

Cross-Collection Map Inference by Intrinsic Alignment of Shape Spaces

Nitzan Shapira¹ Mirela Ben-Chen¹

¹Technion – Israel Institute of Technology

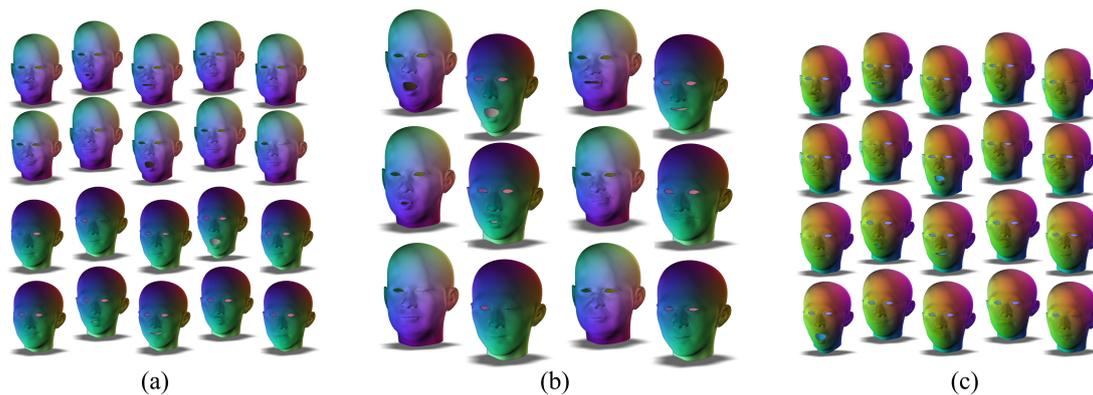


Figure 1: Using our method, we can find correspondences and map functions defined on shapes from two collections, given only maps within the same collection. This map is easy to compute, and provides a meaningful representation of the relation between shapes from different collections, for which a point-to-point map is difficult to obtain and is not well defined. (a) original collections, (b) collection alignment, (c) functional map approximation between all shapes in both collections.

Abstract

Inferring maps between shapes is a long standing problem in geometry processing. The less similar the shapes are, the harder it is to compute a map, or even define criteria to evaluate it. In many cases, shapes appear as part of a collection, e.g. an animation or a series of faces or poses of the same character, where the shapes are similar enough, such that maps within the collection are easy to obtain. Our main observation is that given two collections of shapes whose “shape space” structure is similar, it is possible to find a correspondence between the collections, and then compute a cross-collection map. The cross-map is given as a functional correspondence, and thus it is more appropriate in cases where a bijective point-to-point map is not well defined. Our core idea is to treat each collection as a point-sampling from a low-dimensional shape-space manifold, and use dimensionality reduction techniques to find a low-dimensional Euclidean embedding of this sampling. To measure distances on the shape-space manifold, we use the recently introduced shape differences, which lead to a similar low-dimensional structure of the shape spaces, even if the shapes themselves are quite different. This allows us to use standard affine registration for point-clouds to align the shape-spaces, and then find a functional cross-map using a linear solve. We demonstrate the results of our algorithm on various shape collections and discuss its properties.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: —Shape Analysis.

1. Introduction

Shape correspondence and shape collection analysis are fundamental tasks in geometry processing, at the core of many applications, such as animation, geometric modeling, 3D s-

canning and analysis of medical data, to mention just a few. In many cases, correspondences between shapes are represented as a point-to-point map, taking points on the first shape to points on the second. Such maps are appropriate

when the shapes are similar, e.g. different poses of the same person. For more complicated cases, e.g. the same pose of different characters, it is not always clear which points should be in correspondence. In such cases, it is easier to model a correspondence as a function-to-function map [OBCS*12], taking functions on the first shape to functions on the second. This approach is more flexible and allows us to encode the uncertainty inherent to the solution of an ill-posed problem.

Often, shapes do not appear in isolation, but in collections of related shapes with certain *structure*. For example, a densely sampled animation between two poses has a different structure than a set of unrelated poses. If the collection is homogeneous, i.e., the shapes are similar to each other, it is often feasible to obtain a good correspondence between shapes within the collection, and then leverage this information for further analysis of the collection’s structure.

We address the following problem: given two homogeneous shape collections with intra-collection maps, we seek a correspondence between the shapes (namely which shape in collection A maps to which shape in collection B), as well as the cross-map between all the shapes (see Fig. 1). While this seems harder than finding a map between every pair of shapes in the two collections, we demonstrate that the structure within each collection can be extracted and represented concisely, such that if the structures are similar, the collections can be aligned. This allows us to compute only maps between *corresponding* shapes, which are easier to obtain.

To align two collections of shapes whose geometries are different, we need to define a shape representation such that the similar structure can emerge. Thus, we need a representation which can encode relations like “shape A is to B , like C is to D ”, and allow us to compare *differences of differences* of shapes. Recently, exactly such a representation was introduced [ROA*13], where the difference between two shapes, for which we have a correspondence, can be represented using a *linear operator* which acts on functions on one of the shapes. Comparing two such operators provides a meaningful measure of the shape difference as it can encode not only whether two shapes are different, but also *where* they differ. This brings all the “shape differences” in the same collection to a common ground, and allows us to use standard techniques for Euclidean point clouds to perform the analysis.

We assume that each collection includes shapes which are sampled from a low-dimensional shape space manifold. We therefore use a standard non-linear dimensionality reduction technique [CL06] to find a low-dimensional Euclidean embedding which reproduces as best as possible the original intrinsic distances on the shape space manifold. We repeat this procedure for both collections, and then align the resulting point clouds using an affine registration method [MS10].

Once the collections are aligned, we can use the assumption on the common structure, to specify constraints of the type “ A is to B as C is to D ” as linear constraints on a functional map which maps a shape from the first collection to

a shape in the second. These constraints are enough to recover the cross-collection functional map *without requiring any additional descriptors*. We then extend this map to the rest of the collection, through composition, which puts both collections in correspondence.

1.1. Related Work

The concept of registering two shape collections using dimensionality reduction techniques has not (to the best of our knowledge) been proposed before, nor has the task of finding a cross-collection functional map. Nevertheless, the tasks of finding a map between two shapes and the analysis of shape collections have been widely studied in different contexts.

The task of finding a map between two shapes has been studied from different points of view, under various assumptions, e.g. in [KLF11, OBCS*12]. See [TCL*13] for a recent survey. It is worth noting that our method incorporates much more information, as the whole collection is considered when computing the map.

Considering the entire collection when computing maps *within* the collection has also been addressed previously. In [NBCW*11, HZG*12, HG13, HWL14], the authors add the constraint of global map consistency in order to improve a set of initial maps, using different optimization methods. In addition, structural information from the collection is commonly used for co-segmentation, e.g. in [HKG11, KLM*13, ZCOAM14], among others. In general, we differ from these methods by our assumption of having *two* homogeneous shape collections, with good maps within the collection, instead of having a single heterogeneous collection. This allows us to assume there exists common structure, and use it to align the collections as a whole.

Spectral analysis of shape collections using dimensionality reduction techniques has been used in [ESK07, TSK09, TESK11] for shape de-noising. In [ROA*13], the authors proposed a way to represent the intrinsic shape-space of a collection using *functional operators*. This allows to represent differences between shapes as *comparable objects* instead of merely distances, and then to embed the shape-space in two dimensions, e.g. for visualization. However, the dimensionality reduction was performed using Principal Component Analysis, which assumes the underlying shape-space manifold is *linear*. In many cases, though, such an assumption is too restrictive, and projecting the shape-space on a linear manifold might destroy its structure. We use non-linear dimensionality reduction instead to overcome this difficulty.

Finally, the registration of two collections was studied in the context of video sequences [CI00, CPSK04, LBPW05]. There, however, the problem is better posed, as there are considerably less parameters and more available data.

To conclude, the main difference between previous work



Figure 2: Similar structure in two collections. Although the geometry of the cat (left) differs from the geometry of the lioness (right), the difference between the cats is similar to the difference between the lionesses.

and our approach is that none of the previous approaches attempted to register directly two shape collections. The main obstacle to doing that is finding a common representation such that an alignment is possible. By leveraging the functional approach, namely considering maps and shape differences as linear operators, we can map both shape-spaces to a common space, where alignment is possible. Furthermore, we can use this alignment to compute a functional map, which takes into consideration both collections as a whole.

Algorithm outline. Given two shape collections A and B with internal correspondences, we do the following:

- Choose a *base shape* in each collection, and use it to calculate the shape differences representation (Section 2).
- Reduce the dimensionality of the collections using diffusion maps on the shape differences representation to obtain two point clouds, then use affine registration to align the clouds, and find corresponding shape pairs (Section 3).
- Define *shape-analogies constraints* between corresponding pairs and the base shapes, and obtain an approximated functional map between the base shapes (Section 4).

2. Collection Representation

Given two shape collections, we assume they are sampled from two low-dimensional space-shape manifolds. Our goal is to represent each shape as a point in \mathbb{R}^n , such that Euclidean distances in this representation have some intrinsic geometric meaning, and we can later align the resulting point clouds. If we are given additionally maps between every two shapes *within* each collection, we can use these maps to compute a notion of a *shape difference*. This is a linear operator, which encodes the *variation* induced by the map. Assuming that shapes which are corresponding in two collections undergo a similar transformation under a map from some *base shape*, e.g. the change from a neutral face to a frowning face is similar for two different characters, such a shape difference would provide the intrinsic representation we require.

Assumptions. For our algorithm to be applicable, we must make a few assumptions on the input shape collections. First, we assume that both collections are *homogeneous*, e.g. represent the same character or the same object in different pos-

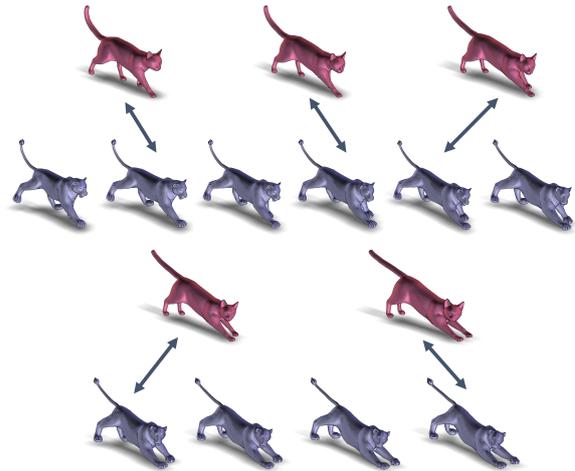


Figure 3: Two similar animations of different shapes, sampled at different rates (30 shapes and 60 shapes). Some shapes in the dense collection are considered as outliers, as they do not have compatible matches in the sparse collection. After removing the outliers, the remaining shapes are matched correctly.

es, and we are given maps between all pairs. Next, we assume that the collections have *similar structure*. Specifically, this means that there is a subset of shapes in both collections which can be paired, such that the differences between them are similar. For example, in Figure 2 we show two shapes (a cat and a lioness) which have different geometries, however the difference between the two cats is similar to the difference between the two lionesses. Furthermore, we assume that for each collection there exists a *base shape* which can be used to evaluate the shape differences to all the other shapes in the collection. Finally, we assume that the low dimensional Euclidean embeddings of the two collections are *not symmetric*, and thus can be reliably registered.

We emphasize that the two collections can differ in their number of shapes. In such cases, some of the shapes in the larger collection are expected to be considered as outliers, enabling the other shapes to be matched successfully. Figure 3 shows an example of the alignment of two collections with a different number of shapes: a dense and a sparse sampling from an animation sequence.

2.1. Functional Maps and Shape Differences

The functional maps framework [OBCS*12] is used to represent maps between surfaces. Namely, given two surfaces M and N , a map $T : N \rightarrow M$ between them induces a map between function spaces $F : L^2(M) \rightarrow L^2(N)$, where $L^2(\cdot)$ is the set of square integrable real-valued functions on a surface. This *functional map* F takes a function $f : M \rightarrow \mathbb{R}$ and maps it to $g : N \rightarrow \mathbb{R}$, and is defined using $g = F(f) = f \circ T$. As explained in [OBCS*12], F is a *linear* transformation be-

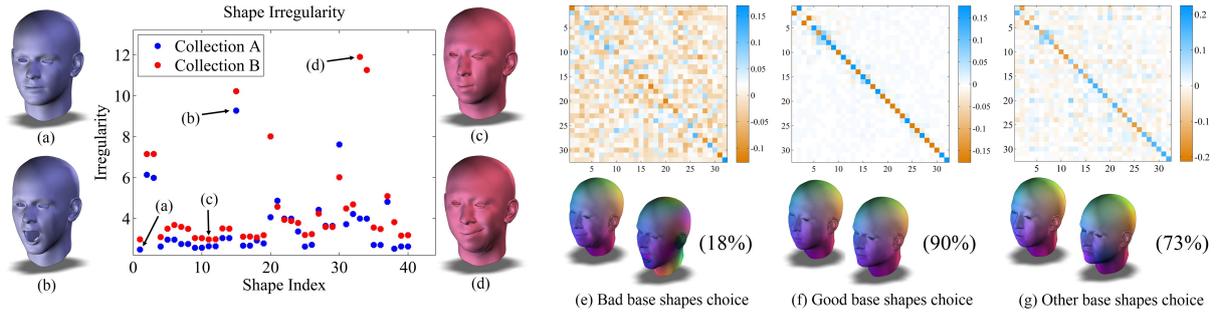


Figure 4: Shape irregularities in two collections, the chosen base shapes, the alignment results (in percentages) and the approximated map. (a, b) Best and worst base shapes in collection A. (c, d) Best and worst base shapes in collection B. (e) A bad pair of base shapes yields a bad map, as can be seen by pushing a coordinate function. (f) Choosing a good pair results in a good map. (g) Choosing non-optimal base shapes (#19 in A and #40 in B) yields sub-optimal, yet reasonable, results.

tween function spaces. Therefore, given a choice of basis, it can be represented as a matrix in the discrete setting.

A *shape difference* [ROA*13] is a linear operator which encodes the disparity between two shapes M and N under a given functional map F . We use the two types of shape differences defined in [ROA*13], one based on the area distortion and another based on the conformal distortion, as induced by the map. Together these two shape differences completely encode the map F .

The *area-based shape difference* is marked $V_{M,N}$ and the *conformal-based shape difference* is marked $R_{M,N}$, where the respective map is usually clear from the context. We emphasize that both $V_{M,N}$ and $R_{M,N}$ are not numbers but *operators*. Note that two shape differences, V_{M,N_1} and V_{M,N_2} , represent linear operators with the same domain and range, $L^2(M)$, even if $N_1 \neq N_2$. Hence, the shape difference between M and N_1 is comparable to the shape difference between M and N_2 , as they are both linear operators acting on functions on M .

In order to represent a functional map discretely, we need to pick a basis for the space of discrete functions on meshes. We choose the eigenvectors of the Laplace-Beltrami operator, as proposed in [ROA*13], as it provides a multi-scale basis which allows to represent smooth functions with a small number of basis functions. As described in [ROA*13], given a functional map F we compute the shape differences using:

$$V_{M,N} = F^\top F, \quad \text{and} \quad R_{M,N} = (D^M)^{-1} F^\top D^N F, \quad (1)$$

where $D^M = \text{diag}(-\{\lambda_i^M\})$, λ_i^M is the i^{th} eigenvalue of the Laplacian of M , and similarly for N . We typically use between 30 and 70 eigenfunctions for the representation.

Finally, the *shape difference distance* (SDD) between two shapes N_1, N_2 , given a base shape M is defined as:

$$d_M(N_1, N_2) = \sqrt{\|V_{MN_1} - V_{MN_2}\|_F^2 + \|R_{MN_1} - R_{MN_2}\|_F^2} \quad (2)$$

2.2. Base Shape Selection

We represent all the shapes in a collection as a shape difference to a chosen *base shape* M . Specifically, each shape is represented as a linear operator which takes functions on M and returns functions on M , therefore, all the shape differences are encoded in the basis of M .

It has been shown in [ROA*13], that if there is cycle consistency in the collection (namely, given any three shapes M, N, K , and functional maps $F_{MN} : L^2(M) \rightarrow L^2(N)$ and similarly for F_{NK}, F_{MK} , we have $F_{MK} = F_{NK}F_{MN}$), then it is possible to transport shape differences between different base shapes by applying a change of basis. Specifically, given a functional map G from M_1 to M_2 we can compute $\tilde{V}_{M_1N_1} = G^{-1}V_{M_2N_1}G$. If additionally G is orthogonal (namely, the map is volume preserving), then we have:

$$V_{M_1N_1} - V_{M_1N_2} = \tilde{V}_{M_2N_1} - \tilde{V}_{M_2N_2} \quad (3)$$

hence the distance between the shape differences as viewed on M_1 are equivalent to those viewed on M_2 . The further G is from being orthogonal, the more influence the choice of base shape will have on the resulting distances, which can potentially be harmful for our registration process.

Therefore, there are two practical problems. First, since we only use the first k eigenvectors of the Laplace-Beltrami operator, we lose cycle consistency, and Equation 3 does not hold anymore. Second, the distances are “distorted” by G , and therefore there is a dependence on the choice of base shape in the two collections. For example, if G has a non-trivial kernel, e.g. if there exists a part on M_1 which does not exist on M_2 , then there is loss of information when changing base shapes. Effectively, the difference between shapes which differ at the missing part cannot be represented using the base shape M_2 .

Therefore, we would like to choose, in both collections, base shapes on which the differences between all the shapes are well represented. To do that, we search for a shape M such that F_{MN} for all the shapes N in the collection is close

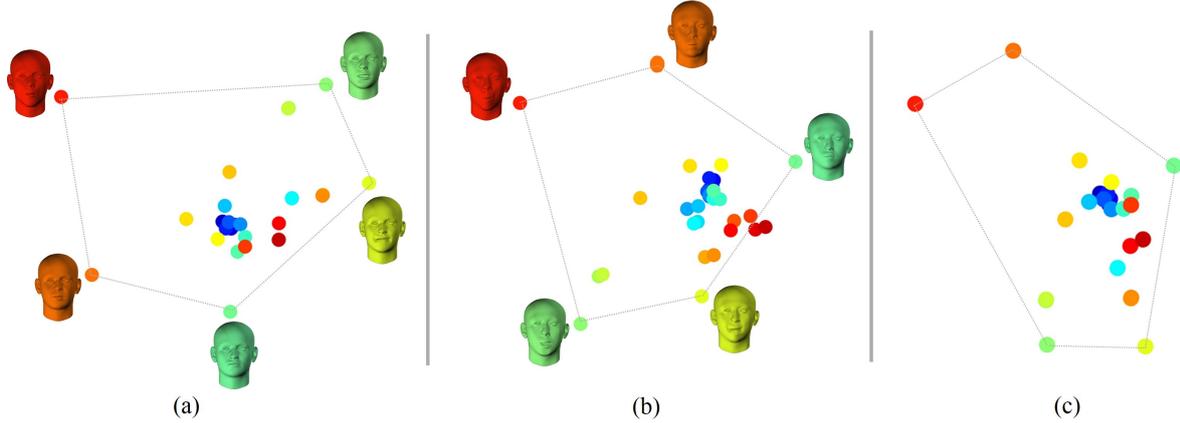


Figure 5: Two collections of 40 blend shapes after reducing their dimensionality using diffusion maps and projecting the resulting 9-dimensional cloud into 2D: (a) collection A cloud, (b) collection B cloud, (c) A to B alignment using an affine transformation with reflection. Even though additional energy is contained in higher dimensions, some of the similarities can be seen in 2D, such as the corresponding shapes (in matching colors) along the edges of the marked polygon.

to an orthogonal matrix. We define this concept as the *shape irregularity*. Specifically, we compute:

$$\arg \min_{M \in A} \sum_{N \in A, N \neq M} \sum_{i=1}^k |\sigma_i - 1|^2 \quad (4)$$

where σ_i is the i -th singular value of the functional map F_{MN} from M to N , and k is the number of basis vectors we are using for the representation.

Figure 4 demonstrates the effect the choice of base shape has on our algorithm as described in Sections 3 and 4. When the worst shapes (b, d) according to the shape irregularity measure are chosen as base shapes, the resulting approximated functional map is not satisfactory (e), as can be seen from the density of the matrix, the errors in transferring a smooth function between the source and target shapes, and the alignment results (only 18% of the shapes were paired correctly). For a good choice of base shapes (a,c), the resulting alignment is 90%, and the functional map is close to the ground truth (f). For a choice of base shapes which is non-optimal, we still get a reasonable, yet sub-optimal, functional map (g). Hence, while our algorithm is dependent on the choice of base shapes, this is done automatically in a manner which optimizes the resulting functional map between the collections. Furthermore, the result is stable under a choice of sub-optimal base shape.

After choosing the base shapes in the collections, we compute the shape difference representation for every shape, and compute the intrinsic distances between the shapes using Equation (2).

3. Collection Alignment

After obtaining the shape differences, we would like to find a correspondence between the two collections. However, the

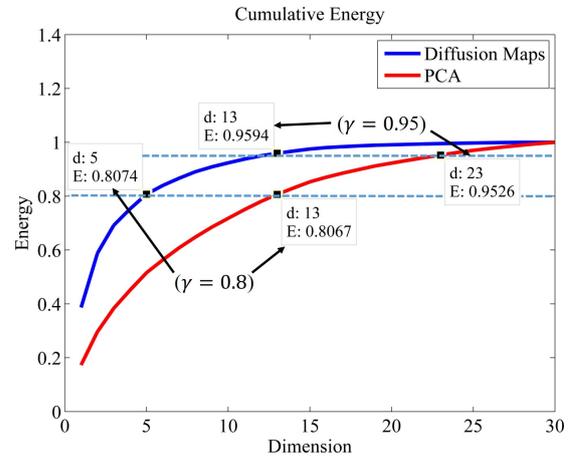


Figure 6: Dimensionality reduction allows us to reveal the intrinsic dimension of a shape collection. Here we can see the cumulative energy (E in Equation 5) of a collection of 40 blend shapes when applying PCA or diffusion maps and the estimated dimension of the data. Choosing a higher value for γ would yield a higher estimated dimension. Since diffusion maps is a non-linear technique, it is capable of recovering the true, non-linear structure of the data (9-dimensional). PCA, on the other hand, assumes a linear structure and therefore identifies an higher intrinsic dimension of 19.

shape differences in different collections cannot be compared directly since different base shapes are used. We therefore assume that each shape collection is a point sampling from a low-dimensional shape space, and use the intrinsic shape difference distances to embed this point cloud in Euclidean space. We then align the resulting point clouds.

Diffusion maps. The “diffusion maps” algorithm [CL06] is a widely known method for non-linear dimensionality re-

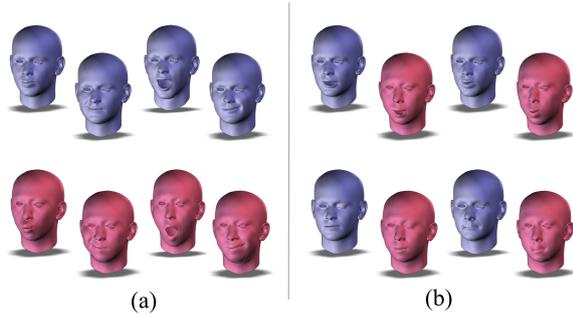


Figure 7: Aligning two collections of 40 blend shapes: (a) identified outliers, (b) correct matches. See also Figure 10.

duction which has been used in many diverse fields, such as computer vision, medical imaging and shape analysis. It has also been used for the analysis of shape collections [T-SK09].

In diffusion maps, we first construct a symmetric weighted graph where each node corresponds to a data point. The weights of the edges represent the similarities between the data points. In our setting, these weights are determined according to the SDD between the shapes, as defined in Equation (2). Then, we calculate the diffusion matrix by normalizing the rows of the matrix of the graph. Taking powers of the diffusion matrix allows us to observe the data at different scales and see the global connectivity of the data set. Figure 5 (left, middle) shows two collections after reducing their dimensionality and projecting the result into 2D.

Coherent point drift. When aligning two point clouds we need to assume some prior on the allowed transformations between them. In general, since our sampling is relatively sparse compared to the dimension (e.g. 40 shapes in dimension 9), we need to assume a somewhat restrictive prior to avoid over-fitting. Assuming the transformation between the point clouds is rigid (i.e. rotation and translation) is too restrictive, as is uniform scaling. Allowing an affine map between the point clouds allows the algorithm to tolerate some error in the SDDs between the shapes (e.g. because the collections are not exactly aligned, or the choice of base shape is not optimal), while still avoiding over-fitting. In addition, we allow reflection, as the diffusion map embedding is only defined up to isometries. We use “coherent point drift” (CPD) for the alignment, which is a state-of-the-art registration algorithm that supports affine registration. Fig. 5 (right) shows the cloud of collection A after aligning it to the cloud of collection B using the resulting affine transformation.

Intrinsic dimension estimation. In many cases, we do not know the intrinsic dimension d in advance. In such cases, we can estimate d from the data. We use a method similar to the one proposed in [SW12]. We set a threshold γ between 0 and 1. Then, we sum the energy along increasing dimensions until the ratio to the total energy exceeds the chosen threshold.

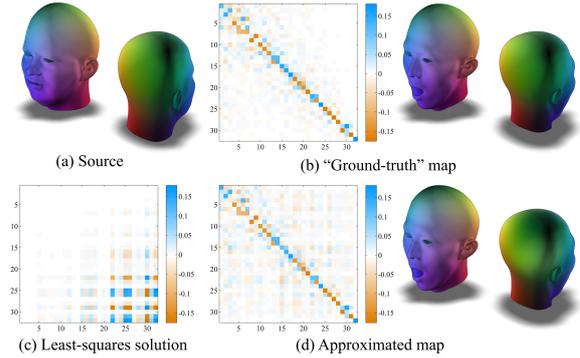


Figure 8: Functional map approximation: (a) source shape in collection A (b) “ground-truth” map to the target shape in collection B , used for comparison and computed from a manually created point-to-point map, (c) least-squares solution G to the map between the base shapes, (d) approximated functional map using iterative refinement and map composition as described in Section 4.

Namely, we choose the minimal d such that:

$$E(d) = \frac{\sum_{u \in Y^d} \|u\|}{\sum_{u \in Y^n} \|u\|} > \gamma \quad (5)$$

where Y^d and Y^n are the result of reducing Y to d or n dimensions (no reduction), respectively. For example, $\gamma = 0.9$ indicates that the chosen dimension consists at least 90% of the total energy of the data set. In our setting, given two shape collections A and B with estimated intrinsic dimensions d_A and d_B , we choose $d = \max\{d_A, d_B\}$. This way, we do not lose information about either collections.

Figure 6 demonstrates the effect of γ on the resulting estimated dimension, as well as the advantage of a non-linear dimensionality reduction technique over a linear one (PCA), using the discussed dimension estimation technique.

Shape pairing. An important observation is that CPD is an asymmetric registration method. Namely, cloud B is registered to cloud A or vice-versa. However, in our setting, we do wish for a symmetric registration. Therefore, we perform the registration in the following way: first, we do not allow a point in the source cloud to match more than one point in the target cloud. If a source point matches more than one target point, we choose the target point which is closer as the match. Second, we match both A to B and B to A , and then choose the direction which yields more matching points. Finally, a point which does not match any other point after the described process is considered to be an *outlier*. A result of this symmetric alignment is presented in Figure 7.

4. Functional Map Inference

Shape analogies constraints. So far we have used the shape differences for computing distances between shapes

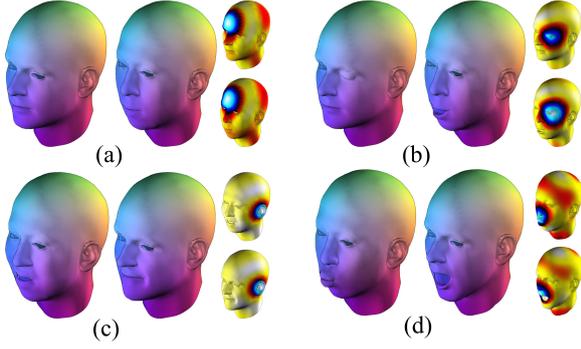


Figure 9: Pushing coordinates functions (left \rightarrow right) and Gaussians (top \rightarrow bottom) through approximated functional maps between collections A and B: (a) A base shape to B base shape, (b) A arbitrary shape to B arbitrary shape, (c) B arbitrary shape to A arbitrary shape, (d) B outlier to A outlier. Notice that a functional map between two outliers is approximated successfully.

within the same collection. However, shape differences encode more information, which can be leveraged for computing a functional map *between the collections*.

Specifically, if we know that two shapes $M_A, N_A \in A$ correspond to two shapes $M_B, N_B \in B$, and we assume that the collections have similar structure, we can additionally assume that the shape differences correspond. Namely that $V_{M_A N_A}$ is similar to $V_{M_B N_B}$, and similarly for R . In the previous section we computed a pairing between shapes in both collections, hence, given such pairs we can pose constraints which enforce this similarity.

Specifically, let $M_i \in A$ and $N_i \in B$ be such that (M_i, N_i) are a corresponding shape pair. Further, let M_A be the base shape on A, and N_A its corresponding shape on in B. Finally let G be the unknown functional map between M_A and N_A . Since we cannot compare $V_{M_A M_i}$ with $V_{N_A N_i}$ directly as they are defined on different function spaces, we apply G on the left and on the right such that all operators take functions on M_A and return functions on N_A . This leads to the following energy:

$$\arg \min_G \sum_{i=1}^K \left(\|GV_{M_A M_i} - V_{N_A N_i}G\|_F^2 + \|GR_{M_A M_i} - R_{N_A N_i}G\|_F^2 \right) \quad (6)$$

where K is the number of matching pairs. In order to minimize this energy, we solve a set of equations which are linear in the elements of G . This is a homogeneous problem and thus it can be solved using SVD. Intuitively, these constraints enforce *shape analogies*, namely, M_A is to M_i as N_A is to N_i . Note, that in [ROA*13] similar constraints were used for finding corresponding shapes given the map G , whereas we solve for the map given the corresponding shapes. Note that these constraints are completely automatic, as the only input they require is the shape pairing between M_i and N_i and between M_A and N_A .

submitted to EUROGRAPHICS 2014.

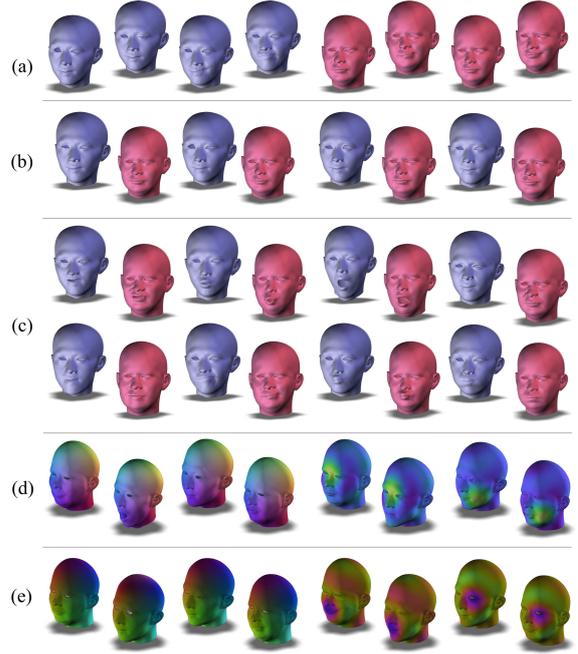


Figure 10: Correspondence and maps between two testers from the FaceWarehouse database [CWZ*13]. Functional maps were approximated correctly for all shapes. (a) identified outliers (17%), (b) wrong matches (19%), (c) correct matches (64%), (d) maps between shapes which are part of a correct match, (e) maps between shapes which are outliers or part of a wrong match.

Iterative refinement. In general, the matrix G which minimizes the energy in Equation (6) does not correspond to a bijection, as we did not enforce any additional constraints beyond the shape analogies. However, we can proceed using a post-processing iterative refinement algorithm, as proposed in [OBCS*12], used to refine a given matrix to make it closer to a point-to-point map. We refer to G_0 as an initial estimate to G and denote the Laplacian eigenvectors matrices of A and B by ϕ_A and ϕ_B . As noted in [OBCS*12], if $G_0 : M \rightarrow N$ is a functional map corresponding to a volume preserving map, then G_0 should be such that each column of $G_0 \phi_A^M$ coincides with some column of ϕ_B^N . We treat ϕ_A and ϕ_B as two point clouds with dimensionality equal to the number of eigenvalues which we used. In addition, for a volume preserving map we also expect the mapping matrix G_0 to be orthonormal, thus we can perform a rigid alignment between ϕ_A and ϕ_B by the following iterative algorithm:

1. For each column v of $G_0 \phi_A^M$ find its closest \tilde{v} in ϕ_B^N .
2. Find the orthonormal G which minimizes $\sum \|Gv - \tilde{v}\|$.
3. Set $G_0 = G$ and iterate for a fixed number of iterations.

This algorithm is effectively ICP in eigenspace, using the minimizer of Equation (6) as the initial solution.

Using this method, we are able to reconstruct an approximated functional map. Note that since this is a homogeneous problem the solution will be up to a constant multiplication (positive or negative). We can ignore the scaling factor – the functional map is a linear operator and we normalize every function pushed through it. However, the sign of this constant *does* affect the resulting G . Therefore, we apply iterative refinement separately for G_0 and $(-G_0)$ and choose the solution which minimizes the noted sum of distances.

We note again that G is a functional map between the *base shape* in A and its corresponding shape in B . In order to get the functional map between two arbitrary shapes $M_i \in A$ and $N_j \in B$, we compose the functional maps to the base shape M_A and to its corresponding shape N_A . We mark $F_{M_i M_A} : L^2(M_i) \rightarrow L^2(M_A)$ and $F_{N_A N_j} : L^2(N_A) \rightarrow L^2(N_j)$ and compose them with G :

$$F_{M_i N_j} = F_{N_A N_j} \cdot G \cdot F_{M_i M_A} \quad (7)$$

Note that using Equation (7) we can compute a functional map between *any* two shapes in the collections, including shapes which were considered outliers or were not matched during the registration step. Figure 8 demonstrates the process of approximating a map between two shapes using the algorithm described above.

In order to calculate the functional maps in the opposite direction (namely, from B to A), we simply produce the corresponding equations by swapping A and B , and proceed as described above. Figure 9 shows approximated maps between various shapes in both collections. The maps are evaluated between shapes which belong to a matching pair, as well as between shapes which were classified as outliers or were a part of wrong match.

5. Experimental Results

We tested our algorithm on different data sets. We present the results and compare them to “ground-truth” results:

- A known point-to-point map between the two collections, if such exists, is used to compute a “ground-truth” functional map for comparison purposes only. We compare our results to this map.
- If the correspondence between the two collections is known (for example, corresponding facial expressions), we demonstrate our registration results with respect to this known correspondence: correct matches, wrong matches, outliers (shapes which were not matched at all) and the corresponding percentages.

Our parameters setting was as follows. We used 32 eigenvalues of the Laplacian for the computation of functional maps and shape differences. The parameters for diffusion maps were $t = 1$, $\sigma = 1$ and $\gamma = 0.9$ for the intrinsic dimension estimation. For CPD we used $\omega = 0.1$ and default values for the other parameters, as described in [MS10].

Limitations. First, our algorithm assumes a similar structure in both collections – if the two given collections do not have a similar structure we will not be able to align them. Second, as explained in [ROA*13], the shape differences are based on externally supplied maps between shapes, and they therefore depend on the quality of these maps. Another requirement is for the collection to contain a minimum amount of shapes (e.g. at least 30). Given a smaller amount of shapes, the collection alignment is not feasible, since the point cloud is too sparse compared to its dimension. In addition, a small collection means that the number of terms in Equation (6) will be smaller, leading to a larger approximation error. Finally, the algorithm depends on several parameters which must be chosen in advance.

Blend shape collections. As presented throughout the paper, we tested our method on two collections of 40 blend shapes each.

- Diffusion maps produced two 9-dimensional point clouds.
- Registering the two collections resulted in 36 correct matching pairs (90%), no wrong matches and 4 outliers in each collection (10%). The results are shown in Figure 7.
- A functional map approximation was recovered and successfully extended to all the shape pairs, including the non-matching shapes in each collection. The results are presented in Figure 9.

FaceWarehouse database. FaceWarehouse [CWZ*13] is a database of 150 individual testers. Each collection consists of 47 different facial expressions and the collections are in correspondence. As mentioned before, we used this known correspondence only for comparison purposes. We tested our algorithm on four different pairs of testers. A summary of all the different pairs is presented in Figure 11.

- *Testers pair 1.* Specific results are presented in Figure 10. Diffusion maps produced two 11-dimensional point clouds. Our algorithm identified correctly 30 matching facial expressions (64%), 8 outliers (17%) and 9 pairs were wrong matches (19%). As can be seen in Figure 10, most pairs which were wrong matches are indeed similar.
- *Testers pair 2:* Diffusion maps produced two 11-dimensional point clouds. Our algorithm identified correctly 28 matching facial expressions (60%), 6 outliers (13%) and 13 pairs were wrong matches (27%).
- *Testers pair 3.* Diffusion maps produced two 11-dimensional point clouds. Our algorithm identified correctly 17 matching facial expressions (36%), 18 outliers (38%) and 12 pairs were wrong matches (26%).
- *Testers pair 4.* Diffusion maps produced two 13-dimensional point clouds. Our algorithm identified correctly 22 matching facial expressions (47%), 11 outliers (23%) and 14 pairs were wrong matches (30%).

We discuss the performance of our algorithm according to the irregularity of its shapes as defined in Section 2.2. Fig-

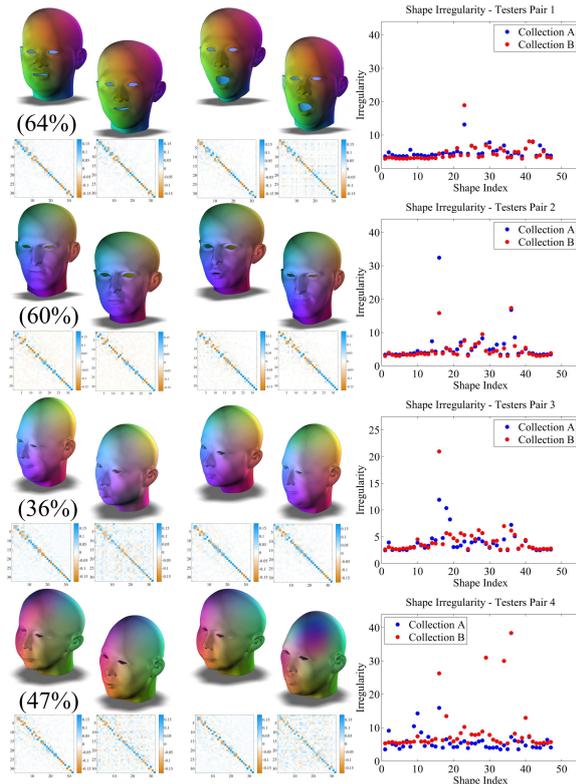


Figure 11: Map approximations between testers pairs 1-4 of the FaceWarehouse database. For each pair we see the percentage of correct matches, two maps between arbitrary shapes (A to B and B to A), the true functional map matrix and the approximated one, and the shape irregularities when the two collections are aligned perfectly. The functional map was approximated successfully for all pairs. Notice testers pair 4 which resulted in a higher approximation error, corresponding to its shape irregularities graphs.

Figure 11 shows the shape irregularities of the different tester pairs. As we can see, when considering two collections after alignment, the shape irregularities graph provides a measure of the similarity between their structures. According to our experiments, we can see that correspondence in the shape irregularities graph predicts successful alignment and functional map approximation.

To summarize, we were able to align the collections of different testers with various percentage rates and functional map approximations were successful for all the tester pairs. Dissimilarities in the shape irregularities graph lead to a higher approximation error, as can be seen in the case of testers pair 4.

Varying collection size. This experiment is intended to measure the effect of the size of the collections on registration and map approximation. We used the blend shape collections presented throughout the article, but took only

a subset of the shapes (the same subset in both collections). The alignments results were as follows:

- Using 20 shapes: 35% correct matches.
- Using 25 shapes: 36% correct matches.
- Using 35 shapes: 63% correct matches.
- Using 40 shapes: 90% correct matches.

The approximated maps corresponding to the size of the collections are presented in Figure 12. As we can see, using small collections leads to poor alignment, since the clouds are very sparse compared to their dimension. As we increase the number of shapes, alignment becomes feasible and thus the approximated map improves.

Small collections with perfect alignment. This experiment is intended to test the approximated map in a rough setting when using small collections with perfect alignment (namely, we provided the matching shape pairs in advance). We used collections of 10 shapes from the Summer and Popović database [SP04]. The resulting map was noisy but was still able to capture some of the information. The results are presented in Figure 13.

6. Conclusion and Future Work

We presented a novel approach for aligning two shape collections and approximating the functional cross-collection map, using only the maps within the collection as prior knowledge. We use shape differences to assess the distances between shapes intrinsically and generate a low-dimensional shape-space embedding, as well as for posing shape analogies constraints for recovering the cross-collection functional map. We demonstrated the effectiveness of our algorithm on various collections, achieving smooth informative functional maps.

Our work provides a glimpse at the possibility of using existing shape analysis tools, such as dimensionality reduction and point registration, for analysing shape-space manifolds. The key to making the leap from shapes to shape spaces is having an intrinsic way to represent differences between shapes, which we achieved by using the shape difference linear operator. It is interesting to consider other functional operators for this task, as well as consider applying other common geometry processing tools directly to the shape-space manifold. Finally, as research progressed from analysing shapes in isolation to analysing collections of shapes, it is possible that the next layer of abstraction is analysing collections of collections. This can serve as a convenient way to model heterogeneous shape collections, simply as a collection of shape-spaces.

Acknowledgements The authors would like to acknowledge FP7-CIG grant 303511 and ISF grant no. 699/12.

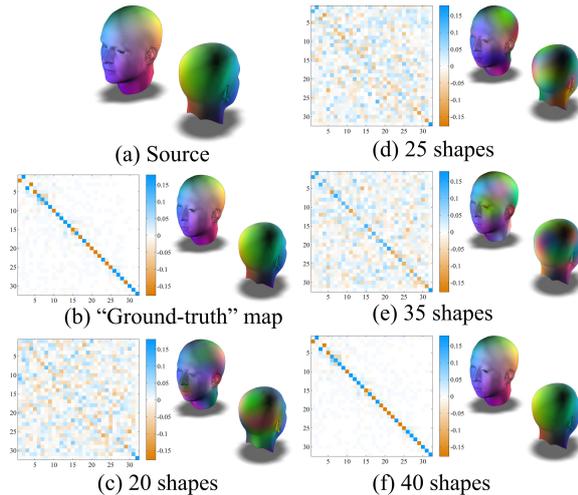


Figure 12: Alignment and map approximations for different subsets of the 40 blend shapes collections. We show the maps between the base shapes (collection A to collection B). (a) source shape in A, (b) “ground-truth” map, (c) using 20 shapes, 35% correct matches, (d) using 25 shapes, 36% correct matches, (e) using 35 shapes, 63% correct matches, (f) using 40 shapes, 90% correct matches.

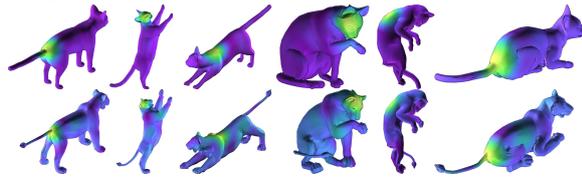


Figure 13: Functional map approximation for small collections (10 shapes) from the Sumner and Popović database [SP04]. Since the collections are small, registering them as point clouds is not feasible. However, given an optimal registration, a rough functional map can still be approximated. This functional map captures a certain amount of the data, but is noisy due to the small size of the collections.

References

- [CI00] CASPI Y., IRANI M.: A step towards sequence-to-sequence alignment. In *Proc. Conf. Computer Vision and Pattern Recognition*. (2000), vol. 2, IEEE, pp. 682–689.
- [CL06] COIFMAN R. R., LAFON S.: Diffusion maps. *Applied and computational harmonic analysis* 21, 1 (2006), 5–30.
- [CPSK04] CARCERONI R. L., PÁDUA F. L., SANTOS G. A., KUTULAKOS K. N.: Linear sequence-to-sequence alignment. In *Proc. Conf. Computer Vision and Pattern Recognition*. (2004), vol. 1, IEEE, pp. 1–746.
- [CWZ*13] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: a 3d facial expression database for visual computing.
- [ESK07] ETYNGIER P., SEGONNE F., KERIVEN R.: Shape priors using manifold learning techniques. In *Proc. Intl. Conf on Computer Vision* (2007), IEEE, pp. 1–8.
- [HG13] HUANG Q.-X., GUIBAS L.: Consistent shape maps via semidefinite programming. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 177–186.
- [HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 125.
- [HWL14] HUANG Q.-X., WANG F., LEONIDAS G.: Functional map networks for analyzing and browsing large shape collections. In *ACM SIGGRAPH*, to appear (2014).
- [HZG*12] HUANG Q.-X., ZHANG G.-X., GAO L., HU S.-M., BUTSCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 167.
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 79.
- [KLM*13] KIM V. G., LI W., MITRA N. J., CHAUDHURI S., DIVERDI S., FUNKHOUSER T.: Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 70.
- [LBPW05] LAPTEV I., BELONGIE S. J., PEREZ P., WILLS J.: Periodic motion detection and segmentation via approximate sequence alignment. In *Proc. Intl. on Conf. Computer Vision*. (2005), vol. 1, IEEE, pp. 816–823.
- [MS10] MYRONENKO A., SONG X.: Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, 12 (2010), 2262–2275.
- [NBCW*11] NGUYEN A., BEN-CHEN M., WELNICKA K., YE Y., GUIBAS L.: An optimization approach to improving collections of shape maps. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1481–1491.
- [OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30.
- [ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 72.
- [SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 399–405.
- [SW12] SINGER A., WU H.-T.: Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics* 65, 8 (2012), 1067–1144.
- [TCL*13] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *Visualization and Computer Graphics, IEEE Transactions on* 19, 7 (2013), 1199–1217.
- [TESK11] THORSTENSEN N., ETYNGIER P., SEGONNE F., KERIVEN R.: Diffusion maps as a framework for shape modeling. *Computer Vision and Image Understanding* 115, 4 (2011), 520–530.
- [TSK09] THORSTENSEN N., SEGONNE F., KERIVEN R.: Pre-image as karcher mean using diffusion maps: Application to shape and image denoising. In *Scale Space and Variational Methods in Computer Vision*. Springer, 2009, pp. 721–732.
- [ZCOAM14] ZHENG Y., COHEN-OR D., AVERKIOU M., MITRA N. J.: Recurring part arrangements in shape collections. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 115–124.