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.


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

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 related shapes. 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 Guitbas 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 Trumble 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 model-

ing [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 $S = \{S_1, S_2, \ldots, S_N\}$ of the same class as
input, and simultaneously estimates the object pose shown in $I$ and reconstructions 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-
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 \( p_{\alpha} \in J \) and the associated basis functions \( B_{\alpha}(\cdot) \). Given a point \( x \in \mathbb{R}^3 \), its deformed counterpart is a linear combination of the control points:

\[
D(x) = \sum_{p_{\alpha} \in J} B_{\alpha}(x) p_{\alpha}. \tag{1}
\]

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 \( D_i^* \).

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 \( D_i^* \) and their inverse deformation-\( s D_i^{-1} \). For each shape \( S_i \) and each neighboring shape \( S_j \), we transform the original control point \( p_{\alpha_j} \) (i.e., in the rest state) of \( D_i^* \) to \( (D_i^{-1} \circ D_j^*)(p_{\alpha_j}) \). Let \( c_{i,j} \) be the vector that collects all transformed control points. Let \( c_{i,j}^0 \) be the original control points. Then each neighboring shape \( S_j \) gives rise to a deformation sample \( c_{i,j} - c_{i,j}^0 \). 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 \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \) and principal directions \( u_1, u_2, \cdots, u_M \) of the deformation space. The prior model on the deformation of each shape is given by

\[
\text{Prior}(D) = -\sum_{j=1}^{M} \frac{\sigma_j}{\sigma_j + \epsilon} \left( (c_i - c_i^0)^T u_j \right)^2, \tag{2}
\]

Intuitively, a deformation leads to a small objective value if it follows the majority of the deformation samples. Note that the shift \( \epsilon \) 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 \( C = (R, t, z_f, s_x, s_y) \). Here \((R, t)\) specifies the rigid motion from the common coordinate system \( \Sigma^C \) associated with the input shapes to the camera coordinate system \( \Sigma^C \); \( z_f \) 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 \( q = (q_x, q_y, q_z) \) in \( \Sigma^C \), its corresponding pixel coordinate \( p = (p_x, p_y)^T \) is given by

\[
p_x = \frac{q_x z_f}{s_x q_z}, \quad p_y = \frac{q_y z_f}{s_y q_z}, \quad q' = R^T(q - t). \tag{3}
\]

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

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

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

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 \( t \) 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 \( C_{\text{cand}} \) collect all candidate camera configurations. For each \( C \in C_{\text{cand}} \), we denote \( I_C \) 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 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):

$$C^* = \arg\min_{C \in C_{\text{const}}} \sum_{i=1}^{N} \exp(-\|f(I) - f(I_i^C)\|^2/2\sigma^2),$$

where $f(\cdot)$ is a given feature descriptor, $\sigma = \min_{i \in \{1, \ldots, n\}, C \in C_{\text{const}}} \|f(I) - f(I_i^C)\|$, 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 $f$ 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 pairwise 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), where 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 $S_1, \ldots, S_K$.

**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 $f$ and $I_i^C$.

After alignment, we derive the initial pixel-shape correspondences from the overlaid image objects. With $M_i \subset I \times S_i$ we denote the initial correspondences from $I \rightarrow S_i$. 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 correspondences is to make them consistent with the optimal deformations $\{D_i^C\}$ that align the input shapes. As both the consistency property and the smoothness prior only involve pairs of correspondences, we formulate the correspondence initialization as a Markov Random Field (MRF) problem.

We introduce a binary random variable $x_{ci} \in \{0, 1\}$ for each initial correspondence $c \in \bigcup_{i=1}^{K}M_i$, where $x_{ci} = 1$ if $c$ is selected and $x_{ci} = 0$ otherwise. We then define the two types of pair-wise potential functions. For each correspondence pair $c_i = (p, q_i) \in M_i$ and $c_j = (p', q_j) \in M_j$, we define a consistency potential:

$$\phi(x_{c_i}, x_{c_j}) = \begin{cases} -\infty & x_{c_i} = x_{c_j} = 1, \|D_i^C(q_i) - D_j^C(q_j)\| \geq \delta \\ 1 & \text{otherwise} \end{cases}$$

where $\delta$ is set as the 0.05 times the averaged shape diameter. For each pair of correspondences $c = (p, q_i) \in M_i$ and $c' = (p', q_j) \in M_j$, where $p, p'$ and $q_i, q_j$ 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.
Then the total potential function simply sums all pair-wise potential functions
\[ f = \sum_{(c,c') \in P} \phi(x_c, x_{c'}), \]  
where \( P \) 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 \( M_i \), 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 \( p = (x_p, y_p) \) (in the camera coordinate system of \( C^* \)) by averaging \( z \)-coordinates of the corresponding points of similar shapes:
\[ z_p = \sum_{(p,q) \in M(p)} q'_z / |M(p)|, \quad (q'_x, q'_y, q'_z)^T = R(q - t), \]
where \( R, t \) are given by \( C^* \). 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 \( P = \{ C(p, z_p) | p \in I \} \) 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 \( \{ z_p | p \in I \} \) and the deformation of each shape \( D_i, 1 \leq i \leq K \); 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:
\[ f_{data} = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{|M_i|} \sum_{(p,q) \in M_i} w_p \|C(p, z_p) - D_i(q_i)\|^2. \]
Here, \( w_p \) is a weight that is adjusted to be higher for more reliable correspondence \( (p, q_i) \). We set \( w_p = 1 \) for interior points and \( w_p = 20 \) 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:
\[ f_{regu} = \frac{1}{|N|} \sum_{(p,p') \in N} (z_p - z_{p'})^2. \]

Finally, the third term applies the key deformation priors learned in the preprocessing stage:
\[ f_{prior} = \frac{1}{K} \sum_{i=1}^{K} \text{prior}(D_i). \]

Combining \( f_{data}, f_{regu}, \) and \( f_{prior} \), the energy minimization problem in the continuous optimization step takes the form:
\[ \min_{\{z_p\}, C_i, \{D_i\}} f_{data} + \lambda_r f_{regu} + \lambda_p f_{prior}. \]  
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 \( D_i \):
\[ \min_{C, \{D_i\}} f_{data} + \lambda_r f_{regu} \]  
We then fix \( C \) and \( D_i \) to optimize the \( z \)-coordinates \( z_p \):
\[ \min_{\{z_p\}} f_{data} + \lambda_r f_{regu}. \]

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 \( P = \{ C(p, z_p) | p \in I \} \) and deformed shapes, we proceed to optimize the image-shape correspondences. We first convert each deformed shape \( D_i(S_i) \) into a point-cloud \( S'_i \) by simulating a scan from the current camera configuration. We then initialize \( M_i \) to collect closest point-pairs:
\[ M_i^{init} = \{(p, q) | q = \arg \min_{q' \in S'_i} \|p - q'\| \} \]  
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
\[ M_i = \{(p, q) | \|p - q\| \leq 2\sigma, (p, q) \in M_i^{init}\}, \]
where \( \sigma \) is the median of \( \|p - q\| \) among each \( M_i^{init} \). 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 $\mathcal{P}_{\text{Kinect}}$:

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

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

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

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

<table>
<thead>
<tr>
<th>Category</th>
<th>#shapes</th>
<th>$\epsilon_{\text{haus}}/\sigma_{\text{haus}}$</th>
<th>$\epsilon_{\text{bs}}/\sigma_{\text{bs}}$</th>
<th>$\epsilon_{\text{deviation}}/\sigma_{\text{deviation}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chair</td>
<td>7.3K</td>
<td>0.17 / 0.15</td>
<td>0.06 / 0.03</td>
<td>0.11 / 0.10</td>
</tr>
<tr>
<td>Table</td>
<td>4.2K</td>
<td>0.14 / 0.12</td>
<td>0.06 / 0.01</td>
<td>0.12 / 0.13</td>
</tr>
<tr>
<td>Cup</td>
<td>1.1K</td>
<td>0.15 / 0.11</td>
<td>0.05 / 0.04</td>
<td>0.09 / 0.09</td>
</tr>
<tr>
<td>Lamp</td>
<td>2.0K</td>
<td>0.13 / 0.15</td>
<td>0.06 / 0.03</td>
<td>0.10 / 0.11</td>
</tr>
<tr>
<td>Car</td>
<td>1.7K</td>
<td>0.12 / 0.11</td>
<td>0.05 / 0.03</td>
<td>0.09 / 0.08</td>
</tr>
</tbody>
</table>

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 $d(p, \mathcal{P})$ 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, chairs 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 estimate 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 tends 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-
Timing. All experiments were conducted on a standard desktop running time for processing an image object was 3507 s, which takes around the y-axis. The deformation subspace has only 1D, which is a coordinated transformation space is low-dimensional and points are generally restricted to move parallel to meaningful axes or scale in a coordinated manner. Thus, such constraints make sure that the local deformation maintain global symmetries. Evidently, in Figure 10, because the problem of single image depth reconstruction is intrinsically under-determined, 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.

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 make sure that the local deformation maintain global symmetries. Evidently, in Figure 10, because the problem of single image depth reconstruction is intrinsically under-determined, 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 3507 s for chairs, 2184 s for tables, 516 s for cups, 1223 s for lamps and 1106 s for cars. The camera initialization stage takes on average 0.3 s 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 ~7s in average, with ~1s on correspondence initialization and ~6s on correspondence pruning. The point cloud optimization stage took ~25s in average. The total running time for processing an image object was ~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, 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 [Marr 1971] 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 symmetry for the cup body and a plane symmetry for the handle, which al-

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.

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.

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.

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).

so 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.

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, and the ERC Starting Grant SmartGeometry (335373).

References


