}(f_M(x), f_N(y)) \quad &= \quad \|f_M^{\mathrm{HKS}}(x) - f_N^{\mathrm{HKS}}(y)\|_1 \\ &+ \quad \|f_M^{\mathrm{WKS}}(x) - f_N^{\mathrm{WKS}}(y)\|_1 \end{aligned} \tag{4}$$

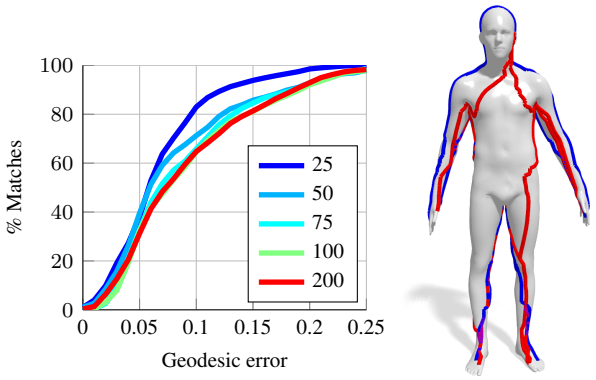


Figure 6. Sensitivity of the spectral features to increasing number of eigenfunctions. On the right we show typical solutions obtained when using 25 (blue) and 200 (red) eigenfunctions.

if $f_M^{\text{SEG}}(x) = f_N^{\text{SEG}}(y)$, and set $\text{dist}(f_M(x), f_N(y)) = \tau$ otherwise. Here, $\tau > 0$ is a large positive value to prevent matching points belonging to different regions. In our experiments, we used $\tau = 10^3$.

3.1. Sensitivity analysis

We performed a sensitivity analysis of our elastic matching method on a subset of our FAUST-derived dataset, under different parametrizations of the spectral features. In most shape analysis applications, only the first k eigenfunctions of Δ are used to define f^{WKS} and f^{HKS}. In the classical 3D-to-3D setting, for large k the resulting descriptors tend to be more accurate, but at the same time become more sensitive to the lack of isometry relating the two shapes.

We observed a similar trend in our 2D-to-3D setting, as reported in Fig. 6. From this analysis we selected $k = 25$ as the fixed number of eigenfunctions for all subsequent experiments. In the figure, we plot cumulative curves showing the percent of matches which have geodesic error (3) smaller than a variable threshold.

3.2. Runtime

We implemented our method in C++, and ran it on an Intel Core i7 3.4GHz CPU. In Fig. 7 we show the execution times of our method on the FAUST dataset (10 queries and 100 targets). The plotted results show that for 3D shapes of fixed size, in practice the runtime grows linearly with the number of 2D query points m.

3.3. Sketch-based shape retrieval

We consider a particular shape retrieval setting in which the dataset is assumed to be a collection of 3D shapes, and the query is a 2D silhouette (possibly drawn by a human). Differently from previous techniques [7, 9, 14, 30], our method does not use learning to compute features, and most importantly, we allow the shapes to deform in a non-rigid fashion. The 2D-to-3D shape similarity is obtained

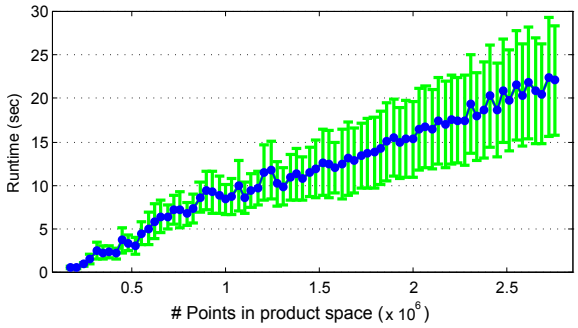


Figure 7. Top: Runtime of our matching method on 3D targets with ∼7K vertices (FAUST dataset). On the x axis we vary the size of the 2D query from 25 to 400 points.

	Ours	ShapeDNA [20]	Consensus [24]
cat	**1.0000**	0.1310	0.1050
human	**1.0000**	0.9078	0.6532
dog	**1.0000**	0.1066	0.3330
horse	**0.5062**	0.0611	0.0629
wolf	**1.0000**	0.0379	0.0302
MAP	**0.9012**	0.2489	0.2369

Table 1. Retrieval results on the 2D-to-3D TOSCA dataset. For each method we show per-class AP and, in the last row, the MAP.

by considering the minimal value of the energy $E(\varphi^*)$ obtained by our optimization problem.

We evaluated the performance of our retrieval pipeline on the full 2D-to-3D TOSCA dataset. As baselines for our comparisons we use the spectral retrieval method of [20] and a pure region-based retrieval technique using segments computed with [24]. The rationale of these experiments is to show that the specific features we use are not sufficient to guarantee good retrieval performance. However, using these quantities in our elastic matching pipeline enables promising results even in challenging cases.

The first baseline method we compare against is ShapeDNA [20], using the (truncated) spectrum of the Laplace-Beltrami operator as a 'global' isometry-invariant shape descriptor. We apply this method to compare targets in the 3D database with flat tessellations of the 2D queries.

The second method used in the comparisons is a simple evaluation of the matching cost obtained when putting the consensus regions into correspondence via linear assignment (see Fig. 5). Since this step typically produces good coarse 2D-to-3D matchings, it can be used as a retrieval procedure per se.

The results of the shape retrieval experiments are reported in Table 1. We used average precision (AP) and mean average precision (MAP) as measures of retrieval performance[2]. Additional qualitative examples of solutions obtained with our method are shown in Fig. 8.

[2] *Precision* measures the percentage of correctly retrieved shapes.

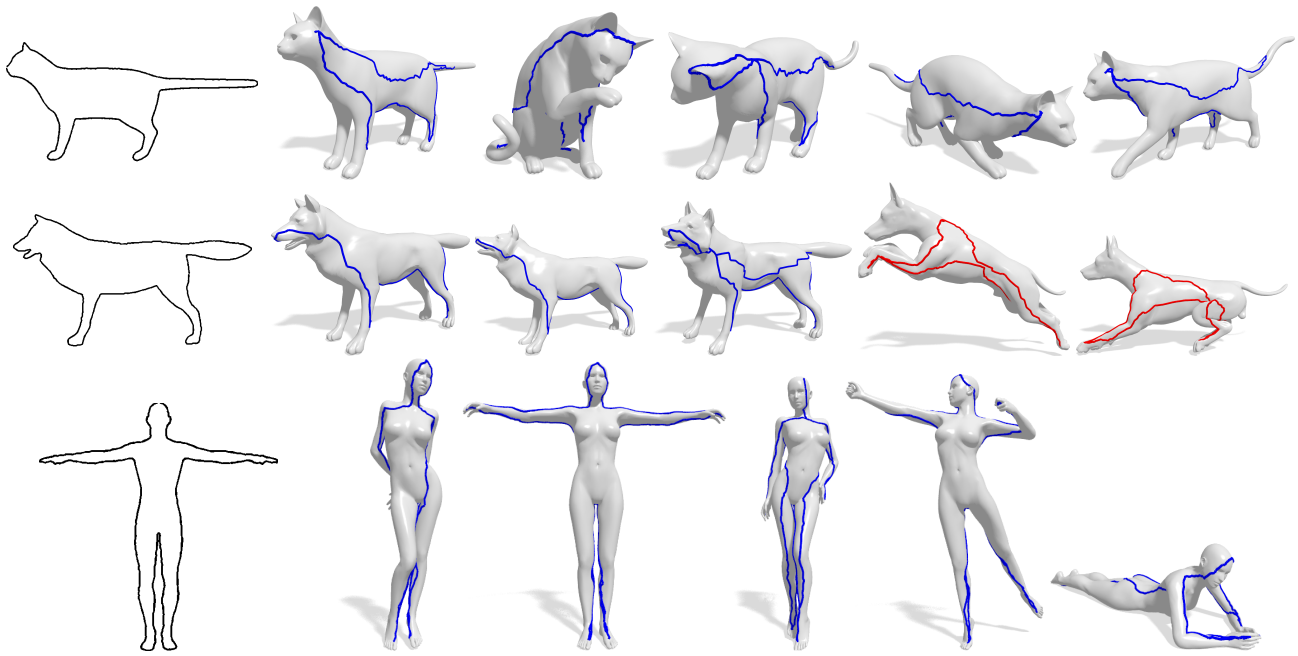


Figure 8. Retrieval examples on the TOSCA dataset. *Left:* Two of the three 2D queries (cat and wolf) have missing parts (two legs in contrast to Fig. 2). Each row shows the top 5 results (ranked by matching energy) provided by our method. The corresponding matching curves are shown on top of the 3D targets. Note that the dataset only contains 3 wolf shapes, which show up as the top 3 matches. The next matches are shapes of the class "dog", which is a semantically similar class.

4. Discussion and conclusions

We proposed the first polynomial-time solution for matching deformable planar curves to 3D shapes. We prove that the worst-case complexity of this algorithm is $\mathcal{O}(mn^2 \log(n))$, where m and n denote the number of samples on the query curve and the 3D shape respectively. We show experimentally that the running time remains linear with respect to m, even when employing a branch-and-bound strategy, making it a very efficient approach that matches curves of hundreds of vertices to 3D shapes with thousands of vertices in a few seconds. We also would like to stress that the branch-and-bound strategy is so effective due to the presented features. Removing the segment feature f^{SEG} from our framework, increases the running time by 2–3 orders of magnitude.

Our algorithm provides a powerful tool for shape analysis, and in particular has great potential in applications such as 3D shape retrieval from 2D sketches.

Limitations. The main limitation of our method is the assumption that the query 2D shape is a closed planar curve. In some situations, this may limit the 'expressivity' of the sketch and pose a disadvantage compared to image-based approaches in sketch-based retrieval applications.

Future research directions. One notable drawback of our discrete optimization is the use of Dijkstra's algorithm for finding shortest paths on the product manifold, which in some situations may not be a consistent discretization of the geodesic distance. In follow-up works, we will explore the use of consistent fast-marching-like schemes.

Acknowledgments

We thank Aneta Stevanović and Matthias Vestner for useful discussions. We gratefully acknowledge the support of an Alexander von Humboldt Fellowship, the ERC Starting Grant No. 307047 (COMET), the ERC Starting Grant "ConvexVision" and the ERC Consolidator Grant "3D Reloaded".

References

[1] B. Appleton and C. Sun. Circular shortest paths by branch and bound. *Pattern Recognition*, 36(11):2513–2520, 2003. 2, 5

[2] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV*, pages 1626–1633, 2011. 6

[3] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, June 2014. 5

[4] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008. 5

[5] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape Google: Geometric words and expressions

for invariant shape retrieval. *Trans. Graphics*, 30(1):1, 2011. 1

[6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006. 2

[7] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *Trans. Graphics*, 31(4):1–10, 2012. 1, 2, 7

[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *Trans. Graphics*, 22(1):83–105, 2003. 1

[9] T. Furuya and R. Ohbuchi. Hashing cross-modal manifold for scalable sketch-based 3D model retrieval. In *Proc. 3DV*, 2014. 1, 2, 7

[10] R. Herzog, D. Mewes, M. Wand, L. Guibas, and H.-P. Seidel. LeSSS: Learned shared semantic spaces for relating multi-modal representations of 3D shapes. *Computer Graphics Forum*, 34(5):141–151, 2015. 2

[11] M. Hueting, M. Ovsjanikov, and N. J. Mitra. CrossLink: joint understanding of image and 3D model collections through shape and camera pose variations. *Trans. Graphics*, 34(6):233, 2015. 1, 2

[12] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proc. CVPR*, 2015. 2

[13] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Proc. ICCV*, 2005. 2

[14] B. Li, Y. Lu, C. Li, et al. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *CVIU*, 131:1 – 27, 2015. 1, 2, 7

[15] M. Maes. Polygonal shape recognition using string-matching techniques. *Pattern Recognition*, 24(5):433–440, 1991. 2

[16] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005. 2

[17] J. Munkres. Algorithms for the assignment and transportation problems. *J. SIAM*, 5(1):32–38, 1957. 4, 6

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

[19] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. *Computer Graphics Forum*, 32(2):459–468, 2013. 2

[20] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as 'shape-DNA' of surfaces and solids. *Comput. Aided Design*, 38(4):342–366, April 2006. 7

[21] E. Rodolà, A. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. CVPR*, 2012. 2

[22] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proc. CVPR*, 2014. 2

[23] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 2016. 2, 6

[24] E. Rodolà, S. Rota Bulò, and D. Cremers. Robust region detection via consensus segmentation of deformable shapes. *Computer Graphics Forum*, 33(5):97–106, 2014. 6, 7

[25] E. Rodolà, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. In *Proc. ICCV*, 2013. 2

[26] F. R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *Proc. ICCV*, 2007. 2

[27] F. R. Schmidt, E. Töppe, and D. Cremers. Efficient planar graph cuts with applications in computer vision. In *Proc. CVPR*, June 2009. 2

[28] T. Schoenemann and D. Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *Trans. PAMI*, 32(7):1153–1164, 2010. 2

[29] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom. Theory Appl.*, 22(1-3):21–74, 2002. 5

[30] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV*, 2015. 1, 2, 7

[31] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP*, pages 1383–1392, 2009. 6

[32] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008. 1

[33] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 2

[34] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3D shapes: A linear programming solution. In *Proc. ICCV*, 2011. 2