

Fine-Grained Semi-Supervised Labeling of Large Shape Collections

Qi-Xing Huang Hao Su Leonidas Guibas

Stanford University

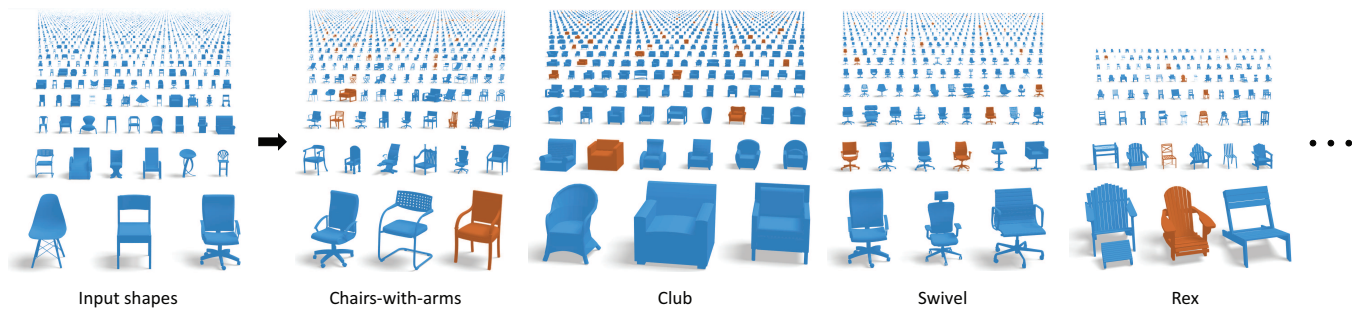


Figure 1: The proposed approach takes a large set of shapes with sparse and noisy labels as input; it outputs cleaned and complete labels for each shape, facilitating organization and search of the shape collection. Labeled chair sets are shown, with training shapes in orange.

Abstract

In this paper we consider the problem of classifying shapes within a given category (e.g., chairs) into finer-grained classes (e.g., chairs with arms, rocking chairs, swivel chairs). We introduce a multi-label (i.e., shapes can belong to multiple classes) semi-supervised approach that takes as input a large shape collection of a given category with associated sparse and noisy labels, and outputs cleaned and complete labels for each shape. The key idea of the proposed approach is to jointly learn a distance metric for each class which captures the underlying geometric similarity within that class, e.g., the distance metric for swivel chairs evaluates the global geometric resemblance of chair bases. We show how to achieve this objective by first geometrically aligning the input shapes, and then learning the class-specific distance metrics by exploiting the feature consistency provided by this alignment. The learning objectives consider both labeled data and the mutual relations between the distance metrics. Given the learned metrics, we apply a graph-based semi-supervised classification technique to generate the final classification results.

In order to evaluate the performance of our approach, we have created a benchmark data set where each shape is provided with a set of ground truth labels generated by Amazon’s Mechanical Turk users. The benchmark contains a rich variety of shapes in a number of categories. Experimental results show that despite this variety, given very sparse and noisy initial labels, the new method yields results that are superior to state-of-the-art semi-supervised learning techniques.

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

Keywords: shape matching, semi-supervised learning, distance learning, multi-label classification, noisy labels, fine-grained

Links: [DL](#) [PDF](#)

1 Introduction

Shape classification is a fundamental problem in shape analysis. So far most existing works have focused on classifying shapes into different high-level categories, e.g., cars, chairs, desks, etc. With the emergence of large shape collections, however, even the shapes within each category still exhibit significant variation. For example, chair models from the Trimble 3D Warehouse contain dozens of sub-classes, including chairs-with-arms, swivel chairs, rocking chairs, etc. (See Figure 1). Classifying shapes into these fine-grained classes can benefit a variety of applications such as product search, browsing and exploration of shape variability, and interactive shape modeling.

In this paper, we consider a semi-supervised problem setting, where the given input is a set of man-made shapes together with associated sparse and noisy labels (e.g., models from Trimble 3D Warehouse and their associated text), and the output consists of cleaned and complete labels for each input shape. This problem is particularly challenging due to (1) relatively subtle geometric differences between different classes, (2) the availability of only very sparse and often quite noisy labels, (3) the fact that each shape can be associated with multiple labels, and finally (4) the size of the problem, as a shape collection will typically contain thousands of models.

The proposed approach addresses these challenges by combining two simple ideas motivated from recent advances in geometry processing and machine learning. First, inspired by current interest in data-driven shape matching, [Kim et al. 2012; Huang et al. 2012; Kim et al. 2013], we propose to align the input shapes of a given category into a common space, thus implicitly generating a set of correspondences between the shapes. This common space provides us with a convenient framework in which to compare shapes, making features across different shapes more consistent and comparable. For example, the common space allows us to focus at particular neighborhoods of that space and examine local shape variations in those neighborhoods under appropriate similarity metrics. To handle large datasets with high shape variability, we introduce a scalable shape matching approach that is able to simultaneously align many thousands of diverse shapes.

Given the aligned shapes in this common space, a straightforward approach for classification would be to train a classifier for each class, either jointly or independently. However, we found that the performance of such approaches is rather unsatisfactory. This is because, typically, a complex decision boundary has to be learned in order to capture the large geometric variations and yet subtle geometric similarities within each class, a task made harder to accomplish in the presence of sparse and noisy labels. This leads to the second idea of the proposed approach, which combines distance learning [Yang and Jin 2006], which is less sensitive to problematic labels but does not directly produce classification results, and graph based semi-supervised classification [Zhu 2006], which employs unlabeled data to determine the decision boundaries but requires high quality similarity graphs. Specifically, we first jointly learn a distance metric for each class to capture its underlying geometric similarity, e.g., that rocking chairs have similar bases. These distance metrics are then used to construct a similarity graph for each class, on which we finally perform graph-based semi-supervised classification—and we do so jointly over all classes.

We have created a benchmark dataset to evaluate the proposed technique. The benchmark consists of three categories of shapes: cars, chairs, and airplanes, all selected from the Trimble 3D Warehouse. Each category has 2K-6K models, dozens of fine-grained classes, as well as ground-truth labels provided by human experts. We ran the proposed pipeline on sparse labels associated with the input shapes. Experimental results show that the above approach generates results that are superior to state-of-the-art semi-supervised classification methods. We also compared each step of the proposed approach with various alternatives to verify and support our design decisions.

2 Background and Prior Work

2.1 Computer Graphics and Vision

3D shape classification. Shape classification has been studied extensively in the past. We refer the reader to [da Fontoura Costa and Cesar Jr. 2009] as a standard reference for this topic. When classifying large shape collections, most existing works have focused on computing meaningful global shape descriptors (see e.g., [Osada et al. 2002; Kazhdan et al. 2003; Chen et al. 2003]). These global descriptors have proven to be successful for classifying and differentiating shapes from different categories, e.g., chairs and airplanes. However, they are less effective in classifying shapes within the same category, where shapes are typically distinguished by subtle partial or local geometric features.

For shape collections of moderate size, Xu et al. [2010] introduced an unsupervised method for classifying a shape collection into groups of different styles, where shapes in each group are geometrically similar after appropriate part scaling. Recently, Kalogerakis et al. [2012] introduced a probabilistic part-based shape grammar for the purpose of synthesizing new shapes. The shape grammar encodes each shape part using a type set, which consists of parts of different shapes. These type sets are learned from the input shapes in an unsupervised manner. In contrast to these two techniques, we focus on a different problem, whose goal is to classify shapes into human-recognizable fine-grained classes.

Fine-grained classification of images. We note that fine-grained classification/categorization has been popular in the computer vision community in the last few years (e.g., [Loeff et al. 2009; Yao et al. 2011; Deng et al. 2013] and the references therein). The key idea of these approaches is to learn class specific features, which classify the instances of each class. The major difference in our approach is that we learn class specific distance metrics, which are

more robust against noisy and sparse labels. The classification is performed as a separate process. In addition, the representation of 3D shapes is very different from images, and color and texture-based methods do not immediately transfer to 3D geometry. On the other hand, we believe that a 3D approach, which can utilize more complete information about objects, has the potential to generate better results than image-based techniques.

Shape matching. Matching multiple shapes is fundamental problem in geometry processing. Despite some recent advances [Huber 2002; Crandall et al. 2011; Kim et al. 2012; Huang et al. 2012; Kim et al. 2013] on this topic, we found that it is extremely difficult to obtain high-quality correspondences among a shape collection of thousands of shapes. Our approach modifies existing approaches so that they become scalable and effective on large shape collections. Specifically, we introduce a reduced affine transformation model in which the MRF formulation described in [Crandall et al. 2011; Huang et al. 2012] can be applied to globally match large collections of man-made objects. For the local alignment of multiple shapes [Huber 2002], we introduce an objective function that admits an efficient alternating optimization.

2.2 Machine Learning

Semi-supervised learning. Semi-supervised learning addresses the case where the labeled data is sparse. We refer to [Zhu 2006] for a survey on this topic and to [Fergus et al. 2009; Liu et al. 2012] for some recent advances. Roughly speaking, semi-supervised techniques fall into two categories: inductive or transductive [Zhu 2006]. Inductive methods typically extend their supervised counterparts to incorporate unlabeled data. In contrast, transductive methods focus on the input database by propagating labels from labeled data to unlabeled data. Most transductive methods are graph-based, where the propagation naturally happens along graph edges.

Very recently, Wang et al. [2012] introduced semi-supervised learning to the graphics community. They developed a shape segmentation framework which can significantly improve the quality of segmentations among a shape collection using a sparse set of user-specified constraints, i.e., the label sets. In contrast, we apply semi-supervised learning to perform multi-label shape classification.

Multi-label classification. Multi-label classification has drawn a lot of interest in machine learning research for the last several years. We refer to [Tsoumakas and Katakis 2007] for an introduction to this topic. The proposed approach is mostly related to [Amit et al. 2007], which performs multi-label classification of images by jointly learning linear classifiers for each class. In particular, Loeff et al. [2009] extend this approach to the semi-supervised setting by considering a unified similarity graph. In practice, we found that using one similarity graph is insufficient as different classes possess different types of geometric similarities. Multi-label classification has also been studied in the graph-based semi-supervised setting (e.g., [Chen et al. 2008]). However, these approaches are still limited because they utilize a unified similarity graph to propagate label information.

Distance learning. Distance metric learning [Yang and Jin 2006] is another active branch of machine learning. Most methods create a similar set and a dissimilar set. A distance metric is learned to minimize the pair-wise distances within the similar set while maximizing the distances from the dissimilar set, subject to various regularization constraints on the metric. Distance metric learning can also be performed in semi-supervised fashion [Baghshah and Shouraki 2009; Hoi et al. 2010]. Our approach applies the general idea of distance metric learning, but is designed based on the specific problem we are solving.

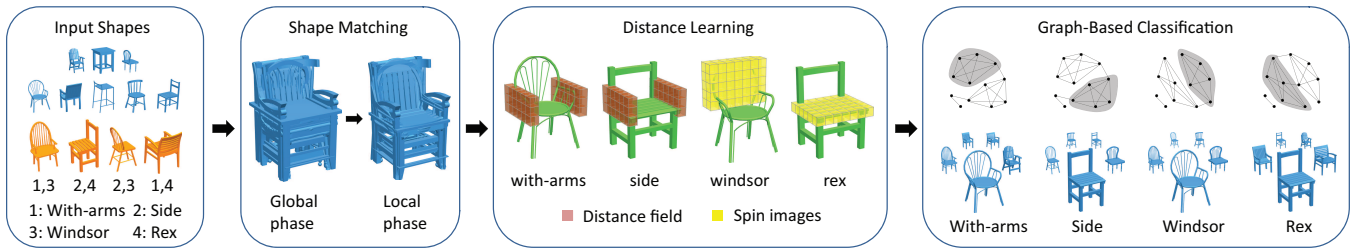


Figure 2: Overview of the shape classification pipeline. The input consists of a shape collection and a sparse labeled set (colored in gold). In the first stage, the input shapes are aligned into a common space. In the second stage, a distance metric is learned for each class to differentiate shapes within this class from other shapes. In the third stage, a similarity graph is constructed for each class and graph-based semi-supervised classification is jointly performed on all graphs to generate the optimal shape set for each class.

3 Overview

The input to the proposed approach consists of:

- A collection of shapes of the same category $\mathcal{S} = \{S_i | 1 \leq i \leq N\}$. Based on the characteristics of 3D Warehouse models, we assume the up-right direction is aligned with the z axis of a world coordinate system Σ . We also normalize each model in Σ so that its bounding box is centered at $(0, 0, 0.5)$, and its ground plane is $z = 0$. As most of man-made objects are reflectionally symmetric, we further assume the reflectional axis of one shape, denoted as S_1 , is aligned with the x axis of Σ .
- A set of object classes $\{c_j | 1 \leq j \leq M\}$ and the corresponding labeled shape sets $\{\mathcal{L}_j^m \subseteq \mathcal{S} | 1 \leq j \leq M\}$ to be used for training. In general we assume that these sets are small compared to \mathcal{S} .

The output consists of:

- The classified shape sets $\mathcal{L}_j^{\text{opt}} \subset \mathcal{S}$ for each class c_j , where $1 \leq j \leq M$. As a shape may be given multiple labels, the sets $\mathcal{L}_j^{\text{opt}}$ corresponding to different classes may overlap.

As illustrated in Figure 2, the proposed approach proceeds via the following three stages. We elaborate on the technical details of these stages from Section 4 to Section 6, respectively.

Shape matching. The first stage aligns the input shapes in the common space Σ , so that corresponding parts on different shapes can be easily compared. We divide this stage into a global phase and a local phase. In the global phase, we jointly compute an affine transformation T_i for each shape S_i so that in the end all shapes are roughly aligned in Σ . This is done by following the principal two-step strategy of matching multiple shapes as in [Huber 2002], where the first step performs pair-wise affine matching to construct a similarity graph \mathcal{G} among the input shapes along with associated relative transformations $T_{(i,j)}$, $(i, j) \in \mathcal{G}$, and the second step jointly computes an affine transformation T_i for each shape by optimizing the consistency between the induced transformations $T_j^{-1} \circ T_i$ and the relative transformations $T_{(i,j)}$. Among existing formulations to this problem, we extend the MRF formulation described in [Crandall et al. 2011; Huang et al. 2012] for our purposes, due to its ability to handle noisy relative transformations. The efficiency of this formulation relies on effectively sampling the transformation space of each shape. To address this issue, we introduce a reduced affine transformation model, which is sufficient to provide an initial starting point for the local phase, and which enables us to perform the MRF optimization for each type of 1D transformation (e.g. the rotation in the xy -plane) in a sequential manner. In this case, we only need to sample a 1D space per-shape in each subproblem.

In the local phase, we proceed to jointly optimize a free-from deformation [Sederberg and Parry 1986] \mathcal{F}_i for each shape S_i to improve

the alignment. To avoid simultaneously optimizing the deformations of all input shapes in large shape collections, we introduce an objective function, which can be optimized in an alternating manner. In particular, at each step the deformation \mathcal{F}_i of each shape can be optimized separately.

Distance learning. In the second stage we jointly learn a distance metric for each class to differentiate shapes within the same class and shapes from different classes. Taking the advantage that the input shapes are already aligned in Σ , we present a linear model in Σ to parameterize distance metrics, i.e., a distance metric is a linear combination of primitive distance metrics, each of which compares shapes in terms of a pre-defined feature descriptor (e.g., spin images [Johnson and Hebert 1999]) and at a spatial location in Σ .

We formulate distance learning as solving an optimization problem that incorporates various objective terms. Similar to standard distance learning techniques [Yang and Jin 2006], we construct similar sets (i.e. pairs of shapes in the same class) and dissimilar sets (i.e., pairs of shapes that belong to different classes) from labeled shapes, and formulate objective terms that minimize(maximize) distances between shape pairs in similar(dissimilar) sets. To handle sparse and noisy labels, we introduce two regularization terms that are derived from analyzing the structure of desirable distance metrics. The first term enforces the consistency of the coefficients of each distance metric, and the second term considers the mutual correlations among all distance metrics. We demonstrate how to formulate these objective terms so that the resulting optimization problem is convex, leading to a global solution. To further improve the quality of the optimized distance metrics, we perform an alternating procedure, where we use the optimized distance metrics to polish the similar and dissimilar sets at each iteration.

Shape classification. In the third and final stage we use the learned distance metrics to construct a similarity graph for each class, and apply graph-based semi-supervised classification to obtain the optimal shape set for each class. To avoid optimizing the association between every shape and every class [Chen et al. 2008], which is inefficient for large-scale datasets, we propose to first pre-compute a candidate set of classifications for each class by performing graph decompositions on the corresponding similarity graph with varying parameters, and then jointly select the best classification. Note that this strategy is particularly efficient because the graph decompositions are performed independently for each class, and the joint optimization is performed on candidate sets, whose sizes are much smaller than the full input shape collection.

In summary, while at a very high level our shape alignment followed by label propagation strategy is based on familiar ideas from graphics and vision, wide shape variations within a category, noisy training labels, and the scale of the problem, have led us to innovate at every step of the way by designing algorithms whose performance have proved essential to the quality of the results we obtain.

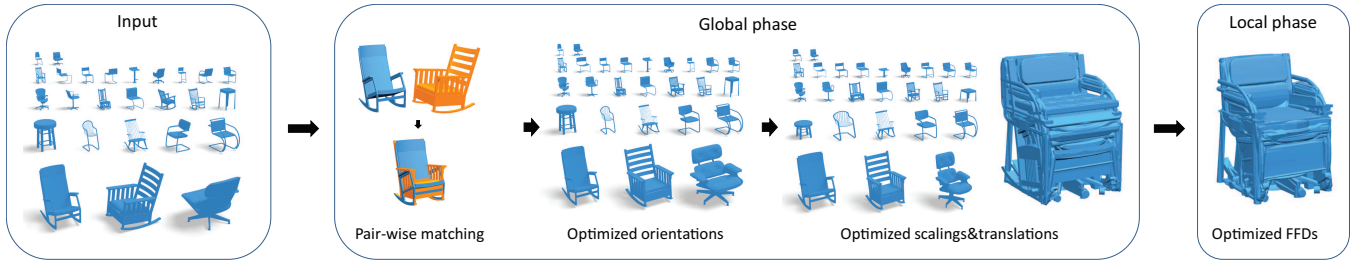


Figure 3: Shape matching procedure. This figure shows the shape matching pipeline, consisting of a global phase followed by a local phase. In the global phase, we first identify pairs of similar shapes and compute optimal affine transformations between them. Then, to embed the shapes into a common space, we apply sequential joint optimizations to optimize their orientations, scalings and translations, in that order. In the final local phase, we optimize a FFD for each shape to refine and improve the alignment.

4 Shape Matching

The proposed classification pipeline begins with aligning all the input shapes. We divide this stage into a global affine matching phase and then a local non-rigid alignment phase (See Figure 3).

4.1 Global Affine Matching

We formulate global affine matching as solving a discrete MRF, which jointly optimizes the transformation of each shape T_i within a discrete set of transformation samples. To make this formulation tractable, i.e, to maintain a small sample set for each shape, we consider a reduced transformation model, under which this MRF optimization can be performed for each type of elementary transformation (e.g., the rotation in the xy -plane) independently.

Reduced transformation model. The reduced deformation model is based on the assumption that we match shapes by first aligning their front orientations and then performing appropriate translation and scaling along each axis. Specifically, we parameterize the affine transformation $T_i : (x, y, z) \in S_i \rightarrow (x', y', z') \in \Sigma$ of each shape S_i using a rotation matrix $R(\theta_i) = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix}$ (i.e., specifying the front orientation with respect to Σ) and a translation $\mathbf{t}_i = (t_i^x, t_i^y)^T$ in the xy -plane, and three scalings (s_i^x, s_i^y, s_i^z) :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_i^x & 0 \\ 0 & s_i^y \end{pmatrix} R(\theta_i) \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_i^x \\ t_i^y \end{pmatrix}, \quad z' = s_i^z z.$$

Accordingly, we represent a relative affine transformation $T_{(i,j)} : (x, y, z) \in S_i \rightarrow (x', y', z') \in S_j$ using 7 parameters:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = S_{(i,j)} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{t}_{(i,j)}, \quad z' = s_{(i,j)}^z z.$$

Here $S_{(i,j)}$ is a 2×2 matrix. Let $S_{(i,j)} = U_{(i,j)} \Lambda_{(i,j)} V_{(i,j)}^T$ be the SVD of $S_{(i,j)}$. It is easy to see that the constraint $T_j^{-1} \circ T_i = T_{(i,j)}$ can be expressed via the following decoupled constraints:

$$\begin{aligned} R(\theta_i - \theta_j) &= U_{(i,j)} V_{(i,j)}^T, \quad s_i^z / s_j^z = s_{(i,j)}^z, \\ \begin{pmatrix} s_i^x / s_j^x & 0 \\ 0 & s_i^y / s_j^y \end{pmatrix} &= R(\theta_j) S_{(i,j)} R(-\theta_i) := \begin{pmatrix} S_{(i,j),\theta}^{xx} & S_{(i,j),\theta}^{xy} \\ S_{(i,j),\theta}^{yx} & S_{(i,j),\theta}^{yy} \end{pmatrix}, \\ \mathbf{t}_i - \mathbf{t}_j &= \begin{pmatrix} s_i^x & 0 \\ 0 & s_i^y \end{pmatrix} R(\theta_j) \mathbf{t}_{(i,j)} := \begin{pmatrix} t_{(i,j),\theta,s}^x \\ t_{(i,j),\theta,s}^y \end{pmatrix}, \end{aligned} \quad (1)$$

where $S_{(i,j),\theta}^{xx}$ and $t_{(i,j),\theta,s}^x$ are introduced to simplify the notations.

Constructing \mathcal{G} via pairwise matching. We adopt a variant of the procedure described in [Kim et al. 2012] for constructing the similarity graph \mathcal{G} , i.e., using descriptor-based nearest neighbor computations and then estimating the associated transformations

using RANSAC. As these steps are rather standard, we leave the details in the supplemental material.

Joint matching via MRF optimization. Based on Equation 1, we decouple the optimization of T_i into the optimizations of $\{\theta_i\}$, $\{s_i^x\}$, $\{s_i^y\}$, $\{s_i^z\}$ and $\{t_i^x\}$ and $\{t_i^y\}$ in this order. For each subproblem, we place $K = 32$ transformation samples per shape (see the table below for details). Let $f : \{1, \dots, N\} \rightarrow \{1, \dots, K\}$ be the map that picks a transformation sample for each shape. We compute the optimal map f^* (which provides the optimized transformations) by solving the following MRF problem:

$$f^* = \arg \max_f \sum_{(i,j) \in \mathcal{G}} \exp(-Q_{ij;f(i)f(j)}), \quad (2)$$

where term $Q_{ij;f(i)f(j)}$ evaluates the difference between the induced transformation and the corresponding relative transformation. The table below specifies the form of Q in each case.

Samples	$Q_{ij;f(i)f(j)}$
$\theta_{i,f(i)} = 2\pi f(i)/K$	$\ R(\theta_{i,f(i)} - \theta_{j,f(j)}) - U_{(i,j)} V_{(i,j)}^T\ _{\mathcal{F}}$
$s_{i,f(i)}^x = \exp(2f(i)/K - 1)$	$2 s_{i,f(i)}^x - s_{j,f(j)}^x S_{(i,j),\theta}^{xx} $
$s_{i,f(i)}^y = \exp(2f(i)/K - 1)$	$2 s_{i,f(i)}^y - s_{j,f(j)}^y S_{(i,j),\theta}^{yy} $
$s_{i,f(i)}^z = \exp(2f(i)/K - 1)$	$2 s_{i,f(i)}^z - s_{j,f(j)}^z s_{(i,j)}^z $
$t_{i,f(i)}^x = 2(2f(i)/K - 1)$	$4 t_{i,f(i)}^x - t_{j,f(j)}^x - t_{(i,j),\theta,s}^x $
$t_{i,f(i)}^y = 2(2f(i)/K - 1)$	$4 t_{i,f(i)}^y - t_{j,f(j)}^y - t_{(i,j),\theta,s}^y $

We solve Equation 2 using the iterative coordinate ascent method described in [Leordeanu and Hebert 2006] due to its simplicity and efficiency. As $Q_{ij;f(i)f(j)}$ only provides relative constraints, we fix $f(1)$ in each subproblem so that T_1 is the identity transformation.

4.2 Local non-rigid registration

In the local phase, we start from the roughly aligned shapes, then for each shape S_i , we optimize a free-form deformation (FFD) \mathcal{F}_i [Sederberg and Parry 1986] to further refine the alignment. Following [Huber 2002], we formulate this step as minimizing the sum of distances between pairs of aligned shapes specified by \mathcal{G} . To formulate the objective function, we first perform pair-wise registration [Li et al. 2008] to establish a set of corresponding point pairs $(\mathbf{p}_{i'k} \in S_i, \mathbf{q}_{i'k} \in S_{i'})$, $k = 1, \dots, n_{i'}$ between each pair of shapes $(S_i, S_{i'}) \in \mathcal{G}$. Then we setup the objective function to minimize the distances between $\mathbf{p}_{i'k}$ and $\mathbf{q}_{i'k}$. To avoid optimizing FFDs over all shapes simultaneously, we introduce a latent point $\mathbf{m}_{i'k}$ for each point pair $(\mathbf{p}_{i'k}, \mathbf{q}_{i'k})$ and setup the optimization problem as

$$f_{\text{multiple}} = \sum_{(i,i') \in \mathcal{G}} \sum_{k=1}^{n_{i'}} (\|\mathcal{F}_i(\mathbf{p}_{i'k}) - \mathbf{m}_{i'k}\|^2 + \|\mathcal{F}_{i'}(\mathbf{q}_{i'k}) - \mathbf{m}_{i'k}\|^2).$$

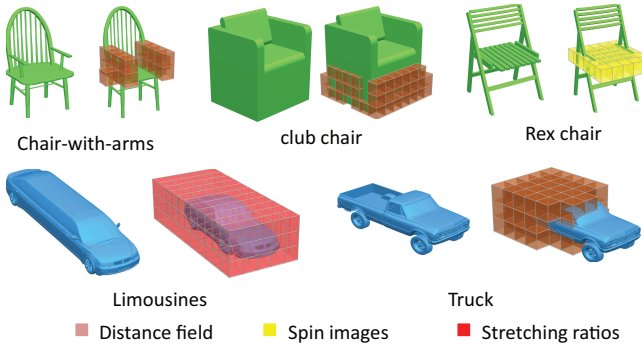


Figure 4: Distance metric gallery. A gallery of the learned distance metrics for various label classes. To illustrate the effectiveness of the method, we highlight the cells contributing most to the distance computation as well as the most influential kernel used there. Note how natural features are automatically selected, e.g., the arms for the chairs, or the loading area of the truck.

The quantity $f_{multiple}$ can be efficiently optimized using alternating optimization. When the \mathcal{F}_i are fixed, we simply set $\mathbf{m}_{i'k} = (\mathbf{p}_{i'k} + \mathbf{q}_{i'k})/2$. When $\mathbf{m}_{i'k}$ are fixed, we can optimize each deformation \mathcal{F}_i independently using [Li et al. 2008].

5 Distance Learning on Aligned Shapes

The central stage of the pipeline is to simultaneously learn a distance metric between aligned shapes for each class c_j , so that each metric captures the underlying geometric similarity of the corresponding class. In the following, we first introduce a linear model for parameterizing the space of distance metrics. Then we present a convex optimization formulation for learning these metrics.

5.1 Linear distance model

We define a distance metric as a linear combination of a set of primitive distance functions, each of which compares a pre-defined volumetric shape descriptor at a spatial location. Specifically, we first voxelize the bounding box of the aligned shapes in Σ . In our implementation, we set the grid size as 0.1. For each cell c and for each type of pre-defined volumetric descriptors $\bar{\mathbf{f}}_{S_i}(\cdot) : \Sigma \rightarrow \mathbb{R}^d, 1 \leq i \leq N$ (to be introduced later in this section), we generate the corresponding primitive distance function as

$$k_c^f(S_i, S_j) = \|\bar{\mathbf{f}}_{S_i}(\mathbf{o}_c) - \bar{\mathbf{f}}_{S_j}(\mathbf{o}_c)\|,$$

where \mathbf{o}_c denotes the center of cell c . Let \mathcal{K} be the collection of all primitive distance functions generated during this process, we define an arbitrary distance metric $d(\cdot, \cdot)$ as

$$d(\cdot, \cdot) = \sum_{k \in \mathcal{K}} x_k k(\cdot, \cdot) = \mathbf{x}^T \mathbf{k}(\cdot, \cdot), \quad \mathbf{x} \geq 0, \quad (3)$$

where $\mathbf{k}(\cdot, \cdot)$ stacks all primitive distance functions in a vector, and \mathbf{x} collects their coefficients.

Volumetric descriptors. For robustness concern, we define each volumetric descriptor $\bar{\mathbf{f}}_{S_i}(\mathbf{x})$ as the surface integral of a surface descriptor $\mathbf{f}_{\mathbf{x}}(\cdot) : S_i \rightarrow \mathbb{R}^d$:

$$\bar{\mathbf{f}}_{S_i}(\mathbf{x}) = \int_{\mathbf{p} \in S_i} e^{-\frac{\|\mathbf{p}-\mathbf{x}\|^2}{2\sigma^2}} \mathbf{f}_{\mathbf{x}}(\mathbf{p}) / \int_{\mathbf{p} \in S_i} e^{-\frac{\|\mathbf{p}-\mathbf{x}\|^2}{2\sigma^2}},$$

where $\sigma = 0.05$. In this paper, we have considered the following surface descriptors (See Figure 4 for their effects):

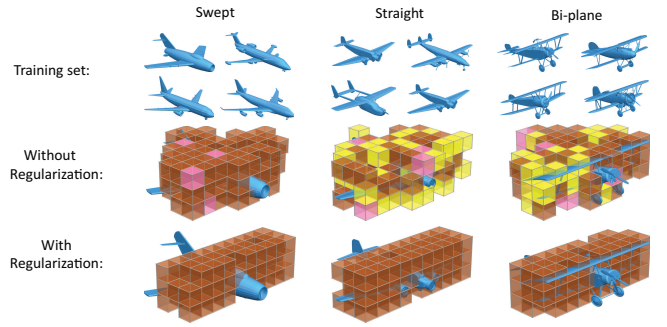


Figure 5: Effect of the regularization terms. We can see that incorporating these terms leads to a more clean and meaningful distance metric than merely optimizing the data term.

- Distance to \mathbf{x} , i.e., $\|\mathbf{p} - \mathbf{x}\|$ for a given point \mathbf{p} . The corresponding descriptor values $\bar{\mathbf{f}}_{S_i}(\mathbf{o}_c)$ effectively define a discrete distance field for each shape S_i . Thus, these primitive distance metrics are expected to compare global geometric similarity.
- Norm of derivatives of \mathcal{F}_i : $\|\frac{\partial \mathcal{F}_i(\mathbf{p})}{\partial x}\|$, $\|\frac{\partial \mathcal{F}_i(\mathbf{p})}{\partial y}\|$ and $\|\frac{\partial \mathcal{F}_i(\mathbf{p})}{\partial z}\|$. These feature vectors are used for classes that have salient local and/or global anisotropic scalings (e.g., Limousines).
- Spin images [Johnson and Hebert 1999], which are used for classes that exhibit local geometric features (e.g., Rex chairs).

5.2 Learning distance metrics

We then jointly learn the distance metric $d_j(\cdot, \cdot) = \mathbf{x}_j^T \mathbf{k}(\cdot, \cdot)$ associated with each class c_j , where $1 \leq j \leq M$. In the following, we first describe the objective function, which consists of a data term and two regularization terms. Then we show how to solve the induced optimization problem. We also present an alternating strategy for updating the data term using the optimized distance metrics.

Data term. Following the principal idea of distance learning [Yang and Jin 2006], we construct for each class c_j a *similar* set $\mathcal{M}_j \subset \mathcal{S} \times \mathcal{S}$ and a *dissimilar* set $\mathcal{D}_j \subset \mathcal{S} \times \mathcal{S}$, which collect pairs of shapes that are expected to have small and large distances with respect to the desired distance metric $d_j(\cdot, \cdot)$, respectively. In our implementation, we initialize both sets from the input shape sets \mathcal{L}_j^{in} :

$$\mathcal{M}_j = \mathcal{L}_j^{in} \times \mathcal{L}_j^{in}, \quad \mathcal{D}_j = \mathcal{L}_j^{in} \times \left(\sum_{j'=1}^M \mathcal{L}_{j'}^{in} \setminus \mathcal{L}_j^{in} \right),$$

As the \mathcal{L}_j^{in} of different classes can overlap, we compute a weight $w_p = 1 - |\mathcal{L}_j^{in} \cap \mathcal{L}_{j'}^{in}| / \max(|\mathcal{L}_j^{in}|, |\mathcal{L}_{j'}^{in}|)$ for each shape pair $p = (S_i \in \mathcal{L}_j^{in}, S_{i'} \in \mathcal{L}_{j'}^{in} \setminus \mathcal{L}_j^{in}) \in \mathcal{D}_j$ to characterize its fuzzy association with \mathcal{D}_j .

The data term is then formulated to minimize the distances between shape pairs in the similar sets, and maximize the distances between shape pairs in the dissimilar sets. In our formulation, we employ the following max-marginal model:

$$f_{data} = \sum_{j=1}^M \left(\frac{1}{|\mathcal{M}_j|} \sum_{p \in \mathcal{M}_j} \mathbf{d}_p^T \mathbf{x}_j + \frac{1}{|\mathcal{D}_j|} \sum_{p \in \mathcal{D}_j} w_p \max(0, 1 - \mathbf{d}_p^T \mathbf{x}_j) \right), \quad (4)$$

where \mathbf{d}_p collects the distances of p with respect to primitive distance functions. Note that due to sparse and noisy input, optimizing the data term alone is typically insufficient (See Figure 5).

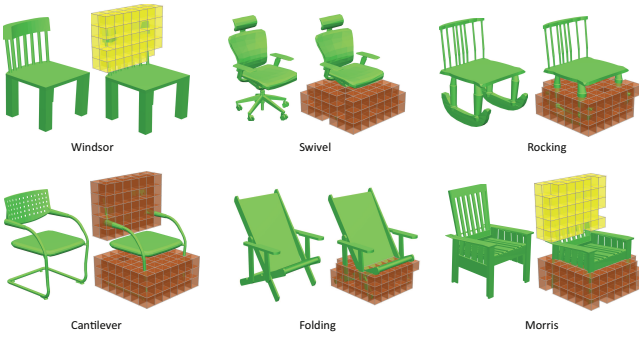


Figure 6: The learned distance metrics on various chair classes. As before, we show the relevant regions and kernels.

Regularization terms. We consider two regularization terms, which are motivated from the properties of desired distance metrics (See Figure 6). The first term f_{coeff} favors the piece-wise constant property, i.e., the coefficients of each desired distance metric remain constant in each of its support region. Intuitively, this forces the coefficients to be determined in groups, which effectively addresses the problem of having sparse and noisy input. Let $\mathcal{N} \subset \mathcal{X} \times \mathcal{X}$ collect pairs of primitive distance metrics defined using the same feature descriptor on neighboring cells. We enforce this piece-wise constant property by minimizing the L1-norm (which prioritizes sparsity) of $\{x_{j,k} - x_{j,k'} | (k, k') \in \mathcal{N}\}$ for each class c_j , where $x_{j,k}$ denotes the coefficient of kernel $k(\cdot, \cdot)$ in $d_j(\cdot, \cdot)$:

$$f_{\text{coeff}} = \sum_{j=1}^M \sum_{(k,k') \in \mathcal{N}} |x_{j,k} - x_{j,k'}| = \sum_{j=1}^M \|J\mathbf{x}_j\|_1, \quad (5)$$

where matrix J is introduced to write down f_{coeff} in the vector form.

The second regularization term f_{rank} considers the mutual relations among the distance metrics [Amit et al. 2007]. In our setting, we assume that there exist a small set of support regions (e.g., the underlying parts), which are shared by all distance metrics. Equivalently, this is to say that the rank of the matrix $X = (\mathbf{x}_1, \dots, \mathbf{x}_M)$ should be minimized. In our formulation, we propose to minimize the nuclear norm [Candès and Recht 2009], which serves a popular convex objective for rank minimization:

$$f_{\text{rank}} = \sum_{k=1}^M \sigma_k(X), \quad (6)$$

where $\sigma_k(X)$, $1 \leq k \leq M$ denote the singular values of matrix X .

As shown in Figure 5, incorporating these two regularization terms leads to significantly improved distance metrics.

Optimization. Combining Equations 4, 5 and 6, we arrive at the following convex problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{j=1}^M \left(\frac{1}{|\mathcal{M}_j|} \sum_{p \in \mathcal{M}_j} \mathbf{d}_p^T \mathbf{x}_j + \frac{1}{|\mathcal{D}_j|} \sum_{p \in \mathcal{D}_j} w_p \max(0, 1 - \mathbf{d}_p^T \mathbf{x}_j) \right) \\ & + \lambda \sum_{j=1}^M \|J\mathbf{x}_j\| + \mu \sum_{j=1}^M \sigma_j(\mathbf{X}), \\ \text{s.t.} \quad & \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_M) \geq 0. \end{aligned} \quad (7)$$

Here parameters λ and μ specify the strength of prior terms. For all of our experiments, we choose $\lambda = 0.5$ and $\mu = 1$, though we also found that the optimal solution is insensitive to these parameters.

For optimization, we employ the alternating directions of augmented multiplier method (ADMM) [Boyd et al. 2011], which has

been proven to be quite effective for solving large-scale convex programs. Please refer to [Boyd et al. 2011] for details.

Updating the data term. In the same spirit as reweighted least squares [Holland and Welsch 1977], we use the optimized distance metrics to update the similar and dissimilar sets used in defining the data term, i.e., the similar and dissimilar sets only include instances that are consistent with the optimized distance metrics:

$$\mathcal{M}_j := \{p | p \in \mathcal{L}_j^{\text{in}} \times \mathcal{S}, \mathbf{d}_p^T \mathbf{x}_p < 2\sigma_j(\mathcal{M}_j)\},$$

$$\mathcal{D}_j := \{p | p \in \mathcal{L}_j^{\text{in}} \times \mathcal{S}, \mathbf{d}_p^T \mathbf{x}_p > 1\},$$

where $\sigma_j(\mathcal{M}_j)$ denotes the medians of d_j among the previous \mathcal{M}_j . We then re-solve Equation 7. This alternating process is iterated until the distance metrics become steady. In practice, we found that 3-5 iterations were sufficient.

6 Graph Based Multi-Label Classification

The final stage of the proposed pipeline employs the learned distance metrics to construct per-class similarity graphs, and performs graph based multi-label classification to extract the classified shapes of each class. To handle large shape collections, we employ a select-from-candidate strategy, i.e., we first generate a set of candidate classifications for each class in isolation via graph partitioning, we then jointly select the optimal classification for each class by solving a MRF [Leordeanu and Hebert 2006].

Similarity Graph Construction. We generate the similarity graph \mathcal{G}_j for each class c_j by connecting k -nearest neighbors ($k = 6$) of each shape with respect to $d_j(\cdot, \cdot)$. The weight associated with each edge $(S_i, S_{i'}) \in \mathcal{G}_j$ is given by

$$w_{(S_i, S_{i'})} = \exp(-d_j^2(S_i, S_{i'})/2\sigma^2),$$

where σ is chosen as the median of $d_j(S_i, S_{i'})$ over all edges in \mathcal{G}_j .

Candidate classifications. We generate candidate classifications by thresholding graph diffusion distances [Coifman et al. 2005] to the labeled shape set, utilizing their power in capturing graph clusters at various scales. Denote $d_{\mathcal{G}_j}^t(S_i, S_{i'})$ as the diffusion distance between S_i and $S_{i'}$ on graph \mathcal{G}_j at scale t (Please refer to [Coifman et al. 2005] for the formula). Given a fixed scale parameter t_j and a distance threshold δ_j , we compute the corresponding candidate classification

$$\mathcal{L}_j = \{S_i | d_{\mathcal{G}_j}^{t_j}(S_i, \mathcal{L}_j^{\text{in}}) = \text{median}\{d_{\mathcal{G}_j}^{t_j}(S_i, S_{i'}) | S_{i'} \in \mathcal{L}_j^{\text{in}}\} < \delta_j\},$$

where the median distance accounts for mislabeled shapes in $\mathcal{L}_j^{\text{in}}$.

To generate all candidate classifications, we first place $L_1 = 8$ uniform samples between 0 and $\text{Diam}(\mathcal{G}_j)$ for the scale parameter t_j . Then for each fixed t_j , we place $L_2 = 8$ uniform samples between 0 and $\max_{S_i \in \mathcal{S}} d_{\mathcal{G}_j}^t(S_i, \mathcal{L}_j^{\text{input}})$ for the distance threshold δ_j . In total, we obtain $L = L_1 L_2 = 64$ candidate classifications for class c_j .

Joint classification selection. Denote $f : \{1, \dots, N\} \rightarrow \{1, \dots, L\}$ as the map that picks one candidate classification from each class, we compute the optimal map f^{opt} (which provides the optimal classification for each class) by maximizing the following second-order MRF potential:

$$Q(f, \theta) = \sum_{j=1}^n \theta_{j, f(j)}^1 + \sum_{1 \leq j < j' \leq n} \theta_{j j', f(j) f(j')}^2. \quad (8)$$

The unary term $\theta_{j, f(j)}^1$ evaluates the disjoint score between $\mathcal{L}_{j, f(j)}$ and $\mathcal{S} \setminus \mathcal{L}_{j, f(j)}$ on graph \mathcal{G}_j . In this paper, we utilize the normalized cut (NCut) score [Shi and Malik 2000] to penalize candidate classifications of small size:

$$\theta_{j, f(j)}^1 = 2 - \text{NCut}(\mathcal{L}_{j, f(j)}, \mathcal{S} \setminus \mathcal{L}_{j, f(j)}).$$

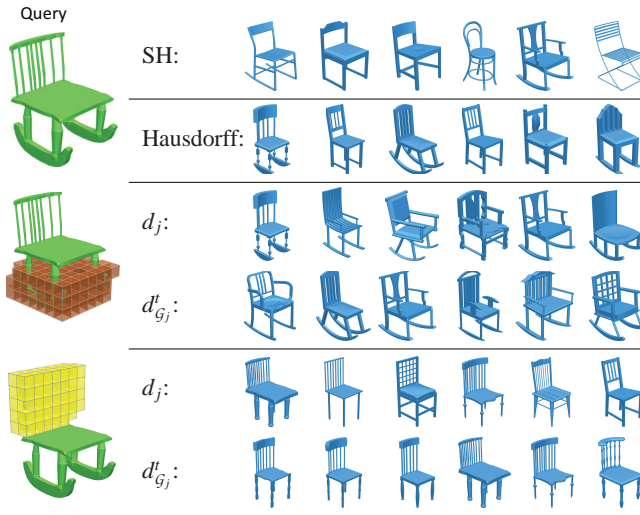


Figure 7: Distance metric comparison. This figure compares different distance metrics for the task of shape retrieval. From top to bottom, we show spherical harmonics (SH) [Kazhdan et al. 2003], Hausdorff distances between aligned shapes, and the distance metrics associated with the rocking and Windsor chair classes, respectively. For the learned distances, we show results using the distance metric directly (top) as well as using the graph diffusion distance (bottom). It is clear that the learned distance metrics capture the shape semantics more fully than traditional ones, and that diffusion distances lead to the best results.

The pair-wise terms $\theta_{j^i, f(j^i)f(j^i)}^2$ favor that the mutual correlations between the output shape sets agree with those of the input shape sets, a criterion used by other multi-label classification techniques [Tsoumakas and Katakis 2007]:

$$\theta_{j^i, f(j^i)f(j^i)}^2 = \gamma(1 - |\text{cor}(\mathcal{L}_{j^i, f(j^i)}, \mathcal{L}_{j^i, f(j^i)}^m) - \text{cor}(\mathcal{L}_j^m, \mathcal{L}_j^m)|),$$

where $\text{cor}(V, V') = |V \cap V'| / |V \cup V'|$, and parameter γ controls the importance of the pair-wise term.

We again employ the iterative coordinate ascent method described in [Leordeanu and Hebert 2006] to optimize Equation 8. The parameter γ is optimized via cross-correlation between the output shape sets $\mathcal{L}_j^{\text{opt}}$ and the input shape sets $\mathcal{L}_j^{\text{in}}$.

Note that after solving Equation 8, we also obtain an optimized scale for defining the diffusion distance on the similarity graph of each shape. These diffusion distances can be easily applied to perform shape retrieval (See Figure 7 for an example).

7 Experimental Evaluation

In this section, we describe the experimental evaluation of the classification pipeline. We begin with introducing the experimental setup in Section 7.1. Then in Section 7.2, we analyze the classification results and compare them with state-of-the-art multi-label and/or semi-supervised classification methods. Finally, we evaluate each stage of the proposed pipeline in Section 7.3.

7.1 Experimental setup

Benchmark. We have created a benchmark shape data set to evaluate the performance of the proposed approach. The benchmark includes three large shape collections: “Chair,” “Automobile” and “Airplane”. We followed the following procedure to create this benchmark. We first determine a candidate set of classes for each

category by searching for sub-categories within WordNet [Miller 1995] (e.g., side chair, Rex chair, sedan, convertible, sports car, bi-plane, propeller airplane). We then use these class names as key words to download models from Trimble 3D Warehouse. As the labels from the 3D Warehouse are quite noisy, we employ Amazon’s Mechanical Turk (AMT) to prune outlier models, i.e., those are not in any category, and to generate ground truth labels. Specifically, we provide for each shape a few views and ask users from AMT to answer questions that determine whether it is an outlier and, if not, the classes it belongs to. When determining fine-grained classes, we also give the AMT users a few examples of each class using Google image search. In total, we obtained 5850 chairs with 26 classes, 1684 cars with 9 classes, and 1206 airplanes with 9 classes.

Baseline methods. As semi-supervised learning techniques fall into transductive and inductive types, we chose a state-of-the-art method from each category as the baseline method to compare to. Among transductive methods, we chose [Chen et al. 2008], which is the multi-label extension of a binary graph based semi-supervised classification method described in [Zhu 2006]. Note that this method requires a similarity graph of all input shapes as input. In the same spirit as [Zhu 2006], we construct this similarity graph by connecting each shape with its $k = 16$ nearest neighbors in terms of a pre-defined global shape descriptor. In our experiments, we set this global shape descriptor as the concatenation of three popular shape descriptors: D2 [Osada et al. 2002], SH [Kazhdan et al. 2003] and the lightfield descriptor [Chen et al. 2003]. Among inductive methods, we chose [Loeff et al. 2009], which is the semi-supervised version of the popular linear classifier based multi-label classification method [Amit et al. 2007]. To make a fair comparison between this approach and our approach, we feed the set of volumetric features used by our approach into [Loeff et al. 2009]. Moreover, we use the same similarity graph as in the transductive baseline method.

Evaluation protocol. We evaluate the performance of a given method in terms of its classification accuracy and precision. Let \mathcal{L}_j^m denote the ground truth set for class c_j . Denote by \mathcal{L}_j the resulting labeled shape set for a given method on c_j . We define the classification accuracy $a(\mathcal{L}_j)$ and the precision $r(\mathcal{L}_j)$ of this method as:

$$a(\mathcal{L}_j) = |\mathcal{L}_j^m \cap \mathcal{L}_j| / |\mathcal{L}_j^m \cup \mathcal{L}_j|, \quad r(\mathcal{L}_j) = |\mathcal{L}_j \cap \mathcal{L}_j^m| / |\mathcal{L}_j|.$$

In the following, we primarily use the classification accuracy to compare different classification methods. We use the precision to evaluate whether a given method reduces the noise level from the input labels or not. In this paper, we report these two measures in percentages. Note that $a(\mathcal{L}_j^{\text{in}})$ and $r(\mathcal{L}_j^{\text{in}})$ describe the percentage of input labels and their precisions, respectively.

7.2 Analysis of classification results

	Input	Transductive	Inductive	Proposed
PlaneType	6.5/81.6	54.7/59.2	75.1/78.2	81.2/87.5
ChairType	3.4/79.8	61.7/67.3	79.3/82.1	83.4/87.1
CarType	6.9/79.0	51.7/59.8	75.3/80.1	81.2/85.7
PlaneStyle	4.7/83.8	45.7/47.2	63.3/67.1	80.1/85.6
ChairStyle	9.9/78.2	44.7/48.7	65.3/68.1	76.8/85.3

Table 1: Average classification accuracies and precisions of the proposed method and two baseline algorithms on classifying types and styles of each dataset. In each entry, we show classification accuracy/precision.

To analyze the classification results, we divide the classes of consideration into a type group and a style group. The type group includes classes, in which shapes exhibit global similarity, e.g.,

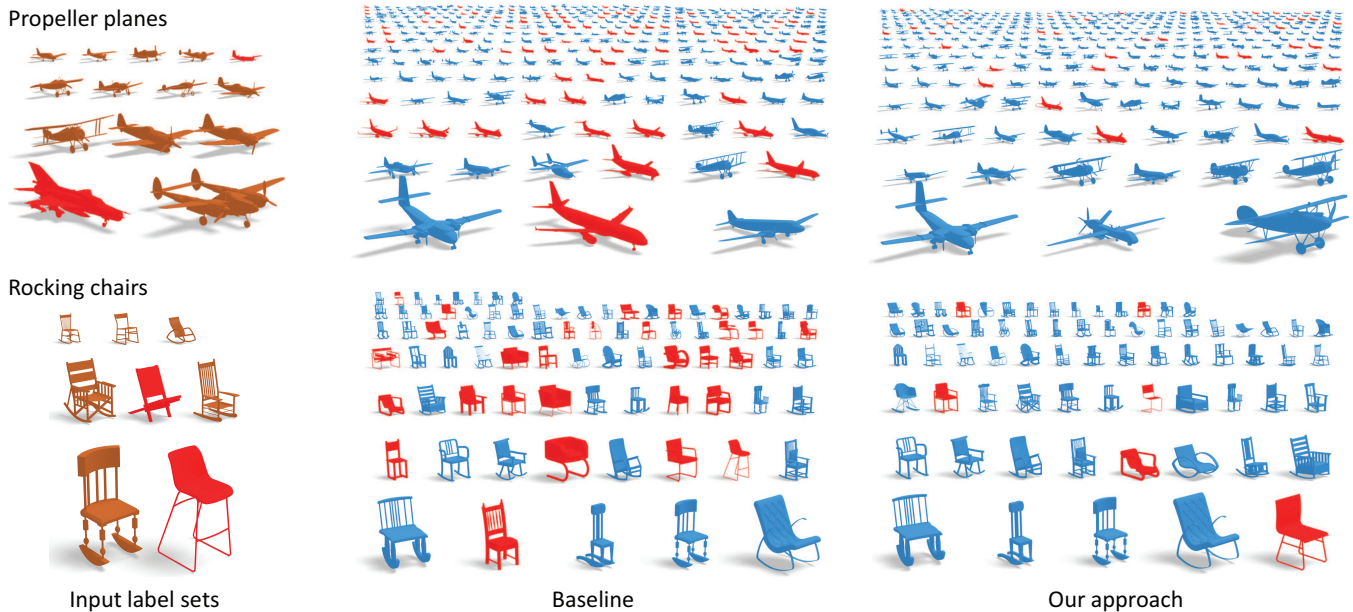


Figure 8: Baseline comparison. Comparison between the linear classifier baseline approach [Loeff et al. 2009] and the proposed approach on classifying propeller planes (Top) and rocking chairs (Bottom). Correctly labeled instances and mislabeled instances are colored in blue and red, respectively. (Left) Input shapes. (Middle) Classification results of the linear classifier based approach. (Right) Classification results of the proposed approach. We see that the proposed approach yields cleaner results.

stools, lounge chairs, sedan and coupe, while the style group includes classes that exhibit partial and local similarity, e.g., rocking chairs, swivel chairs and jet planes. The complete list of each group is provided in the supplemental material.

Table 1 and Figure 9 collect the classification accuracies of various methods. Due to space constraints, we only report results on representative classes as well as the averaged results on each data collection. Please refer to the supplemental material for more details. Overall, our approach delivers good performance on all three collections. In addition, the new approach improves significantly from baseline algorithms and is able to reduce the noise level of the input labels. In the following, we break down the performance of various methods on specific classes.

Type classification. We distinguish between two categories of object types. On objects whose geometric shapes significantly differ from other object types in their category, e.g., limousines and lounge chairs, both the baseline algorithm and the proposed approach lead to comparable high quality results (See Figure 9). This is expected because these object types are known to be well differentiated by comparing either global shape descriptors or aligned shapes. However, on object types that are characterized by relatively small-scale geometric features, e.g., arm-rests that separate side chairs from chairs with arms, the performance of the descriptor-based transductive approach drops significantly. This is due to the fact that these local geometric features are not well captured by comparing global descriptors. In this case, the success of the descriptor-based approach relies on a huge shape collection and densely labeled shapes, so that one can propagate labels among very similar shapes. In contrast, such key small-scale geometric features are nicely localized after aligning shapes, explaining why the proposed approach and the classifier-based approach report high classification accuracy in this case.

Style classification. When classifying object styles, our approach leads to better performance than the linear classifier approach. Figure 8 illustrates the classification results of these two approaches on the style classes of rocking chairs and propeller planes. We can see

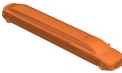







limousines	lounge	side	with-arms
			
89.1/93.2/ 100	91.1/93.2/ 94.5	64.2/83.1/ 84.9	61.2/81.2/ 83.0
rocking	propeller	bar stool	folding
			
32.1/61.2/ 85.1	47.1/56.2/ 84.8	57.1/65.2/ 70.1	41.1/68.2/ 73.1

Figure 9: Results on representative classes. Classification accuracies of the proposed approach and the baseline methods on selected classes. Within each entry, we show transductive baseline approach/inductive baseline approach/proposed approach.

that the linear classifier based approach misclassifies many shapes that are similar to inliers but with respect to a biased notion of geometric similarity. This can be explained by the fact that the nature of geometric variation within each such style class is rather subtle, and it is unlikely to be characterized well using linear classifiers learned from sparse and noisy data. In contrast, the proposed approach utilizes a sub-graph to represent the shapes in each class, which is capable of representing more complex decision boundaries.

7.3 Analysis of classification pipeline

Shape matching. We have evaluated the performance of our shape matching approach on the benchmark dataset described in [Kim et al. 2013]. This dataset consists of four categories of shapes: Seat, Plane, Bike and Helicopter. The size of each category varies from a few hundreds to a few thousands. Within each category, there are 100 shapes with manually labeled feature points for comparing the accuracy of different methods. We run our approach on each complete category of shapes and evaluate its performance on the subset of labeled shapes. Figure 10 compares the performance of our approach with state-of-the-art data-driven shape matching tech-

	Kim13	DL1	DL2	DL3	Chen08	Prop.
PlaneType	76.9	64.6	71.2	69.1	77.9	81.2
ChairType	83.1	69.2	78.1	82.1	82.2	83.4
Car	81.3	69.6	77.1	74.2	77.7	81.2
PlaneStyle	68.6	55.6	65.2	63.8	69.8	80.1
ChairStyle	70.5	60.8	70.1	67.2	77.3	76.8

Table 2: Performance of alternative methods used in the classification pipeline. From left to right: using the shape matching method described in [Kim et al. 2013], three alternative distance learning strategies (DL1, DL2 and DL3), using the method of [Chen et al. 2008] on the learned similarity graphs for classification and the proposed pipeline.

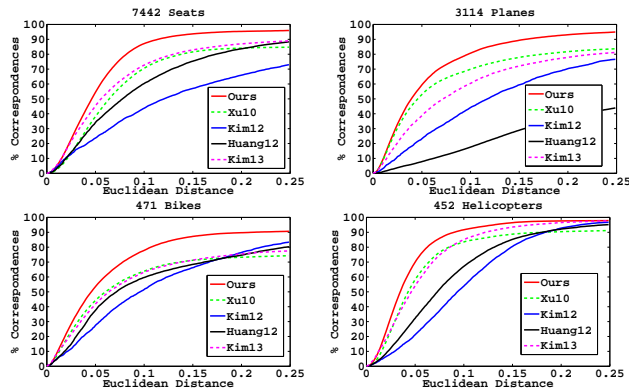


Figure 10: Shape matching quality. This figure demonstrates the shape matching quality of our method (red) relative to prior work: Kim et al. [2013](magenta), Huang et al [2012](black), Kim et al. [2012] (blue) and Xu et al. [2010] (green) on the benchmark provided in [2013]. Our method improves substantially from prior work in terms of both global and local accuracy.

niques [Huang et al. 2012; Kim et al. 2012; Kim et al. 2013] as well as the state-of-the-art pair-wise shape matching method for man-made objects [Xu et al. 2010]. The new approach is considerably better than previous methods in terms of both global accuracy and local accuracy. This shows the advantage of simultaneously matching and aligning shapes in an ambient space.

To understand the influence of the shape matching stage on the final classification results, we have replaced our shape matching method by that of [Kim et al. 2013]. We found that the classification accuracies on object types are similar. However, our method leads to better performance on classifying style classes. This can be understood by the fact that the template-based fitting approach described in [Kim et al. 2013] is insufficient to align style classes that typically exhibit non axis-aligned in-class deformations.

Distance learning. We have tested the final classification accuracies of using three alternative distance learning strategies:

- (DL1): Optimize f_{data} and update $\mathcal{M}_j, \mathcal{D}_j$.
- (DL2): Optimize $f_{data} + \lambda f_{coeff}$ and update $\mathcal{M}_j, \mathcal{D}_j$.
- (DL3): Optimize $f_{data} + \lambda f_{coeff} + \mu f_{rank}$.

As shown in Table 2, both the three objective terms as well as the strategy of updating the similar and dissimilar sets are important for the classification pipeline. Specifically, incorporating the coefficient prior term significantly improves the classification accuracy on all shape collections. The rank prior term, which considers the mutual relations among different classes, is important for style clas-

sification. The strategy of updating the similar and dissimilar sets is also important for style classification as the style labels tend to be sparser and noisier than the type labels (See Table 1).

Graph based classification. An alternative way to perform graph-based multi-label classification is to use the method of [Chen et al. 2008] on the similarity graphs derived from distance learning. As shown in Table 2, we found that the classification accuracy of the proposed approach slightly outperforms that of [Chen et al. 2008] on all three datasets. Moreover, the proposed approach is significantly more efficient since it avoids solving a large-scale linear system. On the chair dataset, the new approach takes 10 minutes, while that of [Chen et al. 2008] takes 210 minutes.

	t_{match}	$t_{feature}$	t_{learn}	t_{cut}	t_{total}
Airplane	3h33m	1h41m	8m	2m	5h24m
Chair	10h42m	4h31m	31m	10m	15h54m
Car	4h11m	2h47m	12m	6m	7h16m

Table 3: Timings of the proposed approach. t_{match} , $t_{feature}$, t_{learn} and t_{cut} represent the shape matching stage, the volumetric feature computation stage, the distance learning stage and the joint graph partitioning stage, respectively.

Timing. Table 3 shows the timing of each stage of the pipeline on a machine with 3.2GHZ CPU and 16G memory. Most of the time at the shape matching stage is spent on performing pair-wise shape matching and local alignment. On the average, matching one pair of fully (partially) similar shapes costs 0.1(1.5) seconds. Non-rigid registration for one pair of shapes takes 0.2 seconds. In the distance learning stage, most of the time is spent on feature computation; the learning process itself takes dozens of minutes. In the joint graph partitioning stage, computing the diffusion distance takes about 2-15 seconds per class. Generating shape sets and solving the optimization problem take less than 10 seconds.

Limitations. Our approach is less effective on classes where geometric similarity is less salient, e.g., folding chairs and bar stools (See Figure 9). In this case, our approach is only able to propagate labels to very similar shapes, and thus requires dense labels for better performance. To classify these classes well, one has to utilize advanced feature vectors that understand shape functionality. This is subject to future research. Moreover, the proposed approach assumes that the input shapes can be aligned in one common space. It does not work well when the part of interest has different repetition counts on some shapes (e.g., wheel classification from cars with different number of wheels). In the future, we plan to address this issue by considering multiple common spaces, each of which accounts for one shared part across multiple shapes.

8 Conclusions

In this paper, given a modest set of labeled shapes in a category, we have described a semi-supervised approach that simultaneously propagates and cleans these labels for others shapes in the category. While there has been extensive work on aligning shapes based on geometric features, the aim of our work has been to learn how to compare shapes so that the resulting geometric similarities parallel those present in a set of semantic class labels of a modest training set. The resulting labeled collection is far easier to organize, search, etc. than before. Furthermore, experimental results show that the performance of the presented approach outperforms state-of-the-art multi-label shape classification techniques.

There are ample opportunities for future research. First, we would like to study how to unify shape matching and shape classification into a single optimization procedure. Intuitively, the classified

shapes and the knowledge of where shapes are similar should help match shapes in a better way. Moreover, we would like to explore other applications of the presented approach to shape modeling and shape editing, or to more sophisticated forms of shape search.

Acknowledgements. This work was supported by NSF grants FODAVA 808515 and CCF 1011228, AFOSR grant FA9550-12-1-0372, ONR MURI award N00014-13-1-0341, a Google Research Award, and the support of the Max Planck Center for Visual Computing and Communications. The authors would also like to thank the anonymous reviewers for the helpful suggestions.

References

- AMIT, Y., FINK, M., SREBRO, N., AND ULLMAN, S. 2007. Uncovering shared structures in multiclass classification. *ICML '07*, 17–24.
- BAGHSHAH, M. S., AND SHOURAKI, S. B. 2009. Semi-supervised metric learning using pairwise constraints. *IJCAI'09*, 1217–1222.
- BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3, 1 (Jan.), 1–122.
- CANDÈS, E. J., AND RECHT, B. 2009. Exact matrix completion via convex optimization. *Found. Comput. Math.* 9, 6 (Dec.), 717–772.
- CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3d model retrieval. *Comput. Graph. Forum* 22, 3, 223–232.
- CHEN, G., SONG, Y., WANG, F., AND ZHANG, C. 2008. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, SIAM, 410–419.
- COIFMAN, R. R., LAFON, S., LEE, A. B., MAGGIONI, M., WARNER, F., AND ZUCKER, S. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *PNAS*, 7426–7431.
- CRANDALL, D., OWENS, A., SNAVELY, N., AND HUTTENLOCHER, D. 2011. Discrete-continuous optimization for large-scale structure from motion. *CVPR '11*, 3001–3008.
- DA FONTOURA COSTA, L., AND CESAR JR., R. M. 2009. *Shape Classification and Analysis: Theory and Practice*, 2nd ed. CRC Press, Inc., Boca Raton, FL, USA.
- DENG, J., KRAUSE, J., AND FEI-FEI, L. 2013. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR'13*.
- FERGUS, R., WEISS, Y., AND TORRALBA, A. 2009. Semi-supervised learning in gigantic image collections. In *NIPS*, 522–530.
- HOI, S. C., LIU, W., AND CHANG, S.-F. 2010. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Trans. Multimedia Comput. Commun. Appl.* 6, 3 (Aug.), 18:1–18:26.
- HOLLAND, P. W., AND WELSCH, R. E. 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics: Theory and Methods A6*, 813–827.
- HUANG, Q.-X., 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.
- HUBER, D. 2002. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- JOHNSON, A. E., AND HEBERT, M. 1999. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 5 (May), 433–449.
- 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.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. *SGP '03*, 156–164.
- KIM, V. G., LI, W., MITRA, N. J., DI VERDI, 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., DI VERDI, 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.
- LEORDEANU, M., AND HEBERT, M. 2006. Efficient map approximation for dense energy functions. *ICML '06*, 545–552.
- LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. In *SGP*, 1421–1430.
- LIU, W., WANG, J., AND CHANG, S.-F. 2012. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE* 100, 9, 2624–2638.
- LOEFF, N., FARHADI, A., ENDRES, I., AND FORSYTH, D. 2009. Unlabeled data improves word prediction. In *ICCV'09*, 956–962.
- MILLER, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38, 39–41.
- OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph.* 21 (October), 807–832.
- SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. *SIGGRAPH '86*, 151–160.
- SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (August), 888–905.
- TSOUMAKAS, G., AND KATAKIS, I. 2007. Multi-label classification: An overview. *Int J Data Ware. and Mining* 2007, 1–13.
- 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, 165.
- XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z.-Q. 2010. Style-content separation by anisotropic part scales. *SIGGRAPH ASIA '10*, 184:1–184:10.
- YANG, L., AND JIN, R. 2006. Distance metric learning: A comprehensive survey.
- YAO, B., KHOSLA, A., AND FEI-FEI, L. 2011. Combining randomization and discrimination for fine-grained image categorization. In *CVPR '11*, 1577–1584.
- ZHU, X. 2006. Semi-supervised learning literature survey. *Computer Sciences TR 1530*, University of Wisconsin Madison.