When 3D Reconstruction Meets Ubiquitous RGB-D Images

Quanshi Zhang†, Xuan Song†, Xiaowei Shao†, Huijing Zhao‡, and Ryosuke Shibasaki†
†University of Tokyo, ‡Peking University

Abstract

3D reconstruction from a single image is a classical problem in computer vision. However, it still poses great challenges for the reconstruction of daily-use objects with irregular[1] shapes. In this paper, we propose to learn 3D reconstruction knowledge from informally captured[2] RGB-D images, which will probably be ubiquitously used in daily life. The learning of 3D reconstruction is defined as a category modeling problem, in which a model for each category is trained to encode category-specific knowledge for 3D reconstruction. The category model estimates the pixel-level 3D structure of an object from its 2D appearance, by taking into account considerable variations in rotation, 3D structure, and texture. Learning 3D reconstruction from ubiquitous RGB-D images creates a new set of challenges. Experimental results have demonstrated the effectiveness of the proposed approach.

1. Introduction

3D reconstruction is a classical area in the field of computer vision, but 3D reconstruction from a single image still has great challenges. In this paper, we re-consider the problem of single-view 3D reconstruction in terms of two CV areas: category modeling and knowledge mining from big visual data (see Fig. 1).

Category modeling & task output: For the bottleneck in single-view 3D reconstruction, *i.e.* the reconstruction of objects with irregular[1] structures, we have to return to the concept of "category modeling".

Therefore, the objective is to train a model to detect objects in the target category in large images, while simultaneously projecting their 2D shapes into the 3D space at the pixel level. The category model encodes the knowledge of how intra-category structure deformation and object rotations affect 2D object shapes.

Mining from big visual data & task input: *Another bottleneck lies in efficiently learning the category-specific knowledge of 3D reconstruction for a huge number of categories.* Ideally, we would need to train a model for each object category in daily use, so as to construct a knowledge base to provide a 3D reconstruction service for arbitrary RGB images. Therefore, we hope to learn from big visual data[2] to avoid the labor of manually preparing training samples (*e.g.* well built 3D models), and thus ensure a high learning efficiency.

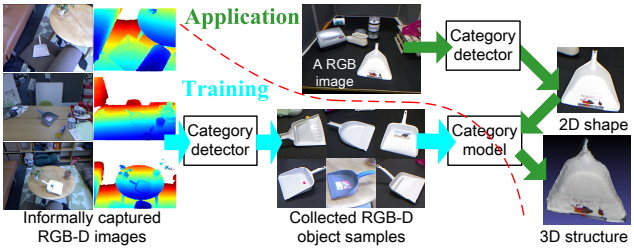

Figure 1. Flowchart for training (cyan arrows) and testing (green arrows) processes. Single-view 3D reconstruction is hampered by objects with irregular[1] shapes. Therefore, for objects in each category, we aim to learn a specific category model for 3D reconstruction. We propose to directly train the category model from "informally captured" RGB-D images (where target objects are NOT aligned) to save human labeling, thereby ensuring high efficiency in model learning. We first mine the category detector from informally captured RGB-D images to collect RGB-D objects, and then use these object samples to train the category model. For testing, the category detector and category model are used in sequence to localize the target object and estimate its 3D structure.

In this paper, we train category models directly from informally captured and "unaligned" RGB-D images. We use the phrase "informally captured" to describe loose requirements for ubiquitous[2] images. They are typical of what can be directly collected by search engines (Figures 1 and 2).

The informally captured images are not manually aligned, and they consist of small objects that are randomly positioned. In particular, these daily-use objects are

[1]The word "irregular" is used to indicate that our approach focuses on general object categories without setting strong assumptions for object shapes. In contrast, Cheeger-set-based methods focus on ball-like surfaces, and perspective-based methods require vanishing points.

[2]Actually, learning from "big visual data" is still in the early stages of development, and its scope is quite extensive. It usually involves two aspects. This first is learning from a large amount of web data, such as the widely used deep learning [17]. The second is learning under challenging conditions of "informally captured" images that contain small objects and are ubiquitous in everyday life, as is the case in our method. [26] also provides detailed description of the second aspect.

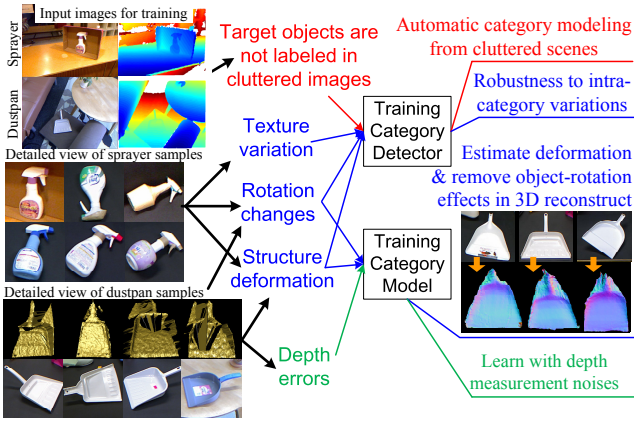

Figure 2. Overview of challenges.

usually designed with texture variations, various rotations, and some structure deformation for commercial purposes. Technically speaking, these images can be loosely regarded as a kind of big visual data[2].

Challenge analysis: Just like the training, the RGB images for testing 3D reconstruction are also such kind of informally captured images in this research. The use of informally captured images raises a number of challenges, as shown in Fig. 2.

For application purposes, the category model should have the ability to detect small target objects with various textures, rotations, and scales in large RGB images before 3D reconstruction. Similarly, the learning process should also be able to handle these challenges to collect RGB-D objects in cluttered RGB-D images. Moreover, this detection has to be accurate to the part level, since 3D reconstruction uses the parts' 2D positions to estimate the pose and 3D structure of an object.

Then, we analyze the challenges of learning 3D reconstruction knowledge from the collected RGB-D objects[3]. We need to simultaneously consider the following three factors that affect the correspondence between the 2D and 3D structure of the object. 1) Rotation changes. This is the primary factor. 2) Deformation of 3D object structures. 3) Depth errors near object edges in training samples (measured by the Kinect)

Proposed method: As shown in Fig. 1, the training of category detectors is the first step. The category detectors collect[3] RGB-D/RGB object samples from the informally captured RGB-D/RGB images, which serves as the basis for further training/testing of 3D reconstruction. We use unsupervised category modeling based on graph matching [28] to train the category detector (specified in [26]). This algorithm automatically discovers a set of relatively distinguishing parts (namely key parts) of objects for each category. Thus, the category detector achieves a part-level detection. We then train the category model to use the parts' 2D positions to determine the object's pose and 3D structure.

Unlike conventional methods that use well built 3D models to provide multi-view object appearances and prior regularity of 3D structure deformation, our approach needs to learn this knowledge in an unsupervised manner. In other words, we need to train the category model to simultaneously remove the effects of viewpoint changes, and estimate the 3D structure deformation using the 2D shape of an object. The category model consists of a number of estimators, each corresponding to a specific position inside the 2D shape. These estimators use the spatial relationship between key object parts to estimate the corresponding 3D coordinates of their positions. We connect neighboring estimators to form a continuous Markov random field (MRF) and thus train these estimators.

The contributions of this paper can be summarized as follows. We regard 3D reconstruction from a single image as a category modeling problem to overcome the difficulties in the reconstruction of irregular[1] shapes. To ensure a high learning efficiency and a wide application, this is the first attempt to learn category models from informally captured RGB-D images, and then apply the category model back to the informally captured RGB images for 3D reconstruction. We explore a new set of challenges raised by our choice of training and testing data, and provide a solution.

2. Related work

3D reconstruction is a large area in the field of computer vision. However, in this paper, we limit our discussion to 3D reconstruction from a single image. Many methods for single-view 3D reconstruction have strong assumptions for the image environment. For example, "shapes from shading" methods [23, 4, 3, 14] had special requirements for lighting conditions and textures. A large number of studies use the perspective principle for 3D reconstruction. They were typically based on the vanishing points in images, and therefore assumed that these images contained enough cues to extract vanishing points [24, 15]. Such methods were mainly applied to large-scale indoor and urban environments. In addition, some studies used the ground/vertical assumption to assist in vanishing point extraction [13, 2, 7, 8]. Saxena *et al.* [19] proposed to learn an MRF for 3D reconstruction in large-scale environments. Fouhey *et al.* [9] proposed to learn 3D primitives from RGB-D images for single-view reconstruction of indoor environment.

A number of methods relied on the assumption that the object structure in question had smooth surfaces. They usually extracted object-level structures from locally captured images that contained no perspective cues. For example, the Cheeger Set approach [18] assumed that the target ob-

[3]In our study, these objects are collected from RGB images (or RGB dimensions of the RGB-D images) using the trained category detectors. The 3D object structure is projected into the 2D shape afterwards.

ject had ball-like surfaces. Given the object contour and the object volume, it computed the 3D structure with the minimum surface area. However, such assumptions for object structure are usually not valid for the reconstruction of irregular[1] shapes. Therefore, many methods [25, 22, 21, 14] combined human interactions with these assumptions to guide the 3D reconstruction.

By and large, the lack of cues for the estimation of irregular[1] object structures is the main challenge for single-view 3D reconstruction. From this perspective, our research is related to the example-based methods [11, 10, 20]. They required a comprehensive dataset of 3D object samples that were captured in all poses from different viewpoints. The 3D structure knowledge of a target object was extracted via a large number of comparisons between the object and the samples in the dataset. Chen *et al.* [6, 5] learned 3D reconstruction knowledge from well built 3D object models. These 3D object models provided knowledge on structure deformation and the multi-view appearance. In this way, they learned to match the 2D shape of a given object to the 3D object models, and thus estimate the object's viewpoint, pose, and 3D structure.

In contrast, we apply much less supervision to idealize the concept of automatic "category modeling" in terms of training. Learning from informally captured RGB-D images without manual alignment saves great labor for preparation for 3D object models or a large 3D-example dataset. The challenges of object detection, rotation changes, and the estimation of 3D structure deformation are all involved in the training process. The trained category model is then applied back to the informally captured RGB images for 3D reconstruction, without example comparison.

3. Training of category detectors

Category detectors are not only trained from, but also applied to informally captured and unaligned images. The category models for 3D reconstruction are designed on the basis of category detectors. In our study, we use the graphical model defined in [26] as the category detector, considering the need for robustness to shape deformations and rotations in the informally captured images. We then apply unsupervised learning for graph matching [28] to mine the category detectors. In reality, we simply use the RGB channels of RGB-D images for training. The detector is trained to encode the pattern for the common objects in all the images. A set of key object parts are discovered to form the pattern for a category (see Fig. 3(c)).

Category detectors: As shown in Fig. 3(a), the image is represented as a complete attributed relational graph (ARG). In Fig. 3(b), continuous object edges (magenta) are discretized into line segments (black), and these line segments form the graph nodes. Thus the objects within it are represented as sub-graphs.

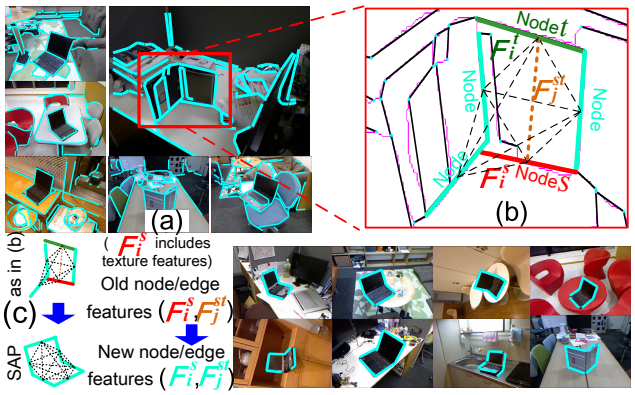

Figure 3. Object detectors. (a) Graph-based image representation. Object edge segments (cyan) form the graph nodes. (b) Notation for the graphical model. The initial graphical model consists of five nodes (colored lines) and edges (dotted lines) between them. (c) Learning the graphical model (category detector). The algorithm modifies (i) the initial graphical model into (ii) the SAP (G). The node set and node/edge features are modified so that 1) the node number (size) of G is maximized and 2) all the node/edge features represent common patterns in all the graphs.

The category detector is a graphical model in [26], which will be automatically trained to be the soft attributed pattern (SAP) among the target sub-graphs (or objects) in different images. Let graph G with node set V denote the model. G is described by a set of node and edge features. Let $\{\mathcal{F}_i^s\}$ and $\{\mathcal{F}_j^{st}\}$ $(i = 1, 2, ..., N_P, j = 1, 2, ..., N_Q)$ denote the i-th node feature for node s and the j-th edge feature for edge (s, t), respectively. Here, N_P and N_Q indicate the feature numbers for each node and edge, respectively. $\mathbf{F}_V = \{\mathcal{F}_i^s\} \bigcup \{\mathcal{F}_j^{st}\}$, $\forall s, t \in V$. Note that $\mathcal{F}_i^s$ includes the texture features on the terminals of lines segments. The category detector thus also contains the textural knowledge. (Please see [26] for details of the feature settings.)

Object detection: Object detection is achieved via graph matching. Given a target RGB image represented by graph G' with a node set V', graph matching is performed to compute the matching assignments between G and G'. Let $\mathbf{x}$ denote the label set, and let the label $x_s \in \mathbf{x}$ denote the matched node in G' for node s in G. Graph matching is formulated as the minimization of the following energy function w.r.t $\mathbf{x}$.

$$E(\mathbf{x}|\mathbf{F}_V,\mathbf{F}_{V'}) = \sum_{s\in V} P_s(x_s|\mathbf{F}_V,\mathbf{F}_{V'}) + \sum_{s,t\in V, s\neq t} Q_{st}(x_s,x_t|\mathbf{F}_V,\mathbf{F}_{V'}) \quad (1)$$

This is a typical formula for graph matching, where the functions $P_s(\cdot|\cdot,\cdot)$ and $Q_{st}(\cdot,\cdot|\cdot,\cdot)$ measure matching penalties (feature dissimilarity) between the two matched nodes and edges. These are generally defined using squared differences. *E.g.* $P_s(x_s|\mathbf{F}_V,\mathbf{F}_{V'}) = \mathbf{w}^T[\|\mathcal{F}_1^s - \mathcal{F}_1^{x_s^k}\|^2, ..., \|\mathcal{F}_{N_P}^s - \mathcal{F}_{N_P}^{x_s^k}\|^2]$. (Please see [28] for details.)

Learning graph matching with SAPs: As shown in Fig. 3(c), [28] proposed an algorithm for mining SAPs. The

SAP is a fuzzy attributed pattern, which describes the common sub-graphs in a set of ARGs, in which node and edge features have considerable variations. In other words, the category detector (*i.e.* the SAP) represents a set of key object parts (nodes) that frequently appear in training images of the entire category. The category detector is trained taking into account the robustness to texture variations and structure deformations.

To start the training process, this method requires people only to label the sub-graph of a single object for initializing the graphical model G. [28] is designed to modify G from the specific shape of the labeled object into the SAP among all graphs. Given a set of N training images, each image is denoted by $\{G'_k\}$ $(k = 1, 2, ..., N)$ with node set V'_k. $\mathbf{F}_{V'_k}$ are the attribute sets of G'_k, and $x^k_s \in \mathbf{x}^k \in \mathbf{X}$ indicates the node in V'_k matched by $s \in V$. The "frequency" of node s in G is defined as the average penalty for matching s to all the $\{G'_k\}$, as follows.

$$E_s(\hat{\mathbf{X}},\hat{\mathbf{F}}_V)=\frac{1}{N}\sum_{k=1}^{N}\Big[P_s(\hat{x}^k_s|\hat{\mathbf{F}}_V,\mathbf{F}_{V'_k})+\sum_{t\in V,t\neq s}Q_{st}(\hat{x}^k_s,\hat{x}^k_t|\hat{\mathbf{F}}_V,\mathbf{F}_{V'_k})\Big]$$

If $E_s(\hat{\mathbf{X}}, \hat{\mathbf{F}}_V)$ is less than a given threshold τ, node s is regarded as a frequently appearing part; otherwise not. Hence, the goal of learning graph matching is to train the node set V and attributes $\mathbf{F}_V$ of model G, in such a manner that 1) all nodes of G frequently appear in graphs $\{G'_k\}$ and 2) the size of G is maximized.

$$\begin{aligned}&\max_{V} \|V\|\\ \text{s.t.}\quad &(\hat{\mathbf{F}}_V, \hat{\mathbf{X}}) = \underset{\mathbf{F}_V,\mathbf{X}=\{\mathbf{x}^k\}}{\operatorname{argmin}} \sum\nolimits_{k=1}^{N} E(\mathbf{x}^k|\mathbf{F}_V, \mathbf{F}_{V'_k}); \quad (2)\\ &\forall s \in V, \quad E_s(\hat{\mathbf{X}}, \hat{\mathbf{F}}_V) \leq \tau.\end{aligned}$$

[28] proposes an EM framework to solve (2).

4. Learning 3D reconstruction

We introduce the preparation of training samples, model learning, and the application of 3D reconstruction in the following three subsections.

4.1. Object sample collection & pose normalization

The preparation of training samples involves two steps, *i.e.* object sample collection and 3D pose normalization. In the first step, we use the trained category detector to collect RGB-D samples from informally captured RGB-D images for training. Each training sample contains its 2D shape and the 3D coordinates of each 2D pixel within it. The 3D coordinates are measured in the global coordinate system of each RGB-D image, rather than a coordinate system *w.r.t* the object. Therefore, in the second step, we compute the 3D pose of the object, and thereby normalize the 3D coordinates into the object coordinate system.

Object sample collection: We use the category detector to match a set of key object parts (graph nodes) in each image by applying (1). However, as mentioned above, the key parts consist of line segments on the object, and do not comprise the entire object body. Therefore, we first recover the entire body area of the object from the detected line segments, before the further learning. Actually, a number of methods for interactive[4] object segmentation are oriented to this application. Nevertheless, in order to make a reliable evaluation of the proposed 3D reconstruction method, we need to simplify the whole system and thereby avoid bringing in uncertainties related to object segmentation. Consequently, given the key line segments of the object, we roughly estimate its body area as the convex hull of the line terminals (see Fig. 4(c)).

3D pose normalization: We define centers and 3D poses for the object samples in the RGB-D images, thus constructing a relative coordinate system for each sample, as shown in Fig. 4(f,bottom). We then project 3D point clouds of the objects onto these coordinate systems, as the ground truth of their 3D structures.

Fig. 4(e) shows the notation. The center of an object is defined as the mean of the 3D coordinates of its key parts $c = \sum_{1\leq s\leq m} \mathbf{p}_s/m$, where m is the part (node) number and $\mathbf{p}_s$ denotes the center coordinates of part s. We define the orientations of the three orthogonal coordinate axes $<\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3>$ as follows.

$$\mathbf{v}_1 = u(\mathbf{v}'_1 + \mathbf{v}'_2),\ \mathbf{v}_2 = u(\mathbf{v}'_1 - \mathbf{v}'_2),\ \mathbf{v}_3 = \mathbf{v}'_1 \times \mathbf{v}'_2 \quad (3)$$

where function $u(\cdot)$ normalizes a vector to a unit vector, $\mathbf{v}'_1 = u(\sum_{1\leq s<t\leq m} u(\mathbf{p}_t - \mathbf{p}_s))$, $\mathbf{v}'_2 = u(\sum_{1\leq s<t\leq m}(\mathbf{o}_t \times \mathbf{o}_s))$, and $\mathbf{o}_s$ denotes the 3D orientation of line segment s.

4.2. Model learning

We train a category model from RGB-D object samples to estimate the pixel-level 3D reconstruction. Essentially, even if we are given a prior 3D structure for a category, 3D reconstruction from a single image still has great challenges. There are two general hypotheses, *i.e.* object rotations and structure deformations, to interpreting a specific 2D object shape in 3D reconstruction. Each hypothesis can independently estimate the 3D object structure from the 2D shape (*e.g.* computing the rotation or 3D deformation of the prior 3D structure that best fits the contour of the 2D shape).

Obviously, the 3D reconstruction needs to combine the both hypotheses. Unlike [5, 6] using well built 3D object models to provide or train prior regularity of structure deformation, we need to train the category model to simultaneously identify the effects of object rotations and structure deformations from 2D shapes, without prior knowledge. This greatly increases the challenge.

[4] The key object parts can be labeled as the foreground.

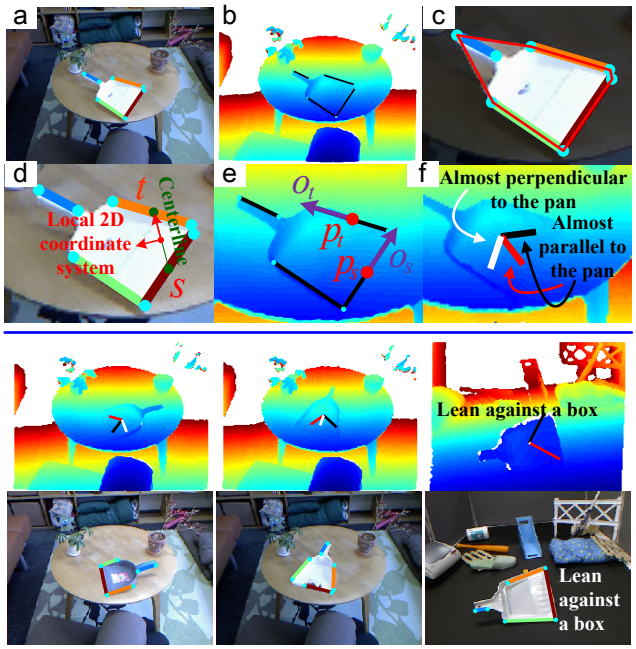

Figure 4. Some steps in learning 3D reconstruction. (a,b) Visualization of the detected key object parts in the RGB and depth image. (c) Recovery of the object body area. (d) Notation for the local 2D coordinate system. (e) Notation for determining the local 3D coordinate system. (f) Three axes of the local 3D coordinate system. (bottom) Local 3D coordinate systems of three objects.

Fig. 5 shows the basic design of the category model. The category model handles both the effects and directly projects each 2D pixel into a 3D space. Considering the model's robustness to shape deformation, we use a set of local 2D coordinate systems to simultaneously localize each pixel inside the 2D shape. Each local 2D coordinate system is constructed using each pair of key object parts. For each point in every 2D coordinate system, we train a local regressor to estimate its 3D coordinates from its point features.

The point features are designed to represent the spatial relationship between the point and key objects parts, as well as the 2D shape of these key object parts. Thus, the point features contain sufficient cues for object rotations and structure deformation to guide the estimation of the 3D coordinates of each 2D point. For point $\mathbf{p}$ in the local 2D coordinate system *w.r.t* key parts s and t of a given object sample, its point features are defined as $\theta_{st}^{\mathbf{p}} = [\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, 1]^T$. ϑ_1 and ϑ_2 denote the lengths of s and t, respectively, and ϑ_3 denotes the angle between s and t. ϑ_4 and ϑ_5 measure the distance between $\mathbf{p}$ and the line segments s and t, respectively. ϑ_1, ϑ_2, ϑ_4, and ϑ_5 are normalized by the centerline length between s and t (Fig. 4(d)).

Then, the local regressor $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ is modeled as a linear regression of the 3D coordinates of $\mathbf{p}$ using $\theta_{st}^{\mathbf{p}}$:

$$\mathcal{F}_{st}^{\mathbf{p}}(\theta_{st}^{\mathbf{p}}) = (\theta_{st}^{\mathbf{p}})^T \mathbf{M}_{st}^{\mathbf{p}} \tag{4}$$

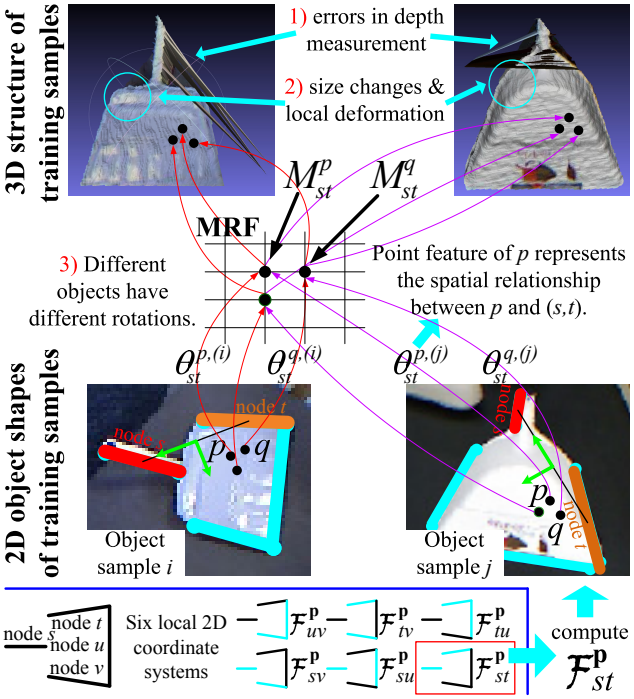

Figure 5. Learning 3D reconstruction. The category model uses different local 2D coordinate systems (bottom) to localize the 2D points, and estimates the 3D structure in these coordinate systems. Given a specific 2D coordinate system *w.r.t* each pair of key parts s and t, we generate a MRF to train the parameter $\mathbf{M}_{st}^{\mathbf{p}}$ for the local regressor $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ in each different point $\mathbf{p}$. $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ uses the point feature $\theta_{st}^{\mathbf{p},(i)}$ in object sample i to estimate 3D coordinates of $\mathbf{p}$, thus overcoming i's specific object rotations, structure deformation, and depth errors.

where matrix $\mathbf{M}_{st}^{\mathbf{p}}$ is the parameter for the function $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$.

The local regressor $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ is the basic element of the category model. The category model is formulated as a linear combination of these local regressors, as follows.

$$\mathcal{Y}(x) = \frac{1}{Z} \sum_{1 \le s < t \le m} w_{st}^{\mathbf{p}} \mathcal{F}_{st}^{\mathbf{p}}(\theta_{st}^{\mathbf{p}})|_{\mathbf{p}=\mathbf{p}_{st}(x)} \tag{5}$$

where $Z = \sum_{1 \le s < t \le m} w_{st}^{\mathbf{p}=\mathbf{p}_{st}(x)}$. x is a pixel inside the 2D object shape, and $\mathcal{Y}(x)$ estimates its 3D coordinates. s and t indicate two key parts of the object; m denotes the key part number. Given the local coordinate system constructed using s and t, function $\mathbf{p}_{st}(x)$ determines the 2D position of x in this coordinate system. Thus, function $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ is the local regressor of 3D coordinates for point $\mathbf{p}$ in the coordinate system of s and t, as mentioned above. $w_{st}^{\mathbf{p}}$ measures the reliability of the local regressor $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ and is regarded as its weight.

Local 2D coordinate system: As shown in Fig. 4(d), given the line segments of each pair of key parts s and t ($1 \le s < t \le m$), we generate a local 2D coordinate system. The line segment connecting the center of s to that of t is

called the centerline between s and t. The origin of the local coordinate system *w.r.t* s and t is defined as the center of the centerline. We take the centerline's orientation as the orientation of the first coordinate axis, and determine the orientation of the second axis using the right-hand rule. The unit vector of the first axis is normalized using the centerline length, and that of the second axis is normalized using the average segment length of s and t.

Model learning based on a continuous MRF: In the local 2D coordinate system determined by the key parts s and t, the local regressors of the neighboring positions are connected to form an MRF. We generate the following objective function to learn the parameters of the regressors.

$$\underset{\mathbb{M}_{st}}{\operatorname{argmax}} \sum_{\mathbf{p}\in\mathcal{V}} \phi_1(\mathbf{M}_{st}^{\mathbf{p}}) + \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{E}} \phi_2(\mathbf{M}_{st}^{\mathbf{p}}, \mathbf{M}_{st}^{\mathbf{q}}) \quad (6)$$

where $\mathbb{M}_{st} = \{\mathbf{M}_{st}^{\mathbf{p}} | \mathbf{p} \in \mathcal{V}\}$; $\mathcal{V}$ and $\mathcal{E}$ denote node and edge sets of the MRF, respectively.

In the MRF, the unary term $\phi_1(\cdot)$ measures the similarities between the ground truth and the 3D coordinates estimated by the regressor for each point $\mathbf{p}$.

$$\phi_1(\mathbf{M}_{st}^{\mathbf{p}}) = \frac{1}{\|S\|} \sum_{i\in S} \exp(-\eta \|\mathcal{F}_{st}^{\mathbf{p}}(\theta_{st}^{\mathbf{p},(i)}) - y_i^{\mathbf{p}}\|) \quad (7)$$

where S is the set of training samples. For each object sample i, $\theta_{st}^{\mathbf{p},(i)}$ and $y_i^{\mathbf{p}}$ denote the point feature and true 3D coordinates of $\mathbf{p}$, respectively. η is a scaling parameter.

The pairwise term $\phi_2(\cdot,\cdot)$ is a smoothing term between neighboring points.

$$\phi_2(\mathbf{M}_{st}^{\mathbf{p}}, \mathbf{M}_{st}^{\mathbf{q}}) = -\lambda(\mathbf{M}_{st}^{\mathbf{p}} - \mathbf{M}_{st}^{\mathbf{q}})^2 \exp[-\alpha(U^{\mathbf{p}} + U^{\mathbf{q}})^2]$$
$$U^{\mathbf{p}} = \frac{1}{\|S\|} \sum_{i\in S} \max_{(\mathbf{p},\mathbf{p}')\in\mathcal{E}} \|y_i^{\mathbf{p}} - y_i^{\mathbf{p}'}\| \quad (8)$$

where $U^{\mathbf{p}}$ use true 3D structures of the training samples to measure the discontinuity around point $\mathbf{p}$. Object areas with high discontinuity, such as object edges, have low weights for smoothing. λ and α are scaling parameters. We use the MCMC method to solve the continuous MRF.

Finally, when the regressor $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ has been trained, the weight for $\mathcal{F}_{st}^{\mathbf{p}}(\cdot)$ in (5) is computed as

$$w_{st}^{\mathbf{p}} = \frac{1}{\|S\|} \sum_{i\in S} \exp[-\eta(\mathcal{F}_{st}^{\mathbf{p}}(\theta_{st}^{\mathbf{p},(i)}) - y_i^{\mathbf{p}})^2] \quad (9)$$

4.3. 3D reconstruction based on category models

As shown in Fig. 1, we use a trained category model to directly perform 3D reconstruction on testing RGB images. We first use the trained category detector to collect target objects in the RGB images, and simultaneously determine the body area of the objects (see Section 4.1). We then use the category model in (5) to estimate the 3D coordinates for each pixel inside the object.

5. Experiments

5.1. Data

We use the category dataset of Kinect RGB-D images [1, 26], which is published as a standard RGB-D object dataset[5] for the learning of graph-matching-based models, such as [26, 27]. These RGB-D images depict cluttered scenes containing objects with different textures and rotations. Four categories—*notebook PC*, *drink box*, *sprayer*, and *dustpan*—in this dataset contain a sufficient number of RGB-D objects for training and are thus chosen in the experiments.

5.2. Implementation details

Training of category detectors: We train the SAP (category detector) from a set of large graphs (informally captured images). Parameter τ controls the graph size of the SAP, or in other word, the number of key object parts contained by the category detector. When we set a higher value for τ, we can extract more key parts for a category, but the key parts are less reliable in part detection. To simplify the learning process, we require the SAPs (detectors) to have four nodes (key parts) for all the four categories. We try different values of τ in the training process, until the category detector satisfies this requirement. Hence, we can detect four key parts for each object, and the rotation and deformation of the object are determined by the complex spatial relationship between the four key parts.

Learning 3D reconstruction: As shown in Fig. 5, in each local 2D coordinate system, we divide the entire object area comprising of 50×50 grids to identify different 2D points. A local regressor with parameter $\mathbf{M}_{st}^{\mathbf{p}}$ is generated for each grid $\mathbf{p}$, and we thus use the local regressors construct the MRF. For the application of 3D reconstruction, many pixels in the 2D shape are not accurately localized in the grid centers. To achieve accurate pixel-level 3D reconstruction, we interpolate the regressor parameter value for each 2D pixel from the trained parameters $\mathbf{M}_{st}^{\mathbf{p}}$ of the nearby grids. We set parameter α as 0.1 for all the categories in model learning, and use different values of λ and η to test the system.

5.3. Quantitative comparison

Competing method: We compare our approach with conventional methods for 3D reconstruction from a single image. As discussed in Section 2, most related techniques have their own specific assumptions for the target image and are thus not suitable for 3D reconstruction of irregular[1] shapes. From this viewpoint, example-based 3D reconstruction is close to our method, but conventional techniques are

[5]Compared to other RGB-D datasets *i.e.* [16, 12], this is one of the largest RGB-D object datasets, and reflects challenges of graph matching.

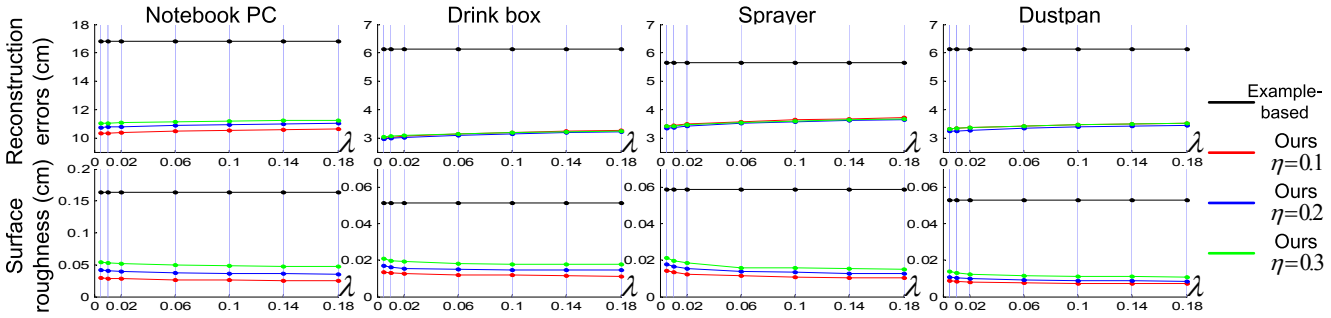

Figure 6. 3D reconstruction comparison. The proposed method exhibits smaller reconstruction errors and surface roughness than the example-based method. The learning of category models is not sensitive to different settings of parameters λ and η. λ and η are not involved in the example-based method.

hampered by the use of informally captured RGB-D images for training. Nevertheless, considering the challenges in the training dataset, we design a competing method that achieves a rough idea of the example-based 3D reconstruction, as follows. First, given a RGB image, we use the category detector trained in Section 3 to detect the target object in it, as well as its key object parts. We then randomly select an RGB-D object from the training sample set as the example for 3D reconstruction. The 3D coordinate estimation is based on (5), just like the proposed method. However, without training, the weight for local regressors ($w_{st}^{\mathbf{p}}$) is set to 1. The local regressors $\mathcal{F}_{st}^{\mathbf{p}}$ is redefined as a constant, *i.e.* the 3D coordinate for point $\mathbf{p}$ on the example, rather than a function in (4) *w.r.t.* point features $\theta_{st}^{\mathbf{p}}$.

Pixel-level evaluation: We evaluate the reconstruction errors and surface roughness (non-smoothness) of the proposed approach at the pixel level. Reconstruction errors are widely used to evaluate 3D reconstruction. We manually prepare the ground the truth of the 3D structure for each object using the Kinect measurement. We measure the distance between the estimated 3D coordinates of each pixel and its true coordinates, as the pixel reconstruction error. We define the reconstruction error of an object as the average reconstruction error of its constituent pixels. The reconstruction error of a category is defined as the average reconstruction error of all its objects. The other evaluation metric is the surface roughness. The roughness of pixel $\mathbf{p}$ is measured as $r_{\mathbf{p}} = \|\mathcal{Y}^{\mathbf{p}} - \frac{1}{N(\mathbf{p})}\sum_{\mathbf{q}\in N(\mathbf{p})}\mathcal{Y}^{\mathbf{q}}\|$, where pixel $N(\mathbf{p})$ is the set of 4-connectivity neighboring pixels of $\mathbf{p}$, $\mathcal{Y}^{\mathbf{p}}$ denotes the estimated 3D coordinates of pixel $\mathbf{p}$ on the object. Just like the reconstruction error, the object roughness is the average of the pixel roughness, and the surface roughness of a category is defined as the average of object-level roughness.

The evaluation is achieved via cross validation. Just as in [26, 27], we randomly select $2/3$ and $1/3$ of the RGB-D images in the entire dataset as a pair of sample pools for training and testing, respectively. Each pair of sample pools can train a category model, and provide values of the reconstruction error and surface roughness for the category. We prepare different pairs of training and testing pools and compute the average performance by cross validation. Fig. 6 and Fig. 7 show the comparison of 3D reconstruction performance. Without sufficient learning, the competing method cannot correctly estimate structure deformation for objects, and suffers from the unavoidable errors in Kinect's depth measurement.

6. Conclusions

In this paper, we proposed to learn a category model for single-view 3D reconstruction, which encoded knowledge for structure deformation, texture variations, and rotation changes. The category model was both trained from and applied to the informally captured images. Experiments demonstrated the effectiveness of the proposed method.

For the training of category detectors, we used various node and edge features. These included texture features on local image patches. Thus, the texture information contributes to the extraction of key object parts. However, textures contribute much less in further 3D reconstruction, as many daily-use objects (*e.g.* the *drink box*) are designed with different textures that are unrelated to object structures for commercial purposes. The spatial relationship between key object parts was, therefore, taken as more reliable cues for 3D reconstruction and used in model learning.

We did not apply object segmentation and thus avoided the uncertainties related to it, so as to simplify the system and enable a reliable evaluation. The category model did not performed so well on object edges as in other object area due to the time-of-the-flight errors and calibration errors between 3D points and RGB images.

References

[1] *Category Dataset of Kinect RGBD Images, http://sites.google.com/site/quanshizhang*. 6

[2] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. *In* ECCV, 2008. 2

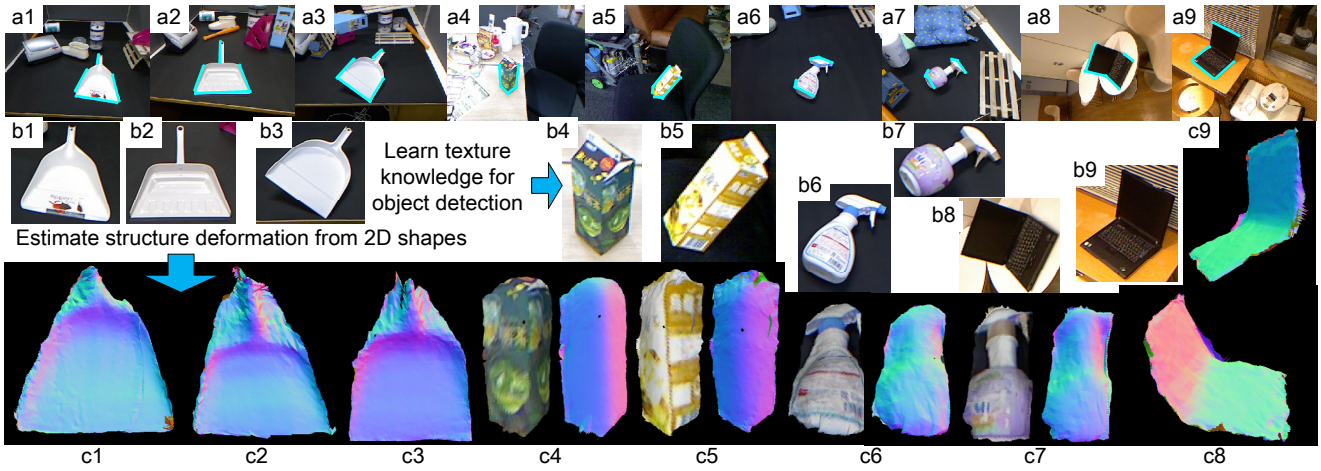

Figure 7. 3D reconstruction performance. (a1–9) Object detection (cyan lines) in RGB images. (b1–9) Detailed view of the target objects. (c1–9) Reconstruction results of the proposed method.

[3] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. *In* ECCV, 2012. 2

[4] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. *In* CVPR, 2012. 2

[5] Y. Chen and R. Cipolla. Single and sparse view 3d reconstruction by learning shape priors. *In* CVIU, 115(5):586–602, 2011. 3, 4

[6] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. *In* ECCV, 2010. 3, 4

[7] E. Delage, H. Lee, and A. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. *In* Robotics Research, 2006. 2

[8] E. Delage, H. Lee, and A. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. *In* CVPR, 2006. 2

[9] D. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. *In* ICCV, 2013. 2

[10] T. Hassner and R. Basri. Example based 3d reconstruction from single 2d images. *In* CVPR workshop, 2006. 3

[11] T. Hassner and R. Basri. Single view depth estimation from examples. *In* CoRR, abs/1304.3915, 2013. 3

[12] E. Herbst, X. Ren, and D. Fox. Rgb-d object discovery via multi-scene analysis. *In* IROS, 2011. 6

[13] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. *In* ICCV, 2005. 2

[14] K. Karsch, Z. Liao, J. Rock, J. T. Barron, and D. Hoiem. Boundary cues for 3d object shape recovery. *In* CVPR, 2013. 2, 3

[15] K. Köser, C. Zach, and M. Pollefeys. Dense 3d reconstruction of symmetric scenes from a single image. *In* DAGM, 2011. 2

[16] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. *In* Proc. of IEEE International Conference on Robotics and Automation (ICRA), pages 1817–1824, 2011. 6

[17] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. *In* ICML, 2012. 1

[18] M. R. Oswald, E. Töppe, and D. Cremers. Fast and globally optimal single view reconstruction of curved objects. *In* CVPR, 2012. 2

[19] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *In* PAMI, 31(5):824–840, 2009. 2

[20] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Gool. Shape-from recognition: Recognition enables meta-data transfer. *In* CVIU, 2009. 3

[21] E. Toppe, C. Nieuwenhuis, and D. Cremers. Relative volume constraints for single view 3d reconstruction. *In* CVPR, 2013. 3

[22] E. Töppe, M. R. Oswald, D. Cremers, and C. Rother. Image-based 3d modeling via cheeger sets. *In* ACCV, 2010. 3

[23] A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3d reconstruction of locally textured surfaces. *In* PAMI, 34(6):1118–1130, 2012. 2

[24] T. Xue, J. Liu, and X. Tang. Symmetric piecewise planar object reconstruction from a single image. *In* CVPR, 2011. 2

[25] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz. Single view modeling of free-form scenes. *In* CVPR, 2002. 3

[26] Q. Zhang, X. Song, X. Shao, R. Shibasaki, and H. Zhao. Category modeling from just a single labeling: Use depth information to guide the learning of 2d models. *In* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 193–200, 2013. 1, 2, 3, 6, 7

[27] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Learning graph matching for category modeling from cluttered scenes. *In* ICCV, 2013. 6, 7

[28] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Attributed graph mining and matching: An attempt to define and extract soft attributed patterns. *In* CVPR, 2014. 2, 3, 4