Category Modeling from just a Single Labeling: Use Depth Information to Guide the Learning of 2D Models

Quanshi Zhang†, Xuan Song†, Xiaowei Shao†, Ryosuke Shibasaki†, Huijing Zhao‡
Center for Spatial Information Science, University of Tokyo†
Key Laboratory of Machine Perception (MoE), Peking University‡
{zqs1022, songxuan, shaoxw, shiba}@csis.u-tokyo.ac.jp zhaohj@cis.pku.edu.cn

Abstract

An object model base that covers a large number of object categories is of great value for many computer vision tasks. As artifacts are usually designed to have various textures, their structure is the primary distinguishing feature between different categories. Thus, how to encode this structural information and how to start the model learning with a minimum of human labeling become two key challenges for the construction of the model base. We design a graphical model that uses object edges to represent object structures, and this paper aims to incrementally learn this category model from one labeled object and a number of casually captured scenes. However, the incremental model learning may be biased due to the limited human labeling. Therefore, we propose a new strategy that uses the depth information in RGBD images to guide the model learning for object detection in ordinary RGB images. In experiments, the proposed method achieves superior performance as good as the supervised methods that require the labeling of all target objects.

1. Introduction

Category model learning is a classical area in the field of computer vision. In this paper, we return to two basic questions. First, for many regular-shape artifacts, it is the structure, rather than the texture, that determines their functions and categories, so *how can we obtain structural knowledge for each object category?* Second, if we idealize the spirit of semi-supervised learning, *can we learn a category model from the minimum labeling (only one labeled object) and casually captured image sample pools?* Here, we use the phrase "casually captured" to describe the loose requirement that training samples do not need to be hand-cropped or carefully aligned, and thus can be easily collected by ordinary people in their daily life. In casually captured image sample pools, the target objects within an image are usually small with large texture variations and various rigid transformations, even including roll rotations (Fig. 1). The minimum labeling meets the efficiency requirement for the construction of a category model base. These category models are expected to be able to detect objects in complex scenes.

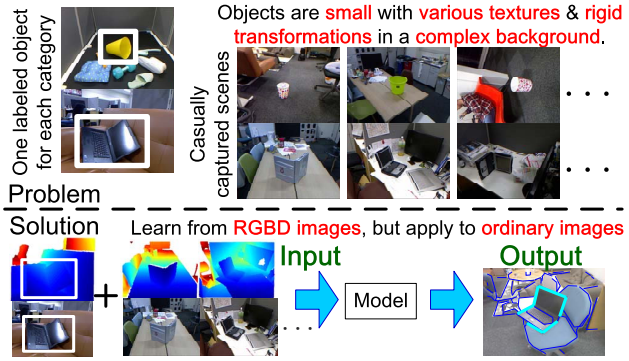

Figure 1. How can a structure-based category model be learnt from one labeled object and a number of casually captured scenes[2]? Accurate part correspondences between target objects are necessary for training the structure-based model, but purely image-based object detection and matching are hampered by texture variations and rigid transformations of objects in these scenes. Therefore, we learn models from RGBD images, but apply them to object detection in ordinary RGB images.

However, the model learning is caught in a dilemma. On the one hand, training the structure-based model requires the collection of small target objects in casually captured scenes, as well as the extraction of part correspondences between these objects. On the other hand, without training, object detection and matching based on the only labeled object is hampered by intra-category texture variations and various rigid transformations, which represent a great challenge for state-of-the-art algorithms. Worse still, bias and errors in object collection in the initial learning steps will affect subsequent steps, and be accumulated into a significant model bias.

Fortunately, the invention of the Kinect [1] has made

[2]The detail definition of the "casually captured scenes" is presented in the first paragraph of Section 1.

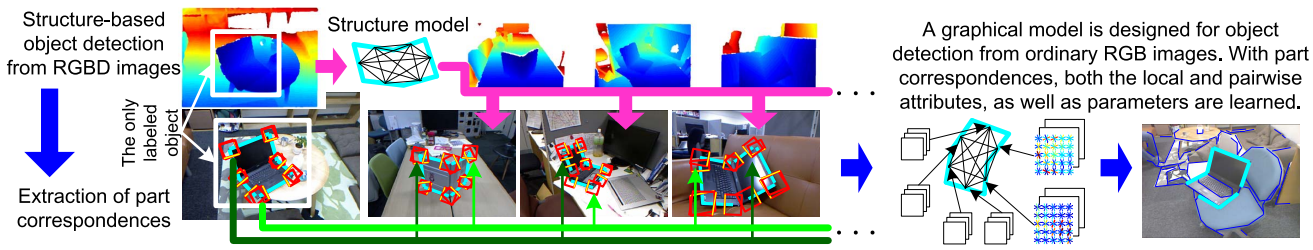

Figure 2. Flowchart of the proposed method. We use the 3D structure of the labeled object to match other objects in RGBD images (purple arrows). We then use the 3D part correspondences (green arrows) to train a model for object detection in ordinary RGB images.

instance-based object detection plausible. The Kinect RGB-D images provide explicit spatial structures of objects that are robust to variations in texture, 2D scale, and viewpoint. In many cases, the 3D structure of a single object is discriminative enough for category detection. Thus, we propose a different model learning strategy in which *we can train the model from RGBD images, and then apply the model to category detection in ordinary RGB images* (Fig. 1). At first, we use structure-based 3D matching to collect objects from RGBD images, simultaneously obtaining part correspondences, in spite of texture variations. Thus, a local codebook of visual words can be learned for each part of the object. The part correspondences in the 3D space are also used to train the 2D structural knowledge in the category model, as shown in Fig. 2. In this way, we use more reliable 3D matching results to guide the learning of not-so-discriminative image-based models, in order to overcome the bias problem in the incremental model learning.

To achieve this learning strategy, we propose a novel graphical model that utilizes an object's edges as a new and concise representation of its structures. Object edges have a stronger relationship than textures to the overall object structure, particularly where large texture variations exist. In this graphical model, we design different attributes to guide both the collection of 3D objects from RGBD images and the training of category models.

Both the 3D object collection and the category-model-based object detection are achieved by graph matching. Conventional algorithms for learning graph matching [2, 3, 4] have focused on training the weights of different graphical attributes, given a template graph (the category model) and multiple target graphs. In contrast, we train the category model by extending the method proposed by Leordeanu *et al.* [2] to estimate the general prototype of model attributes and eliminate the specificity in the labeled object.

The contributions of this paper can be summarized as follows. Facing challenges in the semi-supervised learning of visual models, we propose, for the first time, to use only one labeled object to start learning the structure knowledge from casually captured scenes. We apply the novel strategy—using objects collected from RGBD images to train the RGB-image-oriented model, thus avoiding possible bias problems caused by texture variations and various rigid transformations. A new type of graphical model based on object edges is designed as a concise representation of object structures in RGB and RGBD images.

2. Related Work

Object detection: Texture variations, object rotations, and the use of object structures make the task of object detection a great challenge. Bag-of-words models [5] have exhibited a good performance in image retrieval and recognition without using structural information, and the HOG and silhouette templates[6, 7] have been widely used to represent global structures on the image plane. Later, Hough-style methods [8, 9] were developed as a sophisticated supervised way of encoding the spatial relationship between object parts. [10, 11] proposed the direct use of a 3D model to detect objects and estimate their poses in images. In addition, [12, 13] have used object appearances observed from multiple viewpoints to learn the 3D structure in a supervised manner. Recently, RGBD images made object detection much easier [14, 15, 16], and even the structure discovery [17] or segmentation of indoor environments [18, 19] produced object-level results.

However, in this research, a single labeled object only provides its specific 2D structure and appearance observed from one viewpoint. In this case, the graph matching has the ability to detect objects with various scales and rotations, an approach that has been widely used [20, 21, 22]. Nevertheless, 2D structures of artifacts are not robust to viewpoint changes. Thus, we use the graphical model based on the 3D structure to collect training samples for model learning.

Model learning: We limit our discussion to unsupervised and semi-supervised methods, and analyze them with a view to the construction of a category model base.

The requirement of learning from a single labeling makes this research related to one-shot learning [23]. However, we focus on the extraction of the exact structural model from casually captured scenes, rather than the observing probability of patch textures.

Unsupervised object discovery (reviewed by [24]) was a classical achievement of object-level knowledge mining. Most methods used bag-of-words models [5] for category representation, and others [25, 26, 27, 28] detected repetitive objects with the similar appearance in the environment.

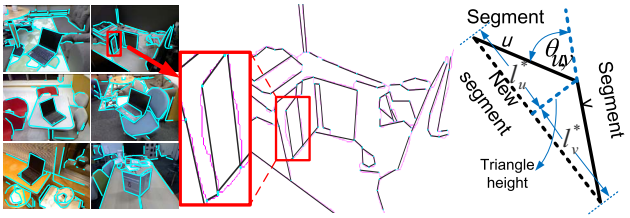

Figure 3. Edge segmentation and illustration of variables.

[29] manually cropped and aligned target objects in images for training, whereas [30, 25, 31] used unsupervised segmentation to generate object candidates, which relied on the foreground-background discrimination.

In contrast to the conventional learning of all categories from a large sample pool, Li *et al.* [5] and Grauman *et al.* [32] proposed semi-supervised learning and active learning to collect objects using an image search engine to sift the raw images. This was found to be a more efficient ways of constructing a category model base.

However, most of the above methods rely on object textures being highly similar, and are thus sensitive to the texture variations of many artifacts. Furthermore, the minimum labeling requirement for model base construction worsens the problem of texture variations. Hence, we focus on structural knowledge and use the depth information in RGBD images to avoid large errors in sample collection.

3. Graphical model of object edge segments and graph matching

Considering the need for robustness to viewpoint variation and roll rotations, we use a graphical model to encode the local and pairwise attributes of the object structure, thus achieving object detection via graph matching. In contrast to conventional studies based on POI in images, or voxels or surfaces [33, 34] in point clouds, we consider the edges of an object as basic elements of their structures. Edges are detected in RGB images using [35] and then discretized into line segments as the graph nodes, as shown in Fig. 3. The concise edge-based structure representation avoids the high computational overhead of matching.

Using edge segments in the only labeled object, we construct a complete undirected graph G as the initial category model, in which parameters will be refined via learning. Given a target scene, we generate a target graph, denoted by G'. The local attributes of vertex i and pairwise attributes of edge ij in G are denoted by $\mathbf{f}_i$ and $\mathbf{f}_{ij}$, respectively. We use a matching matrix $\mathbf{y}$ by $y_{ii'} \in \{0,1\}$ to define the matching assignments between G and G'. $y_{ii'} = 1$ if node i in G maps to node i' in G', otherwise $y_{ii'} = 0$. We set $\sum_{i'} y_{ii'} = 1$ for all i. Thus, the general idea of graph matching is to estimate the best matching assignments as:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, \mathcal{C}, \quad \mathcal{C} = \Big[\sum_{ii'} \rho_{ii'} y_{ii'} + \sum_{ii'jj'} \rho_{ii'jj'} y_{ii'} y_{jj'}\Big] \quad (1)$$

where $\rho_{ii'}$ and $\rho_{ii'jj'}$ are the compatibility for the unary assignment $i \to i'$ and the pairwise assignment $ij \to i'j'$, respectively. These are determined by graph attributes:

$$\rho_{ii'} = \Phi_1(\mathbf{f}_i, \mathbf{f}_{i'}; \mathbf{w}_1), \quad \rho_{ii'jj'} = \Phi_2(\mathbf{f}_{ij}, \mathbf{f}_{i'j'}; \mathbf{w}_2) \quad (2)$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are parameter weightings for attributes.

In our study, some parts of the target objects in the casually captured scenes may be occluded, so some model nodes should not be matched. We use one-to-none matchings to model this case, and thus add a new matching choice—*none*—that is organized as a node in G':

$$\rho_{i,none} = \kappa E(\rho_{ii'}), \quad \rho_{i,none,jj'} = \rho_{ii'j,none} = \kappa E(\rho_{ii'jj'}) \quad (3)$$

where κ ($= 1$, here) controls the matching priority of *none*.

Besides, many-to-one matchings should be avoided, as they introduce errors to the learning of pairwise attributes between those multiple nodes. Considering that the compatibility in (2) is positive in our study, we modify unary compatibility as $\rho_{ii'jj'} = -1$ if and only if $i' = j'$.

By designing different local and pairwise attributes, the graphical model can be used for both object collection from RGBD images and object detection in ordinary images.

Edge segmentation: Edge segmentation is achieved via a local growth strategy. Each pair of neighboring edge points is initialized as a line segment, and then neighboring segments are gradually merged into longer and straighter lines. In particular, edge segments in RGBD images are mapped to the 3D space to represent the 3D object structure.

Local non-smoothness exists on the extracted edges due to low image quality and texture variations. Thus, we design a penalty metric to guide the merging process for reliable segmentation. Suppose neighboring segments u and v are merged into a longer segment, as illustrated in Fig. 3. The penalty of their supplementary angle $\theta_{u,v}$ is calculated as:

$$Pen^{\text{angle}}_{u,v} = \theta_{u,v}(1 - U_{u,v}) \quad (4)$$

where, $U_{u,v} = e^{-\tau \min\{l^*_u, l^*_v\}}$ measures the unreliability of the angle measurement, as angles between shorter segments are more sensitive to local perturbations; τ ($= 0.2$, here) controls the decrease speed, and l^*_u and l^*_v are the projected lengths of segments u and v on the new segment.

As the orientation measurement of long segments suffers less from local non-smoothness than that of short ones, the length penalty is designed to avoid transferring the orientation unreliability from the short to the long segment when merging them:

$$Pen^{\text{length}}_{u,v} = \frac{l^*_u}{l^*_u + l^*_v} \log \frac{l^*_u}{l^*_u + l^*_v} + \frac{l^*_v}{l^*_u + l^*_v} \log \frac{l^*_v}{l^*_u + l^*_v} \quad (5)$$

The total penalty is calculated as follows:

$$Pen_{u,v} = Pen^{\text{angle}}_{u,v} + \eta Pen^{\text{length}}_{u,v} \quad (6)$$

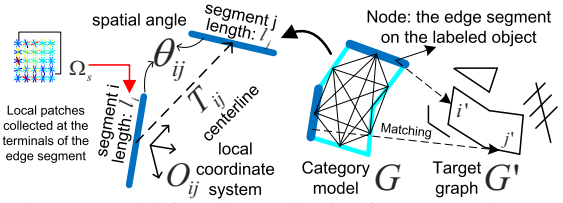

Figure 4. Model for object collection from RGBD images.

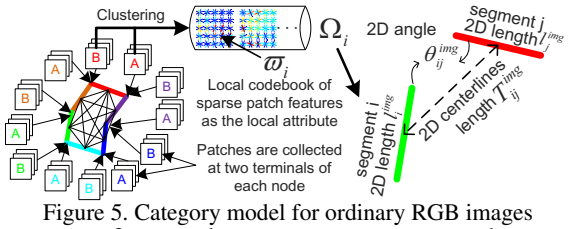

Figure 5. Category model for ordinary RGB images

where η ($= 0.5$, here) is a weighting for the two penalty metrics. Pairs with lower penalty scores are merged earlier. The stopping criterion is that for each merge, the height of the triangle consisting of old and new segments should not be more than six pixel units (Fig. 3). Finally, lines longer than 15 pixel units are selected as reliable segments.

3.1. Model for object collection from RGBD images

The proposed graphical model, as a paradigm, is adapted for collecting objects in RGBD scenes and simultaneously extracting the correspondences of local patches between objects for further learning. The notation for this model is illustrated in Fig. 4.

Spatial length: The spatial length denoted by l_i is taken as a local attribute. The length penalty for assignment $i \to i'$ can be calculated as $|\log \frac{l_{i'}}{l_i}|$. Thus, the compatibility of length attributes is calculated as:

$$P_{ii'}^{length} = e^{-|\log l_i - \log l_{i'}|/\beta} \tag{7}$$

where β ($= 2$, here) controls the deformability level.

Patch features: Two local patches are collected at the terminal points of each edge segment and normalized to their *right* orientations. Their HOG features are also used as local attributes (details follow in Section 4.1). The HOG features extracted from two patches of node i in G are denoted by $\Omega_i = \{\varpi_i^A, \varpi_i^B\}$. We calculate the compatibility of patch features via a Gaussian distribution:

$$P_{ii'}^{patch} = \mathcal{G}([dist(\varpi_i^A, \Omega_{i'}), dist(\varpi_i^B, \Omega_{i'})]^T \quad |\mu = 0, (\sigma^{patch})^2\mathbf{I}) \tag{8a}$$

$$dist(\varpi_i, \Omega_{i'}) = \min_{\varpi_{i'} \in \Omega_{i'}} \|\varpi_i - \varpi_{i'}\|_2 \tag{8b}$$

where $\mathcal{G}(\cdot)$ denotes a Gaussian function, and $(\sigma^{patch})^2$ ($= 1$, here) is the covariance. As we cannot obtain the terminal correspondence from matching, we use the nearest neighboring distance $dist(\cdot,\cdot)$ to $\Omega_{i'}$ of node i' in G'.

Spatial angle: θ_{ij} denotes the spatial angle between nodes i and j in G, and it is a conventional pairwise attribute. Its compatibility is assumed to follow a Gaussian distribution:

$$P_{ii'jj'}^{angle} = \mathcal{G}(\theta_{i'j'}|\mu = \theta_{ij}, (\sigma^{angle})^2) \tag{9}$$

where, $(\sigma^{angle})^2$ ($= 1$, here) denotes the variation in angle.

Centerline: Besides the spatial angle, the relative spatial translation between two nodes is also modeled as a pairwise attribute. We propose the *centerline*—connecting the centers of two node segments—to measure the translation. The centerline is represented in the local 3D coordinate system of the segments, which is independent of the global rotation of the object. Let $\mathbf{o_i}$ and $\mathbf{o_j}$ denote the unit 3D orientation of node segments i and j. We calculate three orthogonal unit vectors to define this coordinate system as $\mathbf{O}_{ij} = [\frac{\mathbf{o_i}+\mathbf{o_j}}{\|\mathbf{o_i}+\mathbf{o_j}\|_2}, \frac{\mathbf{o_i}-\mathbf{o_j}}{\|\mathbf{o_i}-\mathbf{o_j}\|_2}, \mathbf{o_i} \times \mathbf{o_j}]$.

Thus, the 3D translation $\mathbf{T}_{ij}$ between nodes i and j can be measured in the local coordinate system as $\mathbf{d}^{ij} = \mathbf{O}_{ij}^T \mathbf{T}_{ij}$. Note that the orientation of node segment i may be defined as either $\mathbf{o_i}$ or $-\mathbf{o_i}$, so we instead use $\mathbf{c}_{ij} = [\min\{|d_1^{ij}|, |d_2^{ij}|\}, \max\{|d_1^{ij}|, |d_2^{ij}|\}, |d_3^{ij}|]^T$, as the centerline coordinates. The compatibility of centerline coordinates for the matching assignment $ij \to i'j'$ is also assumed to follow a Gaussian distribution:

$$P_{ii'jj'}^{center} = \mathcal{G}(\mathbf{c}_{i'j'}|\mu = \mathbf{c}_{ij}, (\sigma_{ij}^{\text{cen}})^2\mathbf{I}) \tag{10a}$$

$$(\sigma_{ij}^{\text{cen}})^2 = (\alpha\|\mathbf{c}_{ij}\|_2)^2 + (\sigma^{noise})^2 \tag{10b}$$

where the variation is caused by both the structural deformability and noise, which are controlled by $\alpha = 1$ and $\sigma^{noise} = 5$.

Now, we summarize the model for 3D object detection as follows. We define the local and pairwise attributes as $\mathbf{f}_i = [l_i, \Omega_i]$, $\mathbf{f}_{ij} = [\theta_{ij}, \mathbf{c}_{ij}]$, and the parameters as $\mathbf{w}_1 = [\beta, \sigma^{patch}]$, $\mathbf{w}_2 = [\sigma^{angle}, \sigma^{noise}, \alpha]$. Thus, the overall compatibility for unary and pairwise assignments can be calculated as:

$$\begin{aligned} \rho_{ii'} &= \Phi_1(\mathbf{f}_i, \mathbf{f}_{i'}; \mathbf{w}_1) = P_{ii'}^{length} P_{ii'}^{patch} \\ \rho_{ii'jj'} &= \Phi_2(\mathbf{f}_{ij}, \mathbf{f}_{i'j'}; \mathbf{w}_2) = P_{ii'jj'}^{angle} P_{ii'jj'}^{center} \end{aligned} \tag{11}$$

As $\Phi_1(\cdot)$ and $\Phi_2(\cdot)$ are positive bounded functions, the compatibility maximization can be transformed to the energy minimization problem and solved by TRW-S [36].

Finally, we define the matching rate Υ as the simple evaluation of the matching quality: $\Upsilon = N^{detect}/(N^{detect} + N^{none})$, where N^{detect} and N^{none} are the number of nodes matched to real segments in the target images and *none*, respectively. An incorrect matching will produce a large N^{none} and thus a small Υ. Therefore, only those matching results with $\Upsilon \geq 0.7$ are considered to be sufficiently reliable for further model learning.

3.2. Category model for ordinary RGB images

As depth information can no longer be used, we design new local and pairwise attributes for object detection in ordinary images. The notation is illustrated in Fig. 5.

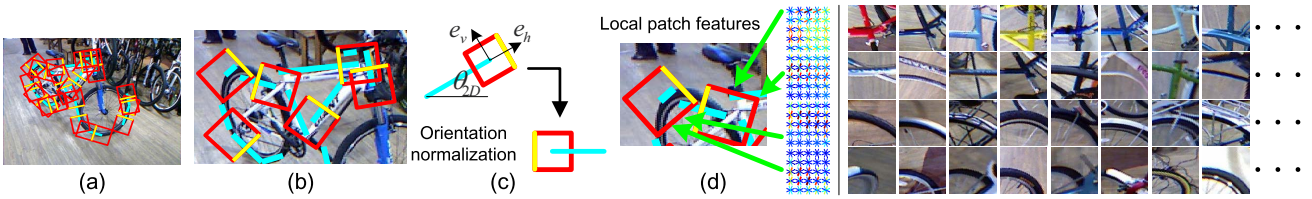

Figure 6. Local codebook extraction. (a) The bicycle is detected by 3D matching. Patches (red) are extracted at terminals of the detected segments (blue). Yellow sides indicate patch orientations. (b) A detailed view. (c) Patch orientation normalization. (d) Patches collected from the same part of objects are clustered to generate a sparse local codebook of patch features.

A local codebook consisting of a set of patch features—$\Omega_i = \{\varpi_i^k\}, (k = 1, 2, ...)$—is learned for each node i in G as the only local attribute (details follow in Section 4.1). Different patch features in the codebook represent different local texture styles, thus overcoming the texture variations.

Three types of pairwise attributes are defined as follows: 1) θ_{ij}^{img} denotes the angle between nodes i and j in G on the image plane; 2) we define $[\lambda_{ij}^A, \lambda_{ij}^B] = \frac{1}{T_{ij}^{img}}[l_i^{img}, l_j^{img}]$ as the relative length, where l_i^{img} denotes the segment length of node i, and T_{ij}^{img} denotes the length of the centerline between nodes i and j in G; 3) $[\theta_{ij}^A, \theta_{ij}^B]$ denote the relative angles between the centerline and line segments of nodes i and j on the image plane, respectively.

As in [2], we absorb local compatibilities into the pairwise compatibilities, $\mathbf{f}_{ij} = \{\theta_{ij}^{img}, \lambda_{ij}^A, \lambda_{ij}^B, \theta_{ij}^A, \theta_{ij}^B, \Omega_i, \Omega_j\}$:

$$\begin{aligned} \rho_{ii'} &= 0 \\ \rho_{ii'jj'} &= \Phi_2(\mathbf{f}_{ij}, \mathbf{f}_{i'j'}; \mathbf{w}) = \\ & e^{-w_1|\theta_{ij}^{img} - \theta_{i'j'}^{img}|^2 - \sum_{k\in\{A,B\}}\left\{w_2|\lambda_{ij}^k - \lambda_{i'j'}^k|^2\right.} \\ & \left. + w_3|\theta_{ij}^k - \theta_{i'j'}^k|^2 + w_4[dist^2(\varpi_{i'}^k, \Omega_i) + dist^2(\varpi_{j'}^k, \Omega_j)]\right\} \end{aligned} \quad (12)$$

The distance between the local codebook Ω_i in G and the patch features $\varpi_{i'}$ in G' is also measured by $dist(\varpi_{i'}, \Omega_i)$ as defined in (8b). Similar to the model for RGBD images, the maximization problem is also solved by TRW-S [36].

4. Model learning

We use matching assignments estimated by relatively reliable 3D matchings to guide the training of the category model for ordinary RGB images, in order to avoid the bias problem. With the part correspondences from 3D matching, we extract a local codebook for each model node that covers all possible texture styles of a local part. We then extend the method proposed by Leordeanu *et al.* [2] for both conventional parameter learning for graph matching and estimation of the general prototype of model attributes.

4.1. Local codebooks extraction

For each node in the category model, we extract a set of patches from its matched node segments in target scenes, as shown in Fig. 6. These patches are extracted at the two terminals of the edge segment, and then normalized to their *right* orientations, thus removing rotation effects. Patches are collected from a square, which should be rotated to the orientation of the edge segment (Fig. 6(c)).

HOG features [6] are extracted from the patches with 5×5 cells, each of which covers half of its neighboring cells. For gradient histogram extraction, the gradient in each cell is encoded into 4 orientation bins ($0°$–$180°$). As the patch is locally collected and suffers only slightly from illumination changes, all the cells can be normalized in a single block.

Patch features corresponding to each node in the model are then clustered via k-means clustering ($k = 5$). Cluster centers are taken as a sparse set of visual words for this node, thus composing the local codebook denoted by Ω_i.

4.2. Model learning

The graph matching based on the category model defined by (1) and (12) can be rewritten as

$$\arg\max_{\mathbf{y}} \mathcal{C} = \arg\max_{\mathbf{y}} \mathbf{y}^T\mathbf{M}\mathbf{y} \quad (13)$$

where $M_{(ii'),(jj')} = \rho_{ii'jj'}$. $\mathbf{y}$ is transformed from a matching matrix to a vector. According to [37], elements of the principal eigenvector $\mathbf{x}$ of $\mathbf{M}$, *e.g.* $x_{ii'}$, can be taken as the confidence value of the corresponding assignment $i \to i'$.

Leordeanu *et al.* [2] proposed to increase the elements corresponding to the correct assignments. At the same time, elements for incorrect assignments will decrease, as $\mathbf{x}$ is normalized. To reduce the large computation, a approximate principal eigenvector is calculated as $\mathbf{x} = \frac{\mathbf{M^n 1}}{\sqrt{(\mathbf{M^n 1})^T(\mathbf{M^n 1})}}$. Thus, the partial derivatives of $\mathbf{x}$ are computed as follows:

$$\mathbf{x}' = \frac{(\mathbf{M^n 1})'\|\mathbf{M^n 1}\| - ((\mathbf{M^n 1})^T(\mathbf{M^n 1})')\mathbf{M^n 1}/\|\mathbf{M^n 1}\|}{\|\mathbf{M^n 1}\|^2} \quad (14)$$

where $(\mathbf{M^n 1})' = \mathbf{M}'(\mathbf{M^{n-1}1}) + \mathbf{M}(\mathbf{M^{n-1}1})'$. We choose $n = 10$, as in [2].

We extend [2] from the pure learning of matching parameters $\mathbf{w}$ to the learning of both the parameters and the model attributes $\{\mathbf{w}, \mathbf{f}\}$ by maximizing the following function:

$$\mathcal{F}(\mathbf{w}, \mathbf{f}) = \sum_{i=1}^{N} \mathbf{x}^{(i)}(\mathbf{w}, \mathbf{f})\mathbf{t}^{(i)} \quad (15)$$

where $i = 1, 2, ..., N$ indicates each target scene used for training; $\mathbf{t}^{(i)}$ denotes the predicted matching assignment.

$\{\mathbf{w}, \mathbf{f}\}$ is initialized using the labeled object, and the maximization of $\mathcal{F}(\mathbf{w}, \mathbf{f})$ can be achieved by modifying $\{\mathbf{w}, \mathbf{f}\}$ in an iterative framework. Intuitively, the matching assignments can be directly predicted as $\mathbf{t}^{(i)} = \hat{\mathbf{y}}^{3D,(i)}$, where $\hat{\mathbf{y}}^{3D,(i)}$ denote the 3D matching assignments in the RGBD image. However, many categories have symmetric 3D structures, *e.g.* notebook PCs, and thus have several potential assignment states. These matching states are equivalent in terms of the 3D structure, but they may show different attributes when the object is projected on the image plane. The matching assignments predicted by the category model (denoted by $\hat{\mathbf{y}}^{img,(i)}$) are not always the same as $\hat{\mathbf{y}}^{3D,(i)}$. Therefore, we use $\hat{\mathbf{y}}^{img,(i)}$ to compute $\mathbf{t}^{(i)}$, and errors in $\hat{\mathbf{y}}^{img,(i)}$ are detected and eliminated by $\hat{\mathbf{y}}^{3D,(i)}$ to avoid the bias problem. If nodes in the target image i are matched by both $\hat{\mathbf{y}}^{img,(i)}$ and $\hat{\mathbf{y}}^{3D,(i)}$, the corresponding assignments in $\hat{\mathbf{y}}^{img,(i)}$ are probably correct. Thus, we get:

$$\mathbf{t}^{(i)} = diag\{a_{jj'}^{(i)}\}\hat{\mathbf{y}}^{img,(i)}, \qquad a_{jj'}^{(i)} = \sum_{j} \hat{y}_{jj'}^{3D,(i)} \tag{16}$$

$a_{jj'}^{(i)} \in \{0, 1\}$ indicates whether node j' has been matched in the 3D matching, as many-to-one matching are avoided.

In iteration k of the EM framework, the matching assignment $\mathbf{t}^{(i),k}$ is estimated by (16), and then the model parameters and attributes are modified via gradient ascent:

$$\begin{aligned} w_j^{k+1} &= w_j^k + \zeta \sum_{i=1}^{N} (\mathbf{t}^{(i),k})^T \frac{\partial \mathbf{x}^{(i),k}(\mathbf{w}, \mathbf{f})}{\partial w_j} \\ f_j^{k+1} &= f_j^k + \zeta \sum_{i=1}^{N} (\mathbf{t}^{(i),k})^T \frac{\partial \mathbf{x}^{(i),k}(\mathbf{w}, \mathbf{f})}{\partial f_j} \end{aligned} \tag{17}$$

5. Experiments

5.1. Data

Various RGBD datasets has been built in recent years. However, according to our scenario of learning from casually captured RGBD images, target objects should not be hand-cropped or aligned, and thus have different scales, textures, and rotations. Each category must contain enough samples for training. Therefore, we build a new dataset containing approximately 900 objects in complex environments. Five large categories—*notebook PC*, *drink box*, *basket*, *bucket*, and *bicycle*—are used, containing 33, 36, 36, 67, and 92 scenes, respectively. Please visit http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343 to download this dataset.

5.2. Results and evaluation

Most of image-based category knowledge mining algorithms are hampered by texture variations and roll rotations. In this case, we compare the proposed method with image-based semi-supervised and supervised learning of graph matching, and five competing methods are used. Pure graph matching based on TRW-S [36] without learning is denoted by *Matching+TRW-S*. Two methods based on [2] learn graph matching in an unsupervised manner, using spectral techniques [37] and the TRW-S [36] to solve graph matching, respectively. However, just like our method, the template graph is also required in [2], so we refer to these as semi-supervised methods: *Semi-supervised+Spectral* and *Semi-supervised+TRW-S*. The remaining two methods achieve supervised learning of the proposed category model. *Supervised* uses the ground truth, instead of 3D matching assignments, to guide the model learning, whereas *Supervised+NIO* uses nonlinear inverse optimization (NIO) for model learning [4, 38].

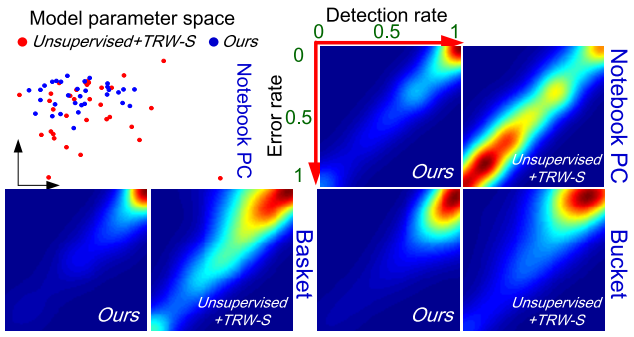

Figure 7. Biased models. (Top left) Model parameters ($\mathbf{w}$) of the *notebook PC* category projected onto a 2D space. Different points indicate $\mathbf{w}$ learned from a different initial labeling. Our method learns more convergent values of $\mathbf{w}$, whereas the outliers provided by *Semi-supervised+TRW-S* indicate the biased models. (The others) Distribution of the detection and error rates of the learned models. *Semi-supervised+TRW-S* provides more biased models.

Matching+TRW-S does not learn the matching weights defined in (12), so we simply set $\mathbf{w} = \mathbf{1}$. *Supervised* transforms semi-supervised learning into supervised learning by redefining $a_{jj'}$ in (16) as 1 or 0 depending on whether node j' in the scene is a true object part according to the ground truth. This kind of supervised learning is also formulated in [2]. Finally, in *Supervised+NIO*, the NIO [38] is used to estimate the model parameters and attributes that minimize the compatibility gap between the true assignments and predicted assignments, as $\arg\min_{\mathbf{f}_{ij}, \mathbf{w}} \sum_{k=1}^{N} \{ \max_{\mathbf{y}} \mathcal{C}(\mathbf{f}_{ij}, \mathbf{w}, \mathbf{y} | G'^{(k)}) - \mathcal{C}(\mathbf{f}_{ij}, \mathbf{w}, \mathbf{y}_{truth}^{(k)} | G'^{(k)}) \}$. $\mathcal{C}(\cdot)$ is the matching compatibility in (1), and $G'^{(k)}$ and $\mathbf{y}_{truth}^{(k)}$ are the graph and the matching ground truth of scene k.

The object detection performance is evaluated by the cross validation. We use each RGBD image to start a single model learning process as follows. We label edge segments on the target object in this image, and randomly select $2/3$ and $1/3$ of the remaining RGBD images in this category as a training set and a testing set, respectively. Thus, we learn a number of models for each category, and use each of them to test object detection.

Detection rate / Error rate (%)	Notebook PC	Drink box	Basket	Bucket	Bicycle
Matching+TRW-S[36]	56.17 / 42.82	84.84 / 14.93	74.12 / 24.67	73.43 / 22.76	67.62 / **18.31**
Semi-supervised+Spectral[2, 37]	41.89 / 58.16	78.01 / 21.99	61.69 / 39.11	74.60 / 30.17	76.28 / 23.72
Semi-supervised+TRW-S[2, 36]	43.57 / 54.43	77.95 / 20.89	62.87 / 30.83	69.47 / 22.41	61.37 / 20.40
Ours	**74.24 / 25.98**	**98.03 / 1.97**	**88.04 / 13.22**	**87.99 / 17.77**	**81.56** / 18.44
Supervised[2]	73.13 / 27.08	**98.61 / 1.39**	87.21 / 14.15	**87.69 / 18.04**	80.98 / 19.02
Supervised+NIO[4, 38]	**78.11 / 22.13**	95.54 / 4.46	**92.05 / 9.42**	79.08 / 25.55	**82.68 / 17.31**
3D matching (in **RGBD** images)	93.68 / 6.49	90.57 / 9.43	90.35 / 11.00	96.12 / 10.57	93.87 / 4.58

Table 1. Detection rate and error rate of object detection. Our method learns from a minimum labeling, but achieves similar performances to these supervised methods that require to manually label all the training samples.

We use the detection rate ($DR = \frac{N^T}{\min\{N^{model}, N^{target}\}}$) and error rate ($ER = \frac{N^B}{N^{model}}$) to evaluate each single detection of objects. N^T and N^B denote the number of nodes in the model that are matched to the target object and the background; N^{model} and N^{target} indicate the total number of segments in the model and the target object. Note that $N^T + N^B \leq N^{model}$, as some model nodes may be matched to *none*.

Thus, the average values of DR and ER indicate the overall detection performance for each category. The averaging is applied across all detections produced by all the learned models in the cross validation.

Results: Fig. 8 illustrates object detection using the learned category models, and Table 1 gives the quantitative results. Table 1 also proves that the performance of 3D matching from RGBD images is superior enough to guide the learning of category models. Conventional semi-supervised methods suffer greatly from the bias problem, as shown in Fig. 7. For some categories, our method exhibits a better performance than the *Supervised* method. This is because the manual labeling of the ground truth only determines a set of correct object segments for detection in target scenes for the *Supervised* method, whereas our method uses 3D matching to provide more exact matching assignments that fit the target model, in spite of some matching errors. Moreover, the learning algorithm [2] is not sensitive to outliers in training samples for the regression of the prototype model, so our method performs even better than the 3D matching for the *drink box* category.

6. Discussion and conclusions

In this paper, we proposed a method for category model learning from a single labeled object and a number of casually captured RGBD images, and the learned model was expected to be applied to object detection in ordinary RGB images. The minimum labeling greatly saves human labor in model base construction. The depth information in RGB-D images helps the semi-supervised learning framework to overcome the bias problem. Our experiments have demonstrated the effectiveness of the proposed.

Using graph matching, the model cannot detect multiple objects for each time. As artifacts for daily use usually have regular shapes and various textures, the proposed category model mainly focuses on structural information, namely object edge segments. This design makes the model robust to texture variations, but at the same time unsuitable for largely occluded objects and those with highly deformable or irregular shapes, such as natural scenes and animals.

ACKNOWLEDGMENT

This work was supported by Microsoft Research, a Grant-in-Aid for Young Scientists (23700192) of Japans Ministry of Education, Culture, Sports, Science, and Technology (MEST), and Grant of Japans Ministry of Land, Infrastructure, Transport and Tourism (MLIT).

References

[1] *Introducing Kinect for Xbox 360,* http://www.xbox.com/en-US/Kinect/, 2011. 1

[2] M. Leordeanu and M. Hebert, "Unsupervised learning for graph matching", *In* CVPR, 2009. 2, 5, 6, 7

[3] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching", *In* PAMI, 2009. 2

[4] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization", *In* ECCV, 2008. 2, 6, 7

[5] L.-J. Li, G. Wang, and F.-F. Li, "Optimol: automatic online picture collection via incremental model learning", In *IJCV*, vol. 88, no. 2, pp. 147–154, 2010. 2, 3

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", In *CVPR*, 2005. 2, 5

[7] M. Leordeanu, M. Hebert, and R. Sukthankar, "Beyond local appearance: category recognition from pairwise interactions of simple features", *In* CVPR, 2007. 2

[8] N. Razavi, J. Gall, P. Kohli, and L. v. Gool, "Latent hough transform for object detection", *In* ECCV, 2012. 2

[9] K. Liu, Q. Wang, W. Driever, and O. Ronneberger, "2d/3d rotation-invariant detection using equivariant filters and kernelweighted mapping", *In* CVPR, 2012. 2

[10] K. Lai and D. Fox, "Object recognition in 3d point clouds using web data and domain adaptation", *In* IJRR, vol. 29, no. 8, pp. 1019–1037, 2010. 2

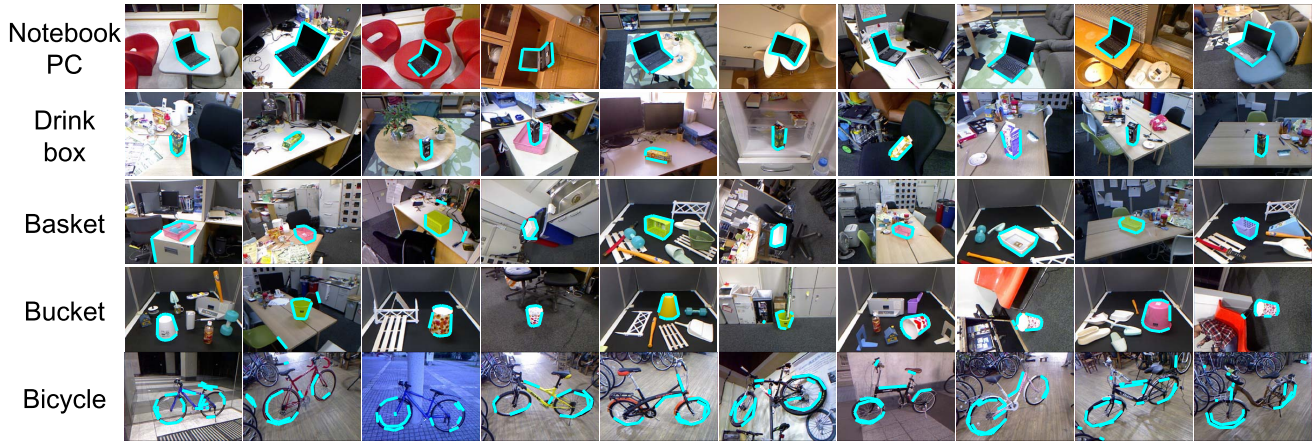

Figure 8. Object detection results.

[11] E. Hsiao, A. Collet, and M. Hebert, "Making specific features less discriminative to improve point-based 3d object recognition", *In* CVPR, 2010. 2

[12] W. Hu, "Learning 3d object templates by hierarchical quantization of geometry and appearance spaces", *In* CVPR, 2012. 2

[13] B. Pepik, P. Gehler, M. Stark, and B. Schiele, "3d2pm—3d deformable part models", *In* ECCV, 2012. 2

[14] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition", *In* ECCV, 2012. 2

[15] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information", *In* ICRA, 2011. 2

[16] W. Susanto, M. Rohrbach, and B. Schiele, "3d object detection with multiple kinects", *In* ECCV, 2012. 2

[17] A. Collet, S. S. Srinivasay, and M. Hebert, "Structure discovery in multi-modal data: a region-based approach", In *ICRA*, 2011. 2

[18] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms", *In* CVPR, 2012. 2

[19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images", *In* ECCV, 2012. 2

[20] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization", *In* ICCV, 2011. 2

[21] K. I. Kim, J. Tompkin, M. Theobald, J. Kautz, and C. Theobalt, "Match graph construction for large image databases", *In* ECCV, 2012. 2

[22] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting", *In* CVPR, 2012. 2

[23] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories", *In* PAMI, vol. 28, no. 4, pp. 594–611, 2006. 2

[24] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine, "Unsupervised object discovery: A comparison", In *IJCV*, vol. 88, no. 2, pp. 284–302, 2010. 2

[25] H. Kang, M. Hebert, and T. Kanade, "Discovering object instances from scenes of daily living", In *ICCV*, 2011. 2, 3

[26] C. Li, D. Parikh, and T. Chen, "Automatic discovery of groups of objects for scene understanding", In *CVPR*, 2012. 2

[27] A. Faktor and M. Irani, ""clustering by composition" - unsupervised discovery of image categories", In *ECCV*, 2012. 2

[28] J.-Y. Zhu, J. Wu, Y. Wei, E. Chang, and Z. Tu, "Unsupervised object class discovery via saliency-guided multiple class learning", In *CVPR*, 2012. 2

[29] Y. J. Lee and K. Grauman, "Shape discovery from unlabeled image collections", In *CVPR*, 2009. 3

[30] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery", In *CVPR*, 2011. 3

[31] Z.Liao, A.Farhadi, Y.Wang, I.Endres, and D.Forsyth, "Building a dictionary of image fragments", In *CVPR*, 2012. 3

[32] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds", In *CVPR*, 2011. 3

[33] C. Olsson and Y. Boykov, "Curvature-based regularization for surface approximation", *In* CVPR, 2012. 3

[34] H. Liu and S. Yan, "Efficient structure detection via random consensus graph", *In* CVPR, 2012. 3

[35] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation", *In* PAMI, vol. 33, no. 5, pp. 898–916, 2011. 3

[36] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization", *In* IEEE PAMI, vol. 28, no. 10, pp. 1568–1583, 2006. 4, 5, 6, 7

[37] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints", *In* ICCV, 2005. 5, 6, 7

[38] Z. Popović C. K. Liu, A. Hertzmann, "Learning physics-based motion style with nonlinear inverse optimization", *In* SIGGRAPH, 2005. 6, 7