Functional Map Networks for Analyzing and Exploring Large Shape Collections

Qixing Huang Fan Wang Leonidas Guibas

Stanford University

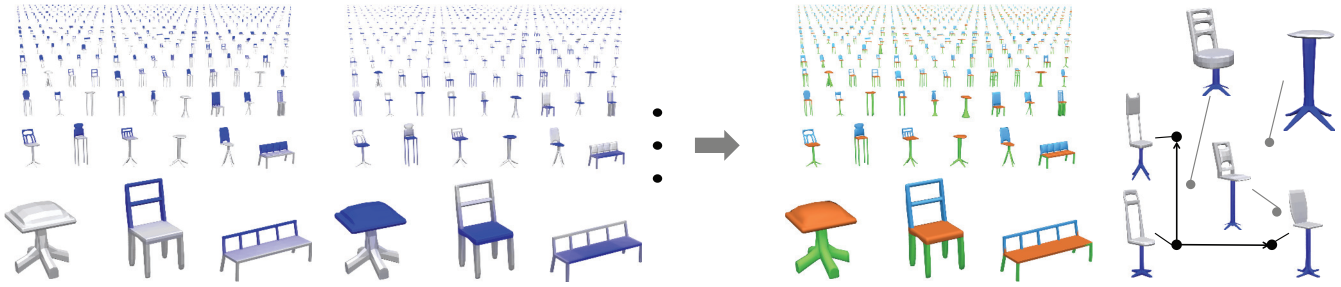


Figure 1: *We introduce a computational framework for constructing functional map networks that can be used to capture structural similarities within heterogeneous shape collections. The shared structure emerges in the network as consistent basis functions across the shape collection. This network representation enables many joint shape analysis tasks, including co-segmentation and shape exploration.*

Abstract

The construction of networks of maps among shapes in a collection enables a variety of applications in data-driven geometry processing. A key task in network construction is to make the maps consistent with each other. This consistency constraint, when properly defined, leads not only to a concise representation of such networks, but more importantly, it serves as a strong regularizer for correcting and improving noisy initial maps computed between pairs of shapes in isolation. Up-to-now, however, the consistency constraint has only been fully formulated for point-based maps or for shape collections that are fully similar.

In this paper, we introduce a framework for computing consistent functional maps within heterogeneous shape collections. In such collections not all shapes share the same structure — different types of shared structure may be present within different (but possibly overlapping) sub-collections. Unlike point-based maps, functional maps can encode similarities at multiple levels of detail (points or parts), and thus are particularly suitable for coping with such diversity within a shape collection. We show how to rigorously formulate the consistency constraint in the functional map setting. The formulation leads to a powerful tool for computing consistent functional maps, and also for discovering shared structures, such as meaningful shape parts. We also show how to adapt the procedure for handling very large-scale shape collections. Experimental results on benchmark datasets show that the proposed framework significantly improves upon state-of-the-art data-driven techniques. We demonstrate the usefulness of the framework in shape co-segmentation and various shape exploration tasks.

CR Categories: I.3.5 [Computing Methodologies]: Computer Graphics—Computational Geometry and Object Modeling;

Keywords: functional maps, shape analysis, shape exploration

Links: DL PDF

1 Introduction

As large on-line shape collections (e.g., Trimble 3D Warehouse) are becoming available, there is a growing need to design effective algorithms for organizing, searching and exploring such collections in order to make them accessible and useful. Towards this goal, an important task is to estimate high-quality maps connecting the shapes. These maps enable us to transport information between the shapes, perform joint shape understanding [Kim et al. 2012; Kim et al. 2013; Huang et al. 2013], and design data-driven solutions to other geometry processing tasks such as shape reconstruction [Nan et al. 2012] and shape modeling [Funkhouser et al. 2004; Kalogerakis et al. 2012].

Yet there are two fundamental challenges when computing maps within large and diverse on-line shape collections. The first challenge is finding a suitable map representation. Most previous work has focused on building point-based maps (either point-to-point or fuzzy) between pairs of shapes. Although point-based representations have proven to be effective on organic shape collections (e.g., mammals and humans), they become less suitable for typical on-line shape repositories of inanimate objects (e.g., furniture or vehicles), which exhibit greater geometric and structural variability.

The second challenge is generating high-quality maps between the shapes. Most existing work has focused on matching pairs of shapes. However, these algorithms are largely designed for matching very similar shapes, and they could easily fail when matching typical shape pairs in heterogeneous shape collections. Recently, there has been a series of works on data-driven shape matching [Nguyen et al. 2011; Huang et al. 2012; Kim et al. 2012; Huang and Guibas 2013], which utilize a cycle-consistency constraint among networks of maps (i.e., the compositions of maps along cycles should approximate the identity map) to improve the maps computed between pairs of shapes in isolation. These algorithms have shown great potential in ameliorating map quality, but all of them are limited to small-scale shape sets and/or fully similar shapes.

To address these two challenges, we introduce in this paper a framework for computing consistent functional maps among het-

ACM Reference Format
Huang, Q., Wang, F., Guibas, L. 2014. Functional Map Networks for Analyzing and Browsing Large Shape Collections. ACM Trans. Graph. 33, 4, Article 36 (July 2014), 11 pages. DOI = 10.1145/2601097.2601111 http://doi.acm.org/10.1145/2601097.2601111.

Copyright Notice
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Copyright © ACM 0730-0301/14/07-ART36 $15.00.
DOI: http://doi.acm.org/10.1145/2601097.2601111

erogeneous shape collections. Unlike point-based maps, functional maps equip each shape with a dual linear functional space and encode the relations between shapes as linear operators between these functional spaces [Ovsjanikov et al. 2012]. Although the functional map representation was initially developed on organic shapes, we argue that it is also quite suitable for man-made objects, as we can easily encode part correspondences as correspondences between the corresponding indicator functions. With appropriate basis functions, functional maps can be represented as small-size matrices. This enables us to apply rich and efficient linear algebra techniques for both encoding and computing functional maps.

To apply the data-driven shape matching idea in the functional map setting, we need to define a proper consistency criterion among functional maps, particularly in the presence of partially similar shapes. The common idea of constraining map compositions along cycles to yield the identity map is only sensible on fully similar collections. To address this fundamental issue, we propose a novel definition of cycle-consistency via constrained matrix factorization. Specifically, if we form a large matrix that stores individual pairwise functional maps in blocks, then that matrix can be factored into the product of a matrix where each row encodes a set of consistent latent basis functions across the input shapes, with another matrix that stores block-wise pseudo-inverses of the first matrix. These latent basis functions provide the vocabulary of common elements shared between the shapes. The partial relations among shapes are naturally encoded in this formulation — each sub-column is filled with a zero function if that latent function does not appear or participate on the corresponding shape. Computationally, since the number of underlying latent basis functions is generally small, this formulation leads to a robust framework for computing consistent functional maps via low-rank matrix recovery techniques. Another prominent feature of this formulation is that it enables us to develop a multi-level algorithm that is scalable to large shape collections.

We have conducted extensive evaluation on various benchmark datasets. Experimental results show that the proposed algorithms significantly outperform existing data-driven shape matching techniques. We also present two applications to demonstrate the usefulness of the resulting functional maps. The first application utilizes the optimized latent basis functions to derive consistent segmentations of the shapes into parts. The key feature of this method is that the segmentations are completely driven by the input shape collection — we do not need to perform initial segmentations or to specify the number of desired segments. In the second application, we show that the quality of the resulting functional maps enables us to apply the shape difference operator [Rustamov et al. 2013] for visually exploring shape collections.

Contributions. In summary, we present three major contributions in this paper.

- The idea of using the functional map representation for matching large heterogeneous man-made shape collections.
- A scalable computational framework that computes consistent functional maps among shapes that are only partially similar.
- Several applications of such consistent functional map networks in shape co-segmentation and interactive shape exploration.

1.1 Related Work

Map representation. Traditional shape matching methods optimize point-to-point maps (continuous or discrete) between shapes. However, these approaches are only suitable for organic shapes or man-made shapes with small geometric variation. They become less effective on heterogeneous shape collections that exhibit large geometric and topological variability.

In [Ovsjanikov et al. 2012], the authors introduced the functional map representation for matching isometric shapes. In contrast, we focus on partially similar shapes and consistent functional maps among many shapes in a large collection. In between point-to-point maps and functional maps, several variations have been proposed, such as fuzzy maps [Kim et al. 2012] or soft maps [Solomon et al. 2012]. They encode probabilistic correspondences between points on the shapes. In fact, fuzzy maps and soft maps can both be considered as special cases of functional maps, by using the hat functional basis defined at each mesh vertex. However, the functional map representation is more general and flexible, as it allows us to choose appropriate bases for encoding the maps.

Map network construction. How to build cycle-consistent maps among a collection of shapes is an active research area in geometry processing. Existing approaches have primarily focused on point-to-point maps [Nguyen et al. 2011; Huang et al. 2012; Kim et al. 2012; Huang and Guibas 2013] and rigid/affine low-dimensional transforms [Wang and Singer 2013]. In [Kovnatsky et al. 2013; Bronstein et al. 2014], the authors proposed to jointly diagonalize graph Laplacians so as to obtain common bases. However, these works still rely on fully similar shapes.

Recently, in a computer vision setting, [Wang et al. 2013; Wang et al. 2014] proposed a regularizer that expresses consistency as map commutation with certain latent basis functions for the image functional spaces. In particular, [Wang et al. 2013] addressed the problem of image co-segmentation using functional maps under the assumption that all images contain an entity from a common class. Although in the full similarity case the latent basis commutation constraint is similar to our cycle-consistency constraint, this formulation only leads to an approximate solution in the partial similarity case that [Wang et al. 2014] addressed. Such an approximate solution cannot be used to develop the multi-level algorithm proposed in this paper. More importantly, this formulation yields non-convex optimization problems, and the proposed alternating strategies of these works may converge to a local minimum. In contrast, the low rank matrix recovery technique introduced in this paper, which is based on convex optimization, delivers much better maps in practice, as we demonstrate later.

Joint shape segmentation. Shape segmentation is one of the fundamental tasks in geometry processing. With the emergence of large shape collections, there is a growing interest in simultaneously segmenting a collection of shapes (co-segmentation), enforcing the consistency of segmentations across the shape collection in order to improve the segmentation of individual shapes [Sidi et al. 2011; Huang et al. 2011; Hu et al. 2012; Wang et al. 2013]. A key task in co-segmentation is to establish good correspondences across shapes, either explicitly or implicitly. The works of [Sidi et al. 2011; Hu et al. 2012] proposed to co-segment shapes via clustering feature descriptors, where correspondences are established between points with similar features. The performance of these approaches heavily depends on various parameters including the set of features chosen and the number of segments.

Beyond clustering-based methods, Huang et al [2011] introduced an optimization approach that jointly estimates segment-level maps between all pairs of shapes with the constraint that the selected segments are similar geometrically and topologically, and form proper segmentations on each shape. However, due to computational complexity, this approach is limited to small-scale data sets only. Wang et al [2013] computed consistent functional maps in the image domain and utilized sharp edges in images to generate the final segmentations. This approach is not applicable in our

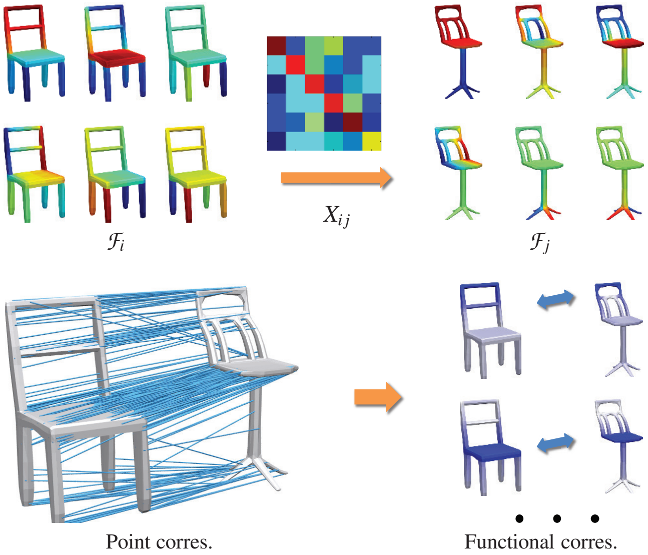


Figure 2: Illustration of the functional map representation. (Top) Functional maps are linear maps between functional vector spaces defined on shapes. (Bottom) Functional maps (shown as functional correspondences) can be derived from point (or part) correspondences.

setting as sharp edges on man-made shapes may not always indicate segment boundaries (e.g., edges along the sides of a chair leg). In contrast, our co-segmentation approach is based on identifying parts by observing variability within the collection. As a consequence, both the part structures and the number of parts per shape are automatically determined by the shape collection itself.

2 Consistent Functional Maps

In this section, we describe how to formulate the cycle-consistency constraint on functional maps among partially similar shapes. We begin with reviewing the functional map representation, with a focus on modifications that handle partially similar shapes. Then we discuss the cycle-consistency constraint and its properties.

2.1 Functional Map Representation

The functional map representation [Ovsjanikov et al. 2012] provides a framework for representing maps between shapes (see Figure 2). Namely, given two shapes S_i and S_j, a functional map $X_{ij} : L^2(S_i) \rightarrow L^2(S_j)$ is a map between spaces of integrable real-valued functions defined on the two shapes. As pointed out in [Ovsjanikov et al. 2012], X_{ij} is a linear map, and can be represented as a finite matrix in the discrete setting. In the remainder of this paper, we will use X_{ij} to represent both the linear map, and the matrix with respect to the default basis functions of each functional space.

The functional map framework is quite flexible because correspondences between points and parts are naturally encoded as correspondences between their indicator functions. It allows us to convert map computation into optimizing matrices and utilize rich numerical tools solve the induced optimization problems. Note that although the framework was originally introduced for organic shapes, the idea nicely extends to man-made shapes, where shape similarities are captured by matching part indicator functions. In the following, we describe the modifications that are necessary for handling partially similar shapes.

Rank-deficient functional maps. When S_i is partially similar to S_j, X_{ij} is no longer a full rank matrix. Take a chair S_i and a stool S_j for example. X_{ij} would map all functions, whose supports lie within the back of chair S_i, to a zero function on the stool S_j. In addition, the inverse of X_{ij} is no-longer unique. In this paper, we choose the inverse of X_{ij} to be the Moore-Penrose pseudo-inverse $X_{ij}^{+} := V\Sigma^{-1}U^T$, where U, Σ, V are given by the singular value decomposition of $X_{ij} := U\Sigma V^T$. This choice inherits several nice properties from the full similarity case, as we show in Section 5 when using the shape difference operator [Rustamov et al. 2013] for shape exploration.

Reduced functional spaces. An important characteristic of the functional representation is that most interesting functions (e.g., indicator functions of points and parts) are well characterized by their projections onto certain low-dimensional Laplacian eigenspaces [Ovsjanikov et al. 2012; Wang et al. 2013]. For shapes that admit manifold structure, we follow [Ovsjanikov et al. 2012] and use the first $K = 30$ eigenvectors of the Laplace-Beltrami (or LB) operator. For other shapes (e.g., those from Trimble Warehouse), we replace the LB basis by the first $K = 30$ eigenvectors of the graph Laplacian of the graph that connects each mesh vertex to its k-nearest neighbors ($k = 12$). In the remainder of this paper, we will approximate each function by its projection in the corresponding reduced functional space and encode it as a vector of coefficients over that basis. Although indicator functions are only approximate in reduced functional spaces, if necessary, they can be rounded into exact indicator functions via appropriate rounding procedures (see Section 4 and 5 for details).

From functional correspondences to functional maps. As described in [Ovsjanikov et al. 2012], the functional map X_{ij} from shape S_i to shape S_j can be computed using a set of functional correspondences written as matrices C_{ij}, D_{ij}, where each pair of corresponding columns of C_{ij} and D_{ij} represents one functional correspondence. These functional correspondences typically come from various shape descriptors, or from indicator functions derived from point and part correspondences estimated between pairs of shapes using off-the-shelf algorithms. Without losing generality, we normalize the corresponding columns of C_{ij} and D_{ij} (i.e., functional correspondences) so that the columns of C_{ij} have the same magnitude and the maximum singular value of C_{ij} is 1.

A natural way to derive the underlying functional map is to solve the following least-square problem:

$$X_{ij}^{\star} = \arg\min_{X_{ij}} \|X_{ij}C_{ij} - D_{ij}\|_F^2, \qquad (1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. However, It turns out that merely using certain functional correspondences between pairs of shapes may not be sufficient. The least square formulation (1) requires that functional correspondences to be clean, which is rare in practice. To address this issue, we propose to study a proper definition of the cycle-consistency constraint in the functional map setting, which utilizes the map network to rectify noisy pairwise functional correspondences.

2.2 Consistency Constraint

We begin by defining the notion of functional map network.

Definition 1. *A functional map network is defined as a connected directed graph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$, whose vertices $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_N\}$ are functional vector spaces, and each edge $(i, j) \in \mathcal{E}$ is decorated with a linear functional map $X_{ij} : \mathcal{F}_i \rightarrow \mathcal{F}_j$. Moreover, the edges come in symmetric pairs, i.e., $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$.*

In the full similarity case, [Wang et al. 2013] proposed to formulate the cycle-consistency constraint as the fact that every function,

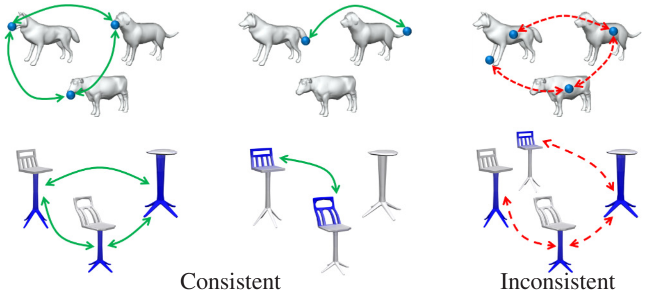


Figure 3: *Illustration of the cycle-consistency constraint in the partial case. (Top) Point-based maps. (Bottom) Functional Maps.*

when transported along a loop, should go back to the original function. In the partial similarity case, the technical challenge is to model the shared sub-spaces between different functional spaces. In the case of point-based maps (i.e., functional spaces are characterized by hat functions on points), we can define the cycle-consistency constraint as the fact that a point, when transferred along a loop, either disappears due to partial similarity or goes back to the original point (see Figure 3 for illustration). To handle general functional maps, we extend this idea to consider latent orthogonal basis functions, as detailed below.

Definition 2. *A functional map network $\mathcal{G}$ is cycle-consistent, if there exists orthogonal basis $B_i = \left(\mathbf{b}_{i1}, \cdots, \mathbf{b}_{idim(\mathcal{F}_i)}\right)$ for each functional space $\mathcal{F}_i$ so that*

$$X_{i_1 i_2}\mathbf{b}_{i_1 j} = \mathbf{b}_{i_2 j'} \text{ or } \mathbf{0} \quad \forall 1 \leq j \leq dim(\mathcal{F}_{i_1}),\ \exists j', \qquad (2)$$

$$X_{i_k i_1}\cdots X_{i_1 i_2}\mathbf{b}_{i_1 j} = \mathbf{b}_{i_1 j} \text{ or } \mathbf{0} \quad \forall \begin{array}{c} 1 \leq j \leq dim(\mathcal{F}_{i_1}), \\ (i_1 \cdots i_k i_1) \in \mathcal{L}(\mathcal{G}), \end{array} \qquad (3)$$

where $\mathcal{L}(\mathcal{G})$ denotes the set of all loops of $\mathcal{G}$.

Here bases $\mathbf{b}_{ij}$, which are expected to be indicator functions of shape parts, are extensions of hat functions; (2) extends the map constraint, where point-wise correspondences are replaced by functional correspondences; (3) is the consistency criterion in the functional setting. Note that the orthogonal constraint on bases $\mathbf{b}_{ij}$ is not mandatory. However, enforcing it strengthens the cycle-consistency constraint for correcting noisy pair-wise maps. Moreover, it also allows us to derive an equivalent but concise formulation (see Appendix A for the proof), which leads to effective optimization algorithms.

Proposition 3. *A functional map network $\mathcal{G}$ is consistent, if and only if there exist row-orthogonal matrices $Y_i = (\mathbf{y}_{i1}, \cdots, \mathbf{y}_{iL})^T \in \mathbb{R}^{L\times dim(\mathcal{F}_i)}, 1 \leq i \leq N$ such that*

$$X_{ij} = Y_j^{+} Y_i, \quad \forall (i,j) \in \mathcal{G}. \qquad (4)$$

It is clear that matrices Y_i also specify maps between pairs of shapes that are not neighbors in the original network:

$$X_{ij} = Y_j^{+} Y_i, \quad \forall (i,j) \notin \mathcal{G}. \qquad (5)$$

If we now let X be a big matrix that encodes the pair-wise map matrices in blocks, then we can express the relation between X and matrices Y_i as

$$X := \begin{pmatrix} X_{11} & \cdots & X_{N1} \\ \vdots & \ddots & \vdots \\ X_{1N} & \cdots & X_{NN} \end{pmatrix} = \begin{pmatrix} Y_1^{+} \\ \vdots \\ Y_N^{+} \end{pmatrix} \begin{pmatrix} Y_1 & \cdots & Y_N \end{pmatrix}. \qquad (6)$$

Discussion. We can interpret Y_i and the factorization in (6) as follows. The rows of $Y = (Y_1, \cdots, Y_N)$ essentially describe a latent functional space $\mathcal{L}$, such that Y_i represents the map from each functional space $\mathcal{F}_i$ to this latent space, and Y_i^{+} characterizes the inverse map. In addition, each row of Y represents a set of *consistent basis functions* across the shape collection. Note that making the rows of Y_i orthogonal is important in the partial similarity case. This forces some of these rows to be zero, characterizing the fact that the corresponding consistent basis does not appear on shape S_i.

Moreover, the latent spaces $\mathcal{L}$ essentially encode distinct shared structures across the shapes. As a consequence, for a collection of related shapes, the dimension of $\mathcal{L}$ is much smaller than the dimension of X, indicating X is generally a low-rank matrix. This low-rank property exhibits two characteristics that are useful for encoding and computing maps. First, it allows us to concisely encode the maps among a shape collection by simply storing Y, which is of much smaller size. Second, it motivates us to adapt robust low-rank matrix recovery techniques to estimate X from noisy initial functional correspondences.

Finally, as we operate within reduced functional spaces, where the original functions are only approximated by their projections, the constraint that the rows of each Y_i are orthogonal to each other is only approximately satisfied. Thus when computing Y_i, we only enforce this orthogonality constraint in a soft manner.

3 Algorithm

In this section, we present algorithms that start from noisy functional correspondences between pairs of shapes and output consistent functional maps encoded as latent basis functions (i.e., $X_{ij} = Y_j^{+} Y_i$). Because of scalability concerns, we introduce two algorithms. The first algorithm (called the single-level algorithm) deals with shape collections of moderate size (e.g., dozens of shapes). The multi-level algorithm is designed to handle large shape collections. It builds a hierarchical structure among the input shapes, so that the single-level algorithm can be performed in a distributed manner.

3.1 Single-Level Map Computation

Input. The input to the single-level algorithm consists of noisy functional correspondences $(C_{ij}, D_{ij}), (i,j) \in \mathcal{G}$ presented between pairs of shapes as specified by a graph $\mathcal{G}$. For example, these initial functional correspondences could come from point or part correspondences estimated using off-the-shelf pair-wise matching algorithms [Huang et al. 2008; Kim et al. 2011; Huang et al. 2011]. As the single-level algorithm deals with shape collections of moderate size, we let the graph $\mathcal{G}$ connect all pairs of shapes. However, one can also choose to only connect shapes with similar shape descriptors [Kim et al. 2012].

Algorithm overview. The technical challenges of computing consistent functional maps are how to handle (i) outliers in the input functional correspondences and (ii) the additional constraints presented in the factorization (6). The key idea of the proposed algorithm is to address these two challenges in two steps (see Figure 4). In the first step, inspired by recent successes of low-rank matrix recovery techniques [Candès et al. 2011; Wang and Singer 2013], we treat the input functional correspondences as noisy measurements of the entries of the map matrix X, and compute X by recovering a low-rank matrix. This robust formulation enables us to compute X only from mostly inliers in the input functional correspondences, and thus the resulting X is close to admitting the factorization (6). In the second step, we proceed to compute the factorization by optimizing a small perturbation of X.

Map computation via low-rank matrix recovery. Adopting the robust principal component analysis (or RPCA) framework [Candès

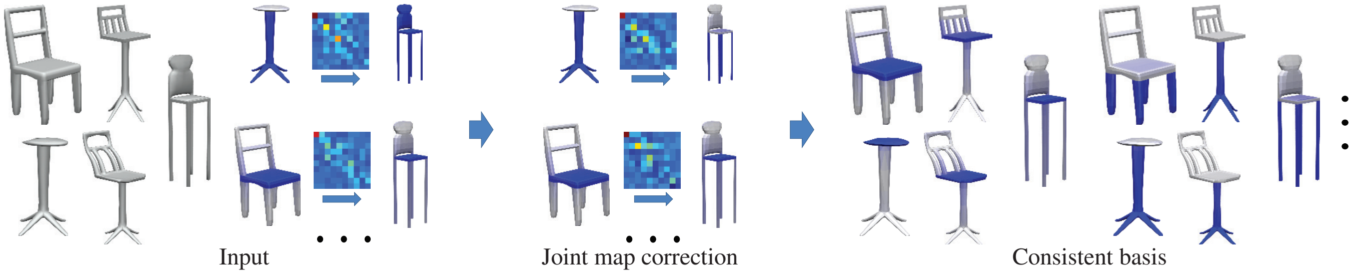


Figure 4: ***Map Computation Pipeline.*** *The single-level map construction algorithm consists of two steps. The first step takes a shape collection and (noisy) initial functional maps between pairs of shapes as input and solves a low-rank matrix recovery problem to correct pair-wise maps. The second step then extracts consistent basis functions from optimized pair-wise maps.*

et al. 2011; Wang and Singer 2013], we formulate the following convex program to compute the map matrix:

$$X^\star = \arg\min_X \|X\|_\star + \lambda \sum_{(i,j)\in\mathcal{G}} \|X_{ij}C_{ij} - D_{ij}\|_{2,1}. \quad (7)$$

The objective function essentially consists of two types of matrix norms. The first component is called the trace-norm defined as $\|X\|_\star = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ are singular values of X. As discussed in depth in [Candès et al. 2011], the trace-norm is a convex proxy for the rank of a matrix. The second component utilizes the $L_{2,1}$ norm, i.e., $\|A\|_{2,1} = \sum_i \|\mathbf{a}_i\|$, where $\mathbf{a}_i$ are columns of matrix A. This $L_{2,1}$ norm, which is a special group-lasso objective [Yuan and Lin 2006], prioritizes the fact that the optimal value of X is insensitive to outlier functional correspondences. An alternative is to use the element-wise L_1 norm. However, we found that the $L_{2,1}$ norm delivers slightly better results. A similar idea is used in [Wang and Singer 2013] for synchronizing rotations.

In [Candès et al. 2011], the authors analyze the robustness of the basic RPCA formulation, where the $L_{2,1}$ norm is replaced by the element-wise L1 norm. They prove that, for a fixed parameter $\lambda = 1/\sqrt{DK}$, where D is the expected vertex degree of graph $\mathcal{G}$[1], RPCA recovers with high probability the ground-truth low-rank matrix if the fraction of entry-wise outliers is below a constant ($20\% - 30\%$ in practice). In our experiments, we found that this robust recovery applies to the general formulation in (7) as well, i.e., with $\lambda = 1/\sqrt{D}$[2], we can recover cycle-consistent maps if the fraction of outlier functional correspondences is below a constant.

For optimization, we employ the alternating direction method of multipliers (or ADMM) described in [Wen et al. 2010]. On a standard PC, it takes about 200 seconds to process a matrix of dimension 900 (30 shapes and 30 functional basis per shape). Please refer to the supplemental material for the algorithmic details.

Latent basis function extraction. In practice, due to (i) the particular noise behavior of the functional correspondences, and (ii) the fact that we do not force the factorization of (6) in (7), the recovered matrix X is typically close to, but does not necessarily admit a factorization as (6). To address this issue, we solve an optimization problem whose objective is to compute a small perturbation of X so that it admits the factorization:

$$\sum_{1\le i,j\le N} \|X^\star_{ij} - Y_j^+ Y_i\|_F^2 + \mu \sum_{i=1}^{N} \sum_{1\le k<l\le L} (\mathbf{y}_{ik}^T \mathbf{y}_{il})^2, \quad (8)$$

where the first term penalizes the perturbation, and the second term ensures the row-orthogonal property of Y_i. In our experiments, we choose $\mu = 100$. We also found that the optimal solution of Y is insensitive to the particular choice of μ as $X^\star$ is usually close to admitting a factorization (6). The initial values and dimensions of Y_i (which also determine the dimension of the latent space) are given by the SVD of $X^\star$, which is exact if $X^\star$ admits the low-rank factorization:

$$(Y_1, \cdots, Y_N) = \Sigma^{\frac{1}{2}} V, \quad X^\star \approx U\Sigma V, \quad (9)$$

where we remove the singular values of $X^\star$ that are smaller than $\sigma_{max}(X^\star)/10$. In other words, U, Σ, V only contain the remaining singular values and corresponding vectors.

As (8) consists of non-linear least squares, we employ the Gauss-Newton method. The key technical step is to derive the derivative of the pseudo-inverse of a matrix with respect to the matrix itself. As the expression is rather complicated, we have placed the technical details in the supplemental material.

3.2 Multi-Level Map Computation

For collections with more than several hundred shapes, performing the map correction step, i.e., solving (7), tends to be intractable. To address this issue, we introduce a scalable approach, which constructs a hierarchical structure among the input shapes. This hierarchical structure allows us to perform the single-level algorithm recursively but only on a smaller number of shapes or latent shapes each time. For simplicity, we describe the algorithm with two levels. The extension to multiple levels is straightforward.

As illustrated in Figure 5, the basic idea is to decompose the input shape collection into sub-collections $\mathcal{S}_1, \cdots, \mathcal{S}_{\overline{N}}$, and then apply the single-level algorithm on each sub-collection $\mathcal{S}_k$ to generate the corresponding latent functional space $\mathcal{L}_k$ and the latent maps $Y_{i,k} : \mathcal{F}_i \rightarrow \mathcal{L}_k$. These latent functional spaces can be considered as the functional space associated with each sub-collection. We then apply the single-level algorithm on these latent spaces to generate a unified latent space $\mathcal{L}$ and the latent maps $\overline{Y}_k : \mathcal{L}_k \rightarrow \mathcal{L}$ at the

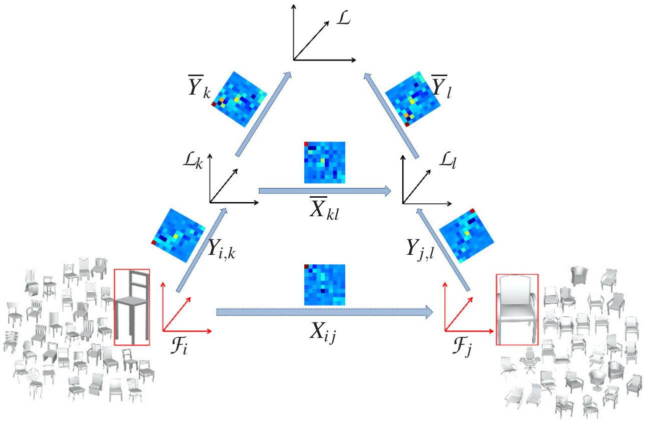


Figure 5: *Map data structure of the multi-level map construction algorithm.*

[1] The analysis assumes a generic random model (see [Candès et al. 2011]).

[2] The new λ reflects the difference between the $L_{2,1}$ norm and the entry-wise L_1 norm.

second level. With this map structure, we can write the latent map from each shape to the unified latent space as

$$Y_i = \overline{Y}_k Y_{i,k} : \mathcal{F}_i \rightarrow \mathcal{L}, \quad \forall S_i \in \mathcal{S}_k, 1 \leq k \leq \overline{N}. \tag{10}$$

The remainder of this section describes the details about how to generate the sub-collections and how to generate initial maps at the second level (in order to apply the single-level algorithm).

Sub-collection generation. Ideally, each sub-collection should only contain very similar shapes. However, thanks to the robustness of the joint map computation procedure, we can afford less than perfectly homogeneous sub-collections, and it is sufficient to use a generic shape descriptor, such as D2 [Osada et al. 2002], to group shapes together. Specifically, we build a k-nearest neighbor graph with respect to the D2 shape descriptor and apply region growing to divide the input shape collection into sub-collections of size around 30. In the experimental evaluation section, we will show that the quality of the resulting maps is insensitive to sub-collection size.

Joint map optimization at the second level. The joint map optimization procedure is mostly similar to that of the single-level algorithm, except for the computation of initial maps between latent spaces. Note that for each pair of shapes $S_i \in \mathcal{S}_k, S_j \in \mathcal{S}_l$, we can write down the function map between them in terms of the functional map $\overline{X}_{kl}$ from $\mathcal{L}_k$ to $\mathcal{L}_l$:

$$X_{ij} = Y_{j,l}^{+} \overline{X}_{kl} Y_{i,k}. \tag{11}$$

This motivates us to compute $\overline{X}_{kl}^{\text{init}}$ using the functional correspondences between shapes in $\mathcal{S}_k$ and shapes in $\mathcal{S}_l$:

$$\overline{X}_{kl}^{\text{init}} = \underset{X_{kl}}{\arg\min} \sum_{(S_i,S_j)\in\mathcal{S}_k\times\mathcal{S}_l} \|Y_{j,l}^{+}\overline{X}_{kl}Y_{i,k}C_{ij} - D_{ij}\|_{2,1}. \tag{12}$$

In our experiments, we employ the CVX software [Grant and Boyd 2011] for optimization.

Since obtaining initial functional correspondences between all pairs of shapes in $\mathcal{S}_k \times \mathcal{S}_l$ is expensive, we only optimize (12) with 60 randomly picked shape pairs for all of our experiments.

After computing initial maps between latent spaces, we feed them into the single-level map optimization algorithm to compute the unified latent space and the associated latent maps.

4 Experimental Evaluation

In this section, we evaluate the proposed data-driven functional map construction pipeline on benchmark datasets and compare it against state-of-the-art data-driven shape matching techniques. In addition, we also show that the proposed approach is insensitive to various parameters (e.g., the number of basis functions).

4.1 Experimental Setup

Datasets. We consider three popular datasets to evaluate the performance of data-driven shape matching techniques (see Figure 6): SHREC07-UnSym [Huang et al. 2012], BCHP13 [Kim et al. 2013] and CoSeg12 [Wang et al. 2012]. These datasets are chosen to include both man-made and organic shape collections with varying in-class variation. Specifically, SHREC07-UnSym contains 5 categories of organic shapes (Armadillo, Fourleg, Hand, Human and Fish), where initial maps are given by blended intrinsic maps [Kim et al. 2011]. BCHP13 contains 4 categories of man-made shapes (Bicycle, Chair, Helicopter and Plane), and the initial maps are derived from pair-wise non-rigid matching [Huang et al. 2008]. These two datasets are provided with manual point-wise correspondences for evaluation. CoSeg12 is a dataset for evaluating shape co-segmentation. We choose two representative collections (Chair (400 shapes) and Vase (300 shapes)) for evaluating the quality of segment-level correspondences. As initial maps are not given in advance, we perform pair-wise shape co-segmentation [Huang et al. 2011] to generate part-wise initial correspondences.

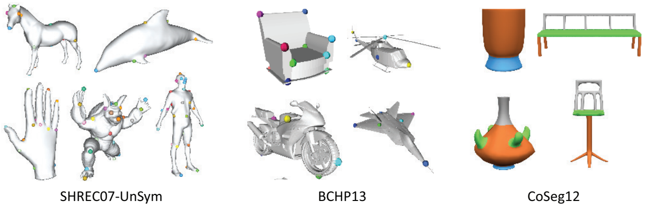


Figure 6: Experimental Evaluation. We perform experimental evaluation on three benchmark datasets: SHREC07-Unsym [Giorgi et al. 2007], BHCP13 [Kim et al. 2013] and CoSeg12 [Wang et al. 2012]. Each benchmark contains multiple shape collections, and each shape collection is provided with either point-wise (SHREC07-Unsym and BHCP13) or segment-level (CoSeg12) ground truth correspondences for evaluation.

As described in Section 2.1, we pre-compute $K = 30$ basis functions on each shape. The initial functional correspondences are generated by placing 60 uniform samples on each shape and converting each point-wise correspondence into a functional correspondence. As a result, the functional correspondences between each pair of shapes is given by a pair of 30 × 60 matrices (C_{ij}, D_{ij}).

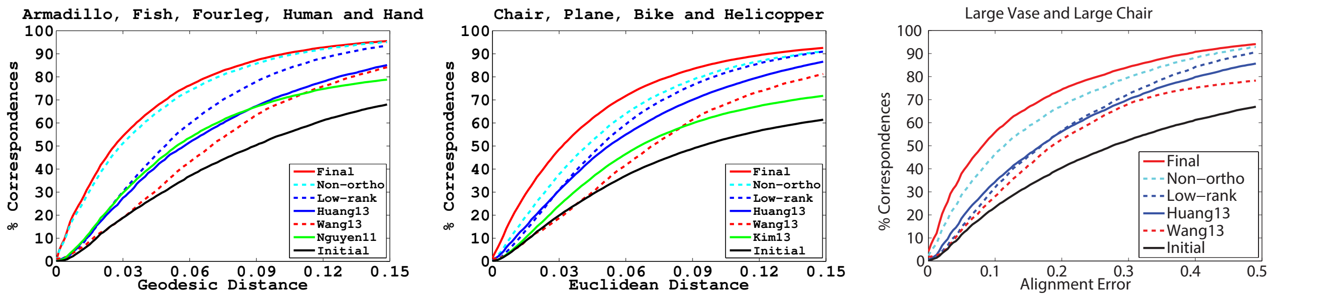
Baseline methods. We compare the proposed approach with existing top performing data-driven shape matching techniques. For organic shapes (i.e. SHREC07-UnSym), we compare with [Nguyen et al. 2011] and [Huang and Guibas 2013], which exhibit the best local and global accuracy, respectively. For man-made shapes (i.e., CoSeg12 and BCHP13), we compare with [Huang et al. 2013] and [Kim et al. 2013], which deliver the best performance on man-made shapes. We also compare the proposed approach with Wang et al. [2013], which formulates cycle-consistency as a regularization term, and performs alternating optimization to optimize the functional maps. To show the effectiveness of enforcing the factorization (6) via (8), we also evaluate the maps obtained from the low-rank matrix recovery step in (7) (referred as Low-rank) and from dropping the orthogonality constraint (referred as Non-ortho).

Evaluation protocols. We apply different evaluation metrics to assess the quality of a shape matching method given point-based or segment-based ground-truth correspondences. In the presence of point-based ground-truth correspondences, we apply the protocol described in [Kim et al. 2011], which plots the percentage of correspondences whose deviations from ground-truth correspondences fall within a varying threshold. The functional maps are converted to point-to-point maps using the procedure in [Ovsjanikov et al. 2012] (i.e., picking the target point with the highest score).

For evaluation with segment-based correspondences, we propose to evaluate the inner product of segment indicator functions which is applicable to both point-based and functional maps. Consider a map X_{ij} from S_i and S_j and a ground-truth segment correspondence $(s, s') \subset S_i \times S_j$. Suppose X_{ij} maps segment $s \subset S_i$ to segment $s'' \subset S_j$, we define the alignment error between map X_{ij} and segment correspondences (s, s') as

$$e(X_{ij}, (s, s')) = 1 - \frac{\langle f_{s'}, f_{s''}\rangle}{\langle f_{s'}, f_{s'}\rangle^{\frac{1}{2}}\langle f_{s''}, f_{s''}\rangle^{\frac{1}{2}}}, \tag{13}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product on shapes, and f_s denotes the indicator function for a segment s on the corresponding mesh. For function maps, we compute the indicator function $\overline{f}_{s''} = X_{ij}\overline{f}_s$,



(a) SHREC07-UnSym (b) BCHP13 (c) CoSeg12

Figure 7: ***Benchmark Evaluation.*** *Comparison between the proposed algorithm and baseline algorithms on each benchmark dataset. A higher curve means better performance. Our approach is seen to be superior to state-of-the-art data-driven shape matching algorithms in terms of both local and global accuracy. Here "Low-rank" corresponds to the functional maps obtained by performing low-rank matrix recovery, and "Non-ortho" corresponds to the functional maps obtained by dropping the orthogonality term.*

where $\overline{f}_s$ ($\overline{f}_{s''}$) is the projection of f_s ($f_{s''}$) in the reduced space $\mathcal{F}_i$ ($\mathcal{F}_j$). The evaluation is shown by plotting the percentage of correspondences whose alignment errors are below a varying threshold.

4.2 Analysis of Results

Figure 7 shows the comparison between our approach and baseline algorithms. The overall performance of our approach is significantly better than that of existing data-driven shape matching techniques. The results are analyzed in detail below.

Global behavior. The proposed approach can significantly improve the map quality. On both SHREC07-UnSym and HCBP13, it approximately recovers all ground-truth correspondences. Moreover, we find that the global behavior of the proposed approach is governed by the low-rank matrix recovery step. This shows the robustness of the formulation. It is important to note that the proposed approach is slightly better than [Huang and Guibas 2013], which is also robust to noisy input maps. This is because the formulation in [Huang and Guibas 2013] is sample-based, and is thus affected by the consistency of samples across different shapes.

Local behavior. Thanks to the continuous nature of functional maps, our approach is significantly better than existing approaches in terms of local accuracy. Even without the local optimization step, the local accuracy of the proposed approach is comparable to the best existing techniques. This behavior justifies the advantage of utilizing functional maps for matching heterogeneous shape collections.

Segment-based maps. As shown in Figure 7(c), utilizing functional maps to compute and encode segment-level correspondences exhibits clear advantage over previous techniques. For example, 93.2% segment correspondences computed by our approach have alignment error less than 0.5. While with the same threshold, the best previous technique [Huang et al. 2013] only achieved 83.3%. This ability of capturing consistent segment level correspondences serves as a strong foundation for co-segmentation, which we demonstrate in the next section.

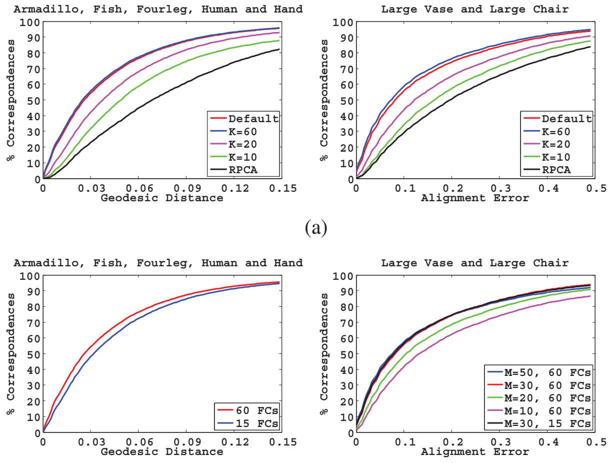
Comparison with Wang el.al [2013]. As illustrated in Figure 7, although the methods of Wang el.al [2013] improve from the initial maps, the quality of the maps computed using the proposed low-rank matrix recovery technique is significantly better than these methods. This is due to the fact these alternating formulations do not enforce exact cycle-consistency, and the alternating strategy tends to arrive at local minima.

4.3 Discussion

Aligning functional correspondences is crucial. An alternative strategy to compute X is to first compute an initial functional map between each pair of shapes using (1), and then apply the robust principal component analysis (or RPCA) formulation [Candès et al. 2011]. As shown in Figure 8(a), the formulation of aligning functional correspondences via the $L_{2,1}$-norm is crucial, since outlier functional correspondences are isolated in the objective function.

The orthogonality constraint improves map quality. As shown in Figure 8, enforcing the orthogonality constraint among rows of each Y_i improves the map quality, particularly on man-made shape collections BCHP13 and CoSeg12. As discussed in Section 2.2, this is due to the fact that the orthogonality constraint helps identify partial similarities across shapes.

Insensitivity to basis dimension. Figure 8(a) shows the results of using different number of bases, i.e., $K = 10, 20, 60$. We can see that increasing the basis dimension from 10 to 20 can substantially improve the performance of the algorithm, while the improvement



(a)

(b)

Figure 8: ***Alternative Formulations and Parameters.*** *This figure shows the performance of using alternative formulations and parameters on each dataset. RPCA: Applying RPCA[Candès et al. 2011] on the initial pair-wise functional maps estimated from noisy functional correspondences (i.e, using (1)). K: the dimension of the functional space on each shape. M: group size at the first level for the multi-level construction approach (only applied on CoSeg12).*

gained from using 60 basis functions is not that salient. This is due to the nature of the Laplacian basis, i.e., the low-frequency signal of a function can be captured by its projection in the space spanned by the first few Laplacian basis functions. On the other hand, one does not need a lot of basis functions to obtain an accurate approximation.

Sparse initial functional correspondences are sufficient. As the low-rank recovery formulation implicitly propagates maps among the shape collection, we find that using sparse initial functional correspondences is sufficient. As shown in Figure 8(b), the performance of using 15 functional correspondences per pair of shapes is similar to that of using 60 functional correspondences. This agrees with the common knowledge in compressive sensing [Candès et al. 2006] and low-rank matrix recovery [Candès et al. 2011] — sparse or low-rank signals can be recovered from sparse samples.

Insensitivity to sub-group size. To test the sensitivity of the sub-group size N_s in the multi-level construction algorithm, we have measured the performance on CoSeg12 under four different values $N_s = 10, 20, 30, 50$ (See Figure 8(b)). We can see that the quality of the resulting maps is insensitive to the sub-group size. The subtle difference is due to the limitation in the current shape grouping step — as D2 is not discriminative enough, a small sub-group may contain very different shapes so that the noise level in the pair-wise initial maps exceeds what can be handled in the data-driven stage.

Running time. In terms of running time, the bottleneck of the proposed consistent functional map recovery technique is the low-rank matrix recovery. For a shape collection with 30 shapes and 30 basis functions per-shape, the low-rank matrix recovery step takes 196 seconds on a single 3.2G HZ core. The refinement step takes 64 seconds. For larger shape collections, the time complexity of the low-rank matrix recovery step roughly scales in the order of $O(M^3)$, where M is the dimension X. The cubic order comes from the complexity of performing a SVD at each iteration of the ADMM. In the future, we plan to explore various ways to improve the efficiency of the ADMM solver such as performing partial SVDs to explore the low-rank nature of X (c.f. [Wen et al. 2010]).

5 Applications

5.1 Co-Segmentation

A straightforward application of computing consistent basis functions is shape co-segmentation. Essentially we can group consistent basis functions into consistent functional subspaces, where each subspace corresponds to a set of consistent parts across shapes.

Algorithm. Intuitively, we should group two latent basis functions (or groups of latent basis functions) if they co-exist on the same set of shapes. We thus define a merging score between two groups of latent functions $I, J \subset \{1, \cdots, L\}$ (specified by basis indices) as

$$score(I,J) = n(I \cup J)/\max(n(I), n(J)), \tag{14}$$

where $n(I)$ denotes the number of shapes that contain I. It is clear that if I and J always co-exist, then $score(I,J) = 1$. On the other hand, if $score(I,J)$ is small, then I and J correspond to different parts. To compute consistent subspaces, we simply start from individual basis functions and iteratively group the two subsets I and J with the largest value of $score(I,J)$ until this value is smaller than 0.9.

After determining the set of consistent functional subspaces, we proceed to generate consistent segmentations. This is done by converting the induced functional subspaces associated with each shape into corresponding segments. As the functional subspaces are consistent, we perform this conversion on each shape independently by solving a standard multi-cut problem. Specifically, let $\mathcal{F}_{i,1}, \cdots, \mathcal{F}_{i,L_i}$ be the functional subspaces associated with shape S_i, and introduce a random variable $z_p \in 1, \cdots, L_i$ for each face p on S_i, where $z_p = l$ if and only if p belongs to the segment associated with $\mathcal{F}_{i,l}$. We then optimize these random variables by minimizing a combination of unary and pair-wise potentials:

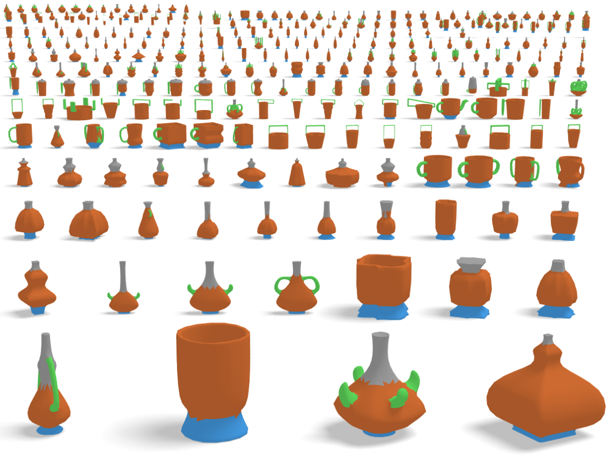


Figure 9: Co-segmentation result on the Vase dataset from [Wang et al. 2012].

$$\underset{z_p, p \in S_i}{\text{minimize}} \sum_{p \in S_i} \phi(z_p) + \sum_{p' \in \mathcal{A}(p)} \phi(z_p, z_{p'}) \tag{15}$$

where $\mathcal{A}(p)$ denotes the adjacent faces of p. Here each unary term measures the projection distance between the corresponding face indicator function and subspaces:

$$\phi(z_p = l) = \|f_p - f_p^{\mathcal{F}_{i,l}}\|, \tag{16}$$

where f_p denotes the face indicator function of p, and $f_p^{\mathcal{F}_{i,l}}$ denotes its projection on $\mathcal{F}_{i,1}$. The pair-wise term is similar to the ones used by other shape segmentation techniques (e.g. [Sidi et al. 2011]), which prefers that segment boundaries fall on sharp edges:

$$\phi(z_p, z_{p'}) = -\log(\theta(p,p'))\delta(z_p - z_{p'}), \tag{17}$$

where $\theta(p,p')$ denotes the angle between p and p'. We apply tree-reweighted BP [Szeliski et al. 2008] to solve this MRF problem.

Results. Figure 1 and Figure 9 show the co-segmentation results on two large shape collections Chair and Vase from [Wang et al. 2012]. As we are able to obtain high-quality functional maps across each shape collection, the resulting segmentations are consistent and agree with the underlying part structures. Table 1 compares the segmentation accuracy of various approaches. We can see that the proposed approach is superior to state-of-the-art techniques [Sidi et al. 2011] and [Kim et al. 2013] on these two datasets. This shows the advantage of using consistent functional bases across the shape collection to derive consistent segmentations.

Note that the success of the proposed approach relies on the fact that (i) the shape collection is large so that data-driven shape matching helps establish high-quality functional maps across the shape collection, and (ii) the variation in the shape collection is significant, so that segments can be identified because they are missing on certain shapes. On the other hand, the performance of our approach drops on homogenous shape collections. This is expected, since our approach would return each shape as a single segment if all input shapes were the same. In other words, our approach is less competitive on shape collections of small size and variation (see Table 1).

	Sidi	Hu	Kim-Auto	Kim-Man	FMap
Chairs(400)	80.2	N/A	91.2	96.9	**98.1**
Vase (300)	69.9	N/A	85.6	81.2	**94.3**
Lamps	94.3	90.7	95.2	**97.6**	96.1
Candelabra	84.4	**93.9**	82.4	87.9	93.1
Chairs	84.8	89.6	97.6	**98.4**	93.9
Vase	87.4	80.2	81.3	83.2	**88.5**
Guitars	97.2	**98.0**	88.5	96.5	93.4
Goblets	98.2	**99.2**	97.6	98.1	91.2
FourLegged	77.3	**88.7**	86.9	87.1	74.8

Table 1: Segmentation Accuracy. *Each entry records fraction of area that was labeled correctly by a segmentation technique. We compare to the consistent segmentation techniques by Sidi et al. [2011], Hu et al. [2012] and Kim et al. [2013] (With both automatic initialization and manual initialization).*

5.2 Exploring Shape Collections

Another application of constructing a consistent functional map network is to use the shape difference operator [Rustamov et al. 2013] for shape exploration. In [Rustamov et al. 2013], the authors demonstrated a few browsing operations on fully similar organic shapes. We show that with the functional maps generated by the proposed algorithms, we can perform similar operations on shape collections that exhibit large variation and partial similarity.

Shape Differences in Map Networks. The shape difference operator [Rustamov et al. 2013] provides a flexible framework for comparing shapes with respect to different measures and at different scales. Specifically, let $X_{12} : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ be a functional map from shape S_1 and S_2. Suppose matrices H_1 and H_2 encode the inner products of functions with respect to a given surface measurement on S_1 and S_2, respectively. The shape difference operator (itself a self-map on S_1) associated with X_{12} is defined as

$$D_{1,2} := H_1^{-1} X_{12}^T H_2 X_{12}. \tag{18}$$

Note that although [Rustamov et al. 2013] only considers isomorphic functional maps, it can be shown that (18) is also proper on rank-deficient functional maps (see Appendix B for details).

When computing the shape difference operators between all pairs of shapes in a shape collection, it is expensive to apply (18) due to the cost in either computing or storing X_{ij}. However, it turns out that in a consistent functional map network $\mathcal{G}$, the shape difference operators can be easily computed using the embedding maps and their inverses $Y_i, Y_i^+, 1 \leq i \leq N$. Specifically, let H_i specify the inner product of a given measure on shape S_i, we can write the shape difference operator between S_j and S_i as

$$D_{i,j} = H_i^{-1} X_{ij}^T H_j X_{ij} = \underbrace{(H_i^{-1} Y_i^T)}_{H_{\mathcal{L},i}} \underbrace{(Y_j^{+^T} H_j Y_j^+)}_{H_{\mathcal{LL},j}} Y_i. \tag{19}$$

In particular, if we are interested in the shape difference between S_j and S_i within a region of interest (ROI) (specified by the indicator function g) on S_i , then the corresponding difference vector is

$$D_{i,j} g = H_{\mathcal{L},i} H_{\mathcal{LL},j} (Y_i g). \tag{20}$$

As the vector $Y_i g$ is independent of shape S_j, computing $D_{i,j} g$ for each shape S_j only needs two matrix-vector multiplications. This enables us to apply the shape difference operator to explore large shape collections as demonstrated below.

Shape Exploration. The dataset we use is a large-scale heterogeneous chair dataset (8401 shapes) that merges all chair shapes in [Huang et al. 2013] and [Kim et al. 2013]. For map computation, we applied a three-level procedure, where each group at each level (except the top level) has 30 shapes or 30 latent functional spaces (each of which corresponds to a group of shapes). The computation was parallelized on a machine with 32G Memory and 12-core 3.2G HZ CPU. The total running time was 32 hours and 42 minutes. Note that the shape difference operator is defined with respect to a measure on each shape. Unless otherwise specified, we assume the default measure is the area measure.

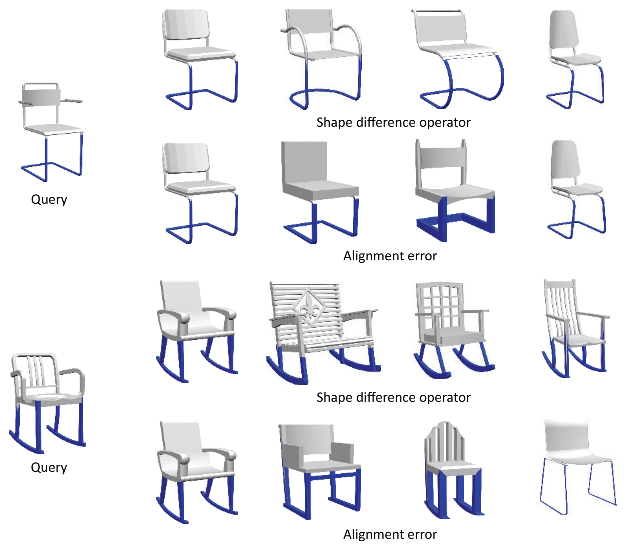


Figure 10: Shape Retrieval. *This figure shows the shape retrieval results on a rocking chair and a cantilever chair with user specified ROIs. We can see that the shape difference operator returns more meaningful results than the approach based on rigid alignment error [Kim et al. 2012].*

As a sanity check, we first show that the shape difference operator can retrieve similar shapes. The input to each retrieval task is a ROI on a query shape S_i. With $\mathbf{g}$ we denote the corresponding projected indicator function. The retrieval procedure then computes for each other shape S_j the corresponding shape difference vector $D_{ij}\mathbf{g}$ using (20) and sort the shapes based on distances $\|D_{i,j}\mathbf{g} - \mathbf{g}\|$. Figure 10 shows the retrieval results of two query shapes: a cantilever chair and a rocking chair. It is clear that the retrieved shapes all belong to the same class. We have compared to the retrieval technique described in [Kim et al. 2012], which measures the extrinsic distance between rigidly aligned ROIs. We can see that the extrinsic distance based approach is sensitive to deformations in shapes, and may return shapes from different classes.

The second example demonstrates that we can extrapolate/interpolate the difference between the corresponding regions of a shape pair. As described in [Rustamov et al. 2013], the basic idea is to look for shapes S_j whose shape difference vectors $D_{i,j}\mathbf{g}$ stay close to the line between the $\mathbf{g}$ and $D_{i,i'}\mathbf{g}$. Figure 11 illustrates the results of interpolating/extrapolating a Windsor chair with 4 beams and another Windsor chair with 6 beams. Note that to make interpolation/extrapolation insensitive to the radius of each beam, we utilize the mean curvature function when defining the inner product, i.e., $\langle f, g \rangle = \int f(x) g(x) |H(x)| dx$, where $H(x)$ is the mean curvature at point x. As shown in Figure 11, we can see that shapes S_j, whose shape difference vectors $D_{ij}\mathbf{g}$ are close to the middle point of $(\mathbf{g} + D_{i,i'}\mathbf{g})/2$, are Windsor chairs of 5 beams. Similarly, we can obtain Windsor chairs with around 8 beams, by extracting shape difference vectors that are close to $2D_{i,i'}\mathbf{g} - \mathbf{g}$.

The third example shows a new application of the shape difference operator in classifying shapes with respect to a ROI. The idea is to define a feature descriptor for each shape by combining its shape difference vectors from a number of landmark shapes. In

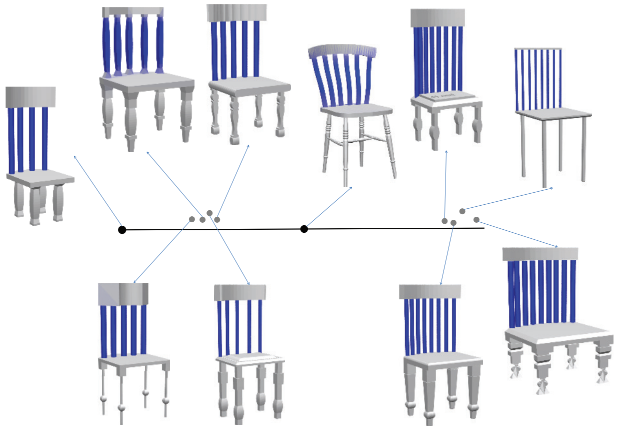


Figure 11: Shape Interpolation/Extrapolation*. Black dots represent the source (4 beams) and target (6 beams) shapes, and gray dots represent interpolated and extrapolated results. The location of each dot is parameterized by its distance to the line that connects the source and target shapes as well as its projection on the line.*

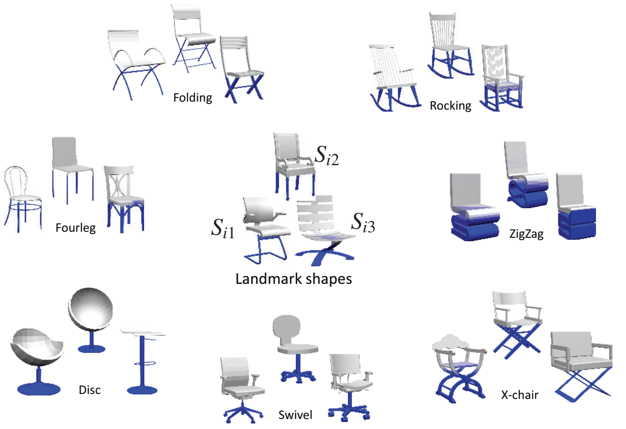


Figure 12: Shape Classification*. Classifying shapes based on descriptors that combine shape difference vectors to landmark shapes. We use the standard K-means for clustering. The resulting clusters nicely capture the underlying classes.*

the example shown in Figure 12, we pick three landmark shapes S_{i1}, S_{i2}, S_{i3} with different chair legs. The feature vector for shape S_j is then given by $(D_{i_1,j}g, D_{i_2,j}g, D_{i_3,j}g)$. Figure 12 shows the K-means classification results on 21 shapes from seven classes, i.e, Swivel, Fourleg, Rocking, Folding, Zigzag, Disc and X-chair. We can see that the underlying classes are nicely captured by the clusters.

6 Conclusions and Future Work

In this paper, we have introduced an algorithm for computing cycle-consistent functional maps among shape collections that exhibit large variation and partial similarity. The framework can be applied at multiple levels, making it scalable for large collections. Experimental results on benchmark datasets show the advantages of the functional representation over point-based representations for capturing various types of similarities across diverse shape sets. The usefulness of the presented approach is demonstrated in applications of co-segmentation and shape collection exploration.

There are many future directions to explore. For example, it would be interesting to study additional applications that are enabled by consistent functional map networks, e.g., fine-grained categorization and shape recognition, all of which require good correspondences across shapes. As the functional map representation is quite flexible, another direction is to apply it on other types of data, such as 3D scenes, medical and biological data, and even between different modalities of data (e.g., images and shapes). The challenges are how to construct meaningful functional spaces and how to compute the initial functional correspondences.

7 Acknowledgement

The authors would like to thank the reviewers for their valuable comments. This work is supported by NSF grants CNS 0832820, CCF 1011228, and DMS 1228304, AFOSR grant FA9550-12-1-0372, and a Google research award.

References

BRONSTEIN, M. M., GLASHOFF, K., AND LORING, T. A. 2014. Making Laplacians commute: multimodal spectral geometry using closest commuting operators. SIAM Journal on Imaging Sciences (SIIS), submitted.

CANDÈS, E. J., ROMBERG, J., AND TAO, T. 2006. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory 52*, 2, 489–509.

CANDÈS, E. J., LI, X., MA, Y., AND WRIGHT, J. 2011. Robust principal component analysis? *J. ACM 58*, 3 (June), 11:1–11:37.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3 (Aug.), 652–663.

GIORGI, D., BIASOTTI, S., AND PARABOSCHI, L., 2007. Shape retrieval contest 2007: Watertight models track.

GRANT, M., AND BOYD, S., 2011. CVX: Matlab software for disciplined convex programming. `http://www.stanford.edu/~boyd/cvx/`.

HU, R., FAN, L., AND LIU, L. 2012. Co-Segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum 31*, 5 (Aug.), 1703–1713.

HUANG, Q., AND GUIBAS, L. 2013. Consistent shape maps via semidefinite programming. *Computer Graphics Forum (SGP) 32*, 5, 177–186.

HUANG, Q., ADAMS, B., WICKE, M., AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. In *Eurographics Symposium on Geometry Processing '08*, 1449–1457.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation using linear programming. *ACM Trans. Graph. 30*, 6 (Dec.), 125:1–125:12.

HUANG, Q., ZHANG, G.-X., GAO, L., HU, S.-M., BUTSCHER, A., AND GUIBAS, L. 2012. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph. 31*, 6 (Nov.), 167:1–167:11.

HUANG, Q., SU, H., AND GUIBAS, L. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. Graph. 32*, 6 (Nov.), 190:1–190:10.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph. 31*, 4 (July), 55:1–55:11.

KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. 2011. Blended intrinsic maps. *ACM Trans. Graph. 30*, 4 (Aug.), 79:1–79:12.

KIM, V. G., LI, W., MITRA, N. J., DIVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. Graph. 31*, 4 (July), 54:1–54:11.

KIM, V. G., LI, W., MITRA, N. J., CHAUDHURI, S., DIVERDI, S., AND FUNKHOUSER, T. 2013. Learning part-based templates from large collections of 3D shapes. *ACM Trans. Graph. 32*, 4 (July), 70:1–70:12.

KOVNATSKY, A., BRONSTEIN, M. M., BRONSTEIN, A. M., GLASHOFF, K., AND KIMMEL, R. 2013. Coupled quasi-harmonic bases. In *Eurographics'13*, 439–448.

NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph. 31*, 6 (Nov.), 137:1–137:10.

NGUYEN, A., BEN-CHEN, M., WELNICKA, K., YE, Y., AND GUIBAS, L. 2011. An optimization approach to improving collections of shape maps. *Computer Graphics Forum 30*, 5, 1481–1491.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph. 21* (October), 807–832.

OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., AND GUIBAS, L. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July), 30:1–30:11.

RUSTAMOV, R. M., OVSJANIKOV, M., AZENCOT, O., BEN-CHEN, M., CHAZAL, F., AND GUIBAS, L. 2013. Map-based exploration of intrinsic shape differences and variability. *ACM Trans. Graph. 32*, 4 (July), 72:1–72:12.

SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph. 30*, 6 (Dec.), 126:1–126:10.

SOLOMON, J., NGUYEN, A., BUTSCHER, A., BEN-CHEN, M., AND GUIBAS, L. 2012. Soft maps between surfaces. *Computer Graphics Forum 31*, 5, 1617–1626.

SZELISKI, R., ZABIH, R., SCHARSTEIN, D., VEKSLER, O., KOLMOGOROV, V., AGARWALA, A., TAPPEN, M., AND ROTHER, C. 2008. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 6 (June), 1068–1080.

WANG, L., AND SINGER, A. 2013. Exact and stable recovery of rotations for robust synchronization. *Information and Inference 2*, 2, 145–193.

WANG, Y., ASAFI, S., VAN KAICK, O., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Active co-analysis of a set of shapes. *ACM Trans. Graph. 31*, 6 (Nov.), 165:1–165:10.

WANG, F., HUANG, Q., AND GUIBAS, L. 2013. Image co-segmentation via consistent functional maps. In *Proceedings of the 14th International Conference on Computer Vision (ICCV)*.

WANG, F., HUANG, Q., OVSJANIKOV, M., AND GUIBAS, L. 2014. Unsupervised multi-class joint image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

WEN, Z., GOLDFARB, D., AND YIN, W. 2010. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Prog. Comput. 2*, 3-4, 203–230.

YUAN, M., AND LIN, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B 68*, 49–67.

A Proofs of Propositions 3

Proof of proposition 3. Suppose Proposition 3 is true. Then we can construct the basis B_i of each $\mathcal{F}_i$ by removing the zero rows in Y_i. In this case, it is clear that each pair-wise functional map X_{ij} would map $\mathbf{b}_{ik}$ to its corresponding basis function of $\mathcal{F}_j$ if it exists or otherwise to the zero function in $\mathcal{F}_j$. Moreover, it is easy to see that each basis function, when transported along any loop, only arrives at the corresponding basis function, or the zero function.

Now suppose the conditions in Definition 2 are satisfied. Then we can build a graph, whose vertices are basis functions on all input shapes, and we connect two vertices with an edge if they can be matched via compositions of the pair-wise functional maps. Now if we consider the disconnected components of this graph, it is easy to see that each component is a clique due to transportability of connectivity. On the other hand, each component can have at most one basis function in each component, since otherwise a basis function, when transformed along a loop, may result in another basis function. This means that we can generate matrices Y_i by filling in appropriate zero functions, so that each row of Y corresponds to one clique. □

B Shape Difference via Rank-Deficient Maps

To show that (18) applies to rank-deficient functional maps as well, we prove that the inverse and composition of shape difference operators, are properly defined. For simplicity, we consider functional basis on S_1 and S_2 such that H_1 and H_2 are identity matrices. In this case, the shape difference operators between them are given by $D_{1,2} = X_{12}^T X_{12}$ and $D_{2,1} = {X_{12}^{+}}^T X_{12}^{+}$. It is easy to see that $D_{1,2}$ and $D_{2,1}$ are related to each other as follows:

$$D_{1,2}^{+} = X_{12}^{+} D_{2,1} X_{12}, \quad D_{2,1}^{+} = X_{12} D_{1,2} X_{12}^{+}. \tag{21}$$

In fact, let us consider the SVD of $X_{12} = U\Sigma V^T$. Then

$$\begin{aligned} X_{12}^{+} D_{2,1} X_{12} &= X_{12}^{+} {X_{12}^{+}}^T X_{12}^{+} X_{12} = V\Sigma^{+} U^T U \Sigma^{+} V^T V \Sigma^{+} U^T U \Sigma V^T \\ &= V {\Sigma^{+}}^2 V^T = (V\Sigma^2 V^T)^{+} = D_{1,2}^{+}. \end{aligned}$$

In other words, $D_{1,2}^{+}$ is identical to $D_{2,1}$ with respect to a generalized similarity transformation.

To check the composition property, let us consider three shapes S_1, S_2, S_3, functional maps X_{12}, X_{23}, and the composed map $X_{23}X_{12}$. Then the induced shape difference operator is

$$\begin{aligned} D_{1,3} &= X_{12}^T X_{23}^T X_{23} X_{12} = X_{12}^T X_{23}^T X_{23} {X_{12}^{+}}^T X_{12}^T X_{12} \\ &= X_{12}^T D_{2,3} {X_{12}^{+}}^T D_{1,2}, \end{aligned}$$

which means $D_{1,3}$ is equal to $D_{2,3}D_{1,2}$ up to a generalized similarity transformation provided by X_{12}.

Functional Map Networks for Analyzing and Browsing Large Shape Collections Supplemental Material

Qixing Huang Fan Wang Leonidas Guibas

Stanford University

1 ADMM for Low-rank Matrix Recovery

For effective optimization, we utilize an equivalent formulation of the trace-norm (cf. [Candes and Plan 2011]):

$$\begin{aligned} \|X\|_\star = \frac{1}{2}\underset{W_1,W_2}{\text{minimize}}\ & \langle I, \begin{pmatrix} W_1 & X \\ X^T & W_2 \end{pmatrix} \rangle \\ \text{subject to}\ & \begin{pmatrix} W_1 & X \\ X^T & W_2 \end{pmatrix} \succeq 0, \end{aligned} \qquad (1)$$

where $\langle \cdot, \cdot \rangle$ denotes the matrix inner product.

To simplify the notations, we denote $\overline{X} = \begin{pmatrix} W_1 & X \\ X^T & W_2 \end{pmatrix}$ and rewrite the optimization problem as follows:

$$\begin{aligned} \text{minimize}\quad & \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \|\mathbf{y}_{ijk}\| + \lambda' \langle I, \overline{X} \rangle \\ \text{subject to}\quad & \mathcal{P}_{ji}(\overline{X})\mathbf{c}_{ijk} - \mathbf{d}_{ijk} = \mathbf{y}_{ijk}, \qquad \begin{array}{c} 1 \leq k \leq N_{ij}, \\ (i,j) \in \mathcal{G} \end{array} \\ & \overline{X} \succeq 0, \end{aligned} \qquad (2)$$

where $\lambda' = \lambda/2$, $\mathcal{P}_{ji}(\overline{X})$ is a linear operator that selects the block X_{ji}.

Introducing dual variables $\mathbf{z}_{ijk}$ and $S \succeq 0$ for constraints $\mathcal{P}_{ji}(\overline{X})\mathbf{c}_{ijk} - \mathbf{d}_{ijk} = \mathbf{y}_{ijk}$ and $X \succeq 0$, respectively, we write down the Lagrangian of (2) as

$$\begin{aligned} & \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} (\|\mathbf{y}_{ijk}\| + \langle \mathbf{z}_{ijk}, \mathcal{P}_{ji}(\overline{X})\mathbf{c}_{ijk} - \mathbf{d}_{ijk} - \mathbf{y}_{ijk} \rangle) + \langle \lambda' I - S, \overline{X} \rangle \\ = & -\sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \langle \mathbf{d}_{ijk}, \mathbf{z}_{ijk} \rangle + \langle \lambda' I + \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \mathcal{P}^\star_{ji}(\mathbf{z}_{ijk}\mathbf{c}^T_{ijk}) - S, \overline{X} \rangle \\ & + \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \min_{\mathbf{y}_{ijk}} (\|\mathbf{y}_{ijk}\| - \langle \mathbf{z}_{ijk}, \mathbf{y}_{ijk} \rangle). \end{aligned} \qquad (3)$$

It is easy to see that

$$\min_{\mathbf{y}_{ijk}} \|\mathbf{y}_{ijk}\| - \langle \mathbf{z}_{ijk}, \mathbf{y}_{ijk} \rangle = \begin{cases} -\infty & \|\mathbf{z}_{ijk}\| > 1 \\ 0 & \text{otherwise.} \end{cases}$$

Thus, to make (3) well defined, we must have $\|\mathbf{z}_{ijk}\| \leq 1$. This allows us to write down the dual problem of (2) as the following constrained optimization problem:

$$\begin{aligned} \text{minimize}\quad & \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \langle \mathbf{z}_{ijk}, \mathbf{d}_{ijk} \rangle \\ \text{subject to}\quad & S - \lambda' I - \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \mathcal{P}^\star_{ji}(\mathbf{z}_{ijk}\mathbf{c}^T_{ijk}) = 0 \\ & \|\mathbf{z}_{ijk}\| \leq 1, \quad (i,j) \in \mathcal{G}, 1 \leq k \leq N_{ij}, \\ & S \succeq 0. \end{aligned} \qquad (4)$$

As described in [Wen et al. 2010], we consider the following augmented Lagrangian problem in order to solve (4):

$$\begin{aligned} \text{minimize}\quad & \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \langle \mathbf{z}_{ijk}, \mathbf{d}_{ijk} \rangle \\ & + \langle \overline{X}, S - \lambda' I - \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \mathcal{P}^\star_{ji}(\mathbf{z}_{ijk}\mathbf{c}^T_{ijk}) \rangle \\ & + \frac{1}{2\mu} \|S - \lambda' I - \sum_{(i,j)\in\mathcal{G}} \sum_{k=1}^{N_{ij}} \mathcal{P}^\star_{ji}(\mathbf{z}_{ijk}\mathbf{c}^T_{ijk})\|^2_F \\ \text{subject to}\quad & \|\mathbf{z}_{ijk}\| \leq 1, \quad (i,j) \in \mathcal{G}, 1 \leq k \leq N_{ij}, \\ & S \succeq 0, \end{aligned} \qquad (5)$$

where μ is a decaying regularizing parameter. In our experiments, we set $\mu^{(0)} = 0.1$, and $\mu^{(t+1)} = 1.1 \cdot \mu^{(t)}$ during the optimization process.

In the same spirit as [Wen et al. 2010], we employ the alternating direction method, which alternates between optimizing $\mathbf{z}_{ijk}$ and S, and updating the primal variable $\overline{X}$. Let $Z_{ij} = (\mathbf{z}_{ij1}, \cdots, \mathbf{z}_{ijN_{ij}})$. At iteration $t+1$, when S and $\overline{X}$ are fixed, we can optimize each Z_{ij} independently via

$$Z^{(t+1)}_{ij} = \underset{\|\mathbf{z}_{ijk}\| \leq 1,\ 1 \leq k \leq N_{ij}}{\arg\min} \langle Z_{ij}, D_{ij} - \mathcal{P}_{ji}(\overline{X}^{(t)})C_{ij} \rangle + \frac{1}{2\mu} \|S^{(t)}_{ji} - Z_{ij}C^T_{ij}\|^2_F. \qquad (6)$$

We apply the barrier method [Boyd and Vandenberghe 2004] to solve the dual program of this convex program efficiently.

When $Z^{(t)}_{ij}$ are fixed, following [Wen et al. 2010], we can write down the optimal value of S and the value of the primal variable $\overline{X}$ at the next iteration as

$$[S^{(t+1)}, -\mu\overline{X}^{(t+1)}] = \text{decomp}(\lambda' I + \sum_{(i,j)\in\mathcal{G}} \mathcal{P}^\star_{ji}(Z^{(t+1)}_{ij} C^T_{ij}) - \mu\overline{X}^{(t)}), \qquad (7)$$

where

$$[A, B] = \text{decomp}(C) \quad \Leftrightarrow \quad A + B = C,\ A, -B \succeq 0.$$

The sub-optimizations problems (7) and (6) are iterated until convergence. Typically only 50-100 alternating iterations are sufficient for convergence.

2 Gauss-Newton Method for Latent Basis Recovery

We begin with introducing the differential of Moore-Penrose pseudoinverse as summarized in the following proposition (c.f. [Constales 1998]).

Proposition 1. *Given a matrix $A \in \mathbb{R}^{n\times m}$ of rank r. Let $A = U \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} V^T$ be the full SVD of matrix $A \in \mathbb{R}^{n\times m}$, i.e., $U \in \mathcal{O}(n), V \in \mathcal{O}(m)$, and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r), \sigma_i > 0$. Let D be the perturbation of A parameterized as $D = U \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} V^T$. Then the differential of $(A+D)^+$ with respect to D is well-defined if $D_{22} = 0$, and the first-order approximation of $(A+D)^+$ with respect to D is given by*

$$(A+D)^+ \approx \mathbf{A}^+ + \mathbf{V} \begin{pmatrix} -\Sigma^{-1} D_{11} \Sigma^{-1} & \Sigma^{-2} {D_{21}}^T \\ {D_{12}}^T \Sigma^{-2} & 0 \end{pmatrix} U^T$$

Now we discuss the details of the Gauss-Newton method. Suppose the current latent basis are $Y_i^{(t)}, 1 \leq i \leq N$. To compute $Y_i^{(t+1)}$, we optimize a small perturbation $dY_i^{(t)}$ of $Y_i^{(t)}$ so that the following objective function is minimized

$$f = \sum_{(i,j)\in\mathcal{G}} \|X^{\star}_{ij} - Y_j^{+} Y_i - dY_j^{+} Y_i - Y_j^{+} dY_i\|_F^2$$
$$+\mu \sum_{k,l} (\mathbf{y}_{ik}^T \mathbf{y}_{jl} + d\mathbf{y}_{ik}^T \mathbf{y}_{jl} + \mathbf{y}_{ik}^T d\mathbf{y}_{jl})^2 + \gamma \sum_{i=1}^{N} \|dY_i\|_F^2, \qquad (8)$$

where both dY_j^{+} and dY_i are parameterized using Prop.1. In this case, f is a quadratic function and can be optimized by solving a linear system. Note that we have added a small regularization term $\gamma \sum_{i=1}^{N} \|dY_i\|_F^2$ to prevent from obtaining degenerate solutions.

When solving the induced linear system, we find that it is more efficient to just use the block-diagonal of the left hand side of the system. More precisely, all terms that are related to multiplications of dY_i and dY_j are dropped. This can be considered as a generalized Jacobian iteration, which leads to a series of small-scale linear systems.

References

BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

CANDES, E. J., AND PLAN, Y. 2011. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inf. Theor. 57*, 4 (Apr.), 2342–2359.

CONSTALES, D. 1998. A closed formula for the moore-penrose generalized inverse of a complex matrix of given rank. *Acta Math. Hungar 80*, 83–88.

WEN, Z., GOLDFARB, D., AND YIN, W. 2010. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Prog. Comput. 2*, 3-4, 203–230.