Path-Invariant Map Networks

Zaiwei Zhang
UT Austin

Zhenxiao Liang
UT Austin

Lemeng Wu
UT Austin

Xiaowei Zhou
Zhejiang University*

Qixing Huang
UT Austin†

Abstract

Optimizing a network of maps among a collection of objects/domains (or map synchronization) is a central problem across computer vision and many other relevant fields. Compared to optimizing pairwise maps in isolation, the benefit of map synchronization is that there are natural constraints among a map network that can improve the quality of individual maps. While such self-supervision constraints are well-understood for undirected map networks (e.g., the cycle-consistency constraint), they are underexplored for directed map networks, which naturally arise when maps are given by parametric maps (e.g., a feed-forward neural network). In this paper, we study a natural self-supervision constraint for directed map networks called path-invariance, which enforces that composite maps along different paths between a fixed pair of source and target domains are identical. We introduce path-invariance bases for efficient encoding of the path-invariance constraint and present an algorithm that outputs a path-variance basis with polynomial time and space complexities. We demonstrate the effectiveness of our approach on optimizing object correspondences, estimating dense image maps via neural networks, and semantic segmentation of 3D scenes via map networks of diverse 3D representations. In particular, for 3D semantic segmentation our approach only requires 8% labeled data from ScanNet to achieve the same performance as training a single 3D segmentation network with 30% to 100% labeled data.

1. Introduction

Optimizing a network of maps among a collection of objects/domains (or map synchronization) is a central problem across computer vision and many other relevant fields. Important applications include establishing consistent feature correspondences for multi-view structure-from-motion [1, 11, 44, 5], computing consistent relative camera poses for 3D reconstruction [20, 17], dense image flows [57, 56], image translation [59, 52], and optimizing consistent dense correspondences for co-segmentation [47, 16, 48] and object discovery [40, 8], just to name a few. The benefit of optimizing a map network versus optimizing maps between pairs of objects in isolation comes from the *cycle-consistency* constraint [31, 18, 15, 47]. For example, this constraint allows us to replace an incorrect map between a pair of dissimilar objects by composing maps along a path of similar objects [18]. Computationally, state-of-the-art map synchronization techniques [3, 7, 15, 19, 43, 16, 18, 25, 57, 58, 29] employ matrix representations of maps [25, 18, 15, 48, 16]. This allows us to utilize a low-rank formulation of the cycle-consistency constraint (c.f. [15]), leading to efficient and robust solutions [16, 58, 43, 19].

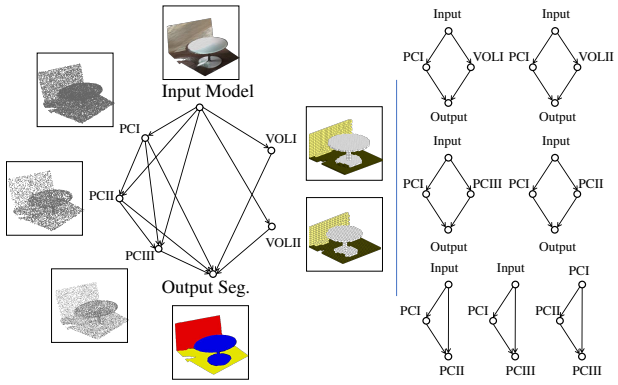

Figure 1: (Left) A network of 3D representations for the task of semantic segmentation of 3D scenes. (Right) Computed path-invariance basis for regularizing individual neural networks.

In this paper, we focus on a map synchronization setting, where matrix-based map encodings become too costly or even infeasible. Such instances include optimizing dense flows across many high-resolution images [30, 24, 41] or optimizing a network of neural networks, each of which maps one domain to another domain (e.g., 3D semantic segmentation [12] maps the space of 3D scenes to the space of 3D segmentations). In this setting, maps are usually encoded as broadly defined parametric maps (e.g., feed-forward neural networks), and map optimization reduces to optimizing hyper-parameters and/or network parameters. Synchronizing parametric maps introduces many technical challenges. For example, unlike correspondences between objects, which are undirected, a parametric map may not have a meaningful inverse map. This raises the challenge

*Xiaowei Zhou is affiliated with the StateKey Lab of CAD&CG and the ZJU-SenseTime Joint Lab of 3D Vision.
†huangqx@cs.utexas.edu

of formulating an equivalent regularization constraint of cycle-consistency for directed map networks. In addition, as matrix-based map encodings are infeasible for parametric maps, another key challenge is how to efficiently enforce the regularization constraint for map synchronization.

We introduce a computational framework for optimizing directed map networks that addresses the challenges described above. Specifically, we propose the so-called *path-invariance constraint*, which ensures that whenever there exists a map from a source domain to a target domain (through map composition along a path), the map is unique. This path-invariance constraint not only warrants that a map network is well-defined, but more importantly it provides a natural regularization constraint for optimizing directed map networks. To effectively enforce this path-invariance constraint, we introduce the notion of a *path-invariance basis*, which collects a subset of path pairs that can induce the path-invariance property of the entire map network. We also present an algorithm for computing a path-invariance basis from an arbitrary directed map network. The algorithm possesses polynomial time and space complexities.

We demonstrate the effectiveness of our approach on three settings of map synchronization. The first setting considers undirected map networks that can be optimized using low-rank formulations [16, 58]. Experimental results show that our new formulation leads to competitive and sometimes better results than state-of-the-art low-rank formulations. The second setting studies consistent dense image maps, where each pairwise map is given by a neural network. Experimental results show that our approach significantly outperforms state-of-the-art approaches for computing dense image correspondences. The third setting considers a map network that consists of 6 different 3D representations (e.g., point cloud and volumetric representations) for the task of semantic 3D semantic segmentation (See Figure 1). By enforcing the path-invariance of neural networks on unlabeled data, our approach only requires 8% labeled data from ScanNet [12] to achieve the same performance as training a single semantic segmentation network with 30% to 100% labeled data.

2. Related Works

Map synchronization. Most map synchronization techniques [20, 17, 54, 31, 15, 57, 16, 7, 49, 5, 58, 2, 53, 19, 34, 43, 14, 56, 59, 52] have focused on undirected map graphs, where the self-regularization constraint is given by cycle-consistency. Depending on how the cycle-consistency constraint is applied, existing approaches fall into three categories. The first category of methods [20, 17] utilizes the fact that a collection of cycle-consistent maps can be generated from maps associated with a spanning tree. However, it is hard to apply them for optimizing cycle-consistent neural networks, where the neural networks change during the course of the optimization. The second category of approaches [54, 31, 57] applies constrained optimization to select cycle-consistent maps. These approaches are typically formulated so that the objective functions encode the score of selected maps, and the constraints enforce the consistency of selected maps along cycles. Our approach is relevant to this category of methods but addresses a different problem of optimizing maps along directed map networks.

The third category of approaches apply modern numerical optimization techniques to optimize cycle-consistent maps. Along this line, people have introduced convex optimization [15, 16, 7, 49], non-convex optimization [5, 58, 2, 53, 19], and spectral techniques [34, 43]. To apply these techniques for parametric maps, we have to hand-craft an additional latent domain, as well as parametric maps between each input domain and this latent domain, which may suffer from the issue of sub-optimal network design. In contrast, we focus on directed map networks among diverse domains and explicitly enforce the path-invariance constraint via path-invariance bases.

Joint learning of neural networks. Several recent works have studied the problem of enforcing cycle-consistency among a cycle of neural networks for improving the quality of individual networks along the cycle. Zhou et al. [56] studied how to train dense image correspondences between real image objects through two real-2-synthetic networks and ground-truth correspondences between synthetic images. [59, 52] enforce the bi-directional consistency of transformation networks between two image domains to improve the image translation results. People have applied such techniques for multilingual machine translation [21]. However, in these works the cycles are explicitly given. In contrast, we study how to extend the cycle-consistency constraint on undirected graphs to the path-invariance constraint on directed graphs. In particular, we focus on how to compute a path-invariance basis for enforcing the path-invariance constraint efficiently. A recent work [55] studies how to build a network of representations for boosting individual tasks. However, self-supervision constraints such as cycle-consistency and path-invariance are not employed. Another distinction is that our approach seeks to leverage unlabeled data, while [55] focuses on transferring labeled data under different representations/tasks. Our approach is also related to model/data distillation (See [38] and references therein), which can be considered as a special graph with many edges between two domains. In this paper, we focus on defining self-supervision for general graphs.

Cycle-bases. Path-invariance bases are related to cycle-bases on undirected graphs [22], in which any cycle of a graph is given by a linear combination of the cycles in a cycle-basis. However, besides fundamental cycle-bases [22] that can generalize to define cycle-consistency bases, it is an open problem whether other types of cycle-bases generalize or not. Moreover, there are fundamental differences between undirected and directed map networks. This calls for new tools for defining and computing path-invariance bases.

3. Path-Invariance of Directed Map Networks

In this section, we focus on the theoretical contribution of this paper, which introduces an algorithm for computing a path-invariance basis that enforces the path-invariance constraint of a directed map network. Note that the proofs of theorems and propositions in this section are deferred to the supplementary material.

3.1. Path-Invariance Constraint

We first define the notion of a directed map network:

Definition 1. *A directed map network $\mathcal{F}$ is an attributed directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $V = \{v_1, \dots, v_{|\mathcal{V}|}\}$. Each vertex $v_i \in \mathcal{V}$ is associated with a domain $\mathcal{D}_i$. Each edge $e \in \mathcal{E}$ with $e = (i, j)$ is associated with a map $f_{ij} : \mathcal{D}_i \to \mathcal{D}_j$. In the following, we always assume $\mathcal{E}$ contains the self-loop at each vertex, and the map associated with each self-loop is the identity map.*

For simplicity, whenever it can be inferred from the context we simplify the terminology of a directed map network as a map network. The following definition considers induced maps along paths of a map network.

Definition 2. *Consider a path $p = (i_0, \cdots, i_k)$ along $\mathcal{G}$. We define the composite map along p induced from a map network $\mathcal{F}$ on $\mathcal{G}$ as*

$$f_p = f_{i_{k-1} i_k} \circ \cdots \circ f_{i_0 i_1}. \tag{1}$$

We also define $f_\varnothing := I$ where $\varnothing$ can refer to any self-loop.

In the remaining text, for two successive paths p and q, we use $p \sim q$ to denote their composition.

Now we state the path-invariance constraint.

Definition 3. *Let $\mathcal{G}_{\text{path}}(u, v)$ collect all paths in $\mathcal{G}$ that connect u to v. We define the set of all possible path pairs of $\mathcal{G}$ as*

$$\mathcal{G}_{\text{pair}} = \bigcup_{u,v \in \mathcal{V}} \{(p, q) | p, q \in \mathcal{G}_{\text{path}}(u, v)\}.$$

We say $\mathcal{F}$ is path-invariant if

$$f_p = f_q, \qquad \forall (p, q) \in \mathcal{G}_{\text{pair}}. \tag{2}$$

Remark 1. It is easy to check that path-invariance induces cycle-consistency (c.f.[15]), but cycle-consistency does not necessarily induce path-invariance. For example, a map network with three vertices $\{a, b, c\}$ and three directed maps f_{ab}, f_{bc}, and f_{ac} has no-cycle, but one path pair $(f_{bc} \circ f_{ab}, f_{ac})$.

3.2. Path-Invariance Basis

A challenge of enforcing the path-invariant constraint is that there are many possible paths between each pair of domains in a graph, leading to an intractable number of path pairs. This raises the question of how to compute a path-invariance basis $\mathcal{B} \subset \mathcal{G}_{\text{pair}}$, which is a small set of path pairs that are sufficient for enforcing the path-invariance property of any map network $\mathcal{F}$. To rigorously define path-invariance basis, we introduce three primitive operations on path pairs merge, stitch and cut(See Figure 2):

Definition 4. *Consider a directed graph $\mathcal{G}$. We say two path pairs (p, q) and (p', q') are* compatible *if one path in $\{p, q\}$ is a sub-path of one path in $\{p', q'\}$ or vice-versa. Without losing generality, suppose p is a sub-path of p' and we write $p' = r \sim p \sim r'$, which stitches three sub-paths r,p,and r' in order. We define the* merge *operation so that it takes two compatible path pairs (p, q) and $(r \sim p \sim r', q')$ as input and outputs a new path pair $(r \sim q \sim r', q')$.*

We proceed to define the stitch operation:

Definition 5. *We define the* stitch *operation so that it takes as input two path pairs $(p, q), p, q \in \mathcal{G}_{\text{path}}(u, v)$ and $(p', q'), p', q' \in \mathcal{G}_{\text{path}}(v, w)$ and outputs $(p \sim p', q \sim q')$.*

Finally we define the cut operation on two cycles, which will be useful for strongly connected graphs:

Definition 6. *Operation* cut *takes as input two path pairs $(\mathcal{C}_1, \varnothing)$ and $(\mathcal{C}_2, \varnothing)$ where $\mathcal{C}_1$ and $\mathcal{C}_2$ are two distinct cycles that have two common vertices u, v and share a common path from v to u. Specifically, we assume these two cycles are $u \xrightarrow{p} v \xrightarrow{q} u$ and $u \xrightarrow{p'} v \xrightarrow{q} u$ where $p, p' \in \mathcal{G}_{\text{path}}(u, v)$ and $q \in \mathcal{G}_{\text{path}}(v, u)$. We define the output of the* cut *operation as a new path pair (p, p').*

Definition 6 is necessary because $f_p \circ f_q = f_{p'} \circ f_q = I$ implies $f_p = f_{p'}$. As we will see later, this operation is useful for deriving new path-invariance basis.

Now we define path-invariance basis, which is the critical concept of this paper:

Definition 7. *We say a collection of path pairs $\mathcal{B} = \{(p, q)\}$ is a path-invariance basis on $\mathcal{G}$ if every path-pair $(p, q) \in \mathcal{G}_{\text{pair}} \setminus \mathcal{B}$ can be induced from a subset of $\mathcal{B}$ through a series of* merge, stitch *and/or* cut *operations.*

The following proposition shows the importance of path-invariance basis:

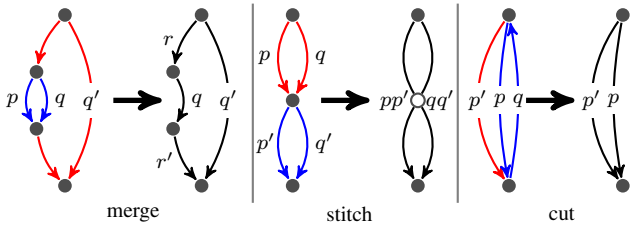

Figure 2: Illustrations of Operations

Proposition 1. *Consider a path-invariance basis $\mathcal{B}$ of a graph $\mathcal{G}$. Then for any map network $\mathcal{F}$ on $\mathcal{G}$, if*

$$f_p = f_q, \quad (p, q) \in \mathcal{B},$$

then $\mathcal{F}$ is path-invariant.

3.3. Path-Invariance Basis Computation

We first discuss the criteria for path-invariance basis computation. Since we will formulate a loss term for each path pair in a path-invariance basis, we place the following three objectives. First, we require the length of the paths in each path pair to be small. Intuitively, enforcing the consistency between long paths weakens the regularization on each involved map. Second, we want the size of the resulting path-invariance basis to be small to increase the efficiency of gradient-descent based optimization strategies. Finally, we would like the resulting path-invariance basis to be nicely distributed to improve the convergence property of the induced optimization problem. Unfortunately, achieving these goals exactly appears to be intractable. For example, we conjecture that computing a path-invariance basis of a given graph with minimum size is NP-hard[1].

In light of this, our approach seeks to compute a path-invariance basis whose size is polynomial in $|\mathcal{V}|$, i.e., $O(|\mathcal{V}||\mathcal{E}|)$ in the worst case. Our approach builds upon the classical result that a directed graph $\mathcal{G}$ can be factored into a directed acyclic graph (or DAG) whose vertices are strongly connected components of $\mathcal{G}$ (c.f. [4]). More precisely, we first show how to compute a path-invariance basis for a DAG. We then discuss the case of strongly connected components. Finally, we show how to extend the result of the first two settings to arbitrary directed graphs. Note that our approach implicitly takes two other criteria into account. Specifically, we argue that small path-invariance basis favors short path-pairs, as it is less likely to combine long path-pairs to produce new path-pairs through merge, stitch and cut operations. In addition, this construction takes the global structure of the input graph $\mathcal{G}$ into account, leading to nicely distributed path-pairs.

Directed acyclic graph (or DAG). Our algorithm utilizes an important property that every DAG admits a topological order of vertices that are consistent with the edge orientations (c.f. [4]). Specifically, consider a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A topological order is a bijection $\sigma : \{1, \cdots, |\mathcal{V}|\} \to \mathcal{V}$ so that we have $\sigma^{-1}(u) < \sigma^{-1}(v)$ whenever $(u, v) \in \mathcal{E}$. A topological order of a DAG can be computed by Tarjan's algorithm (c.f. [46]) in linear time.

Our algorithm starts with a current graph $\mathcal{G}_{cur} = (\mathcal{V}, \varnothing)$ to which we add all edges in $\mathcal{E}$ in some order later. Specifically, the edges in $\mathcal{E}$ will be visited with respect to a (partial) edge order $\prec$ where $\forall (u, v), (u', v') \in \mathcal{E}$, $(u, v) \prec (u', v')$ if and only if $\sigma^{-1}(v) < \sigma^{-1}(v')$. Note that two edges (u, v), (u', v) with the same head can be in arbitrary order.

[1]Unlike cycle bases that have a known minimum size (c.f. [22]), the sizes of minimum path-invariance bases vary

Algorithm 1 The high level algorithm flow to find a path-invariance basis.

input: Directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
output: Path-invariance basis $\mathcal{B}$.

1: Calculate SCCs $\mathcal{G}_1, \ldots, \mathcal{G}_K$ for $\mathcal{G}$ and the resulting contracted DAG $\mathcal{G}_{dag}$.
2: Calculate a path-invriance basis $\mathcal{B}_{dag}$ for $\mathcal{G}_{dag}$ and transform $\mathcal{B}_{dag}$ to $\overline{\mathcal{B}}_{dag}$ that collect path pairs on $\mathcal{G}$.
3: Calculate a path-invariance basis $\mathcal{B}_i$ for $\mathcal{G}_i$.
4: Calculate path-invirance pairs $\mathcal{B}_{ij}$ whenever $\mathcal{G}_i$ can reach $\mathcal{G}_j$ in $\mathcal{G}_{dag}$.
5: **return** $\mathcal{B} = \overline{\mathcal{B}}_{dag} \bigcup \left(\cup_{i=1}^{K} \mathcal{B}_i \right) \bigcup \left(\cup_{ij} \mathcal{B}_{ij} \right)$

For each newly visited edge $(u, v) \in \mathcal{E}$, we collect a set of candidate vertices $\mathcal{P} \subset \mathcal{V}$ such that every vertex $w \in \mathcal{P}$ can reach both u and v in $\mathcal{G}_{cur}$. Next we construct a set $\overline{\mathcal{P}}$ by removing from $\mathcal{P}$ all $w \in \mathcal{P}$ such that w can reach some distinct $w' \in \mathcal{P}$. In other words, w is redundant because of w' in this case. For each vertex $w \in \overline{\mathcal{P}}$, we collect a new path-pair $(p', p \sim uv)$, where p and p' are shortest paths from w to u and v, respectively. After collecting path pairs, we augment $\mathcal{G}_{cur}$ with (u, v). With $\mathcal{B}_{dag}(\sigma)$ we denote the resulting path-pair set after $\mathcal{E}_{cur} = \mathcal{E}$.

Theorem 3.1. *Every topological order σ of $\mathcal{G}$ returns a path-invariance basis $\mathcal{B}_{dag}(\sigma)$ whose size is at most $|\mathcal{V}||\mathcal{E}|$.*

Strongly connected graph (or SCG). To construct a path-invariance basis of a SCG $\mathcal{G}$, we run a slightly-modified depth-first search on $\mathcal{G}$ from arbitrary vertex. Since $\mathcal{G}$ is strongly connected, the resulting spanning forest must be a tree, denoted by $\mathcal{T}$. The path pair set $\mathcal{B}$ is the result we obtain. In addition, we use a $\mathcal{G}_{dag}$ to collect a acyclic subgraph of $\mathcal{G}$ and initially it is set as empty. When traversing edge (u, v), if v is visited for the first time, then we add (u, v) to both $\mathcal{T}$ and $\mathcal{G}_{dag}$. Otherwise, there can be two possible cases:

- v is an ancestor of u in $\mathcal{T}$. In this case we add cycle pair $(P \sim (u, v), \varnothing)$, where P is the tree path from v to u, into $\mathcal{B}$.
- Otherwise, add (u, v) into $\mathcal{G}_{dag}$.

We can show that $\mathcal{G}_{dag}$ is indeed an acyclic graph (See Section A.3 in the supplementary material). Thus we can obtain a path-invariance basis on $\mathcal{G}_{dag}$ by running the construction procedure introduced for DAG. We add this basis into $\mathcal{B}$.

Proposition 2. *The path pair set $\mathcal{B}$ constructed above is a path-invariance basis of $\mathcal{G}$.*

General directed graph. Given path-invariance bases constructed on DAGs and SCGs, constructing path-invariance bases on general graphs is straight-forward. Specifically,

consider strongly connected components $\mathcal{G}_i, 1 \leq i \leq K$ of a graph $\mathcal{G}$. With $\mathcal{G}_{dag}$ we denote the DAG among $\mathcal{G}_i, 1 \leq i \leq K$. We first construct path-invariance bases $\mathcal{B}_{dag}$ and $\mathcal{B}_i$ for $\mathcal{G}_{dag}$ and each $\mathcal{G}_i$, respectively. We then construct a path-invariance basis $\mathcal{B}$ of $\mathcal{G}$ by collecting three groups of path pairs. The first group simply combines $\mathcal{B}_i, 1 \leq i \leq K$. The second group extends $\mathcal{B}_{dag}$ to the original graph. This is done by replacing each edge $(\mathcal{G}_i, \mathcal{G}_j) \in \mathcal{E}_{dag}$ through a shortest path on $\mathcal{G}$ that connects the representatives of $\mathcal{G}_i$ and $\mathcal{G}_j$ where representatives are arbitrarily chosen at first for each component. To calculate the third group, consider all oriented edges between each $(\mathcal{G}_i, \mathcal{G}_j) \in \mathcal{E}_{dag}$:

$$\mathcal{E}_{ij} = \{uv \in \mathcal{E} : u \in \mathcal{V}_i, v \in \mathcal{V}_j\}.$$

Note that when constructing $\mathcal{B}_{dag}$, all edges in $\mathcal{E}_{ij}$ are shrinked to one edge in $\mathcal{E}_{dag}$. This means when constructing $\mathcal{B}$, we have to enforce the consistency among $\mathcal{E}_{ij}$ on the original graph $\mathcal{G}$. This can be done by constructing a tree $\mathcal{T}_{ij}$ where $\mathcal{V}(\mathcal{T}_{ij}) = \mathcal{E}_{ij}$, $\mathcal{E}(\mathcal{T}_{ij}) \subset \mathcal{E}_{ij}^2$. $\mathcal{T}_{ij}$ is a minimum spanning tree on the graph whose vertex set is $\mathcal{E}_{ij}$ and the weight associated with edge $(uv, u'v') \in \mathcal{E}_{ij}^2$ is given by the sum of lengths of $\widehat{uu'}$ and $\widehat{vv'}$. This strategy encourages reducing the total length of the resulting path pairs in $\mathcal{B}_{ij}$:

$$\mathcal{B}_{ij} := \{(\widehat{uu'} \sim u'v', uv \sim \widehat{vv'}) : (uv, u'v') \in \mathcal{E}(\mathcal{T}_{ij})\},$$

where $\widehat{uu'}$ and $\widehat{vv'}$ denote the shortest paths from u to u' on $\mathcal{G}_i$ and from v to v' on $\mathcal{G}_j$, respectively. Algorithm 1 presents the high-level pesudo code of our approach.

Theorem 3.2. *The path-pairs $\mathcal{B}$ derived from $\mathcal{B}_{dag}$, $\{\mathcal{B}_i : 1 \leq i \leq K\}$, and $\{\mathcal{B}_{ij} : (\mathcal{G}_i, \mathcal{G}_j) \in \mathcal{E}_{dag}\}$ using the algorithm described above is a path-invariance basis for $\mathcal{G}$.*

Proposition 3. *The size of $\mathcal{B}$ is upper bounded by $|\mathcal{V}||\mathcal{E}|$.*

4. Joint Map Network Optimization

In this section, we present a formulation for jointly optimizing a map network using the path-variance basis computed in the preceding section.

Consider the map network defined in Def. 1. We assume the map associated with each edge $(i, j) \in \mathcal{E}$ is a parametric map $f_{ij}^{\theta_{ij}}$, where θ_{ij} denotes hyper-parameters or network parameters of f_{ij}. We assume the supervision of map network is given by a superset $\overline{\mathcal{E}} \supset \mathcal{E}$. As we will see later, such instances happen when there exist paired data between two domains, but we do not have a direct neural network between them. To utilize such supervision, we define the induced map along an edge $(i, j) \in \overline{\mathcal{E}}$ as the composition map (defined in (1)) $f_{\widehat{v_i v_j}}^{\Theta}$ along the short path $\widehat{v_i v_j}$ from v_i to v_j. Here $\Theta = \{\theta_{ij}, (i, j) \in \mathcal{E}\}$ collects all the parameters. We define each supervised loss term as $l_{ij}(f_{ij}^{\Theta}), \forall (i, j) \in \overline{\mathcal{E}}$. The specific definition of l_{ij} will be deferred to Section 5.

Besides the supervised loss terms, the key component of joint map network optimization utilizes a self-supervision loss induced from the path-invariance basis $\mathcal{B}$. Let $d_{\mathcal{D}_i}(\cdot, \cdot)$ be a distance measure associated with domain $\mathcal{D}_i$. Consider an empirical distribution P_i of $\mathcal{D}_i$. We define the total loss objective for joint map network optimization as

$$\min_{\Theta} \sum_{(i,j)\in\overline{\mathcal{E}}} l_{ij}(f_{\widehat{v_i v_j}}^{\Theta}) + \lambda \sum_{(p,q)\in\mathcal{B}} \mathop{E}_{v\sim P_{p_t}} d_{\mathcal{D}_{p_t}}(f_p^{\Theta}(v), f_q^{\Theta}(v)) \tag{3}$$

where p_t denotes the index of the end vertex of p. Essentially, (3) combines the supervised loss terms and an unsupervised regularization term that ensures the learned representations are consistent when passing unlabeled instances across the map network. We employ the ADAM optimizer [27] for optimization. In addition, we start with a small value of λ, e.g., $\lambda = 10^{-2}$, to solve (3) for 40 epochs. We then double the value of λ every 10 epochs. We stop the training procedure when $\lambda \geq 10^3$. The training details are deferred to the Appendix.

5. Experimental Evaluation

This section presents an experimental evaluation of our approach across three settings, namely, shape matching (Section 5.1), dense image maps (Section 5.2), and 3D semantic segmentation (Section 5.3).

5.1. Map Network of Shape Maps

We begin with the task of joint shape matching [31, 25, 15, 16, 10], which seeks to jointly optimize shape maps to improve the initial maps computed between pairs of shapes in isolation. We utilize the functional map representation described in [33, 47, 16]. Specifically, each domain $\mathcal{D}_i$ is given by a linear space spanned by the leading m eigenvectors of a graph Laplacian [16] (we choose $m = 30$ in our experiments). The map from $\mathcal{D}_i$ to $\mathcal{D}_j$ is given by a matrix $X_{ij} \in \mathbb{R}^{m\times m}$. Let $\mathcal{B}$ be a path-invariance basis for the associated graph $\mathcal{G}$. Adapting (3), we solve the following optimization problem for joint shape matching:

$$\sum_{(i,j)\in\mathcal{E}} \|X_{ij} - X_{ij}^{in}\|_1 + \lambda \sum_{(p,q)\in\mathcal{B}} \|X_p - X_q\|_{\mathcal{F}}^2 \tag{4}$$

where $\|\cdot\|_1$ and $\|\cdot\|_{\mathcal{F}}$ are the element-wise L1-norm and the matrix Frobenius norm, respectively. X_{ij}^{in} denotes the initial functional map converted from the corresponding initial shape map using [33].

Dataset. We perform experimental evaluation on SHREC07–Watertight [13]. Specifically, SHREC07-Watertight contains 400 shapes across 20 categories. Among them, we choose 11 categories (i.e., Human, Glasses, Airplane, Ant, Teddy, Hand, Plier, Fish, Bird, Armadillo, Fourleg) that are suitable for inter-shape mapping. We also test our approach on two large-scale datasets Aliens (200 shapes) and Vase (300 shapes) from ShapeCOSEG [50]. For initial maps, we employ blended intrinsic maps [26], a state-of-the-art method for shape

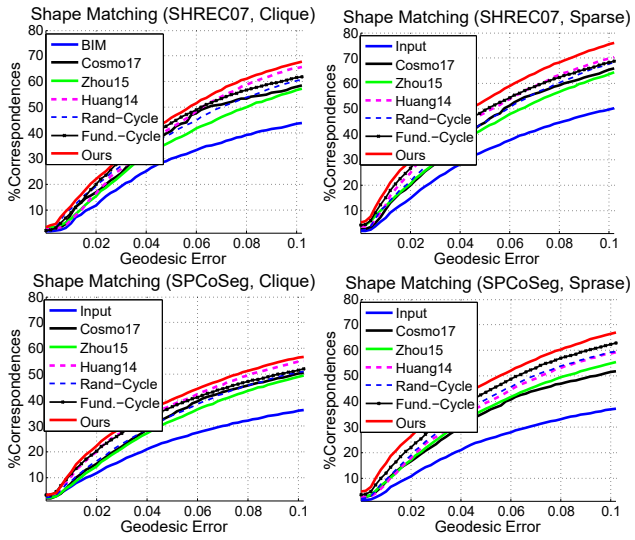

Figure 3: Baseline comparison on benchmark datasets. We show cumulative distribution functions (or CDFs) of each method with respect to annotated feature correspondences.

matching. We test our approach under two graphs $\mathcal{G}$. The first graph is a clique graph. The second graph connects each shape with k-nearest neighbor with respect to the GMDS descriptor [42] ($k = 10$ in our experiments).

Evaluation setup. We compare our approach to five baseline approaches, including three state-of-the-art approaches and two variants of our approach. Three state-of-the-art approaches are 1) functional-map based low-rank matrix recovery [16], 2) point-map based low-rank matrix recovery via alternating minimization [58], and 3) consistent partial matching via sparse modeling [10]. Two variants are 4) using a set of randomly sampled cycles [54] whose size is the same as $|\mathcal{B}|$, and 5) using the path-invariance basis derived from the fundamental cycle-basis of $\mathcal{G}$ (c.f. [22]) (which may contain long cycles).

We evaluate the quality of each map through annotated key points (Please refer to the supplementary material). Following [26, 15, 16], we report the cumulative distribution function (or CDF) of geodesic errors of predicted feature correspondences.

Analysis of results. Figure 3 shows CDFs of our approach and baseline approaches. All participating methods exhibit considerable improvements from the initial maps, demonstrating the benefits of joint matching. Compared to state-of-the-art approaches, our approach is comparable when $\mathcal{G}$ is a clique and exhibits certain performance gains when $\mathcal{G}$ is sparse. One explanation is that low-rank approaches are based on relaxations of the cycle-consistency constraint (c.f. [15]), and such relaxations become loose on sparse graphs. Compared to the two variants, our approach delivers the best results on both clique graphs and knn-graphs. This is because the two alternative strategies generate many long paths and cycles in $\mathcal{B}$, making the total objective function (3) hard to optimize. On knn-graphs, both our approach and the baseline of using the fundamental cycle-basis outperform the baseline of randomly sampling path pairs, showing the importance of computing a path-invariance basis for enforcing the consistency constraint.

5.2. Map Network of Dense Image Maps

In the second setting, we consider the task of optimizing dense image flows across a collection of relevant images. We again model this task using a map network $\mathcal{F}$, where each domain $\mathcal{D}_i$ is given by an image I_i. Our goal is to compute a dense image map $f_{ij} : I_i \to I_j$ (its difference to the identity map gives a dense image flow) between each pair of input images. To this end, we precompute initial dense maps $f_{ij}^{in}, \forall (i,j) \in \mathcal{E}$ using DSP [24], which is a state-of-the-art approach for dense image flows. Our goal is to obtain improved dense image maps $f_{ij}, \forall (i,j) \in \mathcal{E}$, which lead to dense image maps between all pairs of images in $\mathcal{F}$ via map composition (See (1)). Due to scalability issues, state-of-the-art approaches for this task [28, 23, 36, 57] are limited to a small number of images. To address this issue, we encode dense image maps using the neural network f^θ described in [56]. Given a fixed map network $\mathcal{F}$ and the initial dense maps $f_{ij}^{in}, (i,j) \in \mathcal{E}$, we formulate a similar optimization problem as (4) to learn θ:

$$\min_{\theta} \sum_{(i,j)\in\mathcal{E}} \|f_{ij}^{\theta} - f_{ij}^{in}\|_1 + \lambda \sum_{(p,q)\in\mathcal{B}} \|f_p^{\theta} - f_q^{\theta}\|_{\mathcal{F}}^2 \quad (5)$$

where $\mathcal{B}$ denotes a path-invariance basis associated with $\mathcal{F}$; p_s is the index of the start vertex of p; f_p^θ is the composite network along path p.

Dataset. The image sets we use are sampled from 12 rigid categories of the PASCAL-Part dataset [6]. To generate image sets that are meaningful to align, we pick the most popular view for each category (who has the smallest variance among 20-nearest neighbors). We then generate an image set for that category by collecting all images whose poses are within 30° of this view. We construct the map network by connecting each image with 20-nearest neighbors with respect to the DSP matching score [24]. Note that the resulting $\mathcal{F}$ is a directed graph as DSP is directed. The longest path varies between 4(Car)-6(Boat) in our experiments.

Evaluation setup. We compare our approach with Congealing [28], Collection Flow [23], RASL [36], and FlowWeb [57]. Note that both Flowweb and our approach use DSP as input. We also compare our approach against [56] under a different setup (See supplementary material). To run baseline approaches, we follow the protocol of [57] to further break each dataset into smaller ones with maximum size of 100. In addition, we consider two variants of our approach: Ours-Dense and Ours-Undirected. Ours-Dense uses the clique graph for $\mathcal{F}$. Ours-Undirected uses an undirected knn-graph, where each edge weight averages the bi-directional DSP matching scores (c.f. [24]). We employ the PCK measure [51], which reports the percentage of

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
Congealing	0.13	0.24	0.05	0.21	0.22	0.11	0.09	0.05	0.14	0.09	0.10	0.09	0.13
RASL	0.18	0.20	0.05	0.36	0.33	0.19	0.14	0.06	0.19	0.13	0.14	0.29	0.19
CollectionFlow	0.17	0.18	0.06	0.33	0.31	0.15	0.15	0.04	0.12	0.11	0.10	0.11	0.12
DSP	0.19	0.33	0.07	0.21	0.36	0.37	0.12	**0.07**	0.19	0.13	0.15	0.21	0.20
FlowWeb	0.31	0.42	**0.08**	0.37	0.56	0.51	0.12	0.06	0.23	0.18	0.19	0.34	0.28
Ours-Dense	0.29	0.42	0.07	0.39	0.53	0.55	0.11	0.06	0.22	0.18	0.21	0.31	0.28
Ours-Undirected	0.32	0.43	0.07	0.43	0.56	0.55	0.18	0.06	0.26	0.21	**0.25**	0.37	0.31
Ours	**0.35**	**0.45**	0.07	**0.45**	**0.63**	**0.62**	**0.19**	0.06	**0.27**	**0.22**	0.23	**0.38**	**0.33**

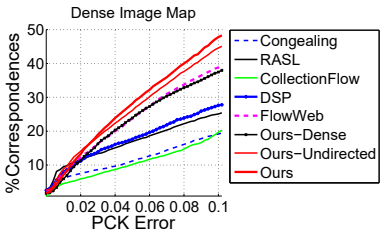

Figure 4: (Left) Keypoint matching accuracy (PCK) on 12 rigid PASCAL VOC categories ($\alpha = 0.05$). Higher is better. (Right) Plots of the mean PCK of each method with varying α

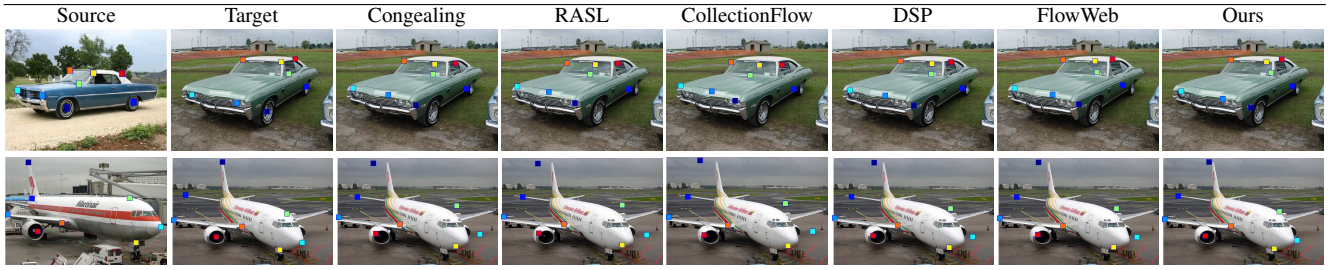

Figure 5: Visual comparison between our approach and state-of-the-art approaches. This figure is best viewed in color, zoomed in. More examples are included in the supplemental material.

keypoints whose prediction errors fall within $\alpha \cdot \max(h, w)$ (h and w are image height and width respectively).

Analysis of results. As shown in Figure 4 and Figure 5, our approach outperforms all existing approaches across most of the categories. Several factors contribute to such improvements. First, our approach can jointly optimize more images than baseline approaches and thus benefits more from the data-driven effect of joint matching [15, 7]. This explains why all variants of our approach are either comparable or superior to baseline approaches. Second, our approach avoids fitting a neural network directly to dissimilar images and focuses on relatively similar images (other maps are generated by map composition), leading to additional performance gains. In fact, all existing approaches, which operate on sub-groups of similar images, also implicitly benefit from map composition. This explains why FlowWeb exhibits competing performance against Ours-Dense. Finally, Ours-Directed is superior to Ours-Undirected. This is because the outlier-ratio of f_{ij}^{in} in Ours-Undirected is higher than that of Ours-Directed, which selects edges purely based on matching scores.

5.3. Map Network of 3D Representations

In the third setting, we seek to jointly optimize a network of neural networks to improve the performance of individual networks. We are particularly interested in the task of semantic segmentation of 3D scenes. Specifically, we consider a network with seven 3D representations (See Figure 1). The first representation is the input mesh. The last representation is the space of 3D semantic segmentations. The second to fourth 3D representations are point clouds with different number of points: PCI (12K), PCII (8K), and PCIII(4K). The motivation of varying the number of points

	PCI	PCII	PCIII	VOLI	VOLII	ENS
100% Label (Isolated)	84.2	83.3	83.4	81.9	81.5	85
8% Label (Isolated)	79.2	78.3	78.4	78.7	77.4	81.4
8% Label + Unlabel (Joint)	82.3	82.5	82.3	81.6	79.0	83.4
30% Label (Isolated)	80.8	81.9	81.2	80.3	79.5	83.2

Table 1: Semantic surface voxel label prediction accuracy on ScanNet test scenes (in percentages), following [37]. We also show the ensembled prediction accuracy with five representations in the last column.

is that the patterns learned under different number of points show certain variations, which are beneficial to each other. In a similar fashion, the fifth and sixth are volumetric representations under two resolutions: VOLI($32 \times 32 \times 32$) and VOLII($24 \times 24 \times 24$). The directed maps between different 3D representations fall into three categories, which are summarized below:

1. Segmentation networks. We use PointNet++ [37] and 3D U-Net[9] for the segmentation networks under point cloud and volumetric representations, respectively.

2. Pointcloud sub-sampling maps. We have six pointcloud sub-sampling maps among the mesh representation (we uniformly sample 24K points using [32]) and three point cloud representations. For each point sub-sampling map, we force the down-sampled point cloud to align with the feature points of the input point cloud [35]. Note that this down-sampled point cloud is also optimized through a segmentation network to maximize the segmentation accuracy.

3. Generating volumetric representations. Each volumetric representation is given by the signed-distance field (or SDF) described in [45]. These SDFs are precomputed.

Experimental setup. We have evaluated our approach on ScanNet semantic segmentation benchmark [12]. Our goal

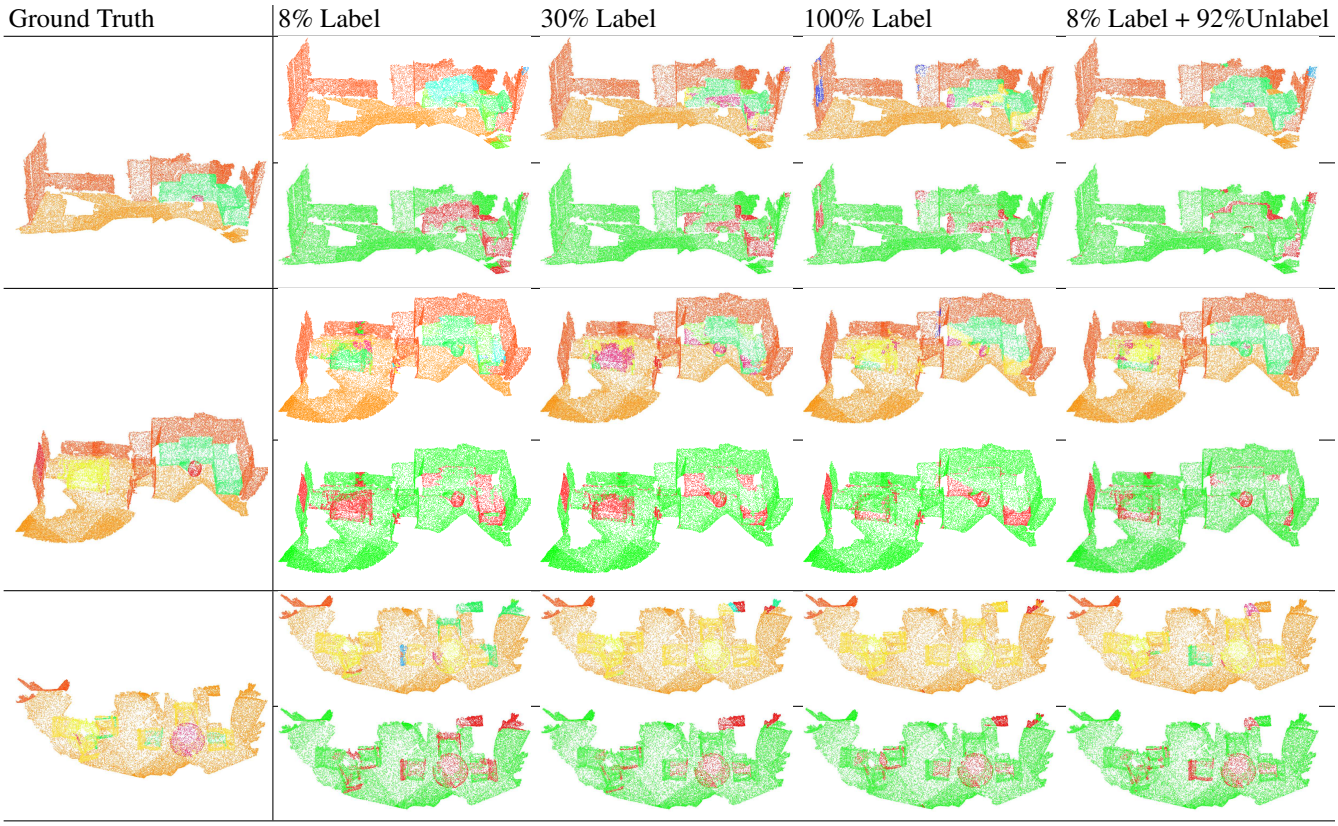

Figure 6: Qualitative comparisons of 3D semantic segmentation results on ScanNet [12]. Each row represents one testing instance, where ground truth and top sub-row show prediction for 21 classes and bottom sub-row only shows correctly labeled points. (Green indicates correct predictions, while red indicates false predictions.) This figure is best viewed in color, zoomed in.

is to evaluate the effectiveness of our approach when using a small labeled dataset and a large unlabeled dataset. To this end, we consider three baseline approaches, which train the segmentation network under each individual representation using 100%, 30%, and 8% of the labeled data. We then test our approach by utilizing 8% of the labeled data, which defines the data term in (3), and 92% of the unlabeled data, which defines the regularization term of (3). We initialize the segmentation network for point clouds using uniformly sampled points trained on labeled data. We then fine-tune the entire network using both labeled and unlabeled data. Note that unlike [55], our approach leverages essentially the same labeled data but under different 3D representations. The boost in performance comes from unlabeled data. Code is publicly available at https://github.com/zaiweizhang/path_invariance_map_network.

Analysis of results. Figure 6 and Table 1 present qualitative and quantitative comparisons between our approach and baselines. Across all 3D representations, our approach leads to consistent improvements, demonstrating the robustness of our approach. Specifically, when using 8% labeled data and 92% unlabeled data, our approach achieves competing performance as using 30% to 100% labeled data when trained on each individual representation. Moreover, the accuracy on VOLI is competitive against using 100% of labeled data, indicating that the patterns learned under the point cloud representations are propagated to train the volumetric representations. We also tested the performance of applying popular vote [39] on the predictions of using different 3D representations. The relative performance gains remain similar (See the last column in Table1). Please refer to Appendix C for more experimental evaluations and baseline comparisons.

6. Conclusions

We have studied the problem of optimizing a directed map network while enforcing the path-invariance constraint via path-invariance bases. We have described an algorithm for computing a path-invariance basis with polynomial time and space complexities. The effectiveness of this approach is demonstrated on three groups of map networks with diverse applications.

Acknowledgement. Qixing Huang would like to acknowledge support from NSF DMS-1700234, NSF CIP-1729486, NSF IIS-1618648, a gift from Snap Research and a GPU donation from Nvidia Inc. Xiaowei Zhou is supported in part by NSFC (No. 61806176) and Fundamental Research Funds for the Central Universities.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, Oct. 2011. 1

[2] Federica Arrigoni, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition. *Computer Vision and Image Understanding*, 174:95–113, 2018. 2

[3] Chandrajit Bajaj, Tingran Gao, Zihang He, Qixing Huang, and Zhenxiao Liang. Smac: Simultaneous mapping and clustering via spectral decompositions. In *ICML*, pages 100–108, 2018. 1

[4] Jrgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - theory, algorithms and applications*. Springer, 2002. 4

[5] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *ICCV*, pages 521–528. IEEE Computer Society, 2013. 1, 2

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1979–1986. IEEE Computer Society, 2014. 6

[7] Yuxin Chen, Leonidas J. Guibas, and Qi-Xing Huang. Near-optimal joint object matching via convex relaxation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 100–108, 2014. 1, 2, 7

[8] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1201–1210, 2015. 1

[9] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016. 7, 16

[10] Luca Cosmo, Emanuele Rodolà, Andrea Albarelli, Facundo Mémoli, and Daniel Cremers. Consistent partial matching of shape collections via sparse modeling. *Comput. Graph. Forum*, 36(1):209–221, 2017. 5, 6

[11] David J. Crandall, Andrew Owens, Noah Snavely, and Daniel P. Huttenlocher. Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2841–2853, 2013. 1

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 1, 2, 7, 8, 16, 19

[13] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track, 2007. 5

[14] Qi-Xing Huang, Yuxin Chen, and Leonidas J. Guibas. Scalable semidefinite relaxation for maximum A posterior estimation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 64–72, 2014. 2

[15] Qixing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, pages 177–186, 2013. 1, 2, 3, 5, 6, 7

[16] Qixing Huang, Fan Wang, and Leonidas J. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph.*, 33(4):36:1–36:11, 2014. 1, 2, 5, 6

[17] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. Graph.*, 25(3):569–578, July 2006. 1, 2

[18] Qi-Xing Huang, Guo-Xin Zhang, Lin Gao, Shi-Min Hu, Adrian Butscher, and Leonidas Guibas. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph.*, 31(6):167:1–167:11, Nov. 2012. 1

[19] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. In *NIPS*, page to appear, 2017. 1, 2

[20] Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image Vision Comput.*, 21(7):637–650, 2003. 1, 2

[21] Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Vigas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. 2

[22] Telikepalli Kavitha, Christian Liebchen, Kurt Mehlhorn, Dimitrios Michail, Romeo Rizzi, Torsten Ueckerdt, and Katharina A. Zweig. Survey: Cycle bases in graphs characterization, algorithms, complexity, and applications. *Comput. Sci. Rev.*, 3(4):199–243, Nov. 2009. 2, 4, 6

[23] Ira Kemelmacher-Shlizerman and Steven M Seitz. Collection flow. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1792–1799. IEEE, 2012. 6

[24] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, pages 2307–2314. IEEE Computer Society, 2013. 1, 6

[25] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Stephen DiVerdi, and Thomas Funkhouser. Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. Graph.*, 31(4):54:1–54:11, July 2012. 1, 5

[26] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 79:1–79:12, New York, NY, USA, 2011. ACM. 5, 6, 17

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5, 15, 16

[28] Erik G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(2):236–250, Feb. 2006. 6

[29] Spyridon Leonardos, Xiaowei Zhou, and Kostas Daniilidis. Distributed consistent data association via permutation synchronization. In *ICRA*, pages 2645–2652. IEEE, 2017. 1

[30] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, May 2011. 1

[31] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas J. Guibas. An optimization approach to improving collections of shape maps. *Comput. Graph. Forum*, 30(5):1481–1491, 2011. 1, 2, 5

[32] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, Oct. 2002. 7

[33] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics*, 31(4), 2012. 5

[34] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1860–1868. Curran Associates, Inc., 2013. 2

[35] Mark Pauly, Richard Keiser, and Markus Gross. Multi-scale Feature Extraction on Point-Sampled Surfaces. *Computer Graphics Forum*, 2003. 7

[36] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2233–2246, Nov. 2012. 6

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5105–5114, 2017. 7, 16

[38] Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, pages 4119–4128. IEEE Computer Society, 2018. 2

[39] Lior Rokach. Ensemble-based classifiers. *Artif. Intell. Rev.*, 33(1-2):1–39, Feb. 2010. 8

[40] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1939–1946. IEEE Computer Society, 2013. 1

[41] Michael Rubinstein, Ce Liu, and William T. Freeman. Joint inference in weakly-annotated image datasets via dense correspondence. *Int. J. Comput. Vision*, 119(1):23–45, Aug. 2016. 1

[42] Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 225–233, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 6

[43] Yanyao Shen, Qixing Huang, Nati Srebro, and Sujay Sanghavi. Normalized spectral map synchronization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4925–4933. Curran Associates, Inc., 2016. 1, 2

[44] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006. 1

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[46] Robert Endre Tarjan. Edge-disjoint spanning trees and depth-first search. *Acta Inf.*, 6(2):171–185, June 1976. 4

[47] Fan Wang, Qixing Huang, and Leonidas J. Guibas. Image co-segmentation via consistent functional maps. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 849–856, Washington, DC, USA, 2013. IEEE Computer Society. 1, 5

[48] Fan Wang, Qixing Huang, Maks Ovsjanikov, and Leonidas J. Guibas. Unsupervised multi-class joint image segmentation. In *CVPR*, pages 3142–3149. IEEE Computer Society, 2014. 1

[49] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2:145–193, Dec. 2013. 2

[50] Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM Trans. Graph.*, 31(6):165:1–165:10, Nov. 2012. 5, 17

[51] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2878–2890, Dec. 2013. 6

[52] Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876. IEEE Computer Society, 2017. 1, 2

[53] Yuxin Chen and Emmanuel Candes. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *https://arxiv.org/abs/1609.05820*, 2016. 2

[54] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433. IEEE Computer Society, 2010. 2, 6

[55] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722. IEEE Computer Society, 2018. 2, 8

[56] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qi-Xing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 117–126, 2016. 1, 2, 6, 15, 16

[57] Tinghui Zhou, Yong Jae Lee, Stella X. Yu, and Alexei A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, pages 1191–1200. IEEE Computer Society, 2015. 1, 2, 6

[58] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*,

pages 4032–4040, Santiago, Chile, 2015. IEEE Computer Society. 1, 2, 6

[59] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251. IEEE Computer Society, 2017. 1, 2