Estimating Image Depth Using Shape Collections

Hao Su Qixing Huang Niloy J. Mitra Yangyan Li Leonidas Guibas
Stanford University University College London Stanford University

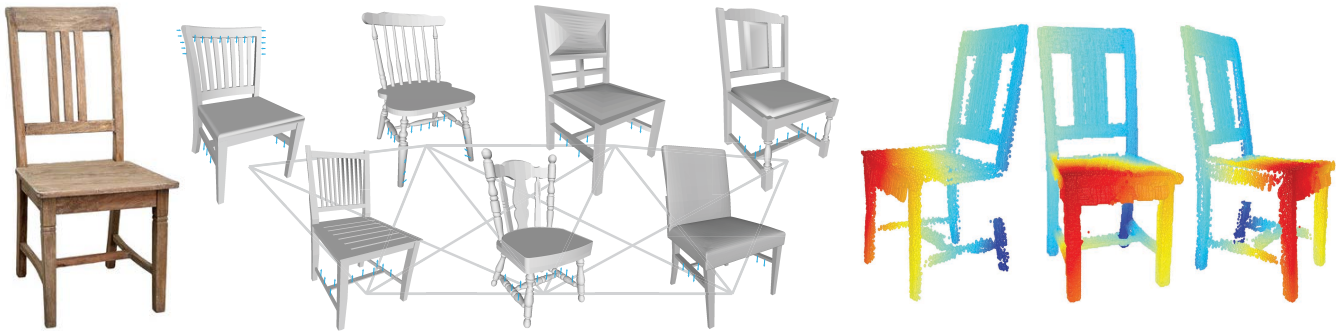

Figure 1: *We attribute a single 2D image of an object (left) with depth by transporting information from a 3D shape deformation subspace learned by analyzing a network of related but different shapes (middle). For visualization, we color code the estimated depth with values increasing from red to blue (right).*

Abstract

Images, while easy to acquire, view, publish, and share, they lack critical depth information. This poses a serious bottleneck for many image manipulation, editing, and retrieval tasks. In this paper we consider the problem of adding depth to an image of an object, effectively 'lifting' it back to 3D, by exploiting a collection of aligned 3D models of related objects shape. Our key insight is that, even when the imaged object is not contained in the shape collection, the network of shapes implicitly characterizes a shape-specific deformation subspace that regularizes the problem and enables robust diffusion of depth information from the shape collection to the input image. We evaluate our fully automatic approach on diverse and challenging input images, validate the results against Kinect depth readings, and demonstrate several imaging applications including depth-enhanced image editing and image relighting.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms.

Keywords: data-driven shape analysis, pose estimation, depth estimation, image retrieval, shape collections

Links: DL PDF WEB VIDEO DATA

1 Introduction

Images remain by far the most popular visual medium. Nowadays they are easy to acquire and distribute, contain rich visual detail, can easily be viewed and understood and, as a result, are ubiquitous in the Web. As 2D projections of our 3D world, however, they may lack certain semantical information. For example, important parts of objects may be occluded, and depth data is typically missing. This poses serious challenges to applications involving image recognition, manipulation, editing, etc. that could greatly benefit from this omitted information. Hence, there is a strong motivation to *lift* images to 3D by inferring attributes lost in the projection. In this paper we are specifically interested in inferring depth for the visible object areas — the key coordinate missing in the projection.

As the problem of recovering depth from single image is naturally ill-posed, various priors have been proposed for regularization. The most common and classical approach is to match the input image to a set of 3D objects in a database (i.e., priors), and use the best matching shape to fill in missing depth information. However, large-scale deployment of such a method is fundamentally limited because only a limited number of 3D models is available. Most often, we do not even have a 3D model of the same or sufficiently similar object from which the image was taken.

In this paper we consider the problem of estimating depth for an image of an object by exploiting, in a novel joint fashion, a *collection* of 3D models of related but largely different objects (see Figure 1). Key to our approach is the estimation of correspondences between the image and multiple models, with the help of correspondences estimated between the models themselves. We address the depth inference problem in its purest form, where we assume that the object image has been segmented from its background (such images are now commonplace in shopping web sites), while our 3D models are typically untextured and come from shape collections, such as the Trimble 3D warehouse.

Our image-based but shape-driven modeling technique is fully automatic and reconstructs a 3D point cloud from the imaged object. The algorithm consists of a *preprocessing* stage, which aligns the input shapes to each other and learns a deformation model for each shape; and a *reconstruction* stage, which uses a continuous optimization to recover the image object pose and reconstruct a point cloud from the image that aligns with relevant 3D models extracted from the collection. We show how to formulate an appropriate objective function, how to obtain an initial solution, and how to effectively refine the solution using an alternating optimization.

In our approach, we jointly match the depth-augmented image, i.e., the *popup* point cloud of the image, with a group of related shapes

in the collection. We pose the task as a joint non-rigid registration problem, in which each shape can be deformed. The formulation has two key features. First, in contrast to utilizing a single similar shape, incorporating a collection of similar shapes offers a better coverage of the relevant neighborhood of shape space. Second, since we have already aligned the 3D models to each other, it enables us to apply consistency constraints [Kim et al. 2012a; Huang and Guibas 2013] to regularize the image to 3D model matching by using the shape-shape correspondences.

In the joint non-rigid registration formulation, we introduce the key concept of *deformation priors*, which govern the deformation of each shape (c.f., [Averkiou et al. 2014]). Intuitively, we aim to preserve the key structural properties of each shape in the deformation, so that round shapes stay round, left-to-right symmetries are preserved, etc. In particular, instead of detecting these properties form each shape alone, which turns out to be unreliable, we learn them from the optimal deformations of each shape to other shapes.

To test the performance of the proposed approach, we have created a benchmark dataset consisting of Microsoft Kinect scans of various categories of objects including chairs, tables, lamps, and cups. Experimental results show that the proposed approach recovers depth information that is close to Kinect scans, and is significantly more accurate than state-of-the-art image-based modeling techniques. Moreover, the proposed approach is robust to variations in textures and lighting conditions.

We demonstrate depth-enhanced image editing to illustrate the possibilities offered by our approach. In addition, we show that our work is a key intermediate step towards the goal of obtaining full 3D models. Using a popup point cloud as input, we can reconstruct in certain cases a full mesh by exploiting shape symmetries learned from the shape network.

Contributions. We present the first, to the best of our knowledge, fully automatic method to utilize a network of related but different 3D objects in order to reconstruct depth information from a single imaged object. The key novelties are:

- showing how a single modestly-sized shape network can help infer depth information for a variety of image objects of the same class;
- using learned deformation models based on an aligned shape network to compensate for the fact that the image is not from a model directly present in the database;
- regularizing model deformations using multi-way 3D alignment between the initial image point cloud and the shapes in a neighborhood of the shape network;

In the process of extracting depth information on an image, we also discover good correspondences between the image and the network shapes, enabling us to connect the image to the network and transfer complementary information back and forth. Example of such information transfer can include textures, segmentations, material properties, labels, etc.

2 Related Work

Data-driven geometry processing. The emergence of large shape collections provides us with a platform to aggregate information from multiple shapes to improve the analysis and processing of individual shapes. Already the Trimble 3D warehouse contains many thousands of example models per category for most indoor objects and some popular outdoor categories such as car and airplane. Recently, we have witnessed the success of data-driven techniques in shape analysis [Huang et al. 2011; Kim et al. 2012a; Kim et al. 2013; Huang et al. 2013; Wang et al. 2013], shape modeling [Chaudhuri et al. 2011; Kalogerakis et al. 2012; Averkiou et al. 2014] and shape reconstruction [Nan et al. 2012; Kim et al. 2012b; Shen et al. 2012]. The key task in data-driven geometry processing technique is to establish high-quality correspondences (at either point- or segment-level) across geometric objects. Although there exist rich techniques for aligning and matching 3D shapes, the problem of matching image objects and 3D shapes, which is the major focus of this paper, is far from being solved.

Image-shape matching. Most existing image-shape matching approaches [Cyr and Kimia 2004; Xu et al. 2011; Wang et al. 2013] convert the problem into an image matching problem, i.e., matching images with projected views of 3D shapes. They typically start from estimating dense correspondences between silhouette curves, and then interpolate correspondences to interior pixels. [Sun et al. 2011] used an ICP-like approach. Recently, Wang et al. [2013] proposed a technique that directly estimates correspondences between entire image objects. The major limitation of these approaches is that a projected view of a 3D shape only contains partial information from the original shape. In practice, these techniques are limited to matching very similar objects. In contrast, we formulate the image shape matching problem as solving a non-rigid alignment problem in 3D, i.e., simultaneously estimating optimizing the depth of the image object and the deformations of 3D shapes to align them in the 3D space. In particular, we show that matching an image with a collection of 3D shapes boosts the matching quality by enforcing consistency between image-shape maps and shape-shape maps.

Pose estimation. There exists a vast body of work to determine the pose of an object in an image relative to a calibrated camera. The problem is commonly formulated as a feature correspondence problem. Thus, they can be distinguished by the type of local image features, such as points, lines, curve segments, whole contours [Chen et al. 2003; Dalal and Triggs 2005; Oliva and Torralba 2006]. Recently, researchers used learning-based scheme to cast it as a classification and learn good features for viewpoint estimation [Zia et al. 2013]. The task of pose estimation is closely coupled with other tasks in the image-shape matching problem, such as depth estimation and point correspondence. We therefore model it as a part of the global optimization problem and iteratively refine it, resulting in large improvement.

Depth estimation. Estimating the depth of an image object is a long standing problem in computer vision and computer graphics. This problem is ill-posed when the input is a single image, and existing approaches typically incorporate additional information such as user interaction [Wu et al. 2008] and shading [Lensch et al. 2003; Goldman et al. 2005], or using abstracted proxy shapes [Zheng et al. 2012]. However, these approaches are designed for objects with simple textures and shapes and/or under specific lighting conditions. In other words, they do not apply well on man-made objects in real images, which exhibit complicated geometries and textures.

With the availability of large collection of depth images, recent depth estimation approaches are based on supervised learning [Hoiem et al. 2005; Saxena et al. 2009]. Given exemplar depth images, these approaches learn conditional probabilistic distributions of pixel depths and relative depths between neighboring pixels, and apply the learned distributions to infer the depth information of new images. We take a different approach. Since obtaining 3D shapes that are similar in global structure to image objects is easy, we estimate depth information in an unsupervised manner, i.e., by directly matching images with 3D shapes, and thus avoiding the tedious task of performing instance specific learning.

3 Pipeline Overview

The proposed image-based but shape-driven modeling approach takes a single image object I segmented from background and a collection of shapes $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$ of the same class as

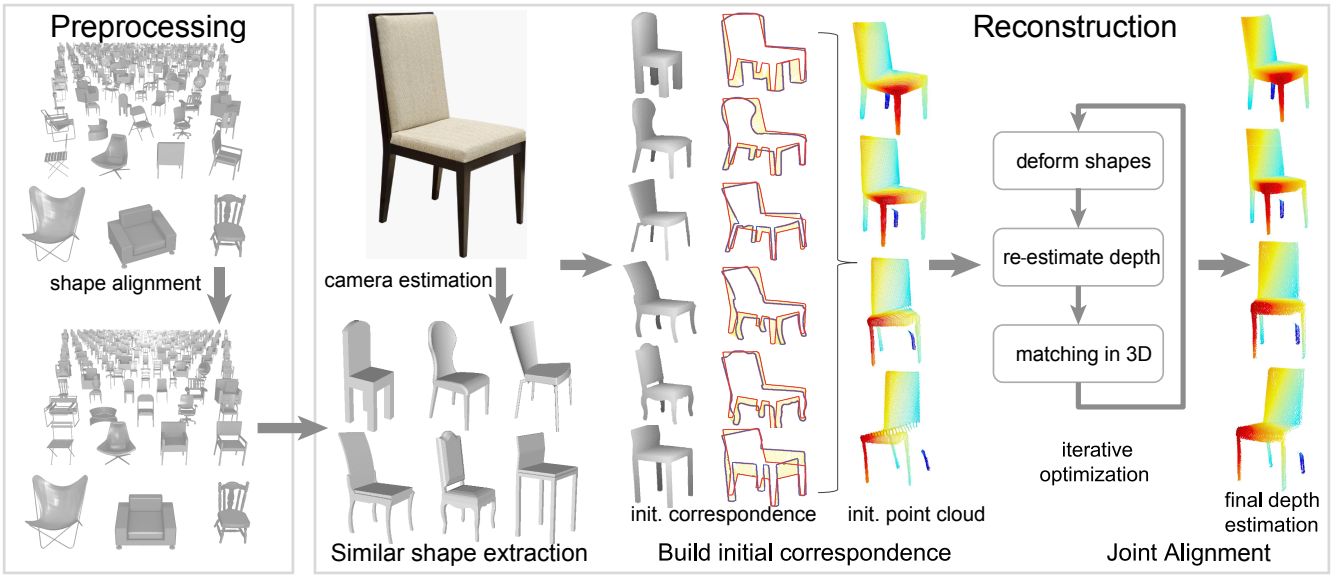

Figure 2: ***Algorithm Pipeline.*** *We reconstruct a 3D point cloud from an image object by utilizing a collection of related shapes. In the preprocessing stage, we jointly align the input shape collection and learn structure-preserving deformation models for the shapes. Then, in the reconstruction stage, we lift a single image to 3D in three steps. The first step initializes the camera pose in the coordinate system associated with the aligned shapes and extracts a set of similar shapes. The second step performs image-image matching to build dense correspondences between the image object and the similar shapes, and generate an initial 3D point cloud. The final step jointly aligns the initial point cloud and the selected shapes by simultaneously optimizing the depth of each image pixel, the camera pose, and the shape deformations.*

input, and simultaneously estimates the object pose shown in I and reconstructs a 3D point cloud P from I. For simplicity, we assume that all input shapes are supported by the same ground plane [Huang et al. 2013], so a common vertical direction is available.

As the shape collection typically does not contain a shape that is exactly same as the object to be reconstructed, we formulate the task as a joint non-rigid alignment problem. The variables to be optimized are the point cloud, parameterized by the camera pose and the z-coordinates (pixel depths) of the image object, and deformations of a set of similar shapes. The objective function minimizes the distance between the point cloud and the deformed shapes. However, there are several challenges. First, the depth coordinates of the point cloud are unconstrained yet we cannot allow the shapes to be deformed arbitrarily, since otherwise both the point cloud and the shapes may be stretched undesirably when being aligned. Second, the success of the non-rigid alignment depends on a good initialization for both the camera pose and the point cloud. Third, even with good initialization, it is challenging to solve the induced optimization problem involving the depth of each pixel effectively. What helps in our situation, and the fundamental difference between the proposed approach and other shape-driven image based modeling techniques, is that we utilize the information provided by the collection to regularize the problem.

As illustrated in Figure 2, the pipeline consists of a *preprocessing* stage and a *reconstruction* stage. The goal of the preprocessing stage is to align the shapes and to learn a smart deformation prior (local model) for each shape. The motivation comes from the fact that plausible deformations of each shape typically lie in a low-dimensional space, when compared with the number of parameters in a general deformation model [Averkiou et al. 2014]. We learn the deformation prior of each shape by performing covariance analysis over its optimized deformations to neighboring shapes. As the deformation prior is directly learned by shapes, it inherits several structure-preserving properties (e.g., symmetry, part structure) from the shape collection. Essentially, we learn the local structure of the shape space.

The reconstruction stage proceeds in three steps, where the first two steps provide an initial solution (a set of similar shapes and an initial point cloud) and the third step optimizes this point cloud to minimize its distance to the deformed similar shapes. Specifically, the first step initializes a camera configuration and extracts a set of similar shapes. This is considered as a pose estimation problem. Although pose estimation using a single shape is hard, we found that when a collection of oriented shapes are available, a simple cumulative score, which sums the weighted similarity scores of the rendered images to the input image, works remarkably well. This can be understood by the fact that the input shapes are aligned, and the best camera pose is voted on by all relevant input shapes together, then the pose tends to be much more stable than those generated from individual shapes. In the same spirit, when generating similar shapes, we combine both the image-shape distances and shape-shape distances to generate a more robust set of similar shapes for later steps of the pipeline.

Given the rendered images of similar aligned shapes, the second step proceeds to initialize the depth information (z-coordinates) by building dense correspondences between the image object and similar shapes and transferring the depth information. Due to the differences between the input image and 3D shapes, we observed that it is extremely hard to obtain reliable correspondences via pair-wise image-shape matching. However, as the input shapes are aligned, we exploit the consistency of correspondences across the set of similar shapes, so that we can obtain much more reliable depth information. This is conceptually similar to state-of-the-art techniques in data-driven shape matching techniques [Kim et al. 2012a; Huang and Guibas 2013] to enforce consistency of correspondences along cycles to improve quality of isolated correspondences.

Finally, in the third step we refine the camera pose and depth information using non-rigid registration formulated as solving a continuous optimization problem. The objective function combines a distance term, which evaluates the distance between the corresponding points on the induced point cloud and the deformed similar shapes, and two prior terms on the deformation models and the depth in-

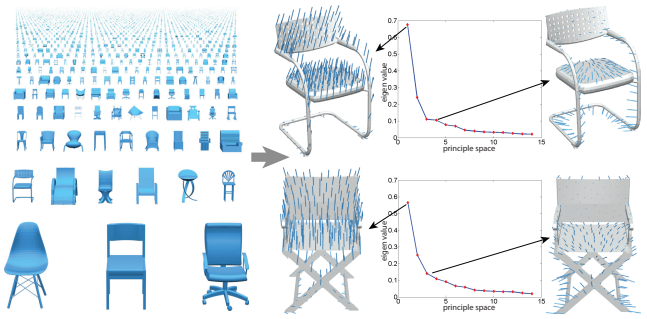

Figure 3: ***Preprocessing Stage.*** *We learn a deformation model of each shape via its optimized deformations to other shapes. Each deformation model is characterized by a small set of typical deformation fields (shown as vectors on model surfaces) derived from covariance analysis. This model serves as the regularizer for the local shape space around each shape and is enforced during the reconstruction stage.*

formation, respectively. Despite the non-linearity and scale of this optimization problem, we show that it can be optimized effectively using an alternating optimization strategy.

4 Preprocessing Stage

The goal of the preprocessing stage is to understand the plausible deformations of each shape in the context provided by the input shape collection. We achieve this goal by aligning all input shapes and then learning a deformation prior for each shape.

Deformation model and joint shape alignment. We use the embedded deformation model [Sumner et al. 2007] to parameterize the deformation of each shape. An embedded deformation consists of a list of control points $\mathbf{p}_\alpha \in J$ and the associated basis functions $B_\alpha(\cdot)$. Given a point $\mathbf{x} \in \mathbb{R}^3$, its deformed counterpart is a linear combination of the control points:

$$\mathcal{D}(\mathbf{x}) = \sum_{\mathbf{p}_\alpha \in J} B_\alpha(\mathbf{x})\mathbf{p}_\alpha. \tag{1}$$

Refer to [Sumner et al. 2007] on how to construct embedded deformation models on shapes. In this paper, we use 200 control points, and hence each shape is controlled by $M = 600$ parameters.

To align the input shapes, we employ the method described in [Huang et al. 2013], which jointly optimizes the deformations of all input shapes to minimize the sum of distances between corresponding points computed using pair-wise alignment. We denote the optimized embedded deformation of shape S_i by $\mathcal{D}_i^\star$.

Deformation-prior learning. We assume that plausible deformations of each shape (parameterized by a vector that collects all control points) lie in a low dimensional space defined by the shape's neighborhood (c.f., [Ovsjanikov et al. 2011]). We learn this space from the optimal deformations of each shape S_i to other shapes, which provide samples of plausible deformations (see Figure 3).

We directly obtain these deformation samples by composing the absolute optimal deformations $\mathcal{D}_i^*$ and their inverse deformations $\mathcal{D}_i^{*-1}$. For each shape S_i and each neighboring shape S_j, we transform the original control point $\mathbf{p}_\alpha$ (i.e., in the rest state) of $\mathcal{D}_i^\star$ to $(\mathcal{D}_j^{\star-1} \circ \mathcal{D}_i^\star)(\mathbf{p}_\alpha)$. Let $\mathbf{c}_{i,j}$ be the vector that collects all transformed control points. Let $\mathbf{c}_i^0$ be the original control points. Then each neighboring shape S_j gives rise to a deformation sample $\mathbf{c}_{i,j} - \mathbf{c}_i^0$. To learn the prior model from similar shapes, we only consider the deformation samples from the 128 most similar shapes to each shape, in terms of the D2 descriptor [Osada et al. 2002].

Given the deformation samples, we perform covariance analysis to extract the principal values $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_M$ and principal directions $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_M$ of the deformation space. The prior model on the deformation of each shape is given by

$$Prior(\mathcal{D}_i) = -\sum_{j=1}^{M} \frac{\sigma_1}{\sigma_j + \epsilon} \left((\mathbf{c}_i - \mathbf{c}_i^0)^T \mathbf{u}_j \right)^2, \tag{2}$$

Intuitively, a deformation leads to a small objective value if it follows the majority of the deformation samples. Note that the shift ϵ is introduced to make the prior term well-defined as we only have limited deformation samples and many principal values are zero.

5 Reconstruction Stage

The reconstruction stage solves a joint optimization to recover the geometry of an image object that aligns with the deformed versions of a set of similar shapes. We begin by introducing the camera model. Then we show how to initialize an approximate solution in Section 5.2 and Section 5.3, and refine it to obtain the final solution in Section 5.4.

5.1 Camera Configuration

We use a simplified nine parameter camera configuration $C = (R, \mathbf{t}, z_f, s_x, s_y)$. Here $(R, \mathbf{t})$ specifies the rigid motion from the common coordinate system Σ associated with the input shapes to the camera coordinate system Σ^C; z_f specifies the focal length; s_x and s_y specify the effective size of the pixels in the horizontal and vertical directions. Given a point $\mathbf{q} = (q_x, q_y, q_z)^T$ in Σ, its corresponding pixel coordinate $\mathbf{p} = (p_x, p_y)^T$ is given by

$$p_x = \frac{q'_x z_f}{s_x q'_z}, \quad p_y = \frac{q'_y z_f}{s_y q'_z}, \qquad \mathbf{q}' = R^T(\mathbf{q} - \mathbf{t}). \tag{3}$$

In the other direction, given a pixel $\mathbf{p} = (p_x, p_y)^T$ and a depth parameter z_p specifying its z coordinate in Σ^C, the corresponding point in Σ is given by

$$\mathbf{q} = R\mathbf{p}' + \mathbf{t}, \quad \mathbf{p}' = \left(\frac{s_x p_x z_p}{z_f}, \frac{s_y p_y z_p}{z_f}, z_p \right)^T. \tag{4}$$

For convenience, we denote the map from $\mathbf{p}, z_p$ to $\mathbf{q}$ as $C(\mathbf{p}, z_p)$.

5.2 Step I: Camera Initialization

Candidate generation. Our candidate camera pose sampling strategy is similar to most pose estimation algorithms [Zia et al. 2013], which sample the viewing direction and fix the rest of the parameters to default values. Specifically, we let the camera position move on a viewing sphere centered at the origin with radius $5d$, where d is the averaged shape diameter. The rest of the parameters are fixed as follows. The focal point $\mathbf{t}$ is placed at the origin. To fix R, we let the up-right direction of the camera system lie in the plane of the viewing direction and the z axis. Finally, we set $z_p = 3d$, and set s_x, s_y so that, on the average, each object occupies half of the rendered image. We generate candidate camera configurations for each shape by uniformly sampling 500 viewing directions on the viewing sphere. Let $\mathcal{C}_{\text{cand}}$ collect all candidate camera configurations. For each $C \in \mathcal{C}_{\text{cand}}$, we denote I_i^C as the rendered image of shape S_i cropped using a tight bounding box surrounding object.

Optimal candidate. When picking the optimal candidate, we follow the common strategy of evaluating rendered images by comparing them with the input image. Due to differences between real images and rendered images, standard single shape based approaches

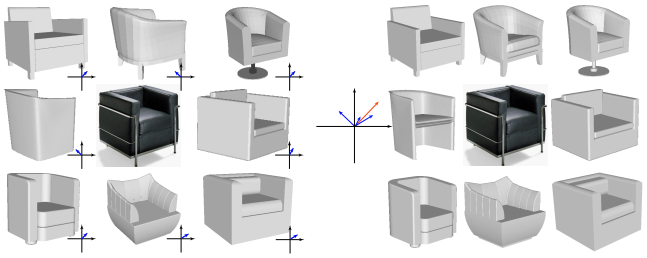

Figure 4: ***Camera Initialization by Voting.*** *Left: given an image object (the sofa in the center), we can find multiple similar shapes, each of which independently proposes a camera pose candidate (the blue arrow for each candidate). Note that some candidates are far from optimal. Right: since shapes are already jointly oriented in the pre-processing step, they can be used to more accurately vote for the optimal pose (the red arrow).*

typically require feature learning. However, we found that when a collection of aligned shapes are present, a simple cumulative similarity score between the input image and the rendered images is sufficient (see Figure 4):

$$C^\star = \underset{C \in \mathcal{C}_{\text{cand}}}{\operatorname{argmin}} \sum_{i=1}^{N} \exp(-\|\mathbf{f}(I) - \mathbf{f}(I_i^C)\|^2 / 2\sigma^2), \qquad (5)$$

where $\mathbf{f}(\cdot)$ is a given feature descriptor, $\sigma = \min_{i \in \{1,\cdots,n\}, C \in \mathcal{C}^{\text{cand}}} \|\mathbf{f}(I) - \mathbf{f}(I_i^C)\|$, and the exponential operator is introduced to down-weight the contribution of images that are less similar to the input image. We have various image descriptors including GIST [Oliva and Torralba 2001], HOG [Dalal and Triggs 2005], and the light-field descriptor [Chen et al. 2003]. Experimentally, we found that the feature descriptor that combines all the three features together yields the best result.

5.3 Step II: Point Cloud Initialization

Given the initial camera configuration C, we generate an initial point cloud P from I. This is done by selecting a set of similar shapes to the input image, and then establishing dense correspondences between I and the similar shapes for transferring depth information. The performance of this step is crucial since it governs the global behavior of the final reconstruction. Although both image-based retrieval and image-image matching have been studied considerably in the past, we found that even state-of-the-art algorithms are insufficient for the purpose of transferring depth. Instead, the key idea of the proposed approach is to utilize the regularization provided by a collection of aligned shapes to boost the performance in each step: i) the similar shapes extracted from matching rendered images have to be similar with each other in the 3D space, and ii) pixels in the rendered images of different shapes corresponding to the same object image pixel should come from points close to each other in the 3D space where the aligned models live. Experimental results show that even with standard pair-wise techniques, the overall performance of joint matching approaches is sufficient for the purpose of depth initialization (see Figure 5).

Similar shape extraction. A naive approach to extract the similar shapes is to compare the input image with rendered images (according to the selected view) one-by-one. However, even with learned feature similarity metrics, such an approach is insufficient due to the diversity in lighting and texture of the input image. Since our input shapes are aligned, we use the distances between shapes to guide the selection of similar shapes.

Specifically, we first use the pairwise similarity score defined in (5) to extract $K_0 = 32$ similar shapes (i.e., an initial similar shape set). We then build a small weighted clique graph, which consists of the input image and the initial set of similar shapes, and use the diffusion distance [Coifman et al. 2005] to sort the initial similar shapes. The weight of each image-shape edge is given by (5), while the image descriptor is replaced by the D2 shape descriptor [Osada et al. 2002] for a shape-shape edge. Given the sorted shapes, we select the top $K = 6$ shapes as the final similar shape set. To simplify the notation let the similar shapes be denoted as $S_1, \cdots, S_K$.

Correspondence initialization. We initialize the image-shape correspondences by matching the input image object (background is removed) and the rendered image object I_i^C of each shape S_i. Given two image objects, we first apply [Munich and Perona 1999] to build dense correspondences between silhouette curves. Treating these correspondences as landmark correspondences, we then employ Laplacian deformation [Sorkine et al. 2004] to align I and I_i^C. After alignment, we derive the initial pixel-shape correspondences from the overlaid image objects. With $\mathcal{M}_i \subset I \times S_i$ we denote the initial correspondences from $I \to S_i$. Note that some pixels may not have correspondences due to 'holes' in the shapes.

Correspondence pruning. So far we only compute the image-shape correspondences between the input image and each shape in isolation. A constraint that we can use to improve these correspondence is to make them consistent with the optimal deformations $\{\mathcal{D}_i^\star\}$ that align the input shapes. More precisely, given two correspondences $(\mathbf{p}, \mathbf{q}_i) \in \mathcal{M}_i$ and $(\mathbf{p}, \mathbf{q}_j) \in \mathcal{M}_j$, if the distance between $\mathcal{D}_i^\star(\mathbf{q}_i)$ and $\mathcal{D}_j^\star(\mathbf{q}_j)$ is large, then at least one of these two correspondences is incorrect. In addition to enforcing this consistency property, we also prioritize the smoothness of correspondences, i.e., given two correspondences $(\mathbf{p}, \mathbf{q}_i), (\mathbf{p}', \mathbf{q}_i') \in \mathcal{M}_i$ where $\mathbf{p}$ and $\mathbf{p}'$ are neighbors and so should be $\mathbf{q}_i$ and $\mathbf{q}_i'$, we favor that either both of them are selected or both of them are pruned.

As both the consistency property and the smoothness prior only involve pairs of correspondences, we formulate the correspondence pruning step as solving a binary second-order MRF problem. We introduce a binary random variable $x_c \in \{0, 1\}$ for each initial correspondence $c \in \cup_{i=1}^K \mathcal{M}_i$, where $x_c = 1$ if c is selected and $x_c = 0$ otherwise. We then define the two types of pair-wise potential functions. For each correspondence pair $c_i = (\mathbf{p}, \mathbf{q}_i) \in \mathcal{M}_i$ and $c_j = (\mathbf{p}, \mathbf{q}_j) \in \mathcal{M}_j$, we define a consistency potential:

$$\phi(x_{c_i}, x_{c_j}) = \begin{cases} -\infty & x_{c_i} = x_{c_j} = 1, \\ & \|\mathcal{D}_i^\star(\mathbf{q}_i) - \mathcal{D}_j^\star(\mathbf{q}_j)\| \geq \delta \\ 1 & \text{otherwise,} \end{cases}$$

where δ is set as the 0.05 times the averaged shape diameter. For each pair of correspondences $c = (\mathbf{p}, \mathbf{q}_i) \in \mathcal{M}_i$ and $c' = (\mathbf{p}', \mathbf{q}_i') \in \mathcal{M}_i$ where $\mathbf{p}, \mathbf{p}'$ and $\mathbf{q}_i, \mathbf{q}_i'$ are two pairs of neighboring pixels, we define a smoothness potential function as

$$\phi(x_c, x_{c'}) = \begin{cases} 1 & x_c = x_{c'} \\ 0 & \text{otherwise.} \end{cases}$$

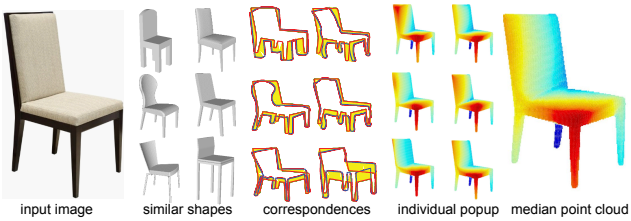

Figure 5: ***Point Cloud Initialization.*** *We start by performing image-image matching to obtain initial dense correspondences between the input image and rendered images of similar shapes. We then use the correspondences between shapes to prune away inconsistent initial correspondences. Finally, we use the rectified correspondences to transfer depth information from similar shapes to the initial point cloud.*

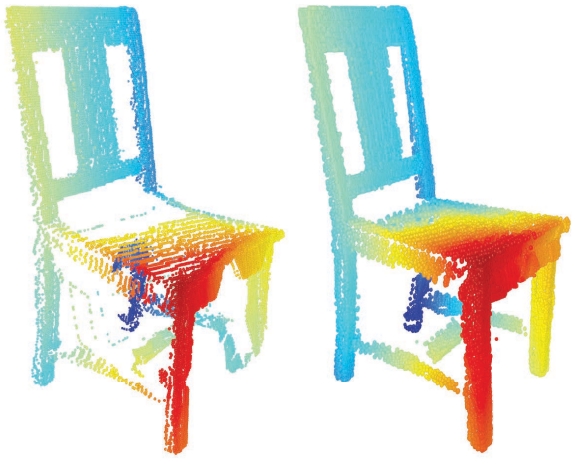

Figure 6: ***Effect of point registration in 2D versus in 3D.*** *We compare the reconstructed point clouds by different strategies of point registration between the input image and the rendered view. Left: registration is done by measuring distance in the image domain (2D). Right: registration is done by distance in 3D.*

Then the total potential function simply sums all pair-wise potential functions

$$f = \sum_{(c,c')\in\mathcal{P}} \phi(x_c, x_{c'}), \tag{6}$$

where $\mathcal{P}$ collects all pairs of correspondences of consideration.

For optimization, we apply tree-reweighted belief propagation (TRBP) [Szeliski et al. 2008], which is very effectively on binary MRF problems. For convenience, we still use $\mathcal{M}_i$ to denote the remaining correspondences between I and S_i after this stage.

Geometry initialization. Using the dense correspondences, we compute the z coordinate of the corresponding point of each pixel $\mathbf{p} = (p_x, p_y)$ (in the camera coordinate system of $C^\star$) by averaging z-coordinates of the corresponding points of similar shapes:

$$z_p = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{M}(\mathbf{p})} q'_z/|\mathcal{M}(\mathbf{p})|, \quad (q'_x, q'_y, q'_z)^T = R(\mathbf{q}-\mathbf{t}), \tag{7}$$

where $R, \mathbf{t}$ are given by $C^\star$. Note that for each pixel p that does not belong to any correspondence, we copy the value of z_p from the closest pixel that has correspondences. We then generate the initial point cloud $P = \{C(\mathbf{p}, z_p)|\mathbf{p} \in I\}$ according to (4).

5.4 Step III: Point Cloud Optimization

We refine the initial image-shape correspondences and the initial point cloud by solving a joint alignment problem, whose objective function minimizes the distance (defined via correspondences) between the point cloud and deformed similar shapes. We employ an ICP-like procedure, alternating between a continuous optimization step, which optimizes the continuous variables including camera configuration C, the z-coordinates of each pixel $\{z_p|\mathbf{p} \in I\}$ and the deformation of each shape $\mathcal{D}_i, 1 \le i \le K$; and a discrete optimization step, which updates image-shape correspondences.

Continuous optimization step. We consider multiple objectives for aligning the induced point-cloud and the similar shapes. The first term evaluates the sum of squared distances between the corresponding points:

$$f_{\text{data}} = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{|\mathcal{M}_i|}\sum_{(\mathbf{p},\mathbf{q}_i)\in\mathcal{M}_i} w_{\mathbf{p}}\|C(\mathbf{p}, z_p) - \mathcal{D}_i(\mathbf{q}_i)\|^2.$$

Here, $w_{\mathbf{p}}$ is a weight that is adjusted to be higher for more reliable correspondence $(\mathbf{p}, \mathbf{q}_i)$. We set $w_{\mathbf{p}} = 1$ for interior points and $w_{\mathbf{p}} = 20$ for points close to silhouettes. Note that another option is to measure the distance in the image domain. However, due to distance distortions in projection, two points that are close to each other in the image domain may be far from each other on the original shape. It turns out measuring the distance in the image domain leads to far less accurate results (see Figure 6).

As f_{data} considers each pixel independently, we next introduce a second term to regularize the z-coordinate of neighboring pixels:

$$f_{\text{regu}} = \frac{1}{|\mathcal{N}|}\sum_{(p,p')\in\mathcal{N}}(z_p - z_{p'})^2.$$

Finally, the third term applies the key deformation priors learned in the preprocessing stage:

$$f_{\text{prior}} = \frac{1}{K}\sum_{i=1}^{K}\text{prior}(\mathcal{D}_i).$$

Combining f_{data}, f_{regu}, and f_{prior}, the energy minimization problem in the continuous optimization step takes the form:

$$\min_{\{z_p\},C,\{\mathcal{D}_i\}} \quad f_{\text{data}} + \lambda_r f_{\text{regu}} + \lambda_p f_{\text{prior}}. \tag{8}$$

In our experiments, we use throughout the same set of parameters: $\lambda_r = 0.01$ and $\lambda_p = 1$.

We again apply alternating optimization to effectively optimize (8). In each step, we first fix the z-coordinates z_p to optimize the camera configuration C and the shape deformations $\mathcal{D}_i$:

$$\min_{C,\{\mathcal{D}_i\}} \quad f_{\text{data}} + \lambda_p f_{\text{prior}}\,. \tag{9}$$

We then fix C and $\mathcal{D}_i$ to optimize the z-coordinates z_p:

$$\min_{\{z_p\}} \quad f_{\text{data}} + \lambda_r f_{\text{regu}}\,. \tag{10}$$

The key advantage of this alternating optimization strategy is that (9) and (10) are either sparse (constraining neighboring pixels) or of small-scale (camera configuration and deformation parameters). This enables us to apply second-order Newton methods to optimize them effectively, i.e., we solve a sparse or a small-scale linear system at each Newton iteration. As the objective terms consists of non-linear least squares, we apply a Gauss-Newton method for optimizing Equations (9) and (10). The derivation is quite standard and we omit the details.

Discrete optimization step. Given the optimized point-cloud $P = \{C(\mathbf{p}, z_p)|\mathbf{p} \in I\}$ and deformed shapes, we proceed to optimize the image-shape correspondences. We first convert each deformed shape $\mathcal{D}_i(S_i)$ into a point-cloud S'_i by simulating a scan from the current camera configuration. We then initialize $\mathcal{M}_i$ to collect closest point-pairs

$$\mathcal{M}_i^{init} = \{(\mathbf{p},\mathbf{q})|\mathbf{q} = \operatorname*{argmin}_{\mathbf{q}'\in S'_i}\|\mathbf{p}-\mathbf{q}'\| \text{ or } \mathbf{p} = \operatorname*{argmin}_{\mathbf{p}'\in P}\|\mathbf{p}'-\mathbf{q}\|\}.$$

As there may only exist partial similarity between the image object I and each shape S_i, we adopt the median thresholding scheme [Rusinkiewicz and Levoy 2001] to remove correspondences that are far from each other, leaving

$$\mathcal{M}_i = \{(\mathbf{p},\mathbf{q})|\|\mathbf{p}-\mathbf{q}\| \le 2\sigma_i, (\mathbf{p},\mathbf{q}) \in \mathcal{M}_i^{init}\},$$

where σ_i is the median of $\|\mathbf{p}-\mathbf{q}\|$ among each $\mathcal{M}_i^{init}$. Figure 8 shows an example of the non-rigid alignment process. In practice, only 4-6 alternating updates are sufficient for good results.

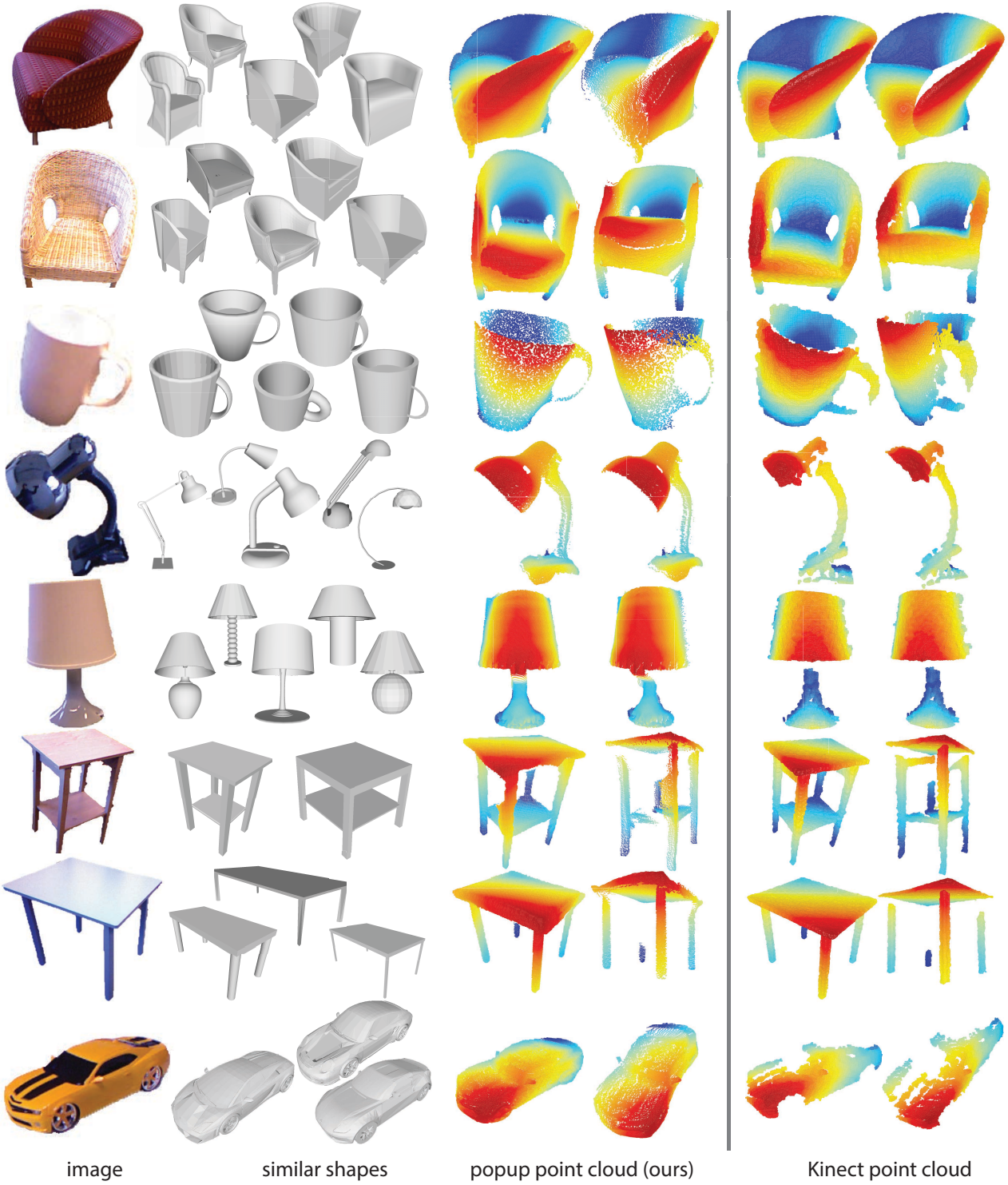

Figure 7: ***Representative Results.*** *We have evaluated our approach on five categories of objects. This figure shows representative results in each category. For every object we show the input 2D image, the extracted similar shapes, the reconstructed point cloud and finally the ground truth Kinect scan.*

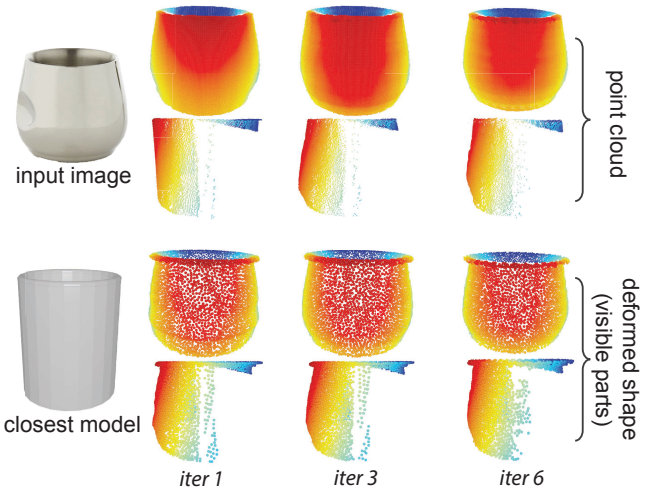

Figure 8: ***Intermediate Optimization Results.*** *We employ an iterative scheme to simultaneously refine point cloud reconstruction and to deform similar shapes. This figure shows the image view (rows 1 and 3) and side view (rows 2 and 4) of the point cloud and the deformed shape in different iterations. Since strong and reliable constraints are imposed along the silhouette of the estimated image view, the results look correct from the original image view at the initialization step. However, the interior of the point cloud is initialized poorly, as can be observed from the side view. When the optimization proceeds, significant improvement can be observed from the side view as the shape deformation subspace prior from similar shapes regularizes the solution and "propagates" the information to the otherwise under-determined interior regions.*

6 Evaluation

We evaluated the proposed shape-driven image-based modeling on various Kinect scans with associated color information.

6.1 Experimental Setup

Images. We consider five categories of objects: chairs, tables, cups, lamps and cars. Each category consists of 4-6 Kinect scans of objects with different shapes. Figure 7 shows representative results in each category. We assume all the objects are captured in our standard setting, where background is easy to remove. The Kinect scans are for evaluation purposes only.

Shapes. The 3D shapes are from Trimble warehouse. Each category contains 2K-7K shapes (see Table 1), where the Chair data set is from [Kim et al. 2013], the Car data set is from [Huang et al. 2013], and the three remaining datasets were collected using a similar strategy to that described in [Kim et al. 2013]. Note that even with thousands of shapes, the shape space is not densely covered as can be seen from the extracted similar shapes (see Figure 7).

Evaluation protocol. We evaluate the reconstructed point-cloud of each object image against the Kinect depth scans. To factor out the free scaling degree of freedom we first compute a similarity transform that aligns the reconstructed point cloud with the Kinect scan. Given the calibrated reconstruction $\mathcal{P}$ and the Kinect scan $\mathcal{P}^{\text{Kinect}}$, we propose two metrics to evaluate the quality of $\mathcal{P}$. The first metric evaluates the Hausdorff distance between $\mathcal{P}$ and $\mathcal{P}^{\text{Kinect}}$:

$$d(\mathbf{p}, \mathcal{P}^{\text{Kinect}}) = \min_{\mathbf{q}\in\mathcal{P}^{\text{Kinect}}} \|\mathbf{p}-\mathbf{q}\|, \forall \mathbf{p} \in \mathcal{P}.$$

The second metric evaluates the deviation between the pair of corresponding points $\mathbf{p}$ and $\mathbf{f}(\mathbf{p})$ in $\mathcal{P}$ and $\mathcal{P}^{\text{Kinect}}$:

$$d(\mathbf{p}, \mathbf{f}(\mathbf{p}))) = \|\mathbf{p}-\mathbf{f}(\mathbf{p})\|, \forall \mathbf{p} \in \mathcal{P}.$$

Table 1: *Statistics on various datasets. Shapes are normalized to have diameter* 1. *ϵ_* and σ_* are the mean and standard deviation of the metrics defined in Sec 6.1. ϵ_*^{bs} and σ_*^{bs} are the numbers for the best matched single shape and those with no superscript are numbers by our algorithm.*

	#shapes	$\epsilon^{\text{bs}}_{\text{haus}}/\sigma^{\text{bs}}_{\text{haus}}$	$\epsilon_{\text{haus}}/\sigma_{\text{haus}}$	$\epsilon_{\text{deviation}}/\sigma_{\text{deviation}}$
Chair	7.3K	0.17 / 0.15	0.05 / 0.03	0.11 / 0.10
Table	4.2K	0.14 / 0.12	0.06 / 0.06	0.12 / 0.13
Cup	1.1K	0.15 / 0.11	0.05 / 0.04	0.09 / 0.09
Lamp	2.0K	0.13 / 0.15	0.06 / 0.03	0.10 / 0.11
Car	1.7K	0.12 / 0.11	0.05 / 0.03	0.09 / 0.08

It is clear that the Hausdorff distance is invariant to interior drifting on the surface, while the correspondence deviation is more strict. For each distance metric we collect statistics on the mean and variance of $d(\mathbf{p}, \mathcal{P})$ over all points (see Table 1). As a baseline, we calculated the Hausdorff metric obtained by the most similar shape.

6.2 Analysis of Results

Table 1 and Figure 7 shows representative results for the proposed approach. Overall the results are reasonably good despite the obvious difficulty of the problem, with 68.2% correspondences falling below 0.02 times the averaged shape diameter. For all datasets, the Hausdorff distance error is considerably lower than that of the deviation error. As the shape of the point cloud is driven by the shape collection, this shows that using the shape collection as a good prior, the distribution of points is restricted to drift along the common shape space. The deviation error is large because the correspondences may glide along the shapes, which are not exactly the same. We next discuss the results for each category.

Chair and tables. We evaluate on the chair and table categories because fine geometric details are present in these shapes. Like other man-made objects, chair and tables usually have strong symmetries, implying a lower-dimensional deformation space. On the other hand, the four legs may introduce matching ambiguities. On these categories, we find that our algorithm is limited when self-occlusion presents: the lower board is occluded by the front leg in Row 6 of Figure 7 and consequently part of it is attached to the leg.

Cups. Cups are relatively small household items and usually have a circular symmetrical body. Interestingly, our method produces visually more pleasing results compared with the Kinect, because the object size is reaching the resolution limit of the sensor and the surface is specular, which is challenging for the structural light mechanism of the Kinect.

Lamps. We choose this category because it has large variations in the possible shapes, particularly in the curvature of the pole. It can be seen that our algorithm succeeds in both lamp examples in Figure 7. The success can be attributed to two reasons. First, we use a data-driven approach to implicitly combine parts from different shapes. Second, we use a non-rigid deformation field, which allows the bending of the pole.

Cars. We choose this category as a common outdoor object having fine geometric details (e.g., wheels, side mirrors). Our algorithm could accurately estimates the depth of cars. On the other hand, the Kinect has problems in detecting windows and wheels, because they are too reflective or too dark respectively.

Comparison to Automatic Pop-Up. Automatic Pop-Up [Hoiem et al. 2005] automatically reconstructs 3D information using a single image and was initially designed for outdoor scenes using plane classifiers. The software assumes simple geometric priors and tend to work poorly for complicated indoor objects with thin and fine features. We show the effect of Automatic Pop-Up on a chair model in Figure 9 using the pre-trained classifiers. Our algorithm is visu-

Figure 9: ***Comparison with Automatic Pop-Up*** *[Hoiem et al. 2005]. That algorithm fails to recover the delicate geometric structure of the chair legs and produces unnatural 3D effects.*

ally significantly better than the output from this software (compare the last column of Figures 2 and 9).

6.3 Discussion

Image-Shape matching versus Image-Image matching. Image-image matching is far less accurate than image-shape matching (Figure 6). Two reasons accounts for the difference. First, the projection from 2D to 3D is perspective and two points close in 2D may be far away in 3D. Second, a 3D point cloud for a shape is obtainable in our setting and projecting it to 2D loses important information. In fact, strong perspective projection might still hurt the performance of our algorithm. For example, the seat of the second row in Figure 7 is estimated to be very thick, which can be attributed to very strong perspective effect close to the chair leg.

Deformation prior is important. We find that our deformation shape prior is key to success. As illustrated in Figure 3, the deformation space is low-dimensional and points are generally restricted to move parallel to meaningful axes or scale in a coordinated manner. Thus, such constraints makes sure that the local deformation maintain global symmetries. Evidently, in Figure 10, because the problem of single image depth reconstruction is intrinsically underdetermined, poor results are obtained without a prior (top). On the contrary, the isotropic symmetry learned by the prior ensures that the deformation subspace has only 1D, which is a coordinated scaling around the y-axis.

Timing. All experiments were conducted on a standard desktop platform with a 2.4GHz Intel Core 2 Duo-core and 12GB of RAM. For each category, the pre-processing stage is shared by all images, which takes 3507s for chairs, 2184s for tables, 516s for cups, 1223s for lamps and 1109s for cars. The camera initialization stage takes on average 0.3s for each input image, and most of the time was spent on extracting image features. For each image object, the point cloud initialization stage took $\sim$7s in average, with $\sim$1s on correspondence initialization and $\sim$ 6s on correspondence pruning. The point cloud optimization stage took $\sim$ 25s in average. The total running time for processing an image object was $\sim$ 33s.

7 Applications

In this section, we use a series of applications to show the usefulness of the reconstructed point cloud, including relighting image object, synthesizing unseen novel views, and depth-aware image composition. In the end, we show that, in some situations, a reasonable full mesh of an image object can be recovered using our point cloud as input, by exploiting shape symmetries.

Relighting. Given an image, we estimate the depth of each pixel and use local PCA analysis to estimate normals and simulate the lighting effects under different illumination conditions. In Figure 11, we assume an ambient light and a diffusion reflection on the surface. Notice that the synthesized image is almost photo-realistic,

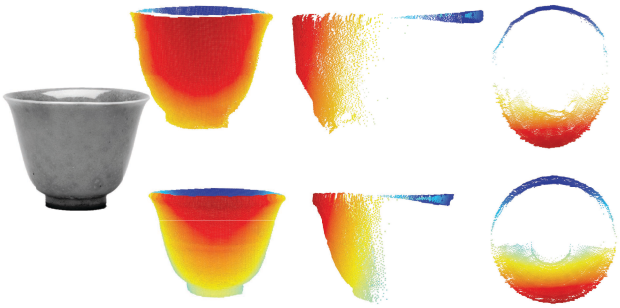

Figure 10: ***Effect of Deformation Prior.*** *For a circular symmetric cup, the top/ bottom row shows the results without/with the prior. We see that the deformation prior guarantees the reconstruction is circularly symmetric.*

except some artifact at the top-right corner due to inaccurate depth estimation.

Novel View Synthesis. Since the full 3D information for each pixel is available, we can simulate the movement of the camera in 3D and synthesize novel views. In Figure 12, the synthesized view from our depth estimation using the inverse warping method [Marcato Jr 1998] is almost photo-realistic. In particular, as we can even accurately recover the depth information of the back mirror of the car, the appearance around the back mirror is quite natural when the car is rotating in the counter-clockwise direction (-30 and -15 deg). Note that the missing parts that are invisible in the image can possibly be recovered by exploiting model symmetry.

Depth-Aware Image Composition. In Figure 13 we demonstrate an experiment in which we compose a 3D model of a woman with a sofa image, so that the woman is 'sitting' on the sofa. A correct composition should make sure that the woman's body and legs cover the back arm, and her hip is covered by the front chair arm. Since the depth of the sofa can be recovered, we can compute this correct occlusion for each pixel (right), as opposed to unnatural occlusion patterns if no depth information is available (left).

Symmetry-based Surface Reconstruction. The inferred point cloud only has points visible from the camera view. However, we show in this experiment that it is a key intermediate step towards full 3D model reconstruction. We hallucinate the missing parts by exploiting the model symmetry. We use [Mitra et al. 2006] to extract symmetry patterns from similar shapes and transport them to the point cloud. In Figure 14, we discover a circular symme-

Figure 11: ***Relighting.*** *Using the inferred depth information, we can build the normal map and simulate different lighting conditions. Leftmost column is the input image and the three columns on the right are the simulated illumination. A directional light source moves from left to right (top row), or up to down (bottom row).*

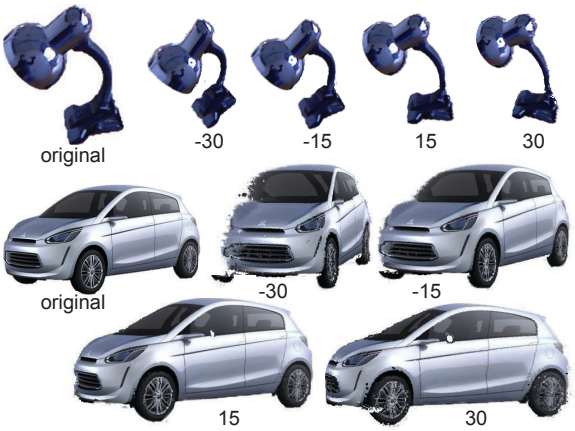

Figure 12: ***Novel View Synthesize.*** *Simulated images by rotating cameras around the y-axis.*

try for the cup body and a plane symmetry for the handle, which also induces a segmentation of the cup. Thus, we can transport the symmetry parameters and achieve full shape recovery using [Mitra et al. 2007]. Finally, we apply Poisson reconstruction [Kazhdan et al. 2006] on the extracted point cloud for surface reconstruction with smoothing. The final result is in the bottom row of Figure 14. We see that the reconstructed mesh generally looks natural from all views. However, because we only apply plane symmetry at the handle part, there is gap at the bottom of the handle and the reconstructed surface is not connected. Discovering better structural predictors to close the small gap is an interesting open problem for further exploration.

8 Conclusion and Future Work

In this paper, we have presented a data-driven algorithm for adding depth information to an image object. The algorithm takes as input an image of a segmented object and a collection of 3D shapes of the same object class, and computes various geometric priors from the shape collection to optimize the depth estimation of the image object. This procedure is fully automatic. We have evaluated the performance of the presented approach on a benchmark that consists of Kinect scans of a variety of common objects taken under different lighting conditions. Experimental results show that our approach produces depth that is close to the ground-truth, and is superior to state-of-the-art depth estimators. We have also shown the usefulness of the our approach for various applications.

Figure 13: ***Depth-Aware Image Composition.*** *Image composition may be tedious as direct overlay may lead to incorrect occlusions (Left). Given an image of a sofa and a 3D model of a sitting woman, occlusions between them can be correctly computed based upon the depth inferred by our algorithm (Right).*

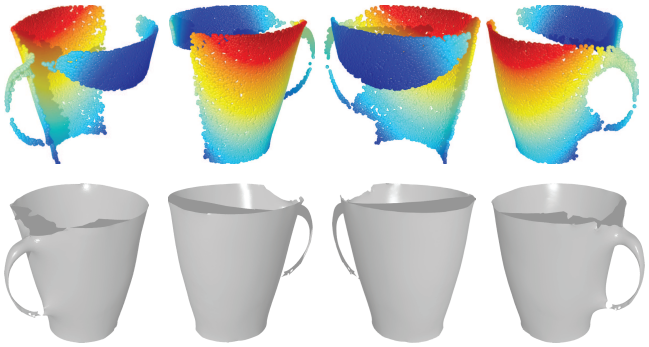

Figure 14: ***Symmetry-based Completion and Surface Reconstruction.*** *Top: point clouds viewed from different angles. Bottom: surface reconstruction results with symmetry-based completion. Original image is from the third row of Figure 7.*

Besides the applications demonstrated in this paper, the presented depth estimator enables a variety of other applications in both computer graphics and computer vision. As an example, the shape collection can serve as the hub that links many image objects. This is particularly useful for retrieving similar image objects that were taken from drastically different view points that cannot be matched well by pure image methods. As another example, with the help of the image-shape network, we can propagate rich image labels for the purpose of categorizing shapes — a challenging problem in shape analysis due to the lack of labeled shapes or of data combining 3D shapes and labels.

Limitations. Of course, as stated, our approach requires a segmented image of an object and a knowledge of the object class. These are well studied problems in computer vision and future work can combine these with our approach.

The presented method works best with man-made objects whose 3D models can be well aligned and where the variation in shape poses is modest. It does not apply well to objects of high variability, such as trees, or buildings, or of high articulation, such as animals. For these objects, it is important to utilize more specialized domain knowledge (i.e., skeletons and regular structures) to establish correspondences and estimate depth. Finally, in our experience, a minimum of a couple of hundreds of shapes is necessary for the algorithm to succeed. The intuition is that each part of the object in the image needs to have multiple correspondences for good regularization.

Future work. There are ample opportunities for future research. While so far we have focused on estimating the depth of a single segmented object, it would be very interesting to generalize this approach to estimate the depth of an entire scene. This would require us to automate the object detection process and to take into account spatial relations among objects.

Acknowledgements. We thank the reviewers for their comments and suggestions on the paper. This work was supported in part by NSF grants IIS 1016324 and DMS 1228304, AFOSR grant FA9550-12-1-0372, NSFC grant 61202221, the Max Plack Center for Visual Computing and Communications, Google and Motorola research awards, a gift from HTC corporation, the Marie Curie Career Integration Grant 303541, the ERC Starting Grant SmartGeometry ((StG-2013-335373), and gifts from Adobe.

References

AVERKIOU, M., KIM, V., ZHENG, Y., AND MITRA, N. J. 2014.

Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. *CGF*.

CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3d modeling. *ACM ToG 30*, 4 (Aug.), 35:1–35:10.

CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3d model retrieval. *CGF 22*, 3, 223–232.

COIFMAN, R. R., LAFON, S., LEE, A. B., MAGGIONI, M., NADLER, B., WARNER, F., AND ZUCKER, S. W. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS 102*, 21, 7426–7431.

CYR, C. M., AND KIMIA, B. B. 2004. A similarity-based aspect-graph approach to 3d object recognition. *IJCV 57*, 1 (Apr.), 5–22.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 886–893.

GOLDMAN, D. B., CURLESS, B., HERTZMANN, A., AND SEITZ, S. M. 2005. Shape and spatially-varying brdfs from photometric stereo. In *Proc. ICCV*, 341–348.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM ToG 24*, 3 (July), 577–584.

HUANG, Q., AND GUIBAS, L. 2013. Consistent shape maps via semidefinite programming. *CGF 32*, 5, 177–186.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation using linear programming. *ACM ToG 30*, 6.

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

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

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. *CGF*, 61–70.

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

KIM, Y. M., MITRA, N. J., YAN, D.-M., AND GUIBAS, L. 2012. Acquiring 3d indoor environments with variability and repetition. *ACM ToG 31*, 6 (Nov.), 138:1–138:11.

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

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM ToG 22*, 2.

MARCATO JR, R. W. 1998. *Optimizing an inverse warper*. PhD thesis, Massachusetts Institute of Technology.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM ToG*, 560–568.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2007. Symmetrization. *ACM ToG 26*, 3, #63, 1–8.

MUNICH, M. E., AND PERONA, P. 1999. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *ICCV*, vol. 1.

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

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

OLIVA, A., AND TORRALBA, A. 2006. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research 155*, 23–36.

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

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3d shapes. *ACM ToG 30*, 4, 33:1–33:10.

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the ICP algorithm. In *3DIM*, 145–152.

SAXENA, A., SUN, M., AND NG, A. Y. 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI 31*, 5 (May), 824–840.

SHEN, C.-H., FU, H., CHEN, K., AND HU, S.-M. 2012. Structure recovery by part assembly. *ACM ToG 31*, 6, 180:1–180:11.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. *CGF*, 175–184.

SUMNER, R. W., SCHMID, J., AND PAULY, M. 2007. Embedded deformation for shape manipulation. *ACM ToG 26*, 3 (July).

SUN, M., KUMAR, S. S., BRADSKI, G., AND SAVARESE, S. 2011. Toward automatic 3d generic object modeling from one single image. In *3DIMPVT*, IEEE.

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

WANG, Y., GONG, M., WANG, T., COHEN-OR, D., ZHANG, H., AND CHEN, B. 2013. Projective analysis for 3d shape segmentation. *ACM ToG 32*, 6 (Nov.), 192:1–192:12.

WU, T.-P., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2008. Interactive normal reconstruction from a single image. *ACM ToG*, 119:1–119:9.

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

ZHENG, Y., CHEN, X., CHENG, M.-M., ZHOU, K., HU, S.-M., AND MITRA, N. J. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM ToG 31*, 4, 99:1–99:11.

ZIA, Z., STARK, M., SCHIELE, B., AND SCHINDLER, K. 2013. Detailed 3d representations for object recognition and modeling. *IEEE TPAMI 35*, 11, 2608 – 2623.