Learning Graph Matching: oriented to Category Modeling from Cluttered Scenes

Quanshi Zhang†, Xuan Song†, Xiaowei Shao†, Huijing Zhao‡, and Ryosuke Shibasaki†
Center for Spatial Information Science, University of Tokyo†
Key Laboratory of Machine Perception (MoE), Peking University‡

Abstract

Although graph matching is a fundamental problem in pattern recognition, and has drawn broad interest from many fields, the problem of learning graph matching has not received much attention. In this paper, we redefine the learning of graph matching as a model learning problem. In addition to conventional training of matching parameters, our approach modifies the graph structure and attributes to generate a graphical model. In this way, the model learning is oriented toward both matching and recognition performance, and can proceed in an unsupervised[1] fashion. Experiments demonstrate that our approach outperforms conventional methods for learning graph matching.

1. Introduction

Attributed graph matching is a fundamental problem ranging across broad fields in computer vision and data mining, and numerous approaches have been proposed for the problem of graph matching optimization [22]. Even so, the literature on learning graph matching remains limited, despite the demonstrated power of learning techniques in this area. The few pioneering studies of learning graph matching mainly aimed to train matching parameters, so as to obtain correct matching assignments for mapping from a graph template to a number of relatively large target graphs.

In this research, we approach the learning of graph matching from the perspective of category modeling. Our aim is to incrementally modify the graph template to produce a graphical model representing the general structural knowledge of the targets objects in target graphs, and not merely train matching parameters. Therefore, this research is of great significance for object knowledge mining from cluttered scenes.

In so doing, we also aim to transform the conventional concept of *learning for graph matching* to *learning based on graph matching for both object matching and recognition*. Here, object recognition refers to determining whether a graph contains the target object, based on the trained model. We call the graphs that contain target objects *positive* graphs, and those that do not *negative* graphs. This idea for learning gives consideration to both matching performance and recognition performance.

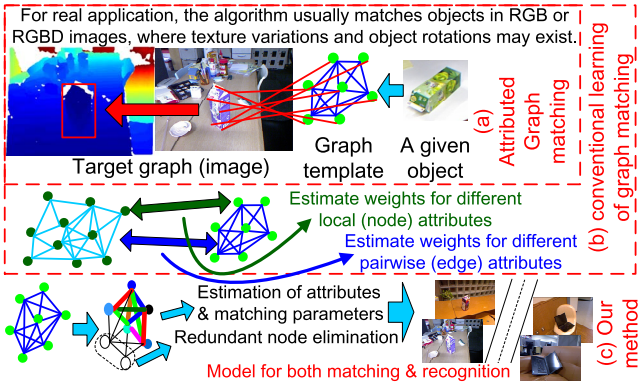

Figure 1. Concept extension from pure attributed graph matching (a) to the proposed learning of graph matching (c). Different from conventional learning of graph matching (b), our method aims to modify the initial graph template into a graphical model, so as to achieve good performance in both matching and recognition. Thus, this research establishes a bridge between the learning of graph matching and the category modeling from cluttered scenes.

The goal of this paper can be described as follows. Given an initial graph template and a number of positive and negative graphs for training, we aim to 1) learn matching parameters in an unsupervised[1] fashion, 2) incrementally refine the local and pairwise attributes of the graph template, and simultaneously 3) modify the structure of the graph template by eliminating incorrect or redundant parts, thus generating a model that achieves good performance in both matching and recognition.

To perform estimates of both matching parameters and model attributes, our work extends the unsupervised[1] learning of graph matching proposed by Leordeanu *et al.* [17]. Meanwhile, our approach to structural modification of the graph template uses a novel technique based on the mecha-

[1]In the context of learning graph matching, the term "unsupervised" [17] refers to the ability to learn the model without manually specifying each individual matching assignment within the graphs, rather than the labeling of positive and negative graphs.

nism of object recognition.

The contributions of this paper can be summarized as follows. We redefine the learning of graph matching as a category modeling problem that is oriented toward not only conventional matching performance, but also object recognition. It includes the estimation of matching parameters, attributes, and graphical structure. In particular, this is the first attempt to use negative graphs in the learning of graph matching, to the best of our knowledge.

2. Related work

Most conventional algorithms for the learning of graph matching are supervised[1] methods that require detailed labeling of each template node's matching assignment in each positive graph for training. [5], [25], and [16] used large-margin methods [26], non-linear inverse optimization [4], and smoothing-based techniques to train matching parameters in a supervised fashion, respectively. Compared to supervised methods, the unsupervised[1] method [17] does not require a large amount of node-level labeling. Thus, it is closer to our approach and resembles, at least philosophically, our idea of category knowledge mining from cluttered scenes. Indeed, we can derive another type of supervised learning from [17] as a compromise, greatly reducing manual interactions by applying object-level labeling, and thereby supporting fairer comparison.

All studies mentioned above are focused on training matching parameters for good matching performance. By contrast, our approach emphasizes ont only the matching rate, but also the recognition performance of the trained, matching-based model. In addition to training the matching parameters, we modify the model (graph template) structure and estimate model attributes.

Then, we give an in-depth discussion on the structural modification, which is the main part of the learning process. To be exact, if we simplify the problem by only considering the matching between two graphs, the structural modification is related to the progressive graph matching [6] proposed by Cho *et al.*, which made a great contribution to the selection of reliable nodes and edges for a more efficient matching. If we do not limit our discussion within the range of general-form graph matching, this problem is also related to category modeling for recognition [18], the model training for the Hough-style matching [27], and common object extraction from two images based on maximal clique mining [28, 31], such as [29, 19]. Most methods for object extraction from multiple images [23, 30, 33, 7, 20] related to clique mining uses a combination of two-image matching results. [13] extracted object models from images based on page-rank mechanisms. [12, 10, 11, 24] aim to learn the maximal frequent subgraph among several graphs with distinct node or edge labels. Above all, most of these methods mentioned above limit their interests to the geometric consistency and similarity of local patches. As thus, they usually need additional data constraints. *E.g.* [18, 27] require there is no roll rotations for matching, and object extraction methods usually require the local features in images to be distinguishing enough to determine a set of potential image matching assignments during the preprocessing.

In contrast, by emphasizing a general algorithm for learning graph matching, our approach remains free of such constraints. We formulate the learning problem strictly under a common paradigm of graph matching based on various local and pairwise attributes, and so are able to apply our method to cluttered scenes containing target objects with different scales, textures, and rotations simply by designing a set of suitable attributes. Nevertheless, we still compare our recognition-oriented structural modification strategy with strategies of the related studies in experiments.

3. Preliminary: graph matching problem

The objective of graph matching is to find correspondences between a graph template (the category model) $G = (V, E, F_V, F_E)$ and an attributed graph $G' = (V', E', F_{V'}, F_{E'})$. V and E denote the node set and the edge set of G, respectively. F_V and F_E denote the attribute sets for local and pairwise attributes. Let G have n_v nodes, $V = \{1, 2, ..., n_v\}$, and G' have $n_{v'}$ nodes, $V' = \{1, 2, ..., n_{v'}\}$. Each node $i \in V$ of G has n^U unary attributes ($f_i^{(k)} \in F_V, k = 1, 2, ..., n^U$) and each edge $(i, j) \in E$ has n^P pairwise attributes ($f_{ij}^{(l)} \in F_E, l = 1, 2, ..., n^P$). The matching assignments between G and G' are represented by a binary matching matrix $\mathbf{Y} \in \{0, 1\}^{n_v \times n_{v'}}$. If node $i \in V$ matches node $i' \in V'$, then $Y_{i,i'} = 1$, otherwise $Y_{i,i'} = 0$. In fact, we use a column-wise vectorized replica of $\mathbf{Y}$, denoted by $\mathbf{y} \in \{0, 1\}^{n_v n_{v'}}$. $y_{ii'}$ in $\mathbf{y}$ corresponds to $Y_{i,i'}$ in $\mathbf{Y}$. Thus, we obtain a typical form [15, 6, 17] of attributed graph matching as follows:

$$\begin{aligned} &\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, \mathcal{C}(\mathbf{y}|G, G'), \quad \mathcal{C}(\mathbf{y}|G, G') = \mathbf{y}^T \mathbf{M} \mathbf{y} \\ &\text{s.t.} \quad \forall i \in V, \sum_{i' \in V'} y_{ii'} \leq 1, \quad \forall i' \in V', \sum_{i \in V} y_{ii'} \leq 1 \end{aligned} \tag{1}$$

This is a quadratic assignment problem, where $\mathcal{C}(\mathbf{x}|G, G')$ is the function measuring the matching compatibility between G and G'. $\mathbf{M}$ is a $(n_v n_{v'})$-by-$(n_v n_{v'})$ compatibility matrix containing non-negative elements. In most cases [15, 6], the matching compatibility $M_{ii',jj'}$ can be represented as a function of attribute distances as follows.

$$\begin{aligned} M_{ii',jj'} &= \begin{cases} \Phi^P(\mathbf{d}_{ii',jj'}|\mathbf{w}^P), & (i,j) \in E, (i',j') \in E' \\ 0, & \text{Otherwise} \end{cases} \\ M_{ii',ii'} &= \Phi^U(\mathbf{d}_{ii'}|\mathbf{w}^U), \quad i \in V, i' \in V' \end{aligned} \tag{2}$$

where $\Phi^U(\mathbf{d}_{ii'}|\mathbf{w}^U)$ is set on the diagonal of $\mathbf{M}$ and measures the unary compatibility for a node pair of $i \in V$ and

$i' \in V'$; the non-diagonal element of $\mathbf{M}$, $\Phi^P(\mathbf{d}_{ii',jj'}|\mathbf{w}^P)$, measures the pairwise compatibility for an edge pair of $(i,j) \in E$ and $(i',j') \in E'$.

We define $\mathbf{d}_{ii'} = [d_{ii'}^{(1)}, d_{ii'}^{(2)}, ..., d_{ii'}^{(n^U)}]^T$ as the Euclidean distances of the unary attributes, $d_{ii'}^{(k)} = \|f_i^{(k)} - f_{i'}^{(k)}\|$. Whereas $\mathbf{d}_{ii',jj'} = [d_{ii',jj'}^{(1)}, d_{ii',jj'}^{(2)}, ..., d_{ii',jj'}^{(n^P)}]^T$ denote the Euclidean distances of the pairwise attributes, $d_{ii',jj'}^{(l)} = \|f_{ij}^{(l)} - f_{i'j'}^{(l)}\|$. $\mathbf{w}^U = [w_1^U, w_2^U, ..., w_{n^U}^U]^T$ and $\mathbf{w}^P = [w_1^P, w_2^P, ..., w_{n^P}^P]^T$ denote the weights for each unary and pairwise attribute, respectively.

As in [17], without loss of generality, we transform (2) to absorb the unary compatibilities into the pairwise compatibilities and leave zeros on the diagonal, achieving better performance.

$$M_{ii',jj'} = \begin{cases} \Phi(\mathbf{d}_{ii'},\mathbf{d}_{jj'},\mathbf{d}_{ii',jj'}|\mathbf{w}^U,\mathbf{w}^P), & (i,j)\in E, (i',j')\in E' \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

Note that when the structure of the graph template G is not well segmented and needs further modification, it is meaningful to bring in a dummy choice—*none*—for the matching assignments of nodes in G. Without an accurate structure, G may have some redundant nodes that should be matched to *none*. Thus, we re-write (1) and (3) as:

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmax}}\, \mathcal{C}'(\mathbf{x}|G,G') \\ \mathcal{C}'(\mathbf{x}|G,G') &= \sum_{i,j\in V\cup\{none\}} c_{ij}(x_i,x_j|G,G') \\ c_{ij}(x_i,x_j|G,G') &= \begin{cases} M_{ix_i,jx_j}, & x_i \neq x_j \in V' \\ -\infty, & x_i = x_j \in V' \\ \frac{\lambda(\mathbf{1}^T\mathbf{M}\mathbf{1})}{n_v^2 n_{v'}^2}, & x_i \text{ or } x_j = none \end{cases} \end{aligned} \quad (4)$$

where x_i indicates the matching assignment of node $i \in V$, and $x_i = i' \in V'$ if and only if $y_{ii'} = 1$. λ is the parameter weighting for the matching compatibility of *none*. The setting of *none* reduces incorrect matching and eases the bias learning problem that commonly afflicts the unsupervised learning of graph matching (which will be explained later).

The maximization of the compatibility function $\mathcal{C}'(\mathbf{x}|G,G')$ can be achieved using various graph matching optimization techniques, and we choose TRW-S [14] here.

4. Learning of graph matching

Given an initial graph template $G=(V,E,F_V,F_E)$, a set of N^+ positive graphs $PG=\{G_k^+|k=1,2,...,N^+\}$, $G_k^+ = (V_k^+,E_k^+,F_{V_k^+},F_{E_k^+})$, and a set of N^- negative graphs, $NG=\{G_l^-|l=1,2,...,N^-\}$, $G_l^- = (V_l^-,E_l^-,F_{V_l^-},F_{E_l^-})$, the objective for learning graph matching is to estimate an induced subgraph of G, $\tilde{G} = (\tilde{V},\tilde{E},F_{\tilde{V}},F_{\tilde{E}})$, $\tilde{V} \subseteq V, \tilde{E} \subseteq E$, as the category model, simultaneously training matching parameters $\{\mathbf{w}^U,\mathbf{w}^P\}$ and modifying model attributes $\{F_{\tilde{V}},F_{\tilde{E}}\}$, so as to achieve good matching performance in positive graphs in PG and high recognition accuracy among graphs in PG and NG.

Algorithm 1 Learning of graph matching

Input: An initial graph template G^*; a set of N^+ positive graphs, PG; a set of N^- negative graphs, NG; the iteration number T for the estimation of matching parameters and attributes; a threshold τ.
Output: The category model $\tilde{G}$.
Set initial leave-one-out (LOO) classification accuracy as $\tilde{A} = 1$ and node reliability of G as $\forall i \in V, \tilde{R}_i = -\infty$.
repeat
1. Initialize the category model $G = G^*$ and the weights for unary and pairwise attributes $\mathbf{w}^U = \mathbf{1}_{n^U\times 1}$, $\mathbf{w}^P = \mathbf{1}_{n^P\times 1}$.
for $iteration = 1$ **to** T **do**
2.1. Use the current G to predict the matching assignments to G_k^+, $\hat{\mathbf{X}}$, based on (4).
2.2. With $\hat{\mathbf{X}}$, update the matching parameters $\mathbf{w}^U$,$\mathbf{w}^P$ and attributes F_V,F_E of G, based on (7).
end for
3. Match the current G to graphs in PG and NG based on (4)[2] and obtain $\hat{\mathbf{X}}$ and $\mathring{\mathbf{X}}$.
4. Given $\hat{\mathbf{X}}$ and $\mathring{\mathbf{X}}$, train the classifier with the normal vector $\mathcal{W}$ and a new LOO accuracy A, based on (10).
5. **If** $A < \tilde{A}$ and $\min_i \tilde{R}_i > \tau$ **then break.**
6. Given $\mathcal{W}$, update node reliability $\tilde{R}_i$, based on (11). Update $\tilde{G} = G$ and $\tilde{A} = A$.
7. Eliminate node $i^* = \operatorname{argmin}_{i\in V} R_i$ from G^* and set it as the induced subgraph $G^* \leftarrow G^*(V\backslash\{i^*\})$.
until $n_v = 2$

4.1. Parameter and attribute estimation

For the matching between the current G and G_k^+ based on (1), let the principal eigenvector of $\mathbf{M}$ be $\mathbf{a}^k$. According to [15], its element $a_{ii'}^k$ can be taken as the confidence value of the corresponding assignment between node $i \in V$ and node $i' \in V_k^+$. Leordeanu *et al.* [17] proposed to increase the elements corresponding to the correct assignments. Meanwhile, as $\|\mathbf{a}^k\|$ is normalized to 1, the elements for incorrect assignments will decrease, thus achieving greater reliability in matching.

To reduce the large computation, an approximate principal eigenvector is calculated as $\mathbf{a}^k = \frac{\mathbf{M}^n\mathbf{1}}{\sqrt{(\mathbf{M}^n\mathbf{1})^T(\mathbf{M}^n\mathbf{1})}}$. Thus, the partial derivative of $\mathbf{a}^k$ is computed as follows:

$$(\mathbf{a}^k)' = \frac{(\mathbf{M}^n\mathbf{1})'\|\mathbf{M}^n\mathbf{1}\| - ((\mathbf{M}^n\mathbf{1})^T(\mathbf{M}^n\mathbf{1})')\mathbf{M}^n\mathbf{1}/\|\mathbf{M}^n\mathbf{1}\|}{\|\mathbf{M}^n\mathbf{1}\|^2}$$
$$(\mathbf{M}^n\mathbf{1})' = \mathbf{M}'(\mathbf{M}^{n-1}\mathbf{1}) + \mathbf{M}(\mathbf{M}^{n-1}\mathbf{1})' \quad (5)$$

Here, we choose $n = 10$, as in [17].

The benchmark method for unsupervised learning of graph matching [17] proposed by Leordeanu *et al.* focus-

es exclusively on learning matching parameters. We extend this method to include the learning of model attributes $\{\mathbf{w}^U, \mathbf{w}^P, F_V, F_E\}$, which is similar to [32]. The objective is to maximize the following function:

$$\mathcal{G}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)=\sum_{k=1}^{N^+}\sum_{\substack{i\in V\\ \hat{x}_i^k\neq none}} a_{i\hat{x}_i^k}^k(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E) \quad (6)$$

where $\hat{\mathbf{x}}^k = \{\hat{x}_i^k \in V_k^+ | i \in V\}$ is the predicted assignments between the current G and G_k^+ based on (4).

As shown in Algorithm 1, we can achieve the maximization of $\mathcal{G}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)$ iteratively. In each iteration, we use the current G to predict the matching assignments $\hat{\mathbf{x}}^k, k = 1, 2..., N^+$, and then modify the matching parameters $\mathbf{w}^U$,$\mathbf{w}^P$ and attributes F_V,F_E via gradient ascent:

$$\begin{aligned}
w_k^U &\leftarrow w_k^U + \zeta\textstyle\sum_{k'=1}^{N^+}\sum_{\substack{i'\in V\\ \hat{x}_{i'}^{k'}\neq none}} \frac{\partial a_{i'\hat{x}_{i'}^{k'}}^{k'}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)}{\partial w_k^U}\\
w_l^P &\leftarrow w_l^P + \zeta\textstyle\sum_{k'=1}^{N^+}\sum_{\substack{i'\in V\\ \hat{x}_{i'}^{k'}\neq none}} \frac{\partial a_{i'\hat{x}_{i'}^{k'}}^{k'}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)}{\partial w_l^P} \quad (7)\\
f_i^{(k)} &\leftarrow f_i^{(k)} + \zeta\textstyle\sum_{k'=1}^{N^+}\sum_{\substack{i'\in V\\ \hat{x}_{i'}^{k'}\neq none}} \frac{\partial a_{i'\hat{x}_{i'}^{k'}}^{k'}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)}{\partial f_i^{(k)}}\\
f_{ij}^{(l)} &\leftarrow f_{ij}^{(l)} + \zeta\textstyle\sum_{k'=1}^{N^+}\sum_{\substack{i'\in V\\ \hat{x}_{i'}^{k'}\neq none}} \frac{\partial a_{i'\hat{x}_{i'}^{k'}}^{k'}(\mathbf{w}^U,\mathbf{w}^P,F_V,F_E)}{\partial f_{ij}^{(l)}}
\end{aligned}$$

4.2. Structural modification

In this subsection, we use the matching results of the current G to train a classifier for object recognition. In order to train the model for good recognition performance, we propose a new method that uses the parameters of the classifier to guide the structural modification of G.

There are a variety of approaches to classification based on graph matching [9, 21], and we will obtain different classification performances by applying different classifiers to different feature. In order to achieve a natural connection between the structural modification of G and the classification mechanism, we select the linear-SVM classifier and attribute distances as our target features.

Feature extraction: Let $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}^k | k = 1, 2..., N^+\}$ and $\mathring{\mathbf{X}} = \{\mathring{\mathbf{x}}^l | l = 1, 2..., N^-\}$ denote a set of predicted assignments matching to positive graphs $G_k^+ \in PG$ and a set of predicted assignments matching to negative graphs $G_l^- \in NG$, based on graph matching[2]. Thus, according to (3), $\mathbf{d}_{i\hat{x}_i^k}$ indicates the distance of the unary attributes for matching node $i \in V$ to node $\hat{x}_i^k$ in positive graph G_k^+. $\mathbf{d}_{i\mathring{x}_i^l}$ is for the matching to negative graph G_l^-. Similarly, $\mathbf{d}_{i\hat{x}_i^k,j\hat{x}_j^k}$ and $\mathbf{d}_{i\mathring{x}_i^l,j\mathring{x}_j^l}$ are for pairwise attribute distances.

[2]Note that graph matching based on (4) is applied by setting $\lambda = -\infty$ in (4) to avoid $\hat{x}_i^k$ or $\mathring{x}_i^l = none$.

Features for object recognition are generated from these attribute distances. We define the feature vector to recognize the matching between G and G_k^+ as follows:

$$\begin{aligned}
&\hat{\mathcal{F}}^k = [\hat{\mathbf{u}}_1^k, \hat{\mathbf{p}}_1^k, \hat{\mathbf{u}}_2^k, \hat{\mathbf{p}}_2^k, ..., \hat{\mathbf{u}}_{n_v}^k, \hat{\mathbf{p}}_{n_v}^k]^T\\
&\hat{\mathbf{u}}_i^k = \mathbf{d}_{i\hat{x}_i^k}^T, \quad \hat{\mathbf{p}}_i^k = \textstyle\sum_{j:j\neq i}\mathbf{d}_{i\hat{x}_i^k,j\hat{x}_j^k}^T / \sum_{\substack{j:(i,j)\in E\\ (\hat{x}_i^k,\hat{x}_j^k)\in E_k^+}} 1
\end{aligned} \quad (8)$$

For the matching between nodes $i \in V$ and $\hat{x}_j^k \in V_k^+$, $\hat{\mathbf{u}}_i^k$ and $\hat{\mathbf{p}}_i^k$, $(i = 1, 2, ..., n_v \in V)$, are two n^U-dimension and n^P-dimension vectors for node $i \in V$, indicating the distance of the n^U unary attributes and the marginal penalty for the distance of the pairwise attributes, respectively.

Similarly, the feature vector for the matching between G and G_l^- is represented as

$$\mathring{\mathcal{F}}^l = [\mathring{\mathbf{u}}_1^l, \mathring{\mathbf{p}}_1^l, \mathring{\mathbf{u}}_2^l, \mathring{\mathbf{p}}_2^l, ..., \mathring{\mathbf{u}}_{n_v}^l, \mathring{\mathbf{p}}_{n_v}^l]^T \quad (9)$$

Both $\hat{\mathcal{F}}^k$ and $\mathring{\mathcal{F}}^l$ are vectors with $n_v(n^U + n^P)$ dimensions.

Classification for object recognition: We train a linear-SVM classifier for object recognition as follows.

$$\begin{aligned}
&\min_{\mathcal{W},\xi,b}\Big\{\frac{1}{2}\|\mathcal{W}\|^2 + C\sum_{k=1}^{N^+ + N^-}\xi_k\Big\},\\
&\text{s.t.}\quad \forall k = 1, 2, ..., N^+, \mathcal{W}\cdot\hat{\mathcal{F}}^k - b \geq 1 - \xi_k,\ \xi_k \geq 0;\\
&\forall k = 1, 2, ..., N^-, -(\mathcal{W}\cdot\mathring{\mathcal{F}}^k - b) \geq 1 - \xi_{k+N^+},\ \xi_{k+N^+} \geq 0
\end{aligned} \quad (10)$$

where $\mathcal{W} = [\boldsymbol{\mu}_1, \boldsymbol{\rho}_1, \boldsymbol{\mu}_1, \boldsymbol{\rho}_1, ..., \boldsymbol{\mu}_{n_v}, \boldsymbol{\rho}_{n_v}]^T$ represent the normal vector to the hyperplane. $\boldsymbol{\mu}_i$ is a n^U-dimension vector and corresponds to the weights for the n^U unary attribute distances in $\hat{\mathbf{u}}_i^k$ and $\mathring{\mathbf{u}}_i^l$. $\boldsymbol{\rho}_i$ is a n^P-dimension vector for pairwise attribute distances in $\hat{\mathbf{p}}_i^k$ and $\mathring{\mathbf{p}}_i^l$.

Classifier-guided structural modification: Here, we combine graph matching and the SVM-based classification to identify reliable and unreliable nodes in G, as follows. Clearly, G should be better matched to positive graphs $G_k^+ \in PG$ than negative ones $G_l^- \in NG$. In other words, attribute distances for matching to positive graphs (*i.e.* $\hat{\mathbf{u}}_i^k$ and $\hat{\mathbf{p}}_i^k$) should be less than those for matching to negative graphs ($\mathring{\mathbf{u}}_i^l$ and $\mathring{\mathbf{p}}_i^l$), when node i is a reliable node in G. Consequently, the weights of node i (*i.e.* $\boldsymbol{\mu}_i$ and $\boldsymbol{\rho}_i$) should be negative, according to (10).

Therefore, we use the following metric to evaluate the reliability of node $i \in V$:

$$R_i = -\frac{\sqrt{n_v}}{\|\mathcal{W}\|}\Big[\sum_{j=1}^{n^U}\mu_i^{(j)} + \sum_{j=1}^{n^P}\rho_i^{(j)}\Big] \quad (11)$$

where $\mu_i^{(j)}, \rho_i^{(j)}$ are the j-th elements of $\boldsymbol{\mu}_i, \boldsymbol{\rho}_i$. R_i is normalized by $\frac{\sqrt{n_v}}{\|\mathcal{W}\|}$ to make it invariable to size changes of G.

We perform structural modification iteratively. In each iteration, after the estimation of matching parameters and attributes (see Section 4.1), we eliminate the node with the

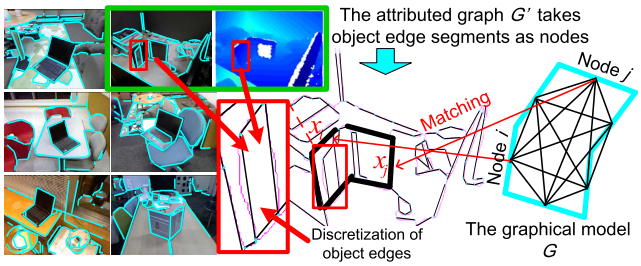

Figure 2. Attributed graph based on line segments of object edges

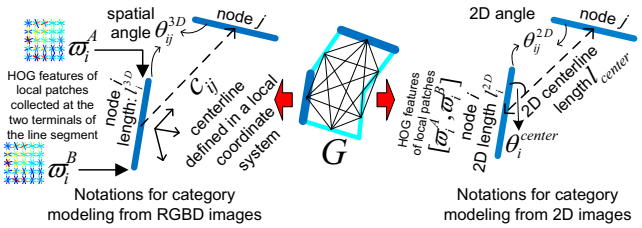

Figure 3. Notation for local and pairwise attributes

lowest reliability $i^* = \text{argmin}_{i \in V} R_i$ from the template, and use this induced subgraph of the template to retrain a new model, thereby replacing current model. Meanwhile, we calculate the classification accuracy in a leave-one-out (LOO) cross validation for each of the models. The stopping condition is that R_{i^*} is greater than a threshold τ and the elimination of node i^* would decrease the LOO accuracy. Please see Algorithm 1 for details.

5. Experiments

The proposed method is especially useful in the field of computer vision, enabling the discovery of general object structures for image matching when the target objects are randomly placed in cluttered scenes. To evaluate our method in this regard, we designed two category modeling experiments, one using ordinary RGB images and the other using RGBD images captured by a Kinect device [1].

We used the category dataset of Kinect RGBD images, published in [32] as a standard RGBD object dataset oriented to graph matching[3]. Four largest categories—*notebook PC*, *drink box*, *basket*, and *bucket*—in this dataset contained enough RGBD objects and were chosen to construct both the positive and the negative graph sets for training. These images depicted cluttered scenes containing target objects with different textures and rotations, and the both experiments were performed on these scenes.

We compared the proposed method with other approaches to learning graph matching and various common strategies in object extraction[4].

5.1. Category modeling from RGB & RGBD images

In cluttered scenes, objects in the same category usually contain a variety of textures, and may be positioned at various rotations. Considering the need for robustness with respect to texture variations, we applied two graphical models proposed in [32], each of which uses [2] to extract object edges and then discretizes continuous edges into line segments to produce the graph nodes (see Fig. 2). The two models use different attributes to represent objects in RGB and RGBD images, respectively. In this subsection, we briefly introduce these attributes. Please refer to [32] for more details.

Experiment 1: For category modeling from ordinary RGB images, one local attribute ($n^U = 1$) and three pairwise attributes ($n^P = 3$) are designed, as illustrated in Fig. 3. The local attribute is the HOG features [8] of two local patches collected at line segment terminals of node i, denoted by $f_i^{(1)} = [\varpi_i^A, \varpi_i^B]$. The first of the three pairwise attributes between nodes i and j is the angle between their line segments, denoted by $f_{ij}^{(1)} = \theta_{ij}^{2D}$. For the edge (i, j), we define the *centerline* as the line connecting the centers of the line segments of i and j. The second pairwise attribute describes the angles between the centerline and the node line segments, denoted by $f_{ij}^{(2)} = [\theta_i^{center}, \theta_j^{center}]$, where θ_i^{center} is the angle between the line segment of i and the centerline. The third pairwise attribute represents relative segment lengths, and is denoted by $f_{ij}^{(3)} = \frac{1}{l_{center}}[l_i^{2D}, l_j^{2D}]$, where l_i^{2D} and l_{center} are the lengths of the line segment of i and the centerline, respectively.

Experiment 2: For category modeling from RGBD images, two local attributes ($n^U = 2$) and three pairwise attributes ($n^P = 3$) are designed (see Fig. 3). The first local attribute is the HOG same feature used in Experiment 1. The second is given by $f_i^{(2)} = \log l_i^{3D}$, where l_i^{3D} is the spatial length of the line segment of node i. The first of the three pairwise attributes, $f_{ij}^{(1)} = \theta_{ij}^{3D}$, denotes the spatial angle between the line segments of nodes i and j in the 3D space. We then measure the centerline in a local 3D coordinate system independent of the global object rotation, as the relative spatial translation between two nodes, denoted by $\mathbf{c}_{ij}$. Based on this, the second and third pairwise attributes, $f_{ij}^{(2)} = \|\mathbf{c}_{ij}\|$ and $f_{ij}^{(3)} = \mathbf{c}_{ij}/\|\mathbf{c}_{ij}\|$, represent the length and local orientation of the centerline, respectively.

5.2. Experiments and quantitative evaluations

For both two experiments, we used the following compatibility function, corresponding to (3).

$$\begin{aligned} &\Phi(\mathbf{d}_{ii'}, \mathbf{d}_{jj'}, \mathbf{d}_{ii',jj'} | \mathbf{w}^U, \mathbf{w}^P) \\ &= \exp\left(-(\mathbf{w}^U)^{\mathbf{T}} \mathbf{d}_{ii'}^2 - (\mathbf{w}^U)^{\mathbf{T}} \mathbf{d}_{jj'}^2 - (\mathbf{w}^P)^{\mathbf{T}} \mathbf{d}_{ii',jj'}^2\right) \end{aligned} \quad (12)$$

There is, in fact, a fair variety of compatibility functions (*e.g.* $\exp(-\mathbf{w}^{\mathbf{T}}\mathbf{d})$ and $\alpha/(\beta + \mathbf{w}^{\mathbf{T}}\mathbf{d})$). Note that our proposed method is not limited to the particular compatibil-

[3]This is one of the largest RGBD object datasets, and fits the requirements of graph matching well. *http://sites.google.com/site/quanshizhang*

[4]Please see Section "Related work" for more discussion.

Figure 4. Object detection performance in RGBD images. We only show depth images corresponding to the first column.

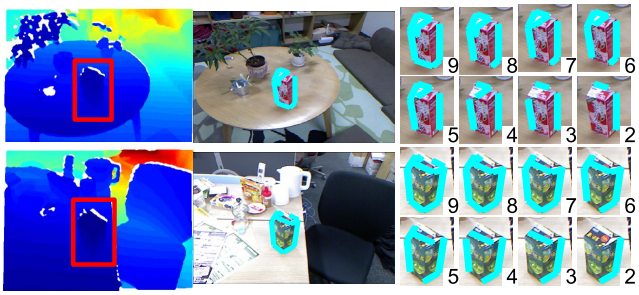

Figure 5. Process of node elimination. The bottom-right number indicates the model node number.

ity function used in these experiments. Any compatibility function that can be cast into the form of (2) or (3) will do.

We set the iteration number as $T = 5$ and the parameter for matching to *none* as $\lambda = 5$. The iteration number T is a general setting for [17] and is applied to all competing techniques of [17], so as to ensure fair comparison.

Cross validation and evaluation metrics: Each labeling of the target object in a given RGB or RGBD image can produce an initial graph template and begin an individual model learning process. We labeled the images for a given category in sequence to begin multiple learning processes. In each of these processes, the remaining (*i.e.* unlabeled) images of this category were used to generate positive graphs. We then randomly selected the same number of images from other categories to generate negative graphs. We used $2/3$ and $1/3$ of these graphs for training and testing in this learning process, respectively. The end result is a set of models for the evaluation of the category.

We used the average matching rate (AMR) to evaluate the matching performance in positive graphs. AMR is widely used to evaluate the learning of graph matching [17, 16]. The matching rate of each individual matching result indicates the proportion of model nodes that are correctly matched to the target object. AMR represents the average of individual matching rates across all matching results produced by the trained models. Similarly, the average recognition accuracy (ARA) (*i.e.* the average value for recognition accuracy in the cross validation) was used to evaluate model-based recognition performance among both positive and negative graphs.

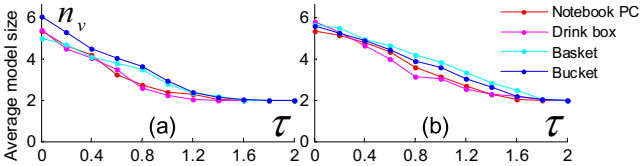

Figure 6. Average model size learned by using different thresholds of τ. (a) For 2D models learned from RBD images and (b) for 3D models learned from RGBD images.

Competing methods: In the first step of the evaluation, we compared our method to several competing approaches to the learning of graph matching across both experiments. First, we performed graph matching without training, denoted by *MA*, to establish a baseline. *MA* uses TRW-S [14] to match the initial graph template to the target objects in images. Second, as the benchmark method for unsupervised learning of graph matching, we used [17] proposed by Leordeanu *et al.*, which does not modify model structure, but rather iteratively train the attribute weights for matching, *i.e.* $\mathbf{w}^U$ and $\mathbf{w}^P$. Two competing approaches were obtained by applying spectral techniques [15] (*LS*) and TRW-S [14] (*LT*), respectively, to solve the matching optimization of (4). Third, we designed a framework that iteratively estimates matching parameters $\mathbf{w}^U, \mathbf{w}^P$ and model attributes F_V, F_E according to techniques presented in Section 4.1, but did not perform structural modification (*WM*). The final competing technique involved supervised learning of graph matching (*SU*) based on [17]. *SU* required to label target objects in positive graphs and regarded matching assignments mapped onto target objects as correct ones.

In the second step of the evaluation, we compared the performances of the proposed method using different structural modification strategies. The first strategy, (*CB*), is based on the matching compatibility/penalty, and has been

	Category	Category modeling from ordinary RGB images						Category modeling from RGBD images					
		MA	LS	LT	WM	SU	Ours^{avg}	MA	LS	LT	WM	SU	Ours^{avg}
Matching	Notebook PC	**56.05**	46.40	48.83	49.93	52.90	50.74	62.02	57.40	61.94	61.91	67.71	**67.79**
	Drink box	56.15	49.46	51.24	58.55	52.70	**89.11**	59.11	55.66	57.91	62.41	60.68	**88.25**
	Basket	55.28	50.25	50.97	59.60	51.68	**81.33**	60.88	56.99	59.59	66.10	61.52	**88.61**
	Bucket	58.02	54.74	56.39	61.76	56.80	**86.85**	61.13	59.15	60.51	64.41	61.90	**89.36**
Recognition	Notebook PC	67.63	67.36	66.60	**77.07**	67.98	75.21	75.41	70.94	72.11	**82.78**	76.79	81.51
	Drink box	72.74	69.73	68.75	82.47	72.05	**89.53**	80.09	76.33	78.36	87.73	78.65	**93.42**
	Basket	63.95	67.13	67.01	74.94	66.32	**76.39**	73.09	70.66	74.59	86.11	71.88	**87.71**
	Bucket	80.90	78.96	77.83	86.12	78.54	**89.07**	75.52	78.34	77.18	84.63	77.36	**87.97**

Table 1. Comparison of average matching rates and average recognition accuracy.

		Methods	Average recognition accuracy				Average matching rate			
			Notebook	Drink box	Basket	Bucket	Notebook	Drink box	Basket	Bucket
The average performance	RGB images	Ours+CB^{avg}	60.96	81.72	72.51	85.58	37.90	87.75	**82.56**	**88.11**
		Ours+WB^{avg}	**75.36**	87.38	73.87	88.18	49.12	82.89	74.20	84.50
		Ours^{avg}	75.21	**89.53**	**76.39**	**89.07**	**50.74**	**89.11**	81.33	86.85
	RGBD images	Ours+CB^{avg}	76.46	85.80	85.67	86.48	63.71	88.06	**93.44**	**90.62**
		Ours+WB^{avg}	80.63	90.64	86.51	86.77	65.12	84.75	87.69	86.89
		Ours^{avg}	**81.51**	**93.42**	**87.71**	**87.97**	**67.79**	**88.25**	88.61	89.36
The best performance	RGB images	Ours+CB^{best}	76.17	85.76	78.94	89.01	50.39	90.72	**85.45**	89.78
		Ours+WB^{best}	80.03	87.91	77.84	89.67	51.71	86.81	76.09	86.29
		Ours^{best}	**82.16**	**90.45**	**82.52**	**91.81**	**56.12**	**94.10**	84.37	**90.22**
	RGBD images	Ours+CB^{best}	84.92	91.32	89.77	88.56	74.40	**93.07**	**96.16**	**91.34**
		Ours+WB^{best}	84.78	91.78	89.06	88.57	67.74	89.59	90.40	89.64
		Ours^{best}	**88.57**	**94.85**	**92.01**	**90.63**	**76.92**	93.02	92.40	90.69

Table 2. Comparison of different structural modification strategies that can be used in the proposed learning framework.

widely used by [10, 30, 29]. *CB* eliminates the node with the lowest average compatibility by replacing (11) with $R_i = \sum_k \sum_{j \in V} c_{ij}(\hat{x}_i^k, \hat{x}_j^k | G, G_k^+)/n_v$. The second strategy [3], (*WB*), is oriented toward linear SVM, and uses weights $\mathcal{W}$ for feature selection. *WB* eliminates the node with the smallest weight amplitude by replacing (11) with $R_i = \|[\boldsymbol{\mu}_i, \boldsymbol{\rho}_i]^T\|$. Note that for this step of the evaluation, all learning components expect the above structural modification strategies are fixed.

Comparison details: Since our method can obtain different models by setting different values of τ for structural modification, we set τ to be $0, 0.2, 0.4, ..., 2$ during training. This produced different matching and recognition performances, *i.e.* different AMRs and ARAs. Fig. 6 shows the changes in model size according to τ, and Fig. 5 illustrates the models in the node elimination process. Larger values of τ indicate stricter structural constraints and lead to smaller models.

We used the average/best performance among all settings of τ (*i.e.* the average/largest values for AMR and ARA) to evaluate the proposed method, denoted by *Ours*^{avg}/*Ours*^{best}. To ensure a fair comparison, for each given τ, *CB* and *WB* were allowed to eliminate nodes until they obtained models with the same size as that produced by *Ours*. Similar to *Ours*^{avg}/*Ours*^{best}, *CB*^{avg}/*CB*^{best} and *WB*^{avg}/*WB*^{best} correspond to the average/best performance among all setting of τ. For the comparison of recognition performance, we trained the proposed classifiers using the matching results produced by the competing methods.

Fig. 4 illustrates the object detection performances, and Table 1 and Table 2 list quantitative comparison. Because the competing methods for learning graph matching (*MA*,*LS*,*LT*,*WM*,*SU*) do not have the ability to refine the topological structure of the graph template, they are sensitive to the bias of the initially labeled graph template (including biased attributes and redundant nodes), especially for the unsupervised methods *LS*,*LT*,and *WM*. This bias may produce matching errors, which, in turn, increase the bias in the unsupervised model learning, thus propagating into a significant model bias. In contrast, our method modifies biased structure in early iterations by eliminating badly matched parts, thereby reducing the prevalence of biased matching in later iterations. As a result, our method exhibits better performance.

6. Conclusions

In this paper, we proposed an algorithm for the learning of graph matching. This method trains the structure and attributes of the graph template, as well as matching parameters, to obtain a graphical model. By including negative graphs in the learning process, we orient the model learning toward both object matching and recognition. Experiments show that our approach outperforms competing methods.

Our strategy for structural modification is based on the recognition mechanisms between positive and negative

graphs, and exhibits better performance than conventional structural modification strategies based only on positive graphs. As the proposed strategy iteratively corrected errors in the topological structure of the initial graph template, it reduced the bias learning problem, which so afflicted pioneering work in the field.

7. Acknowledgement

This work is supported by Microsoft Research, a Grant-in-Aid for Young Scientists (23700192) of Japans Ministry of Education, Culture, Sports, Science, and Technology (MEXT), 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.* 5

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *In* PAMI, 33(5):898–916, 2011. 5

[3] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Feature selection using support vector machines. *In* international conf. on data mining methods and databases for engineering, 2002. 7

[4] Z. P. C. K. Liu, A. Hertzmann. Learning physics-based motion style with nonlinear inverse optimization. *In* SIGGRAPH, 2005. 2

[5] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *In* PAMI, pages 1048–1058. 2

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

[7] M. Cho, Y. M. Shin, and K. M. Lee. Unsupervised detection and segmentation of identical objects. In *CVPR*, 2010. 2

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

[9] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. *In* ICCV, 2011. 4

[10] P. Hong and T. S. Huang. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *In* Discrete Applied Mathematics, 139:113–135, 2004. 2, 7

[11] J. Huan, W. Wang, J. Prins, and J. Yang. Spin: Mining maximal frequent subgraphs from graph databases. In *ACM SIGKDD*, 2004. 2

[12] H. Jiang and C.-W. Ngo. Image mining using inexact maximal common subgraph of multiple args. *In* International conference on visual information system, 2003. 2

[13] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 2

[14] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *In* IEEE PAMI, 28(10):1568–1583, 2006. 3, 6

[15] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *In* ICCV, 2005. 2, 3, 6

[16] M. Leordeanu and M. Hebert. Smoothing-based optimization. *In* CVPR, 2008. 2, 6

[17] M. Leordeanu and M. Hebert. Unsupervised learning for graph matching. *In* CVPR, 2009. 1, 2, 3, 6

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

[19] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. *In* CVPR, 2010. 2

[20] D. Parikh, C. Zitnick, and T. Chen. Unsupervised learning of hierarchical spatial structures in images. In *CVPR*, 2009. 2

[21] B. G. Park, K. M. Lee, S. U. Lee, and J. H. Lee. Recognition of partially occluded objects using probabilistic arg-based matching. *In* CVIU, 90(3):217–241, 2003. 4

[22] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *ECCV*, 2006. 1

[23] H.-K. Tan and C.-W. Ngo. Localized matching using earth movers distance towards discovery of common patterns from small image samples. *In* Image and Vision Computing, 27:1470–1483, 2009. 2

[24] L. Thomas, S. Valluri, and K. Karlapalem. Margin: Maximal frequent subgraph mining. In *Transactions on KDD*, 4(3), 2010. 2

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

[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *In* J. Machine Learning Research, 6:1453–1484, 2005. 2

[27] C. S. Vittorio Ferrari, Frederic Jurie. From images to shape models for object detection. *In* IJCV, 87(3):284–303, 2010. 2

[28] J. Wang, Z. Zeng, and L. Zhou. Clan: An algorithm for mining closed cliques from large dense graph databases. *In* ACM SIGKDD, 2006. 2

[29] H. Xie, K. Gao, Y. Zhang, J. Li, and H. Ren. Common visual pattern discovery via graph matching. *In* ACM MM, 2012. 2, 7

[30] J. Yuan, G. Zhao, Y. Fu, Z. Li, A. Katsaggelos, and Y. Wu. Discovering thematic objects in image collections and videos. *In* PAMI, 21(4):2207–2219, 2012. 2, 7

[31] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Coherent closed quasi-clique discovery from large dense graph databases. *In* ACM SIGKDD, 2006. 2

[32] 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* CVPR, 2013. 4, 5

[33] G. Zhao and J. Yuan. Mining and cropping common objects from images. *In* ACM MM, 2010. 2