

# Exploration of Continuous Variability in Collections of 3D Shapes

Maks Ovsjanikov<sup>†</sup>

Wilmot Li<sup>‡</sup>

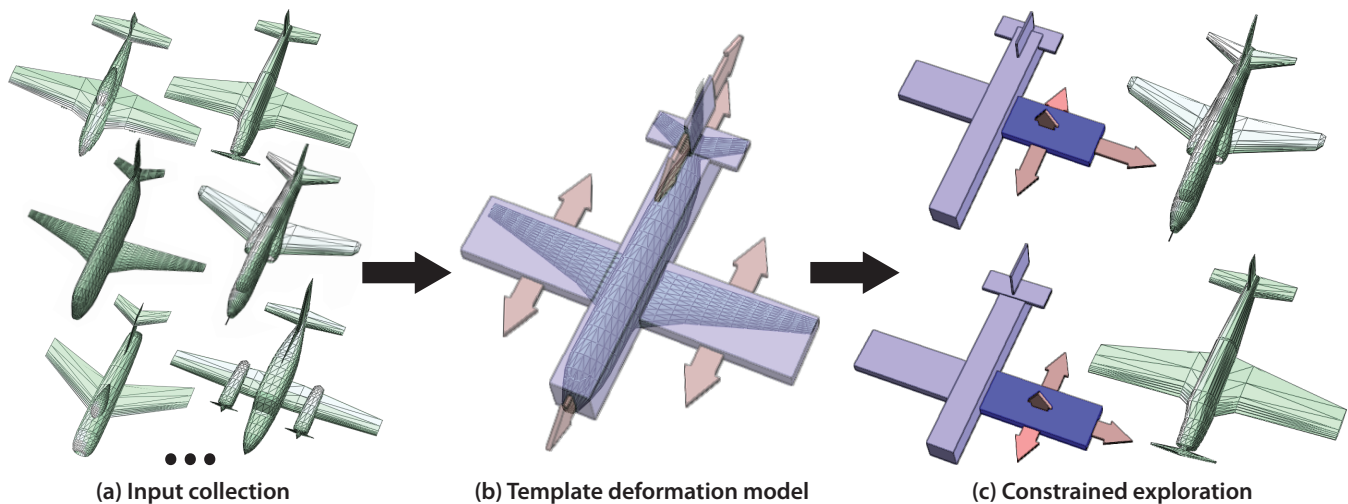
Leonidas Guibas<sup>†</sup>

Niloy J. Mitra<sup>\*</sup>

<sup>†</sup> Stanford University

<sup>‡</sup> Adobe Systems

<sup>\*</sup> KAUST



**Figure 1: Exploring collections of 3D shapes.** We present an approach for learning variability within a set of similar shapes, such as a collection of airplanes, without any labels or correspondences (a). Our analysis automatically extracts a deformation model that characterizes variability based on the spatial arrangement of components in a template shape. Here, the primary mode of variation involves the wings moving along the fuselage in a coupled manner (b). We use this deformation model to provide a constrained manipulation interface for exploring the collection (c). Remarkably, our method avoids establishing correspondences between shapes at any stage of the algorithm.

## Abstract

As large public repositories of 3D shapes continue to grow, the amount of shape variability in such collections also increases, both in terms of the number of different classes of shapes, as well as the geometric variability of shapes within each class. While this gives users more choice for shape selection, it can be difficult to explore large collections and understand the range of variations amongst the shapes. Exploration is particularly challenging for public shape repositories, which are often only loosely tagged and contain neither point-based nor part-based correspondences. In this paper, we present a method for discovering and exploring continuous variability in a collection of 3D shapes without correspondences. Our method is based on a novel navigation interface that allows users to explore a collection of related shapes by deforming a base template shape through a set of intuitive deformation controls. We also help the user to select the most meaningful deformations using a novel technique for learning shape variability in terms of deformations of the template. Our technique assumes that the set of shapes lies near a low-dimensional manifold in a certain descriptor space, which allows us to avoid establishing correspondences between shapes, while being rotation and scaling invariant. We present results on several shape collections taken directly from public repositories.

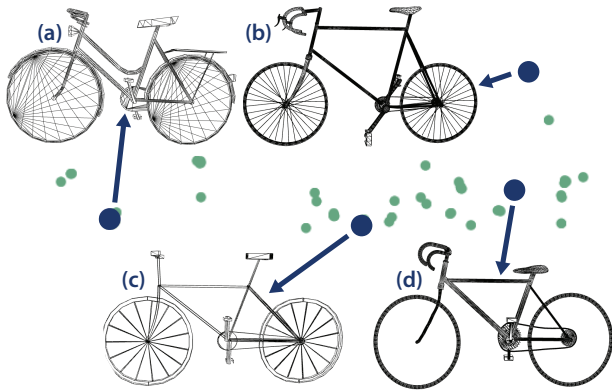
**Keywords:** 3D database exploration, shape descriptors, shape analysis, morphable models, model variability

## 1 Introduction

A growing number and variety of 3D models are becoming available on the web via online repositories. Popular websites such as TurboSquid or Google 3D Warehouse contain hundreds of thousands of models from a wide range of object classes, including airplanes, cars, furniture, etc. One key benefit of these repositories is that they make it possible to incorporate 3D models into a variety of workflows without having to create 3D geometry from scratch. For example, authoring a 3D game or animation often requires modeling the environment where the action takes place. Using repository models to populate these environments significantly reduces the required modeling effort. In addition, 2D graphic designers sometimes incorporate 3D content into their work so that they can tweak perspective and lighting while creating the final image, and thus also benefit from diverse repositories of 3D models.

While the growing availability of 3D models gives users an increasing range of content from which to choose, exploring large repositories can be a challenging task. Most online repositories support text-based search/filtering and return a list of all the matching models. This interface can help users quickly select a class of objects (e.g., all the cars), but it does not support easy exploration of the variations within that class. For example, searching for “car” in the Google 3D Warehouse returns tens of thousands of models on thousands of results pages, and it is difficult to get an overall sense for what types of cars are available or the range of different car shapes without looking at all the results. Furthermore, text-based search does not allow users to explore collections of shapes based on geometric characteristics; for instance, while looking at one car in the collection, a user may want to see if there are similar models with skinnier bodies or larger wheels.

Another approach to exploring collections of 3D models is to organize them based on geometric similarities and differences. The most basic operations in this context are shape comparison and re-



**Figure 2: Navigating in descriptor space.** This plot is an MDS projection of the shape descriptors for a collection of 71 bicycle models, generated using the 100-dimensional smoothed histogram descriptor described in Section 5.1. While the 4 highlighted models are similar, their positions in descriptor space do not indicate any obvious consistent structural variation.

retrieval, which consists of finding models in a collection that are most similar to a given 3D shape. Along these lines, there is a large body of existing work on shape descriptors that attempt to capture important geometric properties of 3D shapes to enable meaningful shape-to-shape comparisons. While analyzing a set of 3D models in descriptor space often makes it easy to identify clusters of similar models and to retrieve models that match a given target shape, these techniques do not directly address the exploration problem. As with text-based filtering, clustering in descriptor space may still result in large, unorganized sets of similar models that are difficult to explore. On the other hand, example-based shape retrieval is most useful when the user already has a sense for what types of models are present in the collection and has at least one example of the desired type of models. As an alternative, the user can explore a collection by navigating directly in descriptor space. However, most descriptors are very high-dimensional, and a standard projection of the space, such as multidimensional scaling (MDS), typically does not result in an intuitive set of parameters for exploration, due primarily to the large amount of geometric detail present in the shapes (Figure 2), which can obscure the global structural variability.

Finally, a different set of methods for navigating a collection of shapes aims at obtaining an explicit representation for the variability observed in the collection, by finding a consistent parametrization for the shapes and then performing statistical analysis in the parameter space. The most common way for parametrizing a set of shapes is by establishing correspondences between points or parts of shapes and measuring their displacements across the shapes in the collection. Active shape models in computer vision [Cootes et al. 1995], statistical shape analysis in biology [Dryden and Mardia 1998] and inverse kinematics in computer graphics [Sumner et al. 2005] all share this spirit. Such methods are useful for navigating within a family of related shapes because they can constrain exploration within the bounds of the observed data. However, obtaining reliable correspondences across even moderately sized collections of shapes is a very challenging problem, especially for the unstructured user-generated shapes found in public repositories.

In this work, we propose a new correspondence-free technique for exploring unorganized collections of 3D models that focuses on learning and presenting variations within a set of similar shapes. Specifically, our method analyzes the input collection to extract and organize continuous variability that can be expressed in terms of the relative sizes and positions of shape parts. Based on this analysis, we provide an exploration interface in which the user interactively

deforms a template shape by dragging arrows that indicate the main variability axes amongst the input shapes. As the template deforms, the system presents the closest matching model in the collection.

There are two key ideas behind our approach:

**I. Exploration via template deformation.** Unlike standard example-based retrieval where users must obtain and specify an example model for each query, we allow users to explore the collection via interactive deformation of a template shape. Our system helps the user select a suitable template and provides simple direct manipulation tools for specifying deformations.

**II. Constrained template deformation model.** To help the user specify deformations that match the actual shape variations in the collection, we propose a novel technique for extracting a constrained template deformation model directly from the collection. Notably, our method does not require correspondences between the models in the collection; instead, we propose a technique that converts variability in the descriptor space representation of the collection into a deformation model of the template shape.

Our strategy of combining a direct manipulation template-based interface with a constrained deformation model addresses many of the limitations of existing exploration techniques. Unlike text-based search, our approach allows the user to explore a collection based on geometric characteristics, and in contrast to previous descriptor-based methods, we represent shape variation in terms of template deformation, with an intuitive, continuous parameter space that allows users to explore the variability in a collection with respect to the configurations of parts that comprise the object. For example, in one of our results, we analyze a collection of airplanes and automatically extract a deformation model in which the positions of the wings vary from the front to the back of the fuselage (Figure 1). Based on this analysis, we allow the user to explore the collection by directly deforming the wings of the template model. Finally, the fact that we do not rely on computing shape-to-shape correspondences makes our method both robust to geometric diversity and efficient, enabling real time exploration of shape collections.

The main assumptions of our method are as follows. First, we assume that the shapes in the input collection all share a common global part structure (e.g., a set of four-legged chairs or two-winged airplanes). However, we do *not* assume the shapes to be similarly segmented, as this is rather unrealistic for general shape repositories. For example, the wing of an airplane may be composed of a single connected mesh component in one model and dozens of components in another. We also assume that most of the shape variability within the collection can be explained in terms of the relative sizes and positions of the shape parts. More specifically, we assume that given any pair of shapes  $A$  and  $B$  in the collection, it is possible to deform  $A$  by modifying the sizes and positions of its parts to align  $A$  close to  $B$ . Finally, we assume that the space of possible deformations is low dimensional and sufficiently densely sampled in the collection. In other words, we assume that we are given a shape collection that lies near a low-dimensional manifold in a deformation space defined with respect to the relative sizes and positions of shape parts.

Our work makes two main contributions:

- We present a template-based interface for exploring collections of similar 3D models via constrained direct manipulation.
- We introduce a novel technique to convert descriptor variability into a deformation model for a template shape without relying on correspondences between shapes.

In the remainder of the paper, we describe the details of our approach and present several results.

## 2 Related Work

**Morphable models and deformation modeling.** A collection of shapes *with* assigned correspondences implicitly encodes a deformation model for the collection. Such shapes can be seen as high dimensional points within a common coordinate system, and their principal modes of variations can be directly extracted using statistical tools. Blanz et al. [1999] in their highly influential work explore this idea in the context of 3D face models. Subsequently, the framework has been extended to analyze shapes of human bodies in consistent poses [Allen et al. 2003], and also for human bodies with both pose and size variations [Angelov et al. 2005]. Sumner et al. [2005] represent meshes as feature vectors of deformation gradients relative to a reference pose, and allow direct mesh manipulation restricted to a nonlinear span of the example feature vectors. Kokkonos and Yuille [2007] learn object deformation models using an active appearance model among objects with manually annotated landmark correspondences, which is similar in spirit to statistical shape analysis in biology [Dryden and Mardia 1998]. In the medical domain, Kim et al. [2008] use a PCA based deformation model between a template and training images for robust registration of MR brain scans. Recently, Berner et al. [2011] extract repetitive object parts where the parts form low dimensional shape spaces. Alternatively, Kilian et al. [2007] create a shape space for *isometric* deformations that allow computation of geodesic paths between model pairs with given correspondence. Mitra et al. [2007] map transformation domain manipulations to indirectly deform the corresponding shapes in the object space.

While all of the above methods can learn useful modes of variation for specific classes of shapes, their reliance on accurate correspondences is a significant limitation for the task of exploring unorganized collections of 3D shapes. In particular, there is typically a large amount of variation in topology and geometric quality across models in public repositories, even those within the same class. For example, as noted by Xu et al. [2010] it is not uncommon to see point sets, polygon soups, and water-tight meshes all within a single collection of models. In the face of such variations, global correspondence detection remains a challenging open problem (see [van Kaick et al. 2010]).

**Exploring shape datasets.** Due to the difficulty of establishing correspondences across unstructured collections of shapes, researchers have proposed several alternative strategies. For example, text keywords (see [Fisher and Hanrahan 2010] for a discussion), or proxies for shape parts [Funkhouser et al. 2004; Chaudhuri and Koltun 2010] are regularly used to query heterogeneous datasets. Alternatively, models can be mapped to an appropriate global descriptor space (e.g., shape distributions [Osada et al. 2002], spherical harmonics [Kazhdan et al. 2003], spherical wavelets [Laga et al. 2006], heat kernels [Ovsjanikov et al. 2009]) and based on the properties of the descriptors (e.g., shape, pose or rotation invariance) shapes are embedded in a consistent descriptor space without requiring explicit object level correspondences.

While clustering in descriptor space produces broad categorizations (e.g., distinguishing between cars and horses) studying finer scale variations within a cluster remains a challenge. Recently, part-based correspondence has been explored as a method for studying variations within such model clusters. For example, Golovinskiy et al. [2009] present an algorithm for simultaneously segmenting while establishing part correspondences across a set of models based on clustering on a graph of potential corresponding model faces, and Kalogerakis et al. [2010] introduce a data-driven approach for learning a consistent segmentation and labeling of model parts using a range of geometric and contextual features. In contrast, we demonstrate that subtle models variations can be learned

and explored from model collections by establishing a two-way mapping between descriptor space and object space *without* computing direct correspondences at the level of features or parts.

There is a significant amount of prior work that supports modeling with the help of a database of existing models [Funkhouser et al. 2004; Chaudhuri and Koltun 2010; Fisher and Hanrahan 2010; Xu et al. 2010]. While some of these techniques propose or use specific ways of searching the database for good candidate models (or suggestions), these methods typically rely on example-based search, which, as discussed earlier, is not ideal for exploring unfamiliar collections of shapes. In many ways, our work is complementary to that of the data-driven modeling community; we focus on techniques that facilitate the exploration of model collections, while they focus on methods for combining various components from different models together.

## 3 Overview

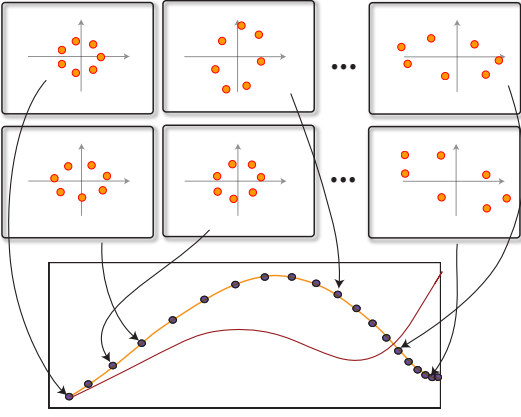
Our approach consists of two main steps. Given a collection of 3D shapes, we first analyze the collection to select a template shape and extract a deformation model that captures the primary modes of continuous variability found in the collection. Then, we incorporate the template and its deformation model into an exploration interface that allows users to browse the collection by interactively deforming the template to find similar shapes. Since the goal of our work is to help users understand and explore variations within a collection of similar shapes, we assume the input models all come from the same class (e.g., a set of shapes returned by text-based filtering or clustering in some descriptor space). Furthermore, we focus on extracting continuous rather than discrete variability. Thus, our approach works best for object classes whose shape variations are characterized by the relative size and positions of the main object parts (e.g., bicycles, cars, planes and other mechanical objects), and we do not attempt to learn variations in the number of parts per shape (e.g., chairs with different numbers of legs). Finally, in our implementation, we assume the input models are represented as polygonal meshes.

To help explain and motivate our analysis technique, we first discuss the problem of extracting a deformation model from a set of shapes in the context of a simple example (Section 4). We then describe the details of our analysis framework (Section 5) and exploration interface (Section 6). Finally, we present results from several experiments (Section 7) and discuss directions for future work (Section 8).

## 4 Problem Formulation and Method Overview

In this section, we illustrate the problem of shape exploration with a motivating example, which underlines both the complexity of shape analysis and the properties of our approach.

In the most basic setting, suppose that we are given a collection of  $s$  shapes, where each shape is simply a set of  $n$  points in 2-dimensional space, as shown in Figure 3 (top). Our overall goal is to discover and expose the *variability* of the shapes in the collection, while factoring out the rotation. Of course, if there is no a-priori information about the shapes, any set of correspondences or deformations is equally likely. Now suppose instead that it is known that every shape in the collection is obtained by deforming the basis shape, Figure 3 (top left), and that moreover, the set of deformations forms a smooth 1-parameter family, although the shapes are not presented in any particular order. For example, the shapes in Figure 3 were obtained by scaling the  $x$  coordinate of each point of the shape shown in the top left, by  $t dx$  and the  $y$  coordinate by  $t dy$ , where  $t$  ranges from 0 to 1, and  $dx, dy$  are fixed. Moreover,



**Figure 3: Overview of our method.** Given a collection of unlabeled shapes (top), we compute a descriptor for each shape, embedding them into a canonical domain (bottom). If the shapes were produced by a 1-parameter family of deformations, their descriptors will trace a curve. Moreover, any such family of deformations of a starting shape will produce a curve in the descriptor space (bottom red). By fitting the curve lying most closely to the data, we recover the original deformation.

an arbitrary rotation was added to each shape. Our goal is then to recover  $dx$  and  $dy$  as well as the value of  $t$  for each shape, while factoring out the rotations.

Note that even given the knowledge that a set of shapes was produced by a smooth deformation, establishing point correspondences is non-trivial due to the combinatorial complexity of the search for both the right order of the shapes and the right correspondences between them. Indeed, a brute-force approach to establishing point-correspondences between  $s$  shapes with  $n$  points would have  $O(s!n!^s)$  complexity, which is doubly exponential if  $s = O(n)$ . One of the main observations of this paper is that if a collection of shapes comes from a 1-parameter family set of deformations (i.e., each shape corresponds to a deformation of some basis shape) then in an appropriately chosen rotation-invariant descriptor space (Section 5.1), the descriptors of the shapes will trace a smooth curve. In other words, if we summarize each shape by a compact rotation-invariant descriptor in  $\mathbb{R}^m$ , which changes smoothly as the shape is deformed, then the set of all descriptors will form a smooth one-parameter family in  $\mathbb{R}^m$ , which is a curve. Moreover, since any 1-parameter family of deformations will produce a set of shapes whose descriptors trace a curve, given a potential value for  $dx$  and  $dy$  we can evaluate how well its curve fits the data in the descriptor space and pick the deformation parameters that are as faithful as possible to the data. In particular, if we have an explicit expression for how the change in the deformation parameters affects the descriptor, we can view the problem of picking the optimal deformation parameters as an energy minimization problem, and solve it using gradient descent or other optimization techniques. Note that in this example we used the descriptor introduced in Section 5.1 and the continuous optimization framework described in Section 5.3 to successfully recover the original deformation.

In the rest of this paper, we show how to apply this idea to recover continuous variability in unlabeled collections of 3D shapes. Namely, if the variability in a collection of shapes can be captured by deformations of some base template shape, our goal is to recover both the template as well as its family of deformations. Moreover, we avoid relying on correspondences across shapes, since obtaining such correspondences is a very challenging problem, which only becomes harder with the growth of shape repositories.

## 5 Shape Analysis and Deformation Modeling

Given a set of related 3D shapes, we apply the high-level approach described in the previous section to extract and model the continuous variations in the collection. To realize this approach, we present an analysis framework that addresses the following challenges:

- Identify a suitable shape descriptor whose variability is related to deformations of the template shape (Section 5.1).
- Select a representative template shape whose deformation will be used to explain the variability of the input shapes (Section 5.2).
- Find variability in the descriptor space and convert it into variability in the deformations of the template shape (Section 5.3).
- Learn a low-dimensional deformation model that captures the template deformations (Section 5.4).

### 5.1 Shape descriptor

As described in Section 4, our general strategy is to extract continuous variability in the descriptor space, in which each shape is represented as a vector of fixed finite dimensions, and use it to infer continuous variability in the relative sizes and positions of shape parts. The key requirement for a finite dimensional descriptor to work in our setting is that continuous (or differentiable) deformations of a shape must result in continuous (or differentiable) changes in each coordinate of the shape descriptor. Moreover, as we show in Section 5.3, we would like to have *explicit* expressions for the change in the descriptor coordinates with respect to shape deformations.

Note that direct adaptations of most shape descriptors do not satisfy these properties. To illustrate this, Figure 4 shows the changes of different descriptors under smooth deformation of a shape, obtained by continuously sliding the wings of a plane model along its fuselage from the very front towards the back (Figure 4 top). We sampled this deformation 71 times and considered how different coordinates of multi-dimensional shape descriptors change under this motion. Figure 4 (bottom left) shows the changes in 3 coordinates of a popular 517-dimensional Spherical Harmonic descriptor [Kazhdan et al. 2003], while Figure 4 (bottom middle) shows the changes in 3 out of 100 coordinates of the Shape Distribution [Osada et al. 2002] represented as a histogram of pairwise distances between 2 million point pairs. To remove randomness in the computation of the histogram, the pairs of points were pre-computed and deformed together with the rest of the shape.

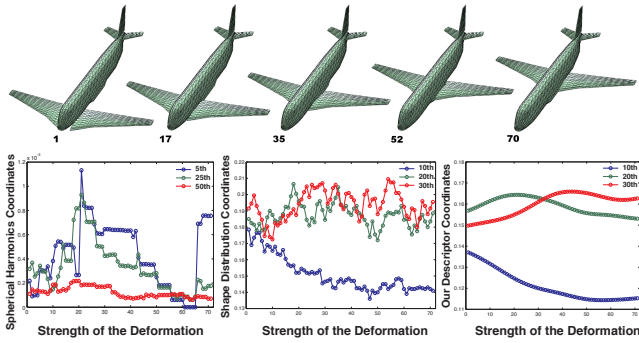
In order to achieve the above-stated goals, we use a different discretization of the Shape Distribution. Specifically, given an input shape  $M$ , we first compute Euclidean distances between  $N$  pairs of points  $(p_{j1}, p_{j2})$ ,  $j \in [1..N]$ , where each  $p_{j1}$  and  $p_{j2}$  is sampled uniformly at random on  $M$ . However, instead of discretizing the distribution of distances as a histogram, we convolve this distribution with a Gaussian kernel of a fixed width  $\sigma$ . Thus, given a shape  $M$  its descriptor  $S(M)$  is a vector in  $\mathbb{R}^m$ , with the  $i^{\text{th}}$  coordinate:

$$S_i(M) = \frac{1}{N} \sum_{j=1}^N \exp\left(-\frac{(\mu_i - d(p_{j1}, p_{j2})/R)^2}{2\sigma^2}\right),$$

where the normalizing constant  $R = (1/N) \sum_{j=1}^N d(p_{j1}, p_{j2})$ , and we use  $\mu_i = 3i/m$  (for an  $m$ -dimensional descriptor vector) and  $\sigma = 0.1$  in our experiments. Note that each coordinate  $S_i(M)$  changes smoothly under smooth changes of the coordinates of the points,  $p_{j1}, p_{j2}$ , Figure 4 (bottom right).

We note that the same discretization technique can be applied to other descriptors based on distributions of quantities that change





**Figure 4:** Changes of different descriptors under smooth deformation of a shape. (Top) A basis shape was deformed by progressively moving the wings along the body. (Bottom) Changes of fixed coordinates of different descriptors. (Left) Spherical Harmonics [Kazhdan et al. 2003], (middle) Shape Distribution (SD) [Osada et al. 2002] represented as a histogram of distances, and (right) our discretization of SD. For simplicity we only present how specific coordinates of each descriptor change under smooth deformations.

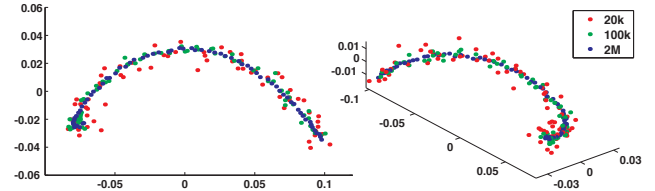
smoothly under shape deformations, such as areas or curvatures. Nevertheless, the smoothness of the discretized descriptor is very easily broken, e.g., by scaling these quantities by their maximum value rather than the mean. Similarly, convolving a distribution *after* binning it to a histogram will not result in a smooth descriptor; since a histogram of a fixed number of values — and thus, its convolved version — has only a finite number of possible states, it cannot possibly vary smoothly under continuous deformations. Moreover, our discretization technique does not necessarily achieve smoothness for all histogram-based descriptors. For example, the Shape Diameter Function [Shapira et al. 2008] measures the lengths of inward-pointing rays within a shape, which can change discontinuously under deformations. Note that it is also possible to produce smoothly varying descriptors by summarizing a distribution using its moments, where  $j^{\text{th}}$  coordinate of the descriptor becomes  $(1/N) \sum_{j=1}^N (d(p_{j1}, p_{j2})/R)^j$ , or to use the empirical characteristic function:  $\sum_j \exp(i t d(p_{j1}, p_{j2}))$  for various values of  $t$ , where  $i$  is the imaginary unit. We leave the comparison of these descriptors and exploration of other possibilities to future work.

It is also important to observe that  $S(M)$  is invariant under rigid deformations as well as global scaling of the sample points. Furthermore, as shown by Boutin and Kemper [2004] almost all  $n$ -point configurations are reconstructable from their distributions of distances. Since  $S(M)$  is simply a convolution of the Shape Distribution with a Gaussian kernel of fixed width, it contains the same amount of information (since it is invertible in the Fourier domain) and can thus discriminate between non-congruent shapes.

## 5.2 Template selection and deformation space

In the first stage of our pipeline, we compute the descriptor for each shape in the given collection. Since this stage is done off-line and only once, we emphasize accuracy over efficiency, and thus, can afford to use a large set of point-pairs for computing the descriptor. In practice, we use 2 million point pairs for every shape, and 100 coordinates for the descriptor  $S(M)$  sampled uniformly between 0 and 3 (recall that the mean distance is always 1).

To illustrate the dependence of the accuracy of the estimation of our descriptor on the number of sample points, we plotted in Figure 5 the 2D and 3D MDS embedding of the descriptors computed for the shapes from the set shown in Figure 4 (top). Unlike the experiment shown in Figure 4, however, we computed the descriptor independently for each pose, rather than fixing the set of point pairs. As a result, since the pairs are sampled randomly, there will



**Figure 5:** The variance in the estimation of the descriptor decreases with increasing sampling density. The points represent 2D (left) and 3D (right) MDS embedding of the descriptors computed for the set of shapes used in Figure 4, but sampled independently for each shape. The curve in the descriptor space is well pronounced already with 100k.

be variance in the estimation of the descriptor. On the other hand, because the shapes are related by a smooth deformation, we expect the descriptors to trace a curve in the descriptor space. As can be seen in Figure 5, the variance in the estimation of the descriptor decreases when increasing sampling density (from 20k to 100k to 2M samples), and the curve becomes well pronounced.

After computing the descriptor for all shapes, our system helps the user select a template shape whose constituent parts define the allowable set of deformations (i.e., the *deformation space*) for our analysis. Since the goal of our analysis is to determine how the relative sizes and positions of object parts vary across the input collection, the chosen template should include all the basic parts for the given object class and ideally be segmented into exactly those parts. As noted by Xu et al. [2010], shapes in public repositories tend to be *over-segmented* when considering simply their mesh connectivity. Thus, we adopt a simple strategy for selecting a template shape in a collection. We first order the shapes by the distance of their descriptor to the average descriptor in the collection. Next, we filter out all shapes that have more than a user-specified maximum number of connected mesh components. Finally, we automatically suggest the first shape to the user as a potential template. With this approach, we end up suggesting a template that closely resembles the average while not being over-segmented.

In most cases, manual inspection by the user is still necessary to ensure that the template components are reasonably segmented and actually correspond to a meaningful set of parts. If the user is not happy with the first suggestion, she can view the other shapes in the ordered list of suggested templates. In addition, we allow the user to modify a template shape by selecting components to group together into a single part. In all of our experiments, we were able to find satisfactory templates amongst the first 2–3 shapes closest to the mean, and for three of our six test datasets, we ended up using the automatically suggested template without any modifications. For the remaining three datasets, we grouped a few components together to simplify the template part structure; this grouping took less than one minute per template. Note that the exact choice of the template shape is not crucial in our method as long as it has the desired global structure, since relations based on deformation are transitive and reflexive — i.e.,  $A$  is deformable onto  $B$  implies that  $B$  is deformable onto  $A$ .

For the chosen template shape, with  $C$  components (or user-defined groups of components), we define the template configuration  $X = (x_1, \dots, x_{6C})$  as the coordinates of the bounding box for every component. We parameterize the deformation space of the template shape by a set of  $6C$  deformation parameters that correspond to 3 translation and 3 scaling parameters for each component. Specifically, given a deformation vector  $D \in \mathbb{R}^{6C}$ , the deformed coordinate of a point  $p = (p_x, p_y, p_z)$  that belongs to component  $c$  becomes  $p(D) = (D_{c1}p_x + D_{c2}, D_{c3}p_y + D_{c4}, D_{c5}p_z + D_{c6})$ , where  $D_{c1} \dots D_{c6}$  are the deformation parameters in  $D$  corresponding to component  $c$ .

Although more flexible parameterizations of the deformation space are possible, e.g., [Sorkine and Alexa 2007], our goal is to capture global variability using as few as possible deformation parameters. Note also that since we view each shape as a point in the high-dimensional deformation space, we need a dense sampling to recover patterns in the deformation space without over-fitting, which becomes progressively more difficult by introducing more complicated and thus higher dimensional deformation parameterizations.

### 5.3 Descriptor variability and template deformations

Given a template shape and a parametrization of its deformation space, our goal is to use it to navigate through other shapes in a collection and to learn their variability. In other words we would like to capture the variability of shapes present in the collection *in terms of* the deformation of the template shape.

We start by identifying local patterns among the set of descriptors of the shapes in the collection. Specifically, since by construction our descriptors change smoothly under deformations of the shape, low dimensional families of deformations get mapped to low dimensional variability of the descriptors. Therefore to capture shape variability, we perform local PCA in the descriptor space around the descriptor of the undeformed template shape. We select a large set of neighbors of the template descriptor, and find the principal directions of variability of the descriptors in this collection. Our goal then is to convert this variability in the descriptor space into variability in the deformation of the template.

To solve this problem, first consider the following simple formulation: suppose we are given descriptors of two shapes  $S(M_1)$  and  $S(M_2)$ , and our goal is to find a deformation of  $M_1$  that would cause its descriptor to move as closely as possible to that of  $M_2$ . This problem can be phrased in terms of energy minimization. For a particular choice of deformation parameters  $D$ , let

$$E(D) = \|S(M_1(D)) - S(M_2)\|^2,$$

where  $M_1(D)$  represents shape  $M_1$  deformed according to  $D$ .

Then, the optimal deformation  $D$  can be defined as  $D_{\text{opt}} = \arg \min_D E(D)$ , and found using non-linear optimization techniques. One key requirement of such techniques is that the gradient  $\nabla E(D)$  exists and preferably can be computed analytically. Note that this is very easy to do with our descriptor, given the knowledge of partial derivatives of the coordinate of each point  $p \in M_1$  with respect to the deformation parameters:  $\partial p(D)_i / \partial D_j$ . In particular:

$$\nabla E(D) = 2(S(M_1(D)) - S(M_2))J,$$

where  $J$  is the Jacobian matrix, such that  $J(k, i) = \partial S(M_1(D))_k / \partial D_i$ , which can be easily computed as:

$$J(k, i) = \frac{1}{N} \sum_{j=1}^N \exp\left(-\frac{(\mu_k - d_j/R)^2}{2\sigma^2}\right) \frac{(\mu_k - d_j/R)}{\sigma^2} \frac{\partial(d_j/R)}{\partial D_i},$$

where  $d_j = d(p_{j1}(D), p_{j2}(D))$ ,  $R = \frac{1}{N} \sum_{j=1}^N d_j$ , and

$$\frac{\partial(d_j/R)}{\partial D_i} = N \frac{\partial d_j / \partial D_i \sum d_j - d_j \sum \partial d_j / \partial D_i}{(\sum d_j)^2},$$

$$\frac{\partial d_j}{\partial D_i} = \frac{p_{j1}(D) - p_{j2}(D)}{d_j(D)} \cdot \frac{\partial(p_{j1}(D) - p_{j2}(D))}{\partial D_i}.$$

In particular, if  $p(D) = (D_1 p_x + D_2, D_3 p_y + D_4, D_5 p_z + D_6)$ , then e.g.  $\partial p(D) / \partial D_1 = (p_x, 0, 0)$ ,  $\partial p(D) / \partial D_2 = (1, 0, 0)$  and similarly for  $D_i, i = 3 \dots 6$ .

To summarize, if we are given a parametrization of the deformation space by a fixed number of parameters encoded in a vector  $D$ , such that the coordinates of all points of the shape change smoothly under the changes of these parameters, then the gradient of the energy

$E(D)$  is well defined and has a simple closed form solution, which can be used to find the optimal deformation  $D$ .

Note, however, that in general multiple solutions may exist for  $D_{\text{opt}}$  since, although our descriptor is informative, because of limited precision and memory multiple shapes may exist sharing the same descriptor. On the other hand, since we are interested in the *variability* among the descriptors, rather than fitting the deformation to a single point in the descriptor space, we would like to fit it to an entire curve (a line in the case of PCA), thereby also alleviating the ambiguity in the solution.

For this, we sample a given principal direction with  $L$  equally spaced sample points starting at the descriptor of the undeformed template shape, and rather than solving for a single deformation  $D$ , we solve for  $L$  deformations  $D_1 \dots D_L$  that minimize the following energy function:

$$\sum_{j=1}^L \|S(M(D_j)) - S_j\|^2 + \alpha \sum_{j=2}^L \|D_j - D_{j-1}\|^2,$$

where the second term is designed to ensure smoothness of the deformations. In practice, we use  $L = 3$  and  $\alpha = 2$  for the examples in this paper.

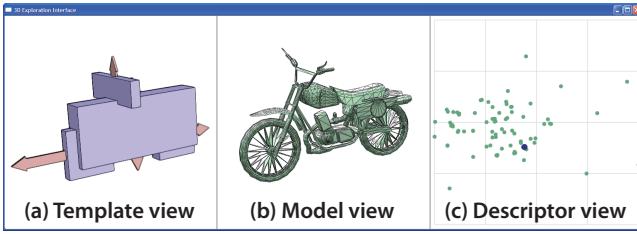
Note that this formulation assumes that uniformly spaced deformations in the deformation space will produce approximately uniform changes in the descriptor space. Although this may not be true in general, we have observed that for medium deformations, this is not a restrictive assumption.

We compute the optimal set of deformations using the same optimization framework as described above, made possible by the smoothness of the energy with respect to the deformation parameters, and thus our ability to compute the analytic gradient  $\nabla E(D)$ .

The output of this procedure is a set of  $2L$  deformations for each principal direction with  $L$  in each positive and negative directions. Using these deformations we then learn a deformation model that is used to navigate the shape collection efficiently.

### 5.4 Learning the deformation model

To combine the  $2L$  template deformations from the previous step into a deformation model, we use an approach similar to the Active-Shape Models of Cootes et al. [1995] and related techniques. First, we convert each deformation  $D$  from a set of translation and scaling parameters into a *deformation vector*  $T = (t_1, t_2, \dots, t_{6C})$  that represents an explicit offset from a given template configuration. While  $D$  and  $T$  are equivalent (i.e., they define the same template deformation),  $T$  is a more convenient representation for learning the deformation model. Next, we perform PCA on all  $2L$  deformation vectors and treat the first  $K$  principal components as a *deformation basis*. This basis represents the core of our deformation model. By restricting the space of allowable deformations to linear combinations of the basis vectors, we ensure that the template always deforms in a way that corresponds with the underlying descriptor variability. To limit how far the template can deform in each basis direction, we project the original  $2L$  deformations onto the basis and compute the minimum and maximum projected values in each basis dimension – in other words, we compute the  $K$ -dimensional bounding box of the original  $2L$  deformations vectors projected onto the basis. We define the space of allowable template deformations as the linear combinations of basis deformations whose coefficients fall within this bounding box. For the examples in this paper, we set  $K = 2$ . Also, to make our learned deformation models a bit more flexible, we increase the range of allowable coefficient values in each basis dimension by a small scale factor.



**Figure 6: Exploration interface.** As the user deforms the template shape using coordinate arrows (a), its descriptor is displayed in the context of the collection (c), and the closest model is shown (b).

## 6 Exploration

Having analyzed a collection of shapes to obtain a template and corresponding deformation model, we provide an exploration interface that visualizes the variability in the collection and allows the user to explore the set of input shapes by interactively deforming the template.

Our interface consists of three linked views (Figure 6):

The *template view* shows the set of  $C$  axis-aligned bounding boxes that correspond to the components of the template shape. This view allows the user to specify a template deformation by directly manipulating the individual template boxes.

The *model view* shows the model in the collection that is most similar to the current template configuration. This view updates automatically as the user deforms the template.

Finally, the *descriptor view* shows a 2D MDS embedding of the descriptors for the entire collection and the descriptor for the current template configuration, which updates as the user deforms the template. This view allows the user to see how template deformations map to descriptor space and provides visual context for how the current template configuration relates to the rest of the collection.

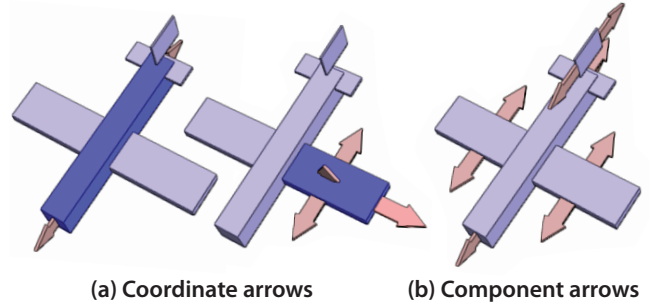
The remainder of this section describes the main features of our interface, and explains how these features help the user understand and navigate the variety of shapes in the collection.

### 6.1 Visualizing variability

To help the user understand the variability in the collection with respect to the template shape, our template view provides two types of visualizations.

**Coordinate arrows.** As discussed in the previous section, we parameterize the deformation space of a template by the  $6C$  coordinate values that define the size and position of each template component. Thus, one way to visualize the variability in the collection is to show the range of allowable values for each of these coordinates under the given deformation model. We show these ranges using arrows, one for each coordinate, that extend from the corresponding template box face to the minimum/maximum allowable value for that coordinate. To reduce visual clutter, the user can tell the system to only show the arrows for a selected template box (Figure 7a). By emphasizing how individual coordinates are allowed to deform, coordinate arrows help the user understand variability in terms of specific template components. For example, Figure 7b shows that in the airplane collection, the length of the wing and its position along the fuselage both vary much more than the wing thickness.

**Component arrows.** A different way to visualize variability is to show how the overall arrangement of template components varies across the collection. To this end, we generate component arrows,



**Figure 7: Arrows.** We generate coordinate arrows and component arrows to visualize variability with respect to the template shape. These images show the variations within a collection of airplanes.

which are simply a subset of all the coordinate arrows for the template. For each template component, we identify the dimension (i.e.,  $x$ ,  $y$  or  $z$ ) whose two corresponding coordinates have the maximum range of allowable values. We then show the corresponding two coordinate arrows. These component arrows allow the user to see, at a glance, the main ways in which the relative positions of components can change under the deformation model. For example, Figure 7b shows that the primary variations in the airplane dataset correspond to the wings moving along the fuselage and the vertical tail fin sliding forwards and backwards.

In order to generate both coordinate and component arrows, we must be able to determine the allowable range of values for a given template coordinate  $x_i$  with respect to the current template configuration  $X$ . To do so, we first construct the offset vector  $u$  that represents how  $X$  changes when we increase/decrease  $x_i$ . We then project  $X$  and  $u$  into the basis space defined by the deformation model, to get a set of basis coefficients  $Y$  and basis offset  $v$ . Since the deformation model specifies the allowable range of basis coefficient values, we can determine the longest distance we can travel along  $v_i$  starting from  $Y$  before we reach the limit of allowable template deformations. Finally, we take this longest distance in basis space and convert it to range of allowable values for  $x_i$ .

### 6.2 Exploring via constrained direct manipulation

To explore the collection, the user deforms the template by interactively dragging along any visible arrow in the template view. As the user drags, we update the template as follows. First, we modify the current template configuration  $X_1$  by updating only the coordinate associated with the dragged arrow. We then project this modified configuration  $X'_1$  into the deformation basis to obtain a set of basis coefficients  $Y_2$ . We clamp each coefficient based on the range of allowable values specified by the deformation model, and then we multiply  $Y_2$  by the basis deformation vectors to reconstruct a new template configuration  $X_2$ . Since we always project the template back into the deformation basis, the user is able to explore the space of allowable deformations for the entire template just by modifying a single coordinate. Furthermore, since dragging on an arrow attempts to modify only the corresponding coordinate, this interaction enables both part scaling and translation, as long as such deformations are supported by the deformation model.

Once the template is updated, we compute the descriptor for the new configuration and then find the most similar model in the collection by performing a nearest neighbor search in descriptor space. We update the model view to show the nearest neighbor model, and we update the descriptor view to show the new template descriptor value. If the user scrolls in the model view, we cycle through the  $k$  nearest neighbor models, where  $k$  is a user-specified parameter.

class	# of shapes	# conn. comp.		% manifold
		mean	std.dev.	
Bicycles	67	96.8	97.3	29.8
Cars	100	75.2	165.3	67
Motorbikes	81	223	207.3	16
Planes	55	52.4	97.8	25.5
Chairs	132	444.9	1931.8	89.4

**Table 1: Properties of our datasets.** *Note that the number of connected components ranges wildly across shapes (from 1 to 20494 in our datasets), and only a small fraction of shapes are manifold.*

One important property of our direct manipulation approach is that it automatically enforces any spatial dependencies between parts that are captured by the deformation model. For example, with the deformation model that we extracted from the collection of airplanes, dragging the left wing of the template automatically moves the right wing as well (see supplementary video). While it is technically possible to explore the collection by deforming each template coordinate independently, we found the learned dependencies to be critical for achieving a practical exploration interface. As an example, Figures 8d–e, show two principal modes of the deformation model for a collection of bicycles and motorcycles. In this model, there is a relationship between interwheel distance and the height of the body component; long motorcycles have lower bodies, while long bicycles tend to have higher bodies. Without the deformation model, the user would have to guess what template deformations might match the collection. In this case, even if the user guesses correctly that there are shapes with a long interwheel distance and slides the front wheel forward (Figure 8e), he would only find long motorcycles and not long bicycles.

## 7 Results

To evaluate our approach, we consider five classes of 3D shapes: airplanes, bicycles, motorcycles, cars and chairs. We chose these objects because they have a well-defined set of parts (e.g., wings, wheels, etc.), and variations within these classes of models are often characterized by the spatial layout of the parts. Furthermore, mechanical shapes and furniture are very popular in online repositories (e.g., 40k “cars” and 14k “chairs” in the Google 3D Warehouse).

For each class, we gathered a dataset of 3D models, ranging in size from 55 to 132 models (Table 1). We obtained most of our examples from the INRIA GAMMA Group repository [Saltel 2008], which aggregates 3D models from a variety of commercial and academic databases (e.g., Foundation 3D, the Princeton Shape database), with a few extra shapes added from the Google 3D Warehouse. Note that we did not prune models based on any low-level geometric properties. Thus, our datasets span a wide range of model complexity and quality, which is typical of online repositories in general. For example, across our test data the number of connected components varies from 1 to 20494, and a significant portion of the models are non-manifold as shown in Table 1. Finally, since the goal of our approach is to find variations within collections of similar shapes, we removed outliers from the datasets by thresholding the distance to measure [Chazal et al. 2010], with  $k = 10$  in our smoothed histogram descriptor space. Table 1 summarizes the shapes we obtained after pruning and outlier detection.

To determine the variations within each class of shapes, we analyzed the datasets as described in Section 5. Note that for template selection, we manually grouped a small number of connected components in the airplane and chair templates, and we used the automatically selected template for the bicycles, motorcycles, and cars. Given these templates, our approach automatically extracted a deformation model for each class of shapes. We then used our

exploration interface to visualize and navigate the variability in the collection.

Figures 8a–e show the main modes of variation for the five different classes of shapes. Notice that our technique captures changes in both the size and position of the template components. For example, the distance between the front and back wheel accounts for most of the variability in the motorcycles, while the length and width of the body varies across the set of cars. Furthermore, as we discussed in Section 6.2, our method automatically learns and encodes the spatial dependencies between components to help the user understand and explore the space of allowable deformations as defined by the data. For example, symmetric components such as wheels and wings deform together in all four learned deformation models. We also capture less obvious dependencies, such as the body of the motorcycle moving downwards as the front wheel moves forward; this relationship is mainly due to several stretched, low-profile “chopper” bikes in the dataset, such as the one shown in the bottom row of Figure 8b.

As an additional experiment, we combined the bicycles and motorcycles into one collection to see if we could learn a single deformation model for these two related classes of shapes. The analysis produced a deformation model with two distinct modes of variation (Figures 8c and 8d). Interestingly, the second mode of variation in the combined set contains a deformation related to stretching the body along its width, so that skinny configurations corresponded to bicycles. Using these deformations, we were able to navigate through both the bicycles and motorcycles in our exploration interface, as shown in the accompanying supplementary video.

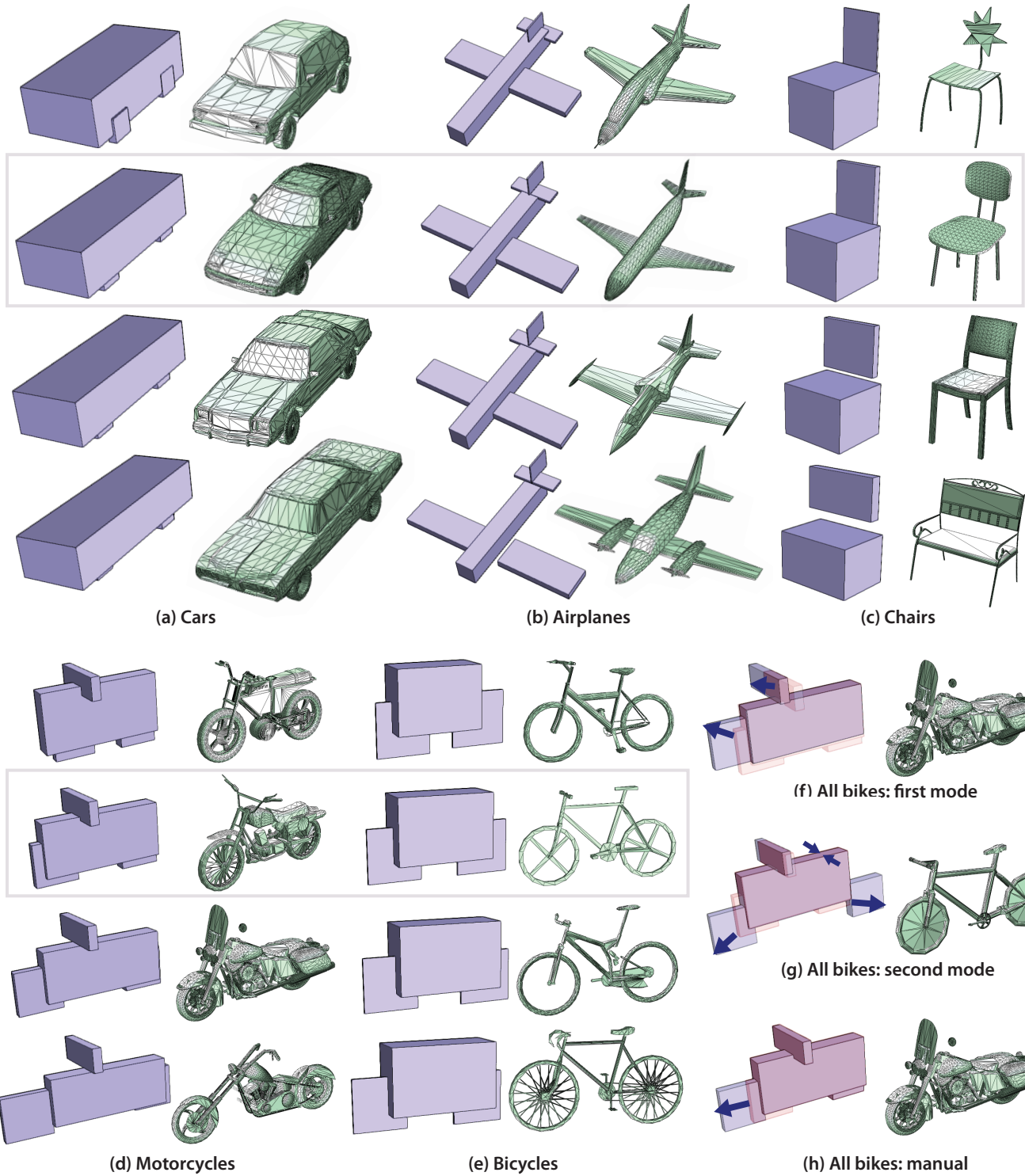
**Timing.** By far, the most time consuming part of our approach is pre-processing, when we compute the descriptor for each shape in the collection. Using a simple MATLAB implementation, this computation took under 18 seconds per shape on average for 2M point pairs on an Intel Core 2 Duo processor. During optimization, we only use 20k samples to compute the descriptor for efficiency. However, prior to optimization we find a set of 20k samples such that the descriptor computed using this set is within 0.5 percent of the descriptor computed using 2M pairs. The time for optimization depends on the number of connected components in the template, but always remained under 1-2 minutes for templates with up to 8 connected components.

## 8 Conclusions and Future Work

In this paper, we introduced a method to capture and explore variability in a collection of shapes without correspondences. Our main insight is to study the relation between the deformation of a shape and its global, high-dimensional descriptor. We introduced a modification of a previously suggested descriptor that varies smoothly under smooth deformations of the shape, and therefore allows us to analyze shape variability in the descriptor space. We described a method for recovering this variability in terms of deformations of a template model and described an exploration interface that allows us to navigate a collection of related shapes via restricted template manipulation.

Although our system has shown encouraging results, we believe that this is only a beginning and there is a number of interesting questions to be answered in the future. First, although our parametrization of the deformation space is low dimensional, the results of the optimization can be non-intuitive with parts “drifting apart.” An explicit encoding of the part connectivity in the optimization may be an interesting way to resolve this issue. Similarly, since we solve a non-convex non-linear problem, we may only find





**Figure 8: Summary of results.** We apply our method to extract templates and deformation models for five classes of shapes (a–e). For each class, we show sample template deformations on the left and the corresponding nearest neighbor shapes from the collection on the right. The grey boxes indicate the selected template shape for each dataset. Notice that the learned deformation models capture variations in both the size and position of components. They also encode dependencies between parts, so that, for example, the wings of the airplane move together. In addition, we visualize the principal modes of variation from a dataset with both bicycles and motorcycles (f–g). These modes capture several non-trivial deformations including movement of the wheels along the length and height of the template, and shrinking of the width of the body in the second mode (g). In contrast, a manual deformation of the template is unlikely to include all of the learned deformations (h).

a local minimum of the energy. A convex formulation of a similar optimization problem is an interesting problem for the future.

Other extensions to our framework may include outlier detection for shape retrieval by verifying similar shapes using curves in the descriptor space that their deformations produce. As mentioned earlier, it is also interesting to consider and compare other descriptors in which shape deformation results in explicit changes. Perhaps most interesting is the case of analyzing the relation of *discrete* variability in the shape (such as adding extra components), in terms of the changes of various descriptors. Finally, we are considering extensions to our exploration interface, where the user can query the system for deformations that explore different parts of the descriptor space.

**Acknowledgements.** We would like to acknowledge the INRIA GAMMA group and especially Eric Saltel for the model database [Saltel 2008], the anonymous reviewers for the helpful comments and suggestions, and Primoz Skraba for the extremely useful feedback and discussions. The work was partially supported by a KAUST visiting student scholarship, an Adobe internship, NSF grants FODAVA 0808515, IIS 0914833 and CCF 1011228, NSF/NIH MathBio grant 0900700, and a grant from Google Inc. Special thanks to Skype for enabling a seamless communication channel during all the stages of the project.

## References

- ALLEN, B., CURELESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. SIGGRAPH*, 587–594.
- ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM SIGGRAPH 24* (July), 408–416.
- BERNER, A., WAND, M., MITRA, N. J., MEWES, D., AND SEIDEL, H.-P. 2011. Shape analysis with subspace symmetries. *CGF (Proc. EUROGRAPHICS) 30*, 2, 277–286.
- BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, 187–194.
- BOUTIN, M., AND KEMPER, G. 2004. On reconstructing n-point configurations from the distribution of distances or areas. *Advances in Applied Mathematics 32*, 4, 709 – 735.
- CHAUDHURI, S., AND KOLTUN, V. 2010. Data-driven suggestions for creativity support in 3d modeling. In *ACM SIGGRAPH Asia*, 183:1–183:10.
- CHAZAL, F., COHEN STEINER, D., AND MÉRIGOT, Q. 2010. Geometric Inference for Measures based on Distance Functions. Research Report RR-6930, INRIA.
- COOTES, T. F., TAYLOR, C. J., COOPER, D. H., AND GRAHAM, J. 1995. Active shape models – their training and application. *Comput. Vis. Image Underst. 61*, 38–59.
- DRYDEN, I., AND MARDIA, K. 1998. *Statistical Shape Analysis*. John Wiley & Sons.
- FISHER, M., AND HANRAHAN, P. 2010. Context-based search for 3d models. In *ACM SIGGRAPH Asia*, 182:1–182:10.
- FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM SIGGRAPH 23*, 652–663.
- GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. Consistent segmentation of 3d models. *Comput. Graph. 33* (June), 262–269.
- KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3d mesh segmentation and labeling. In *ACM SIGGRAPH*, 102:1–102:12.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proc. SGP*, 156–164.
- KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric modeling in shape space. vol. 26, #64, 1–8.
- KIM, M.-J., KIM, M.-H., AND SHEN, D. 2008. Learning-based deformation estimation for fast non-rigid registration. In *CVPR workshop*, 1–6.
- KOKKINOS, I., AND YUILLE, A. 2007. Unsupervised learning of object deformation models. In *IEEE ICCV*, 1–8.
- LAGA, H., TAKAHASHI, H., AND NAKAJIMA, M. 2006. Spherical wavelet descriptors for content-based 3d model retrieval. In *SMI*, 15.
- MITRA, N. J., GUIBAS, L., AND PAULY, M. 2007. Symmetrization. In *ACM SIGGRAPH*, vol. 26, #63, 1–8.
- OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM TOG 21*, 807–832.
- OVSJANIKOV, M., BRONSTEIN, A. M., BRONSTEIN, M. M., AND GUIBAS, L. 2009. Shapegoogle: a computer vision approach for invariant shape retrieval. In *ICCV workshop, NORDIA*.
- SALTEL, E., 2008. INRIA Gamma team research database. <http://www-roc.inria.fr/gamma/download/download.php>.
- SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput. 24*, 249–259.
- SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. SGP*, 109–116.
- SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. *ACM SIGGRAPH 24*, 488–495.
- VAN KAICK, O., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. 2010. A survey on shape correspondence. *Computer Graphics Forum*.
- XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z.-Q. 2010. Style-content separation by anisotropic part scales. In *ACM SIGGRAPH Asia*, 184:1–184:10.