Projective Analysis for 3D Shape Segmentation

Yunhai Wang[⋆∗] Minglun Gong[†⋆] Tianhua Wang[‡⋆] Daniel Cohen-Or[⁎] Hao Zhang[§] Baoquan Chen[⋆♯∗]
[⋆]Shenzhen VisuCA Key Lab/SIAT [†]Memorial University of Newfoundland [‡]Jilin University
[⁎]Tel-Aviv University [§]Simon Fraser University [♯]Shangdong University

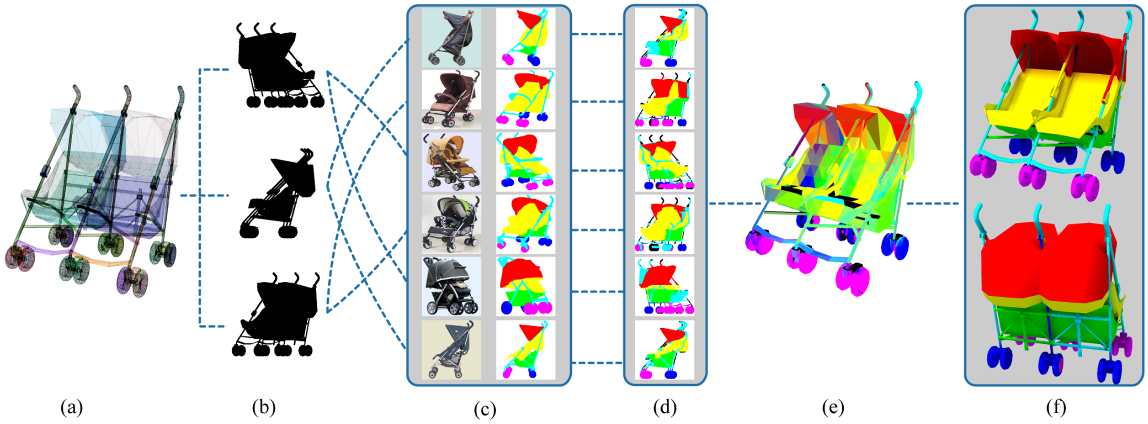


Figure 1: *Our projective analysis treats an input 3D model (a) as a collection of projections (b), which are labeled (d) based on selected images (c) from a pre-labeled image database. Back-projecting 2D labels onto the 3D model forms a probability map (e), which allows us to infer the final shape segmentation and labeling (f). Note how the labeling of the twin stroller is inferred from the images of single strollers.*

Abstract

We introduce *projective analysis* for semantic segmentation and labeling of 3D shapes. The analysis treats an input 3D shape as a collection of 2D projections, labels each projection by transferring knowledge from existing labeled images, and back-projects and fuses the labelings on the 3D shape. The image-space analysis involves matching projected binary images of 3D objects based on a novel *bi-class Hausdorff distance*. The distance is topology-aware by accounting for internal holes in the 2D figures and it is applied to *piecewise-linearly warped* object projections to compensate for part scaling and view discrepancies. Projective analysis simplifies the processing task by working in a lower-dimensional space, circumvents the requirement of having complete and well-modeled 3D shapes, and addresses the data challenge for 3D shape analysis by leveraging the massive available image data. A large and dense labeled set ensures that the labeling of a given projected image can be inferred from closely matched labeled images. We demonstrate semantic labeling of imperfect (e.g., incomplete or self-intersecting) 3D models which would be otherwise difficult to analyze without taking the projective analysis approach.

Links: DL PDF WEB DATA CODE

1 Introduction

Human visual perception of 3D shapes is based on 2D observations [Fleming and Singh 2009], which are typically obtained by projecting a 3D object into multiple views. A collection of the multi-view projective images together forms an understanding of the 3D object. This is the basic premise of multi-view 3D shape reconstruction and recognition [Ferrari et al. 2004], where 2D data as sparse as object silhouettes can be highly effective [Laurentini 1994]. Silhouettes turn out to be one of the most important visual cues in object recognition [Koenderink 1984], while binary images provide enriched shape characterizations. One of the most successful global shape descriptors for 3D retrieval is based on the multi-view light field descriptor (LFD) [Chen et al. 2003], which is computed from projected contour and image data.

In this paper, we propose projective analysis of 3D shapes beyond multi-view object reconstruction or recognition. We focus on the higher-level and more delicate task of semantic segmentation and labeling of 3D shapes. The core idea is to transfer labels from available 2D data by selecting and back-projecting the inferred labels onto a 3D shape. Rather than merely transforming a global shape analysis problem from 3D to 2D [Chen et al. 2003], we perform fine-grained shape matching in the projective space and establish connections between 2D and 3D parts to allow label transfer.

Potential gains offered by projective analysis are three-fold. First, analyzing projected images rather than 3D geometry can circumvent the requirement of having complete and well-modeled 3D shapes with quality surface tessellations, without losing the ability to discriminate or characterize the projected shapes. Second, working in a low-dimensional space, from 3D to 2D, simplifies the 3D shape segmentation task. Last but not least, the approach makes it possible to tap into and leverage the massive availability of image data, e.g., those from online photographs.

In recent years, there has been growing interest in data-driven analysis [Kalogerakis et al. 2010; Sidi et al. 2011] of 3D shapes to ad-

[∗]Corresponding authors: Yunhai Wang (cloudseawang@gmail.com), Baoquan Chen (baoquan.chen@gmail.com)

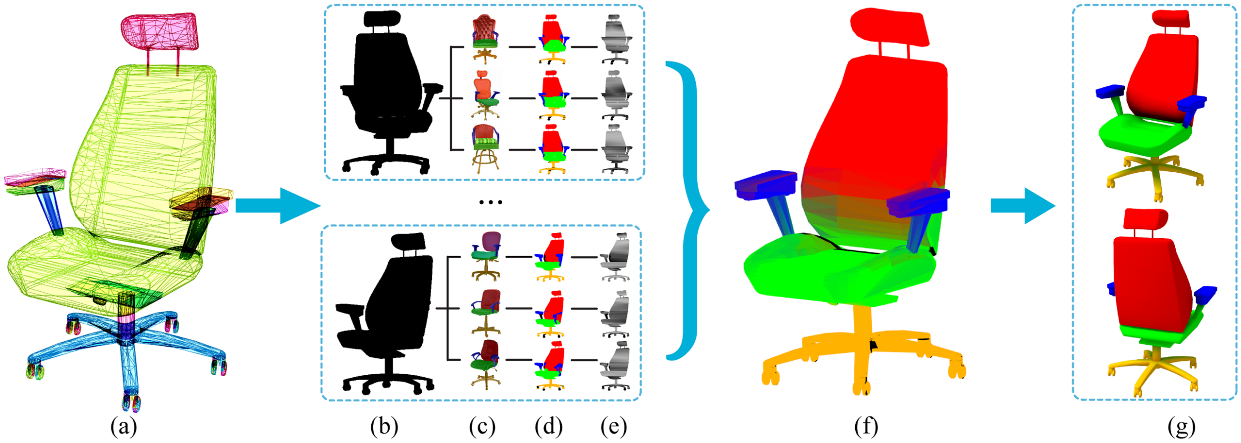


Figure 2: *Algorithm pipeline. Given a 3D shape (a), we produce a set of multi-view projections as binary images (two of them are shown in (b)). Each projection is used to retrieve multiple images from semantically labeled images (three retrieved images are shown in (c)). Label transfer is performed using each labeled image, resulting in a labeled projection (d) and an associated confidence map (e). All labeled projections and confidence maps are back-projected onto the input 3D model to compute the labeling probability map (f). Finally, graph cut segmentation is applied based on the labeling probabilities to produce the final segmentation and labeling (g).*

dress the challenge of learning shape semantics. The success of data-driven analysis rests directly on the quality of the utilized data. Quality datasets as a whole should be sufficiently large in number, as well as dense and rich in variability, so as to cover the variability in the input. At an individual level, each shape should possess an adequate representation quality (e.g., complete or watertight) to allow the computation of widely adopted shape descriptors which require surface (e.g., geodesics) or volume (e.g., shape diameter functions) analyses. Unfortunately, the quality of most community-built 3D models, such as those from the Trimble Warehouse, do not meet such quality criteria. Thus a recurring challenge faced by data-driven 3D shape analysis is the scant availability of quality 3D shapes and quality 3D shape collections.

Our projective analysis is image-driven and addresses the data challenge by utilizing image data. If relevant labeled 3D models exist, they are utilized as well but also in image form as multi-view 2D projections. The density and richness of detail necessary for a successful shape analysis is more likely attained by the large amounts of available image data rather than the less abundant available 3D shape data. Moreover, the incomplete shape information offered by projected images of a 3D object from limited views (in some cases, only a single-view capture is available) can be compensated by the aggregate of a large image collection, more specifically, by images of other similar objects in the collection. Finally and no less importantly, working with projections bypasses various difficulties in computing 3D shape descriptors, particularly over imperfect shapes, allowing the analysis to process them effectively.

Our algorithm segments and labels a 3D shape according to prior knowledge from a dataset of semantically labeled images. Given a 3D shape, we first obtain a series of its 2D projections from multiple views. Each projection is segmented through transferring labels from the most similar samples in a database. A *region-based* shape matching method that operates on binary images representing 2D shapes is used. It is based on a novel *bi-class Hausdorff distance* which is topology-aware by accounting for internal holes in the 2D figures. Moreover, the object images are warped in a piecewise linear fashion to compensate for view discrepancies and non-uniform object scaling. Based on the matchings found, labels from each matched labeled image, along with an associated confidence map, are transferred to the 3D shape via back-projection. Finally, the transferred labels from multiple views are integrated on the 3D input shape to obtain a segmentation and semantic labeling.

The main contribution of this paper is the introduction of projective analysis, which relies on image-space supervised learning, to semantically label a (possibly imperfect) 3D shape. Figure 2 gives an overview of our method. Extensive experiments show that our 2D matching approach can label projections using shapes with similar topology but different part scales and view directions, e.g., see Figure 1. This alleviates in part the density and richness requirements of the labeled set and compensates for using 2D instead of 3D data. We further demonstrate semantic labeling of imperfect models (non-manifolds, incomplete, or self-intersecting as shown in Figure 2) and 3D point clouds, which are difficult to analyze without projective analysis and utilization of 2D labeled data.

2 Related work

Shape Segmentation and labeling. Shape segmentation and labeling are closely related problems which are fundamental to computer graphics and many solutions have been proposed [Shamir 2008]. Earlier approaches focused on finding low-level geometric criteria to form meaningful segments or segment boundaries. However, it is difficult to develop precise mathematical models for what a meaningful shape part is. Recently, more research effort has been devoted to the data- or knowledge-driven approach.

Data-driven analysis. Instead of analyzing an input shape in isolation, the data-driven approach utilizes knowledge gained from labeled data or a shape collection. Representative approaches include supervised learning [Kalogerakis et al. 2010; van Kaick et al. 2011], unsupervised co-segmentation whereby a set of shapes is analyzed together [Golovinskiy and Funkhouser 2009; Xu et al. 2010; Sidi et al. 2011], and semi-supervised segmentation via active learning [Wang et al. 2012]. The novelty of our work lies in the utilization of 2D labeled data and supervised learning in the projective space to facilitate 3D shape analysis.

Common to all data-driven approaches is their reliance on data quality. In terms of data size, the largest mesh segmentation benchmark [Chen et al. 2009] contains 380 meshes on 19 object categories. In contrast, available image segmentation datasets are much larger, e.g., ImageNet [Deng et al. 2009] contains over 15 million well labeled images in over 22,000 object categories. By utilizing ImageNet to train object detectors, Lai et al. [2012] demonstrated that the resulting detector can be used to reliably label objects in 3D scenes. While the amount of available 3D data continues to grow, it is unlikely that it will ever come close to matching the volume of image data. Moreover, compared to 2D images, 3D shapes are inherently more difficult to acquire and process, requiring more effort to label and analyze. Our work demonstrates the advantages of image data, in terms of its sheer volume and relative ease for processing, which can be exploited to address challenges arising from the segmentation of 3D shapes.

Projective shape analysis. Treating a 3D shape as a collection of 2D projections rendered from multiple directions is not new to computer graphics. Murase and Nayar [1995] recognize an object by matching its appearance with a large set of 2D images obtained automatically by rendering 3D models under varying poses and illuminations. Lindstrom and Turk [2000] compute an image-space error metric from these projections to guide mesh simplification. Cyr and Kimia [2001] generate projections from selected view directions and use them to identify 3D objects and their poses. Sketch- or image-based 3D shape retrieval [Eitz et al. 2012] compares object projections with query images or user-drawn sketches in 2D. Similarities among 2D shapes can be evaluated using techniques such as LFD [Chen et al. 2003] and cross-correlation [Makadia and Daniilidis 2010]. Liu and Zhang [2007] embed a 3D mesh into the spectral domain, turning the 3D segmentation problem into a contour analysis one. 3D reconstruction from multi-view images is one of the most fundamental problems in computer vision. Our work applies projective analysis to a new application: semantic segmentation of 3D shapes. Specifically, we fuse labeled segmentations learned from back-projected 2D labels to obtain a coherent semantic labeling of a 3D object.

Image and shape hybrid processing. 3D shape reconstruction often benefits from utilizing available 2D data, e.g., from registered photographs, to improve the quality of 3D scans [Li et al. 2011]. On the other hand, leveraging *a priori* 3D geometry of a given object category can alleviate the ill-posed nature of image analysis from single photographs. Chang et al. [2009] and Pepik et al. [2012] combine the representational power of 3D objects with 2D object category detectors to estimate viewpoints. Xu et al. [2011] take a data-driven approach for photo-inspired 3D shape creation, where the best matching 3D candidate is deformed to fit the silhouette of the object captured in a single photograph. In our work, we also take a hybrid approach where the semantics of 3D shapes is guided by constraints learned via projective shape analysis.

Image retrieval. Measuring image similarity for retrieval is extensively studied in computer vision; see [Xiao et al. 2010] for a systematic study of image features for scene retrieval. Well-known distance measures between 2D shapes include Hamming distance, Hausdorff distance [Baddeley 1992], shock graph edit distance [Klein et al. 2001], distance between Fourier descriptors [Chen et al. 2003], inner distance shape context [Ling and Jacobs 2007], and context-sensitive shape similarity [Bai et al. 2010]. Different from previous attempts, we do not only retrieve a 2D shape but also infer a semantic labeling of its interior. Unlike existing contour-based methods [Ling and Jacobs 2007], our region-based analysis allows shape retrieval and label transfer to be conducted in a coherent manner. Moreover, our image retrieval is not cross-category, but within-category, with the goal of finding shapes with similar topological features to guide part-aware label transfer. To properly evaluate the differences between the corresponding parts of two shapes, we implicitly warp one shape to match the other, before computing dissimilarity using a topology-aware Hausdorff distance measure.

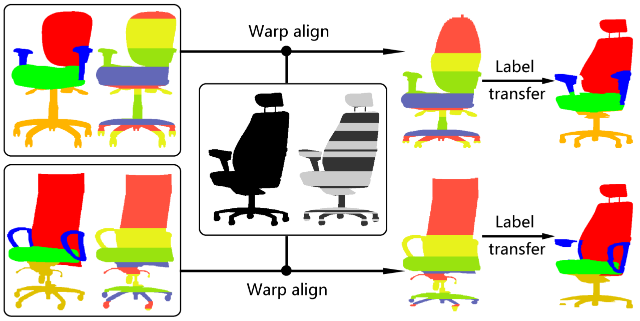


Figure 3: *Region-based matching via warp alignment. Both the labeled images (left column) and query projection (middle column) are cut into axis-aligned slabs. Each labeled image is then warped to match the query projection. The dissimilarity is measured using warp-aligned shapes, allowing the matching to favor the shape with similar topologies (top row) over the one with parts at similar scales and positions (bottom row). Note that although the bottom chair is visually more similar, the top chair is more useful for labeling the armrest area in the query projection.*

Image label transfer. Semantic label transfer is another core problem in computer vision. Existing approaches can be classified into learning-based and non-parametric based. The former ones try learn a model for each object category. A successful method is Textonboost [Shotton et al. 2006], which trains a conditional random field (CRF) model. A problem of learning-based methods is that they do not scale well with the number of object categories. With the emergence of large image databases, non-parametric methods have demonstrated their advantages. Given an input image, Liu et al. [2011a] first retrieve its nearest neighbors from a large database using GIST matching [Oliva and Torralba 2001]; then transfer and integrate together annotations from each of these neighbors via dense correspondence estimated from SIFT flow [Liu et al. 2011b]. Compared to learning-based approaches, this method has few parameters and allows simply adding more images and/or new categories without requiring additional training. When the set of annotated images is small, Zhang et al. [2010] and Chen et al. [2012] further learn an object model from the retrieved nearest neighbors to improve the performance of label transfer. Our approach incorporates the same nearest neighbor idea, but instead of performing label transfer within the whole image domain, we compute semantic labeling for the interior of the 2D shape only. This provides us additional constraints for obtaining a better labeling result. In addition, almost all existing dense correspondence estimation approaches [Liu et al. 2011b; Berg et al. 2005; Leordeanu and Hebert 2005; Duchenne et al. 2011] rely on local intensity patterns and are unsuitable for transferring labels to textureless 2D projections.

3 Overview

Our image-driven shape analysis is based on a dataset of pre-labeled images which captures the semantic knowledge about the relevant class of shapes. The input is a 3D mesh model, possibly non-manifold, incomplete, or self-intersecting. The 3D shape and the labeled images belong to the same semantic class. We assume that both the input and the objects captured in the labeled images are in their upright orientations. In practice, we found the assumption to hold for the vast majority of the data, e.g., almost all chair images found on Google. We apply our multi-view shape matching and back-projection method to obtain a segmentation and semantic labeling of the 3D shape; see Figure 2 for an overview.

Dataset. The labeled dataset consists of a large collection of images gathered from the Web and organized into several semantic classes. The foreground object in each image is extracted using Grabcut [Rother et al. 2004] and different semantic parts of the object are manually segmented and labeled. Note that by using our shape matching technique described below, we are able to transfer labels from processed images to novel ones, providing an initial labeling result that facilitates the manual labeling process. To further enrich the labeled set, multi-view projections of available labeled 3D shapes can be added to the dataset as well.

Shape matching in projective space. The matching and dissimilarity measures between two projected binary images are at the core of our method (Section 4). They are required for the retrieval of labeled 2D objects having a matching shape and pose as well as for image correspondence during label transfer. Our matching technique is region-based and takes advantage of the upright orientation and pose alignment from retrieval. The dissimilarity estimation between two binary images (one query and one labeled) relies on a novel bi-class symmetric Hausdorff (BiSH) distance between the query image and an implicitly warped version of the labeled image.

Specifically, we independently cut each projected image into topologically homogeneous slabs along horizontal and/or vertical directions. Each slab is formed by clustering horizontal or vertical scanlines of the image sharing the same topology. Then given two slabbed images, we apply dynamic programming to match the slabs based on BiSH distance. This is followed by a piecewise linear warp of the slabs from the labeled image to align with the corresponding slabs in the query image; see Figure 3. Finally the dissimilarity measure is computed between the warp-aligned images.

Semantic labeling via back-projection. We extract multi-view projections of the input shape and for each projection, retrieve similar labeled images from the dataset using our shape matching technique. A label map is generated for the projection using each warp-aligned labeled image and a confidence value is calculated for each transferred label, forming a confidence map. Both the transferred label map and confidence map are back-projected to the 3D shape based on correspondences established when projecting the input.

Since different labels may be assigned to the same area of the 3D model, we collect all the labels and the associated confidences to build a probability map over the input shape. A graph cut optimization is applied based on the probability map and typical geometric cues for shape segmentation to produce the final labeling of the 3D shape. This labeling process is robust since it implicitly relies on a voting procedure for back-projecting the labels. The graph cut optimization leads to contiguous and smooth labeled regions.

4 Region-based Shape Matching

State-of-the-art methods for shape matching are contour-based, heavily relying on corresponding features [Belongie et al. 2002; Ling and Jacobs 2007]. However, how to use contour matching results for label transfer between two shapes is a non-trivial problem. Furthermore, in our setting, the contour features may not be descriptive enough to capture the dissimilarity between the shapes, while the topology of shapes (e.g., their holes and protrusions) more prominently dominates their similarity.

The premise that the shapes are given in their upright orientations allows a significant relaxation on matching measurement. The key is that some shapes can only be registered through an axis-aligned warp, and hence the dissimilarity measure needs to be calculated between warp-aligned shapes. Here we represent each shape using an array of axis-aligned rectilinear regions, which are defined as the intersection of vertical and horizontal slabs; see Figure 6. Each slab is a cluster of horizontal (vertical) scanlines of similar shape and topology, and hence can be represented using a single representative scanline. The alignment between two shapes is performed by finding slab correspondence between the two and adjusting the height (width) of horizontal (vertical) slabs to match each other. The dissimilarity between warp-aligned shapes is decomposed into two axial distances, each of which is an aggregate of distances between the representative scanlines of the corresponding slabs. Thus, the fundamental inter-slab region-based dissimilarity measure is reduced to a one dimensional problem. This quantization of the shapes into rectilinear regions significantly accelerates the warping and the calculation of the dissimilarity measure.

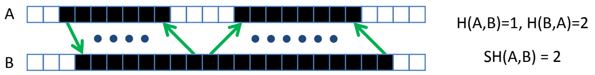


Figure 4: *Hausdorff distance between two lines. The green arrows link selected pixels to their closest pixels on the other line.*

4.1 Matching between 1D shapes

Recall that we want our measure to be topology-aware. Thus, given two streams of black and white pixels, we need to measure the distances not only among the black pixels, but also the white ones, as they represent the holes. Here we build upon the Hausdorff distance and extend it to treat equally the two classes of pixels.

Let A and B represent the set of black pixels in two given lines, respectively. The symmetric Hausdorff distance between A and B is given as $SH(A,B) = \max(H(A,B), H(B,A))$, where $H(A,B) = \max_{a \in A}(\min_{b \in B} dist(a,b))$. As shown in Figure 4, the above definition finds, for each black pixel in one line, the closet black pixel in the other; and then outputs the maximum distance among all matching pairs.

While symmetric Hausdorff distance provides a reasonable distance measure between two 1D shapes, it is not sensitive enough to topology changes. For example, as shown in Figure 5, whether a hole exists or not in one of the shapes, does not affect the value of $SH(\cdot,\cdot)$. To address this problem, we define a bi-class symmetric Hausdorff distance that considers the two classes symmetrically:

$$BiSH(A,B) = \max(SH(A,B), SH(A^c,B^c)), \qquad (1)$$

where A^c and B^c denote all the white pixels in the two lines, respectively. As shown in Figure 5, the existence of a small hole can dramatically change the $BiSH(\cdot,\cdot)$ values, providing sensitivity to topological differences between the two 1D shapes.

Armed with the above dissimilarity measure between scanlines, we can cluster scanlines in a 2D binary shape T into slabs based on their shapes and topologies. Here we simply apply a 1D variant of the hierarchical clustering algorithm [Telea and Van Wijk 1999] to iteratively combine two adjacent slabs that have the minimal weighted dissimilarity between them, until the number of remaining slabs reaches a user-specified number. That is, we merge the i^{th} slab with the $(i+1)^{th}$ one, if

$$i = \arg\min_{0 \le i < n} (\sqrt{h_i + h_{i+1}} \times BiSH(T[r_i], T[r_{i+1}])), \qquad (2)$$

where n is the number of horizontal (vertical) slabs currently used to represent T, r_i is the row (column) number of the representative

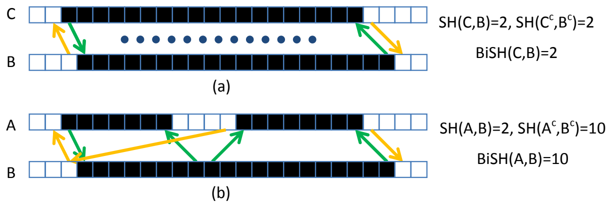


Figure 5: *Bi-class Hausdorff distance between two lines. Green and yellow arrows connect the matching black and white pixel pairs, respectively.*

scanline for the i^{th} horizontal (vertical) slab, and h_i is its height (width). $T[\cdot]$ returns the scanline at a given row (column) of image T. The weighting term defined using the square root function encourages small slabs to be merged first without overpenalizing large slabs. Once a new slab is formed from merging, we pick the middle scanline as its representative. Figure 6 compares the slab quantization results generated using both the conventional Hausdorff definition and our bi-class approach. It shows that our results better respect topology changes between adjacent scanlines. Consequently, the slab generation process always cuts the image at scanlines with topology changes, producing stable slab segmentations.

4.2 Matching between 2D shapes

Given two shapes, labeled S and unlabeled T, the warp-aligned dissimilarity between them is defined as:

$$D(T,S) = \sum_{0\le i<n} h_i \times BiSH(T[r_i], S[r_{W(i)}]), \qquad (3)$$

where n, r_i, h_i, and $T[\cdot]$ follow the definitions in (2). $S[\cdot]$ are scanlines in S and $W(\cdot)$ is an axial-aligned mapping function that scales slabs in T to match those in S. Note that the above definition implicitly requires that $W(\cdot)$ is single valued, that is, a slab in T can only map to a single slab in S. In practice, we deliberately over-segment T into more regions than S (25 slabs for S and 50 for T in our experiments), and hence this many-to-one requirement is likely satisfied; see Figure 7.

To find the optimal slab mapping function $W(\cdot)$ and the corresponding dissimilarity value $D(T,S)$, we first compute a slab matching cost matrix M, where each entry is the distance between corresponding slabs: $M_{i,j} = h_j \times BiSH(T[r_j], S[r_i])$.

As shown in Figure 7, given $M_{i,j}$, the optimal slab mapping function $W(\cdot)$ that gives the smallest dissimilarity value $D(T,S)$ can be found by searching for a path that goes from the top left corner to the bottom right corner of the table. In addition, to satisfy the requirement that $W(\cdot)$ is single valued, the path should step through only one cell at each column and therefore only horizontal and diagonal moves are allowed. Such a path can be efficiently calculated using a variant of the dynamic time warping (DTW) [Berndt and Clifford 1994] technique with restricted moves, and the total cost along the path yields the $D(T,S)$ value.

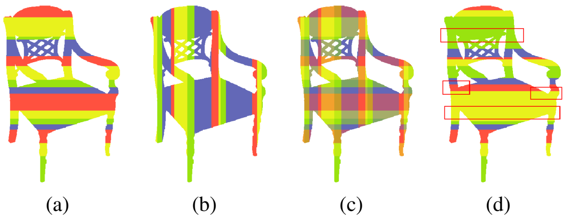


Figure 6: *The horizontal slabs (a), vertical slabs (b), and axis-aligned regions (c) generated for a chair shape using bi-class symmetric Hausdorff distance. For comparison, the horizontal slabs generated using symmetric Hausdorff distance are shown in (d). In areas highlighted by red boxes, scanlines with different topologies are improperly clustered into the same slab.*

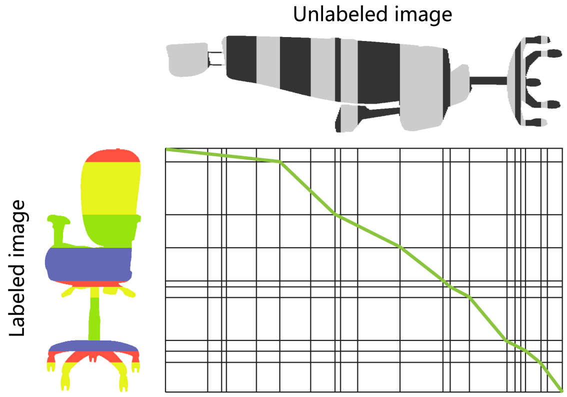


Figure 7: *Finding corresponding slabs between the labeled and unlabeled images; both are squeezed for space saving purpose only.*

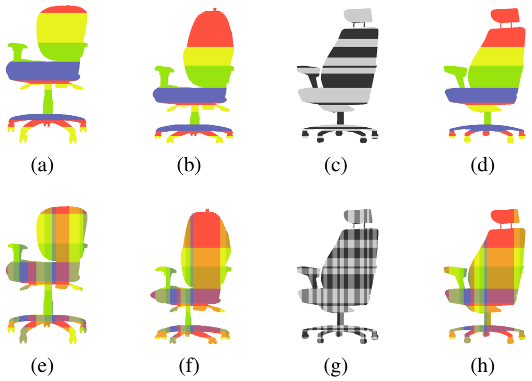
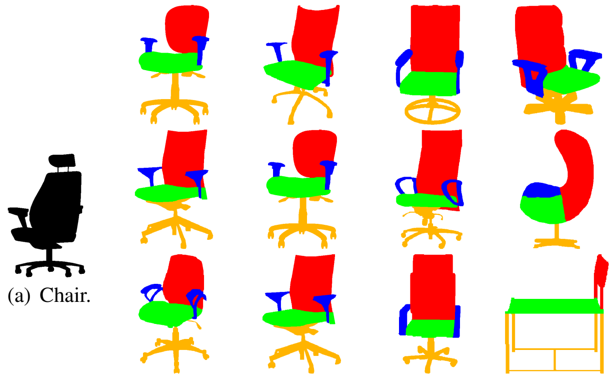


Figure 8: *Warping the labeled image S makes it better aligned with the unlabeled image T. (a) Horizontal slabs of S. (b) Warping result of (a) along vertical direction. (c) Horizontal slabs of T, (d) Recoloring of slabs in (c) based on the matching slabs in (b); note the many-to-one mapping relation between (c) and (d). (e) Regions of S. (f) Warping result of (e) along both directions. (g) Regions of T. (h) Recoloring of regions in (g) based on their matching regions.*

Figure 8(b) demonstrates the effect of warping image S along the vertical direction using the $W(\cdot)$ function shown in Figure 7. Figure 8(f) further shows the results of warping S along both vertical and horizontal directions. While the additional horizontal warping step can be beneficial in cases such as matching between 3-seat and 2-seat sofas, in practice we find that warping horizontal slabs along the vertical direction only, gives satisfactory retrieval results in most cases. Hence, in all the results shown in the paper, only vertical warping is performed. Scaling along horizontal direction is achieved through pixel-to-pixel correspondence search during the label transfer step; see Section 5.2.

Figure 9 compares the shapes retrieved by ours and other representative approaches, including inner-distance shape context



(a) Chair.

(b) Ours. (c) IDSC. (d) GIST. (e) LFD.

Figure 9: *The top three ranked images retrieved by different approaches for a chair shape (a). Both our region-based approach (b) and IDSC (c) provide shapes with similar topologies, e.g. all have armrests and rolling wheels. GIST (d) returns images that are visually very similar (e.g. high back chairs), but topologically different (e.g., no rolling wheels in the top ranked chair). LFD (e) finds objects with incorrect poses and dissimilar shapes.*

(IDSC) [Ling and Jacobs 2007], LFD [Chen et al. 2003], and GIST descriptor [Oliva and Torralba 2001]. Both IDSC and LFD are widely used for shape retrieval, with LFD measuring holistic features whereas IDSC capturing part structures. Sucessfully used for scene level retrieval and label transfer [Liu et al. 2011a], GIST computes a low dimensional representation for the input image. It is used here to retrieve 2D shapes by treating them as binary images.

The visual comparison verifies that measuring bi-class Hausdorff distance over warp-aligned images can effectively find shapes with similar topology. IDSC also retrieves topologically similar shapes, but is much slower than ours. The images found by GIST are visually more similar than the previous ones, but are less optimal for labeling the query projection at part level. Additional comparisons are provided in Figures 2–5 of the supplementary material.

5 Back-projection and label transfer

Now we describe the whole pipeline of our projective analysis algorithm. During the preprocessing stage, we normalize all labeled images in the database to the same size through uniform scaling. We then generate horizontal slabs for each labeled image. Next, to segment a given 3D model P, the following steps are performed.

5.1 Retrieve labeled images using projections

With the upright orientations, the model P is first projected into 2D shapes from a set of pre-determined viewpoints. These viewpoints are chosen to roughly match common views used for capturing the images in the database. For example, we typically set the camera slightly above the model and rotate the model 360 degrees at 6 degrees per step to generate the projections on all sides. Similar with [Chen et al. 2003], we turn off the lighting and apply the orthogonal projection; see Figure 2(b). The obtained projection may be scaled to match the size of the labeled images in the database.

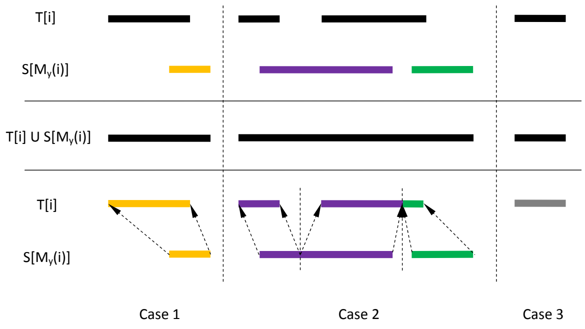


Figure 10: *Pixel-to-pixel mapping between two scanlines.*

For each projection T, we first compute its slab representation and use it to evaluate its dissimilarities to each labeled image S in the database; see Figure 2(c). The top K_2 images with smallest dissimilarities $D(S,T)$ are kept, which are used to compute the average matching cost for projection T. The ensuing label transfer and back-projection operations are only applied to the top K_1 non-adjacent projections with the smallest average matching costs. Hence there are two key parameters: the number of projections (K_1) and the number of labeled images for each projection (K_2). The back-projection uses a total of $K_1 \times K_2$ images.

5.2 Transfer label information to projections

Transferring labels from image S to projection T is performed for each scanline independently. For the i^{th} horizontal scanline $T[i]$ in T, we first find its corresponding scanline $S[M_y(i)]$ in S, where $M_y(\cdot)$ maps scanlines from T to S based on the slab mapping function $W(\cdot)$. Although it is not performed explicitly, this is equivalent to first warping S using $W(\cdot)$ and then matching between scanlines in the corresponding row; see Figure 8.

Next, we setup pixel correspondence between occupied pixels in $T[i]$ and $S[M_y(i)]$ and perform label transfer accordingly. Since there is no cue to infer parts inside the projected shape T, the following heuristic approach is used for pairing the two scanlines. As shown in Figure 10, we first compute the union of the two sets of occupied pixels; then we detect gaps in the union set. The gaps split the scanline into multiple regions, which are processed separately.

Within each region, there could be three different scenarios: i) only one segment exists in both scanlines; ii) more than one segment exists in either scanline; iii) there is no segment in either scanline. In the first case, where a one-to-one relationship exists, we simply scale and shift the segment in $S[M_y(i)]$ to match the one in $T[i]$ and copy the labels over. In the second case, we first split the region using the midpoint of gaps between adjacent segments to obtain a one-to-one mapping relationship and apply shifting and scaling as in case one. Finally, in case three, no label transfer is performed.

To each pixel label transfer $L(i,j)$ we also associate a confidence value $C(i,j)$. The confidence is calculated heuristically using three cost terms: i) a per-image term, $ci = D(T,S)/(\sum_{0\leq i<n} h_i)$, that depends on the overall differences between the two shapes; ii) a per-scanline term based on the dissimilarity between corresponding scanlines: $cs(i) = BiSH(T[i], S[M_y(i)])$; and iii) a per-pixel term based on among of shifting within the scanline: $cp(i,j) = |j - M_x(i,j)|$, where $M_x(i,j)$ is the column number of the pixel in S that is used to label (i,j) in T. The rationale for such a definition is that the higher the cost of matching and the more stretching effort is required to label a pixel, the less confidence we have on the label transfer result. That is:

$$C_{i,j} = \exp\left(-(ci + cs(i) + cp(i,j))^2/\sigma^2\right), \qquad (4)$$

where σ is the Gaussian support (set to 150 in our experiments). Figures 2(d-e) show label transfer results and corresponding confidence maps obtained using each of the retrieved label images.

5.3 Back-project labels and graph cuts optimization

Now we map the label and the confidence of each pixel from the labeled projection backwards to the 3D shape. Here we assume that each primitive μ (triangle in mesh and point in raw scan) in the 3D shape belongs to only one semantic part and hence only carries one label in the final result. Using $p(l|\mu)$ to denote the probability of assigning label l to μ, we derive $p(l|\mu)$ as:

$$p(l|\mu) = \frac{\sum_{(i,j)\in\Omega_\mu} \delta(L(i,j) - l)C(i,j)}{\sum_{(i,j)\in\Omega_\mu} C(i,j)}, \tag{5}$$

where Ω_μ is the set containing all pixels that backproject to primitive μ, and $\delta(\cdot)$ is the Dirac's delta function.

Figure 2(f) visualizes the $p(l|\mu)$ values calculated for different primitives, based on which we employ a multi-label alpha expansion graph-cut algorithm [Boykov et al. 2001] to arrive at the final labeling. Given a shape, we define the graph $G = \{V, E\}$, where the nodes V are given by the primitives of the shape. To deal with imperfect shape representations, such as polygon soups or point clouds, where connectivity information is unavailable, we construct two types of edge networks E_1 and E_2.

The first, E_1, relies on proper connectivity information. If the primitives represented by nodes μ and ν are connected, we add an edge $\{\mu, \nu\}$ to E_1. To further enforce neighboring primitives having coherent labels regardless of their connectivities, E_2 is constructed based on their proximities. That is, we build a kd-tree for all primitives represented by their mass centers and then add edges between each node and its k nearest neighbors to E_2. The parameter k is empirically set to 5 in all our experiments. The optimization is then posed as finding the labeling l that minimizes the energy:

$$\xi(l) = \sum_{\mu\in V} \xi_D(\mu, l_\mu) + \omega \sum_{\mu\nu\in E_1} \xi_{sc}(l_\mu, l_\nu) + \lambda \sum_{\mu\nu\in E_2} \xi_{sd}(l_\mu, l_\nu), \tag{6}$$

where l_μ and l_ν are the labels assigned to nodes μ and ν, respectively. ω and λ are two constants that balance the influences of the data energy term (ξ_D), the connectivity-based (ξ_{sc}) and distance-based (ξ_{sd}) smoothness terms. These terms are defined as:

$$\begin{aligned} \xi_D(\mu, l_\mu) &= -\log(p(l_\mu|\mu)), \qquad (7) \\ \xi_{sc}(l_\mu, l_\nu) &= -\log(\theta_{\mu\nu}/\pi) l_{\mu\nu}, \\ \xi_{sd}(l_\mu, l_\nu) &= -\log(d^2(\mu, \nu)), \end{aligned}$$

where $l_{\mu,\nu}$ is the length of the edge, $\theta_{\mu\nu}$ is the positive dihedral angle and $d(\mu, \nu)$ is the Euclidian distance between nodes μ and ν. Figure 2(g) shows the result of solving the label assignment.

6 Results

In this section, we present experimental results and evaluation for our projective analysis method, over large and varied datasets. Two types of labeled datasets are used. The first consists of projections of labeled 3D models, which is used for comparing with methods that rely on 3D labeled data. By using only projections of available 3D data, no extra knowledge is introduced. The second labeled dataset contains only photos downloaded from the Internet, which were subsequently labeled. These photos are grouped into eleven categories, as shown in Table 1.

The resolution of all the projections and labeled images is set to 512×512. Since our image dissimilarity measure is calculated over a small number of slabs, the core image matching step is fairly efficient. Matching a projection with 500 labeled images takes about 30 seconds. After matching a projection, the label transfer and back-projection steps each takes about one minute on input meshes containing about 20K triangles.

Table 1: *A dataset consisting of labeled online photos only. The number of semantic parts and the size for each category are shown.*

Category	# parts	# photos	Category	# parts	# photos
Chair	4	500	Stroller	6	400
Truck	3	400	Lamp	3	344
Vase	4	300	Table	2	250
Bike	5	181	Pavilion	3	60
Guitar	3	20	Fourleg	5	234
Robot	4	174			

Evaluation and comparison. We performed two large-scale quantitative experiments. The first evaluates our method on the dataset used by the supervised learning method of Kalogerakis et. al [2010]. This dataset consists of 3D shapes that are well modeled, in the sense that they are manifolds, complete, with no self-intersecting pieces. These shapes are classified into 19 categories, seven of which represent objects with upright orientations and hence are used here. Among the seven categories, the first five are rigid objects, whereas the last two are articulated ones.

For a fair comparison between our method and that of Kalogerakis et al. [2010], we used the same segmentation quality measure (in terms of recognition rates) and matched their experiment settings as much as possible. For example, for the "SB3" testing, like in Kalogerakis et al. [2010], we randomly select 3 labeled models from 20 available ones in the corresponding category; the dataset is used to label one randomly selected input model and this test is repeated five times to compute the average recognition rate. To generate the labeled images, each model is projected from 60 views. The top 25 ranked images are used for label transfer ($K_1 = 25$, $K_2 = 1$); and the graph cut parameters ω and λ are set to 10 and 50, respectively. The same set of parameters is used in all tests.

Comparison results between the two methods under different experimental settings (Table 2) show that our method performs slightly better on rigid objects (95% to 93%), but worse on articulated ones (57% to 88%). We argue that the better performance is somewhat surprising since our labeled data consists of partial information, i.e., a subset of projections, from the same set of 3D models available to the competing method. The advantages of using full 3D models for labeling are inherent since, for example, cavities are typically not captured by projections. The poor performance on articulated objects is expected since our shape retrieval and label transfer methods are designed for rigid man-made objects with known upright directions. To properly handle articulation, other shape retrieval methods, such as IDSC [Ling and Jacobs 2007], can be incorporated into the proposed projective analysis framework. While our current approach is limited to rigid objects, it can process imperfect shapes (non-manifold, non-watertight, polygon soups, point cloud, or incomplete), which cannot be handled by existing approaches, like Kalogerakis et al. [2010]. We believe that this is a significant advantage and a useful feature.

Another quantitative evaluation is carried over the large dataset used by Wang et. al [2012]. Two categories of rigid 3D shapes (400 chairs and 300 vases) are used. This time we evaluate the effects of not only the number of available labeled samples (percentage of labeled models used for generating the labeled set) but also the number of images actually selected for label transfer and back-projection (parameters K_1 and K_2). The recognition rates under different settings are plotted in Figure 11. With respect to the effect of available labeled samples, similar trends as the previous test are observed, i.e., our method performs better when more sam-

Table 2: *Comparison between our method and Kalogerakis et al. over seven object categories. Higher recognition rates, indicating higher segmentation quality, are shown in* ***bold****.*

	Ours				Kalogerakis et. al [2010]			
Set	SB19	SB12	SB6	SB3	SB19	SB12	SB6	SB3
Chair	**99.2**	**99.6**	**97.9**	93.4	98.5	98.4	97.8	**97.1**
Table	**99.6**	**99.6**	**99.6**	**99.4**	99.4	99.3	99.1	99.0
Vase	**91.9**	**90.5**	**89.7**	**80.8**	87.2	85.8	77.0	74.3
Mech	94.6	**91.3**	**90.2**	**90.6**	94.6	90.5	88.9	82.4
Cup	99.1	**99.6**	97.5	94.4	**99.6**	96.0	**99.1**	**96.3**
Fourleg	67.9	54.3	59.1	58.6	**88.7**	**86.2**	**85.0**	**82.0**
Human	63.8	55.6	51.1	48.0	**93.6**	**93.2**	**89.4**	**83.2**

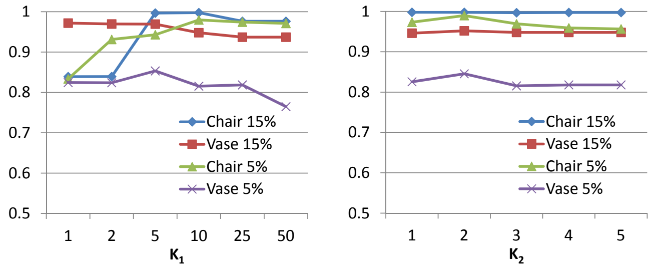


Figure 11: *Recognition rates by our algorithm under different parameters for two shape categories, each with two sampling rates for labeled set generation. Left: Varying the number of projections (K_1) while fixing the number of candidates retrieved for each projection ($K_2 = 2$). Right: Changing K_2, while fixing $K_1 = 10$.*

ples are available (i.e., under 15% sampling rate). The plots also show that the best performance is achieved when K_1 is between 5 and 10 and K_2 is around 2. Further increasing K_1 noticeably degrades the performance. Our hypothesis for this phenomenon is that not all projections carry sufficient topology information for reliable matching. Using more projections with less matching confidences often introduces poor label transfer results and hence hurts the overall performance. Similar phenomena were observed by previous nearest neighbor based approaches, e.g., [Liu et al. 2011a].

Labeling imperfect 3D models. To test the robustness of our method in labeling impefect inputs, we used both 3D meshes from Trimble Warehouse and point clouds. Meshes from Trimble Warehouse are often non-manifolds, incomplete, and self-intersecting, whereas the point clouds are noisy and sparse. Working on 2D projections allows us to bypass these imperfections. Here the labeled set shown in Table 1 is used and 3–10 models are tested for each category. The labeling results are shown in Figures 1, 2, 13 and 12, as well as Figures 6–17 in the supplementary material. The results adequately demonstrate the capability of our algorithm in labeling imperfect shapes with complex topologies and fine-scale parts.

Note that we set $K_1 = 3$ and $K_2 = 2$ in these tests. The value of K_1 is smaller than the optimal one found in the previous evaluation since, unlike projections of labeled models, labeled photos in our labeled set are mostly taken from a limited range of popular view directions. Nevertheless, the default setting of K_1 and K_2 cannot guarantee superior results for all scenarios. As seen in Figure 12(a), the default setting loses to an alternative setting.

While our approach works robustly for most models we tested, it does fail when it cannot find a good match in the dataset for the input projections. As shown in Figure 14, since our labeled dataset does not contain images of two-seat bicycles, the top ranked image has quite a different topology from the input model. As a result, one of the seats is incorrectly identified as handle, whereas the other seat and the handle are labeled as body.

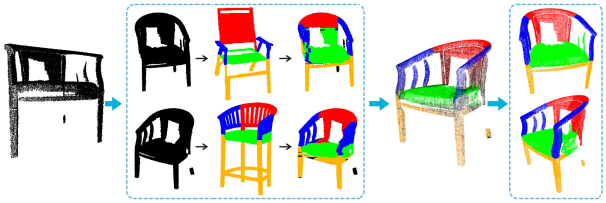


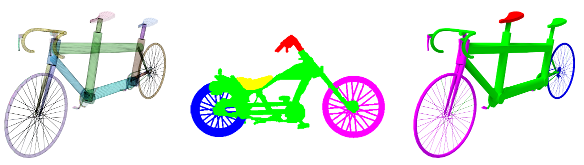
Figure 13: *Labeling results for a raw chair scan, a point cloud input. From left to right we show input point cloud obtained using Kinect, top two ranked images and transferred labels, probability map, and final label results under two different views.*

Learning from mixed categories. It is interesting to test the performance of the method when the given input object is unclassified. That is, its labeling is learned from a training set of mixed categories. To explore the potential of our method in such a setting, we test how well the method can correctly identify the category that a given shape belongs to, i.e., the classical shape retrieval problem.

Here we randomly select 50 images from each of the first eight categories in Table 1 to constitute a testing dataset. The remaining three categories are excluded due to either articulation (Fourleg and Robot) or insufficient number of photos (Guitar). Then, for every image in the database, we match it with all other images and use the top five matches to compute a confusion matrix [Csurka et al. 2004]; see Table 3. The performances of other shape retrieval methods under the same settings are shown in Table 2 and 3 of the supplementary material.

The results show that our shape matching algorithm performs well for categories such as bicycle and table; and hence it can properly label parts for 3D bicycle and table models without prior knowledge about what they are. However, it becomes confused between categories such as chair and stroller. We consider such performance as not robust enough. It should be stressed that, like most shape analysis methods, we used a classified labeled set. How to properly use heterogeneous labeled sets is left for future work.

Labeling 2D images. Finally, we analyze the performance of our approach for labeling 2D images and compare it with Liu et al. [2011a]; see Figure 15. Here a novel input image is labeled using the dataset shown in Table 1. The aforementioned procedure is used to retrieve two best matching images, transfer labels, and evaluate confidences for transferred labels. The labels and confidences are fused into labeling probabilities, which form the data term. Together with a color distance based smooth cost [Blake et al. 2004],



(a) input mesh (b) best matching (c) labeling result

Figure 14: *Imperfect labeling of a two-seat bicycle. The best matched image in the database (b) has a different topology as the input (a), resulting in incorrect labeling results (c).*

Table 3: *Confusion matrix between different shape categories.*

	Stroller	Bike	Chair	Pavilion	Table	Truck	Vase	Lamp
Stroller	**0.74**	0.01	0.13	0.02	0.02	0.04	0.03	0.01
Bike	0.00	**0.99**	0.00	0.00	0.01	0.00	0.00	0.00
Chair	0.09	0.00	**0.74**	0.03	0.03	0.03	0.04	0.04
Pavilion	0.01	0.02	0.02	**0.84**	0.08	0.02	0.01	0.00
Table	0.00	0.00	0.01	0.01	**0.93**	0.04	0.01	0.00
Truck	0.00	0.00	0.00	0.20	0.05	**0.90**	0.03	0.00
Vase	0.01	0.00	0.02	0.04	0.00	0.01	**0.86**	0.06
Lamp	0.02	0.01	0.04	0.00	0.02	0.02	0.08	**0.81**

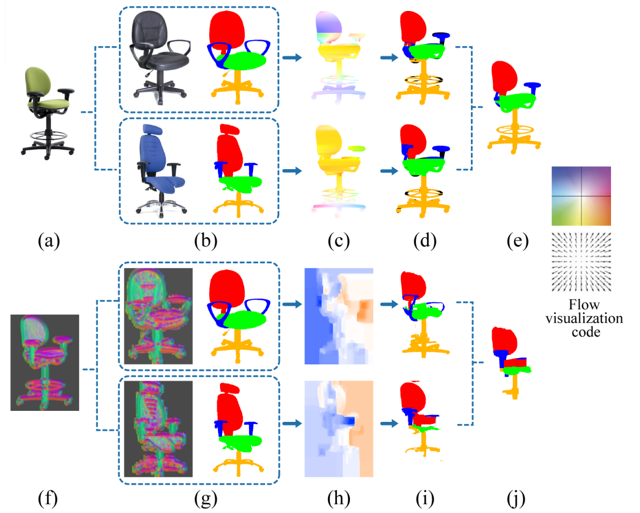


Figure 15: *Labeling of 2D images. Given an input image (a), our approach first retrieves two best matched shapes (b). The dense correspondence between the input and each retrieved image is shown in (c), based on which the labels are transferred (d). The final result (e) is computed using graph cuts. For fair comparison, the label transfer result of Liu et al. [2011a] using the same two images is shown. Their approach first computes dense SIFT for the input (f) and labeled images (g). SIFT flow (h) is then generated, which guides the label transfer (i). Final result (j) is generated using the authors' implementation with default parameters. Dense correspondences in (c,h) are shown using the color scheme in (e).*

we compute optimal labels using graph cuts.

As shown in Figure 15(e), although the final result is imperfect, it is useful for assisting users in creating more labeled images. In comparison, Liu et al.'s approach [2011a] cannot transfer labels at part level effectively since it is designed for labeling objects within the whole image and does not respect the boundary of the 2D shape; see Figure 15(j). Results on another test image are provided in Figure 19 of the supplementary material. It is worth noting that we also tried to use SIFT flow to transfer labels from images to projections, but got very poor results since no SIFT feature can be computed for the textureless interior of 2D projections.

7 Discussion, limitations, and future work

We present a shape analysis algorithm based on the projective approach and demonstrate its strong potential in analyzing 3D shapes. Generally, a 3D shape is treated as a set of projected images, allowing the major analysis task to be performed in the projective space, that is, over a series of 2D images. Then, the partial analysis results are back-projected and fused on the original 3D shape.

A key advantage of the projective approach is that it does not place strong requirements on the quality of the input 3D model, allowing the handling of non-manifold, incomplete, or self-intersecting shapes. Another advantage is that it builds on the rich availability and ease of processing of photos, compared to their 3D counterpart. A key disadvantage is that projections generally do not fully reproduce the 3D shape, e.g., due to concavities. Another disadvantage, at least in the way we realized our approach, is that we compare the labeled and unlabeled image in the spatial domain and not in feature space, which has been shown to be more effective [Kalogerakis et al. 2010; Sidi et al. 2011]. We compensate for it by assuming that the upright orientation is given. We argue that this is not a strong assumption as the upright orientation of shapes in 3D and certainly in photos is known in the vast majority of cases.

An intriguing aspect of projective analysis is that it may allow to transfer labels between shapes that differ significantly in 3D when only their projections are considered. For example, as shown in Figure 1, the labeling of the twin stroller is able to transfer labels from the images of single strollers. While the 3D shapes differ, their projections are more similar. This demonstrates that providing only partial information via projections is not entirely a lost cause. On the other hand, Figure 12 showcases several examples where the labeled images are geometrically quite dissimilar to the input projections. The success of our method in these cases can be attributed to the BiSH distance and warp alignment. They are topology- and feature-aware, while robust to geometric variations that do not alter the topology or feature characteristics of the images.

The technique that we present is inherently supervised since the key idea is to transfer labels from images. However, it is still interesting to consider an unsupervised version, e.g., in a co-analysis setting, where a set of shapes are analyzed together. The projections are co-analyzed and the results are back-projected to the original shapes. This however does not take advantage of the rich availability of photos. Another avenue to consider is to analyze the given shape at the *part* level, where we apply the projective approach on each part of the shape separately. Often, the parts are given but unlabeled and their positions in a particular example might mislead the analysis which considers the whole shape globally.

Currently, we treat the projections as binary images. Additional cues derivable from projections of a 3D shape, e.g., depth, colors, or normal information, may potentially boost performance. However, our initial experimentation indicates that it is not as straightforward as one might expect. We plan to investigate further along this direction. In addition, currently the labeled images are unorganized — all images in the database are searched for a retrieval. To allow more scalability, we plan to organize the images in a hierarchical data structure to accelerate the search.

Last and not least, we believe there is more potential for projective 3D shape analysis. Perhaps the strongest one lies in automatic extraction and processing of usable images from online image search results so that they can be directly used to form large labeled sets for analyzing 3D shapes. We would also like to explore other potential applications under the projective analysis framework, for example, transferring colors or textures from photos to 3D shapes.

Acknowledgements

The authors would like to thank all the reviewers for their valuable comments. This work is supported in part by grants from NSFC (61202222, 61232011, 61025012, 61202223), 863 Program (2013AA01A604), Guangdong Science and Technology Program (2011B050200007), Shenzhen Science and Innovation Program (CXB201104220029A, ZD201111080115A, KC2012JSJS0019A), Natural Science and Engineering Research Council of Canada (293127 and 611370) and the Israel Science Foundation.

References

BADDELEY, A. J. 1992. An error metric for binary images. In *IEEE Workshop on Robust Computer Vision*, 59–78.

BAI, X., YANG, X., LATECKI, L. J., LIU, W., AND TU, Z. 2010. Learning context-sensitive shape similarity by graph transduction. *PAMI 32*, 5, 861–874.

BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape context. *PAMI 24*, 4, 509–522.

BERG, A. C., BERG, T. L., AND MALIK, J. 2005. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 26–33.

BERNDT, D., AND CLIFFORD, J. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, 359–370.

BLAKE, A., ROTHER, C., BROWN, M., PEREZ, P., AND TORR, P. 2004. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, 428–441.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *PAMI 23*, 11, 1222–1239.

CHANG, J. Y., RASKAR, R., AND AGRAWAL, A. 2009. 3D pose estimation and segmentation using specular cues. In *CVPR*, 1706–1713.

CHEN, D., TIAN, X., SHEN, Y., AND OUHYOUNG, M. 2003. On visual similarity based 3D model retrieval. *CGF (EUROGRAPHICS) 22*, 3, 223–232.

CHEN, X., GOLOVINSKIY, A., , AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. Graph. (SIGGRAPH) 28*, 3, 1–12.

CHEN, X., LI, Q., SONG, Y., JIN, X., AND ZHAO, Q. 2012. Supervised geodesic propagation for semantic label transfer. In *ECCV*, 553–565.

CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J., AND BRAY, C. 2004. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, vol. 1, 22.

CYR, C. M., AND KIMIA, B. B. 2001. 3D object recognition using shape similiarity-based aspect graph. In *ICCV*, 254–261.

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

DUCHENNE, O., JOULIN, A., AND PONCE, J. 2011. A graph-matching kernel for object categorization. In *ICCV*, 1792–1799.

EITZ, M., RICHTER, R., BOUBEKEUR, T., HILDEBRAND, K., AND ALEXA, M. 2012. Sketch-based shape retrieval. *ACM Trans. Graph. (SIGGRAPH) 31*, 4, 31–40.

FERRARI, V., TUYTELAARS, T., AND GOOL, L. V. 2004. Integrating multiple model views for object recognition. In *CVPR*, 105–112.

FLEMING, R. W., AND SINGH, M. 2009. Visual perception of 3D shape. In *ACM SIGGRAPH 2009 Courses*, 24:1–24:94.

GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. Consistent segmentation of 3D models. *Computers & Graphics (SMI) 33*, 3, 262–269.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph. (SIGGRAPH) 29*, 3, 1–11.

KLEIN, P. N., SEBASTIAN, T. B., AND KIMIA, B. B. 2001. Shape matching using edit-distance: an implementation. In *SODA*, 781–790.

KOENDERINK, J. J. 1984. What does the occluding contour tell us about solid shape. *Perception 13*, 321–330.

LAI, K., BO, L., REN, X., AND FOX, D. 2012. Detection-based object labeling in 3D scenes. In *ICRA*, 1330–1337.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *PAMI 16*, 2, 150–162.

LEORDEANU, M., AND HEBERT, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 1482–1489.

LI, Y., ZHENG, Q., SHARF, A., COHEN-OR, D., CHEN, B., AND MITRA, N. 2011. 2D-3D fusion for layer decomposition of urban facades. In *ICCV*, 882–889.

LINDSTROM, P., AND TURK, G. 2000. Image-driven simplification. *ACM Trans. Graph. 19*, 3, 204–241.

LING, H., AND JACOBS, D. 2007. Shape classification using the inner-distance. *PAMI 29*, 2, 286–299.

LIU, R., AND ZHANG, H. 2007. Mesh segmentation via spectral embedding and contour analysis. *CGF (EUROGRAPHICS) 26*, 3, 385–394.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. Nonparametric scene parsing via label transfer. *PAMI 33*, 12, 2368–2382.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. SIFT flow: Dense correspondence across scenes and its applications. *PAMI 33*, 5, 978–994.

MAKADIA, A., AND DANIILIDIS, K. 2010. Spherical correlation of visual representations for 3D model retrieval. *IJCV 89*, 2-3, 193–210.

MURASE, H., AND NAYAR, S. K. 1995. Visual learning and recognition of 3-d objects from appearance. *IJCV 14*, 1, 5–24.

OLIVA, A., AND TORRALBA, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV 42*, 3, 145–175.

PEPIK, B., GEHLER, P., STARK, M., AND SCHIELE, B. 2012. $3D^2$pm–3D deformable part models. In *ECCV*, 356–370.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (SIGGRAPH) 23*, 3, 309–314.

SHAMIR, A. 2008. A survey on mesh segmentation techniques. *CGF 27*, 6, 1539–1556.

SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 1–15.

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

TELEA, A., AND VAN WIJK, J. J. 1999. Simplified representation of vector fields. In *IEEE Visualization*, 35–507.

VAN KAICK, O., TAGLIASACCHI, A., SIDI, O., ZHANG, H., COHEN-OR, D., WOLF, L., AND HAMARNEH, G. 2011. Prior knowledge for part correspondence. *CGF (EUROGRAPHICS) 30*, 2, 553–562.

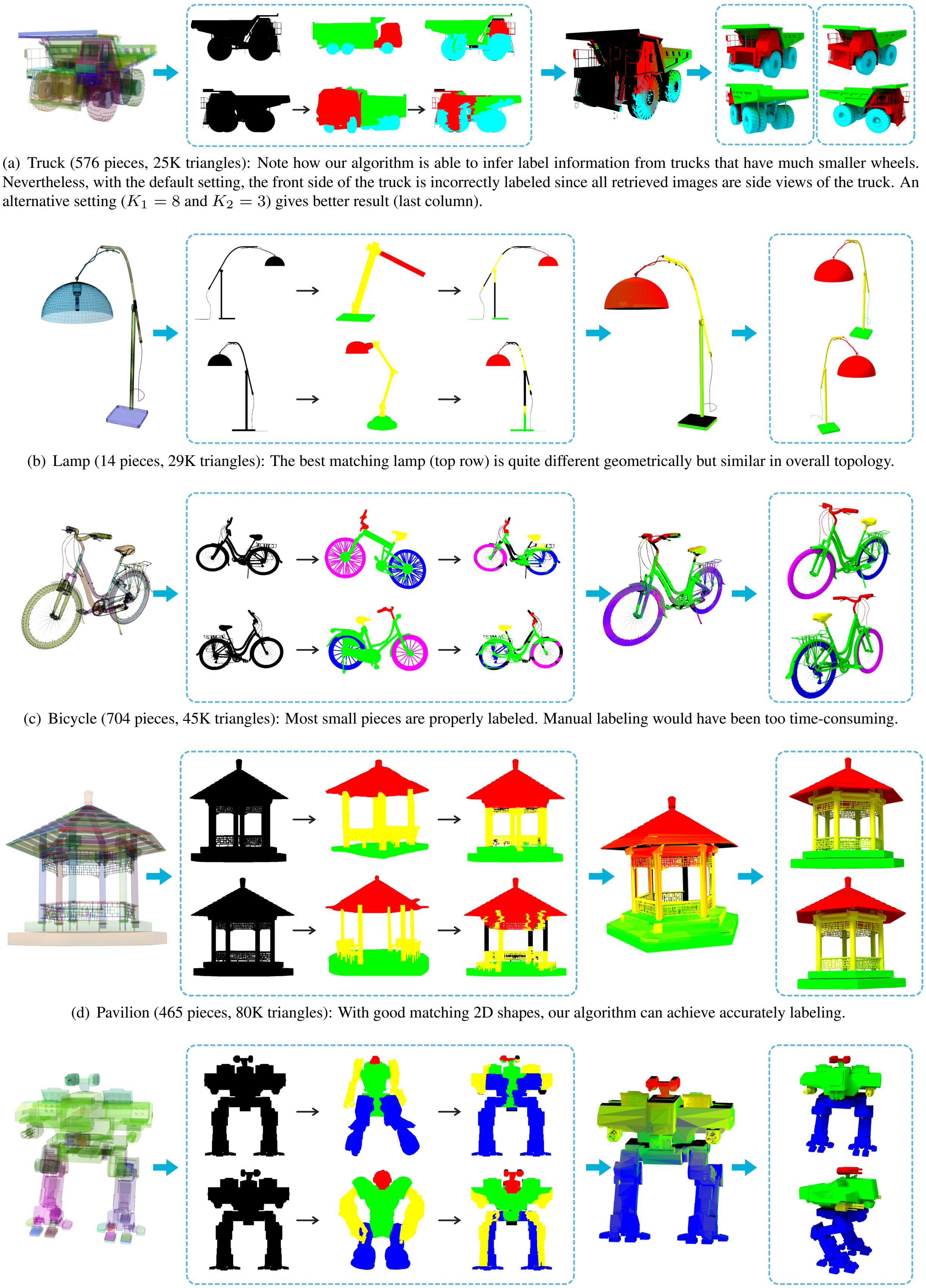
WANG, Y., ASAFI, S., VAN KAICK, O., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Active co-analysis of a set of shapes. *ACM Trans. Graph. (SIGGRAPH Asia) 31*, 6, 165–174.

XIAO, J., HAYS, J., EHINGER, K. A., OLIVA, A., AND TORRALBA, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492.

XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z. 2010. Style-content separation by anisotropic part scales. *ACM Trans. Graph. (SIGGRAPH Asia) 29*, 5.

XU, K., ZHENG, H., ZHANG, H., COHEN-OR, D., LIU, L., AND XIONG, Y. 2011. Photo-inspired model-driven 3D object modeling. *ACM Trans. Graph. (SIGGRAPH) 30*, 4, 80:1–80:10.

ZHANG, H., XIAO, J., AND QUAN, L. 2010. Supervised label transfer for semantic segmentation of street scenes. In *ECCV*, 561–574.



(a) Truck (576 pieces, 25K triangles): Note how our algorithm is able to infer label information from trucks that have much smaller wheels. Nevertheless, with the default setting, the front side of the truck is incorrectly labeled since all retrieved images are side views of the truck. An alternative setting ($K_1 = 8$ and $K_2 = 3$) gives better result (last column).

(b) Lamp (14 pieces, 29K triangles): The best matching lamp (top row) is quite different geometrically but similar in overall topology.

(c) Bicycle (704 pieces, 45K triangles): Most small pieces are properly labeled. Manual labeling would have been too time-consuming.

(d) Pavilion (465 pieces, 80K triangles): With good matching 2D shapes, our algorithm can achieve accurately labeling.

(e) Robot (1248 pieces, 17K triangles): Although designed for rigid objects, the algorithm can handle objects with limited articulation as well.

Figure 12: *Labeling results on various imperfect meshes downloaded from Trimble Warehouse. In each row, from left to right, we show the input shape, the top two ranked images and label transfer results, probability map, and the final labeling under two different views.*