This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Attributed Graph Mining and Matching: An Attempt to Define and Extract Soft Attributed Patterns

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

Abstract

*Graph matching and graph mining are two typical areas in artificial intelligence. In this paper, we define the soft attributed pattern (SAP) to describe the common subgraph pattern among a set of attributed relational graphs (ARGs), considering both the graphical structure and graph attributes. We propose a direct solution to extract the SAP with the maximal graph size without node enumeration. Given an initial graph template and a number of ARGs, we modify the graph template into the maximal SAP among the ARGs in an unsupervised fashion. The maximal SAP extraction is equivalent to learning a graphical model (*i.e. *an object model) from large ARGs (*i.e. *cluttered RGB/RGB-D images) for graph matching, which extends the concept of "unsupervised learning for graph matching." Furthermore, this study can be also regarded as the first known approach to formulating "maximal graph mining" in the graph domain of ARGs. Our method exhibits superior performance on RGB and RGB-D images.*

The code will be published later.

1. Introduction

We categorize this research under the fields of both graph matching and graph mining to better explain the concept extensions that range across these two fields.

Views of graph matching & task introduction: Attributed relational graphs (ARGs) are widely used. For example, in computer vision, ARGs can represent scenes or objects, taking part features and the spatial relationship between the parts as the local and pairwise attributes, respectively. Attributed graph matching aims to estimate node correspondences between a small graph template (an object) and a large ARG (an image), based on the similarity of local and pairwise attributes. In the general case[1] of ARGs, the graph matching is formulated as a quadratic assignment problem (QAP) and requires global optimization.

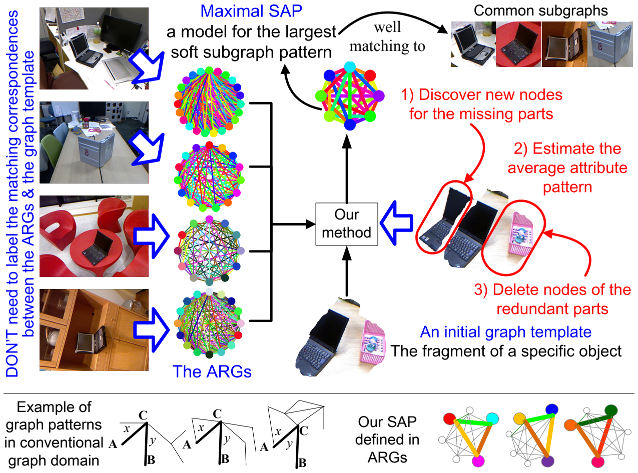
DON'T need to label the matching correspondences between the ARGs & the graph template
Maximal SAP
a model for the largest soft subgraph pattern
well matching to
Common subgraphs
1) Discover new nodes for the missing parts
2) Estimate the average attribute pattern
Our method
3) Delete nodes of the redundant parts
An initial graph template
The fragment of a specific object
The ARGs
Example of graph patterns in conventional graph domain
C
x
y
A
B
Our SAP defined in ARGs

Figure 1. Concept extension. We extract the soft attributed pattern (SAP) from ARGs, and the size of the SAP is maximized. This study overcomes a key challenge in graph mining, as it can be considered as the mining of maximal frequent subgraphs defined in the graph domain of ARGs, rather than conventional labeled graphs. From another perspective, we also extend the concept of unsupervised[2] learning for graph matching. Given an initial graph template and a set of large ARGs, we simultaneously discover the missing nodes, delete redundant nodes, and train attributes, so as to obtain a graphical model with good matching performance.

The learning for graph matching has demonstrated its superior performance in terms of improving matching accuracy. Indeed, the concept of learning graph matching has been extended in recent years. We focus on unsupervised[2] approaches, which are analogous to automatic category modeling from cluttered scenes. [17] mainly trains the attribute weights for matching, and [30, 6] further incorporate structural refinement of the graph template.

In this paper, we propose a new concept of learning graph matching, by considering the discovery of missing graph parts[3]*,* i.e. *the ability to recover a general graphical pattern from a fragmentary graph template, as shown in Fig.3.* This is orthogonal to conventional unsupervised

[1] Unlike ARGs in [12], local attributes in ARGs may not, in general, be sufficiently distinguished to independently provide matching correspondences (candidates) between ARGs without global optimization.

[2] The word "unsupervised" means that people do not need to label the matching correspondences manually for training.

[3] The missing parts appear in different ARGs with similar local attributes, and have a similar pairwise relationship with the graph template.

methods for learning attribute weights and structural refinement. Given a graph template and a number of ARGs, our method simultaneously 1) discovers missing parts of the template, 2) eliminates redundant parts, and 3) adjusts its attributes in an unsupervised manner, so as to grow the initial template into the common subgraph pattern among these ARGs, and achieve good matching performance (Fig. 1).

Views of graph mining & the proposed method: From another perspective, the proposed learning of graph matching can be understood as the mining of maximal[4] graph patterns. In fact, related techniques in graph mining, such as maximal frequent subgraph (MFS) extraction and maximal clique mining, have been extensively investigated and developed in the past.

However, a bottleneck for maximal pattern mining lies in the constraints of the target graphs. It is mainly applied to graphs that are generated from tabular data, and have distinguishing structures, distinct node and edge labels, or a list of pre-determined potential node correspondence candidates.

Therefore, we extend this field to much "fuzzier[1]" ARGs, whose matching is formulated as a QAP based on attribute similarities. As shown in Fig. 1(bottom), conventional graph patterns based on graph isomorphisms can no longer be applied. *Alternatively, we redefine the concept of the graph pattern as a new "soft" attributed pattern (SAP).* Consequently, the learning process actually extracts the S-AP with the maximal graph size among the ARGs.

Moreover, for the mining of maximal graph patterns, existing methods based on node enumeration are hampered in the new graph domain of ARGs. They produce NP-hard solutions. Fortunately, we demonstrate the existence of an approximate solution to the discovery of new nodes from ARGs without any enumeration, using the typical squared penalties in graph matching.

Summary: The contributions of this paper can be summarized as follows. 1) This study redefines the concept of unsupervised[2] learning for graph matching, as it idealizes the spirit of training graphical structures. To the best of our knowledge, this is the first attempt to formulate the discovery of missing parts into the theory of attributed graph matching. 2) In terms of graph mining, this research also extends the target domain for the mining of maximal graph patterns to fuzzy ARGs. 3) In this new graph domain, we propose the maximal SAP to define the graph pattern, and demonstrate the existence of a direct solution that does not require computationally intensive node enumeration.

1.1. Brief introduction of potential applications

This method can be considered as a general platform for model learning from cluttered scenes, where small target objects are unlabeled and randomly localized in large images with some variations in texture, rotation, and structure. Thus, it has many extended applications, *e.g.* learning 3D reconstruction from ubiquitous RGB-D images [31]. In addition, given the appropriate design of unary and pairwise attributes, we can use this method to either recover global object shapes from fragments (Fig. 3) or train models for matching deformable objects.

Of course, the application is not limited within the field of computer vision. We can apply this approach to any problem that is formulated using ARGs.

2. Related work

Views of graph matching: Given a graph template and a number of ARGs, conventional algorithms for learning graph matching [4, 16, 17, 24] mainly train matching parameters, and Cho *et al.* [5] proposed to learn a model for matching. Most of them are supervised methods that require the target subgraphs in ARGs to be labeled for training. Leordeanu *et al.* [17] proposed the first unsupervised method that did not require such manual labeling.

[30] considered the structural refinement as a part of the unsupervised learning for graph matching. [9] utilized a similar idea to mine spatial patterns from ARGs. [6] aimed to learn the node linkage, and this can also be regarded as structural refinement. However, they only consider the matching between two ARGs.

Essentially, structural refinement just deletes "bad" nodes from the graph template, rather than recovering the prototype graph patterns. Therefore, to perfect the learning of a graph structure, we encode the challenging task, *i.e.* the discovery of missing parts from large ARGs, into our definition of learning graph matching.

Views of graph mining: The discovery of missing parts relates this study to the mining of maximal[4] graph patterns, as it is usually meaningful to mine the pattern with the maximal graph size. This idea has been realized by MF-S extraction [22] and maximal clique mining [25, 28] in the field of graph mining (reviewed in [11]).

As shown in Fig. 1(bottom), MFS extraction [22, 12, 10, 23] is based on graph isomorphisms and usually require 1) the distinguishing (topological) structure of the graph pattern, and 2) pre-defined distinct node/edge labels or potential inter-graph node correspondences determined by local consistency. Thus, node numeration is used to mine the MFS. The distinct graph structure and labels are used to prune the enumeration range, thereby avoiding possible NP-hard computation. Similarly, maximal clique mining [25, 28, 26, 19] mainly extracts a dense graph clique to maintain geometric consistency during matching. This method also requires distinguishing local features to pre-determine local matching correspondence candidates among graphs (as discussed in "Views of applications").

In contrast, fuzzily defined ARGs usually have neither

[4] The word "maximal" indicates that we should grow the target graph pattern so as to obtain the maximal graph size.

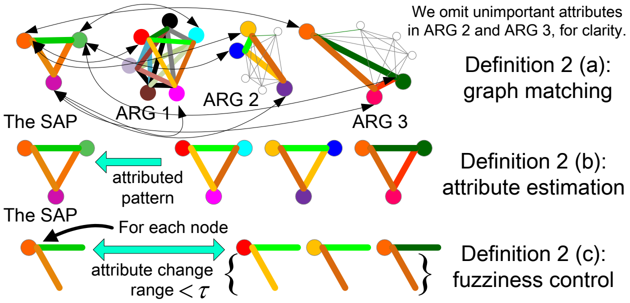
We omit unimportant attributes in ARG 2 and ARG 3, for clarity.
Definition 2 (a): graph matching
The SAP
ARG 1
ARG 2
ARG 3
attributed pattern
Definition 2 (b): attribute estimation
The SAP
For each node
attribute change range < τ
Definition 2 (c): fuzziness control

Figure 2. Visualization of the SAP in Definition 2. Colors in ARGs denote different local and pairwise attributes. Note that in graph matching, we use pairwise attributes (edge colors), rather than only geometric distances between nodes (although such distances can be used as one of the N_Q types of pairwise attributes).

distinguishing structures nor distinct node labels. In many applications, nodes are connected in a uniform style and have not-so-strong local attributes. Thus, it requires a new mining strategy (without node enumeration) to deal with the ARGs. Considering the fuzzy condition[1], the matching between ARGs can only be solved by global optimization (*i.e.* a QAP). Thus, we redefine the graph pattern as the SAP based on the attributes' consistency, rather than a graph isomorphism *w.r.t* the structure and labels.

Views of applications: For some certain graph-matching applications, the iterative methods for estimating common graph structures have been proposed [18, 3], which is a pioneer that discovered common structural patterns of edges from images. In fact, many visual applications involve the detection of common objects (or co-appearing parts) in a set of images. However, they are mainly designed with some data-driven techniques oriented to their own applications. For example, many studies of common object extraction from images [13, 21, 27, 32, 7, 20] use techniques related to maximal clique mining [25, 28] to some extent. They thereby require target objects to have high-quality patch features (with little texture variations), thus pre-determining a set of potential inter-image matching correspondences using local features. In contrast, our approach is formulated in the theory system of attributed graph matching. Thus, it is not limited to some specific CV applications, although we use certain ARGs generated from RGB and RGB-D images for testing.

3. Problem formulation

Definition 1 **(ARG)** *An ARG G is a three element tuple $G = (V, \mathbf{F}_V, \mathbf{F}_{V\times V})$, where V is the node set. Undirected edges connect each pair of nodes to form a completed graph. G contains N_P types of local attributes for each node and N_Q types of pairwise attributes for each edge. $\mathbf{F}_V = \{\mathcal{F}_i^s | s \in V, i = 1, 2, ..., N_P\}$ and $\mathbf{F}_{V\times V} = \{\mathcal{F}_j^{st} | s, t \in V, s \neq t, j = 1, 2, ..., N_Q\}$ denote the local and pairwise attribute sets, respectively. Each attribute corresponds to a feature vector.*

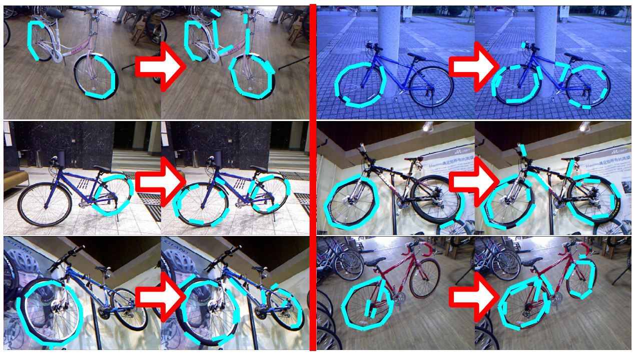

Figure 3. Structure modification from different graph templates (object fragments) to SAPs (fuzziness $\tau = 0.4$).

Actually, this definition can be extended to incomplete graphs with the form $G^* = (V, E, \mathbf{F}_V, \mathbf{F}_E)$. We can transform G^* to our ARG by setting a pairwise attribute $\mathcal{F}_j^{st} = 1$ if edge $(s,t) \in E$, and 0 otherwise.

Attributed graph matching: Given a set of ARGs $GS = \{G'_k | k = 1, 2, ..., N\}$, $G'_k = (V_k, \mathbf{F}_{V_k}, \mathbf{F}_{V_k \times V_k})$, the graph template $G = (V, \mathbf{F}_V, \mathbf{F}_{V\times V})$ represents an attribute pattern among the ARGs in GS, and is not exactly embedded in any G'_k. The matching between G and G'_k aims to compute a set of matching assignments between G and G'_k, denoted by $\mathbf{x}^k = \{x_s^k | s \in V\}$. Each matching assignment $x_s^k \in V_k \cup \{none\}$ maps node s in G to a node in G'_k or a dummy choice $none$. $none$ is used when some nodes in G do not exist in G'_k. The attributed graph matching is formulated as a typical QAP with the following energy function:

$$\mathcal{E}(\mathbf{x}^k | G, G'_k) = \sum_{s\in V} P_s(x_s^k | G, G'_k) + \sum_{(s,t)\in V, s\neq t} Q_{st}(x_s^k, x_t^k | G, G'_k) \quad (1)$$

Function $\mathcal{E}(\mathbf{x}^k | G, G'_k)$ indicates the total matching energy. The functions $P_s(\cdot)$ and $Q_{st}(\cdot,\cdot)$ denote matching penalties for local and pairwise attributes. Various graph matching optimization techniques can solve the energy minimization of $\mathcal{E}(\mathbf{x}^k | G, G'_k)$, and we choose TRW-S [14]. In this study, matching penalties are defined using squared differences.

$$P_s(x_s^k | G, G'_k) = \begin{cases} \sum_{i=1}^{N_P} w_i^P \|\mathcal{F}_i^s - \mathcal{F}_i^{x_s^k}\|^2, & x_s^k \in V_k \\ P_{none}, & x_s^k = none \end{cases} \quad (2a)$$

$$Q_{st}(x_s^k, x_t^k | G, G'_k) = \begin{cases} \frac{\sum_{j=1}^{N_Q} w_j^Q \|\mathcal{F}_j^{st} - \mathcal{F}_j^{x_s^k x_t^k}\|^2}{\|V\|-1}, & x_s^k \neq x_t^k \in V_k \\ +\infty, & x_s^k = x_t^k \in V_k \\ \frac{Q_{none}}{\|V\|-1}, & x_s^k \text{ or } x_t^k = none \end{cases} \quad (2b)$$

where P_{none} and Q_{none} are relatively large constant penalties for matching to $none$. $\|\cdot\|$ is the Euclidean norm. We use infinite penalties to avoid many-to-one matching assignments. w_i^P and w_j^Q denote the weights for local and pairwise attribute differences. We require the pairwise penalty to be symmetric, *i.e.* $Q_{st}(x_s, x_t | G, G'_k) = Q_{ts}(x_t, x_s | G, G'_k)$, and to be normalized by $(\|V\| - 1)$[5].

[5] During the learning process, we insert/delete missing/redundant n-

Definition 2 **(SAP)** *Given a set of ARGs $GS = \{G'_k|k = 1, 2, ..., N\}$ and a threshold τ, a graph template $G = (V, \mathbf{F}_V, \mathbf{F}_{V\times V})$ is an SAP among the ARGs in GS, iff*

(a) $\hat{\mathbf{x}}^k = \text{argmin}_{\mathbf{x}^k}\, \mathcal{E}(\mathbf{x}^k|G, G'_k)$*; we set* $\hat{\mathbf{X}} = \bigcup_{k=1}^{N} \hat{\mathbf{x}}^k = \{x_s^k|s \in V, k = 1, 2, ..., N\}$*;*

(b) $(\mathbf{F}_V, \mathbf{F}_{V\times V}) \leftarrow \text{argmin}_{\mathbf{F}_V,\mathbf{F}_{V\times V}} \sum_{k=1}^{N} \mathcal{E}(\hat{\mathbf{x}}^k|G, G'_k)$*;*

(c) $\forall s \in V$, $E_s(\hat{\mathbf{X}}|G, GS) \leq \tau$*;*

where $E_s(\hat{\mathbf{X}}|G, GS)$ is defined as the average matching penalty related to node s in G among all the ARGs in GS.

$$E_s(\hat{\mathbf{X}}|G,GS)=\frac{1}{N}\sum_{k=1}^{N}\Big[P_s(\hat{x}_s^k|G, G'_k)+\sum_{t\in V,t\neq s}Q_{st}(\hat{x}_s^k,\hat{x}_t^k|G, G'_k)\Big]$$

Maximal SAP: The definition of the SAP can be visualized in Fig. 2, and we introduce the physical meaning of each item in Definition 2, as follows.

Condition (a) directly matches the SAP G to each large ARG G'_k in GS to determine the SAP's corresponding subgraphs embedded in these ARGs.

Condition (b) trains the local and pairwise attributes of the SAP G. G should represent the average attribute pattern among all its corresponding subgraphs determined by ***Condition (a)***. In other words, the SAP's attributes $(\mathbf{F}_V, \mathbf{F}_{V\times V})$ should minimize the total matching energy, given all the matches between G and the ARGs in GS.

Condition (c) sets a threshold τ to control the fuzziness of G. We require each node s in the SAP to have a low average matching penalty among all the matches to ensure that all the SAP's nodes represent the common parts in the ARGs.

With these preliminaries, our goal is to mine the SAP G with maximal graph size $\|V\|$, *i.e.* the largest common subgraph pattern among the ARGs.

4. Algorithm

To extract a maximal SAP, the initial graph template G is modified in the following EM framework. In each iteration, we use the current G to estimate the matching assignments in the ARGs in GS, $\hat{\mathbf{X}}$, and then use $\hat{\mathbf{X}}$ to update the attribute sets of $\mathbf{F}_V$ and $\mathbf{F}_{V\times V}$ of G. The new $\mathbf{F}_V$ and $\mathbf{F}_{V\times V}$ are finally used as feedback to modify the structure of G by (probably) discovering a missing node from the ARGs, or deleting a redundant one. Thus, the initial graph template G is iteratively modified to the maximal SAP (see Fig. 3).

Attribute estimation: According to Definition 2(a), matching assignments $\hat{\mathbf{X}}$ are first estimated based on the current G. We then use Definition 2(b) to estimate the local and pairwise attributes of G, given $\hat{\mathbf{X}}$. As $\mathcal{E}(\hat{\mathbf{x}}^k|G, G'_k)$ is a convex function with respect to $\mathbf{F}'_V$ and $\mathbf{F}'_{V\times V}$ (see (1) and (2)), the minimization problem can be directly solved by $\frac{\partial \sum_{k=1}^{N} \mathcal{E}(\hat{\mathbf{x}}^k|G,G'_k)}{\partial \mathcal{F'}_i^s} = 0$ and $\frac{\partial \sum_{k=1}^{N} \mathcal{E}(\hat{\mathbf{x}}^k|G,G'_k)}{\partial \mathcal{F'}_i^{st}} = 0$. We thus have that G's attributes are equal to the average attributes among all subgraphs matched to G in the ARGs.

$$\mathcal{F}_i^s = \underset{k:\delta(\hat{x}_s^k)=1}{\text{average}}\, \mathcal{F}_i^{\hat{x}_s^k}, \quad \mathcal{F}_i^{st} = \underset{k:\delta(\hat{x}_s^k)\delta(\hat{x}_t^k)=1}{\text{average}}\, \mathcal{F}_i^{\hat{x}_s^k \hat{x}_t^k} \qquad (3)$$

where $\delta(\cdot)$ indicates whether a node in G is matched to $none$. If $\hat{x}_s^k = none$, $\delta(\hat{x}_s^k)$ is set to 0; otherwise 1.

Structure modification: We grow the initial G into the maximal SAP using a greedy strategy (see Algorithm 1). In each iteration, we delete the "worst" (not well matched to the ARGs) node from G, and insert the "most probable" missing node. Both the insertion and elimination depend on the unified requirement for the local matching quality in Definition 2(c). We choose node $\hat{s} = \arg\max_{s\in V} E_s(\hat{\mathbf{X}}|G, GS)$ in G. If $E_{\hat{s}}(\hat{\mathbf{X}}|G, GS) > \tau$, we delete $\hat{s}$ from G; otherwise, this node is retained.

The key part is the node insertion. This involves two issues, i.e. *the attribute estimation of the new node and the determination of its matching assignments to the ARGs. However, this has the appearance of a chicken-and-egg problem.* On the one hand, the local and pairwise attributes related to the new node represent the pattern of their corresponding nodes and edges in ARGs[6]. They are thus determined by the matching assignments of the new node (see

Algorithm 1 Maximal SAP extraction

```
Input: The initial graph template $G = (V, \mathbf{F}_V, \mathbf{F}_{V\times V})$; a set of ARGs $GS = \{G'_k|k = 1, 2, ..., N\}$, where $G'_k = (V_k, \mathbf{F}_{V_k}, \mathbf{F}_{V_k\times V_k})$; a threshold $\tau$ controlling the SAP's fuzziness; the maximum iteration number $M$.
for $iteration = 1$ to $M$ do
    1. Use the current $G$ to estimate matching assignments in all the $N$ ARGs as $\hat{\mathbf{X}}$ (see Definition 2(a)).
    2. Given $\hat{\mathbf{X}}$, update the attribute sets $\mathbf{F}_V$ and $\mathbf{F}_{V\times V}$ of $G$ (see (3) and Definition 2(b)).
    3. With the updated attributes, compute the local matching penalty $E_s(\hat{\mathbf{X}}|G, GS)$ of each node $s$ in $G$ matching the ARGs in $GS$. Select the worst node $\hat{s} = \arg\max_{s\in V} E_s(\hat{\mathbf{X}}|G, GS)$, and if $E_{\hat{s}}(\hat{\mathbf{X}}|G, GS) > \tau$, then delete $\hat{s}$ from $G$ (see Definition 2(c)).
    4. Create a new node $y$ as the potential missing node of $G$, and thus construct $G^{new}$. Estimate the optimal attributes and matching correspondences for $y$ (see (6) and (8)). If $E_y(\mathbf{X}^{new}|G^{new}, GS) \leq \tau$, then insert node $y$ into $G$. (see Definition 2(c))
end for
```

odes into/from G to obtain the maximal SAP. However, this operation will increase/decrease the overall weights for pairwise attributes in both the graph matching (1) and the calculation of $E_s(\hat{\mathbf{X}}|G, GS)$, causing an unstable performance. Therefore, we normalize $Q_{st}(x_s, x_t|G, G'_k)$ using $(\|V\| - 1)$ to prevent such effects.

[6] Conventional enumeration of the new nodes in the graphs cannot en-

Definition 2(b)). On the other hand, the matching of the new node cannot be applied without knowing its attributes.

Fortunately, we have developed an efficient solution that simultaneously determines the attributes and matching assignments of the missing node, thus overcoming the chicken-and-egg problem. Let y be the missing node of G, and let $\mathbf{F}_y=\{\mathcal{F}_i^y|1\leq i\leq N_P\}$ and $\mathbf{F}_{\{y\}\times V}=\{\mathcal{F}_j^{yt},\mathcal{F}_j^{ty}|t\in V,1\leq j\leq N_Q\}$ denote the local and pairwise attribute sets related to y. Consequently, in ARG G'_k, the node matched by y can be denoted by $x_y^k\in V_k\setminus\hat{\mathbf{x}}^k$ ($\hat{\mathbf{x}}^k=\{\hat{x}_s^k|s\in V\}$). Thus, y's matching assignments in all the ARGs are denoted by $\mathbf{X}_y=\{x_y^k|k=1,2,..,N\}$.

We use $G^{new}=(V^{new},\mathbf{F}_{V^{new}},\mathbf{F}_{V^{new}\times V^{new}})$ to denote the dummy enlarged model after node insertion. We define the notation for G^{new} in the same way as that for G: $V^{new}=V\cup\{y\}$, $\mathbf{F}_{V^{new}}=\mathbf{F}_V\cup\mathbf{F}_y$, $\mathbf{F}_{V^{new}\times V^{new}}=\mathbf{F}_{V\times V}\cup\mathbf{F}_{\{y\}\times V}$, $\mathbf{x}_{new}^k=\hat{\mathbf{x}}^k\cup\{x_y^k\}$, $\mathbf{X}^{new}=\bigcup_{k=1}^N\mathbf{x}_{new}^k$.

Thus, the local matching penalty of y is transformed to

$$\begin{aligned}
&E_y(\mathbf{X}^{new}|G^{new},GS)=\mathbf{P}_y+\textstyle\sum_{t\in V}\mathbf{Q}_{yt}\\
&\mathbf{P}_y=\textstyle\sum_{k=1}^N P_y(x_y^k|G^{new},G'_k)/N \qquad (4)\\
&\mathbf{Q}_{yt}=\textstyle\sum_{k=1}^N Q_{yt}(x_y^k,\hat{x}_t^k|G^{new},G'_k)/N
\end{aligned}$$

The goal of node insertion is transformed to

$$\arg\min_{\mathbf{F}_y,\mathbf{F}_{\{y\}\times V}}\sum_{k=1}^N\mathcal{E}(\mathbf{x}_{new}^k|G^{new},G'_k) \qquad (5a)$$

$$\underset{\mathbf{X}_y}{\text{argmin}}E_y(\mathbf{X}^{new}|G^{new},GS) \qquad (5b)$$

Equation 5a corresponds to Definition 2(b). Given the matching assignments of y in the ARGs ($\mathbf{X}_y$), y's attributes ($\mathbf{F}_y,\mathbf{F}_{\{y\}\times V}$) are trained to minimize the matching energy, *i.e.* representing the attributed pattern among the ARGs.

Equation 5b estimates y's matching assignments ($\mathbf{X}_y$) based on Definition 2(c). Given the attributes of y, the nodes in ARGs matched by y ($\mathbf{X}_y$) should have similar attributes to y. In other words, they should minimize the local matching penalty ($E_y(\mathbf{X}^{new}|G^{new},GS)$). If $E_y(\mathbf{X}^{new}|G^{new},GS)<\tau$, then y satisfies Definition 2(c).

Similar to (3), attributes in $\mathbf{F}_y$ and $\mathbf{F}_{\{y\}\times V}$ are represented by x_y^k as the solution to (5a).

$$\begin{aligned}
\mathcal{F}_i^y&=\underset{k:\delta(x_y^k)=1}{\text{average}}\mathcal{F}_i^{x_y^k}=\textstyle\sum_{k=1}^N\mathcal{F}_i^{x_y^k}/N\\
\mathcal{F}_i^{yt}&=\underset{k:\delta(x_y^k)\delta(\hat{x}_t^k)=1}{\text{average}}\mathcal{F}_i^{x_y^k\hat{x}_t^k}=\underset{k:\delta(\hat{x}_t^k)=1}{\text{average}}\mathcal{F}_i^{x_y^k\hat{x}_t^k} \qquad (6)\\
\mathcal{F}_i^{ty}&=\underset{k:\delta(\hat{x}_t^k)\delta(x_y^k)=1}{\text{average}}\mathcal{F}_i^{\hat{x}_t^k x_y^k}=\underset{k:\delta(\hat{x}_t^k)=1}{\text{average}}\mathcal{F}_i^{\hat{x}_t^k x_y^k}
\end{aligned}$$

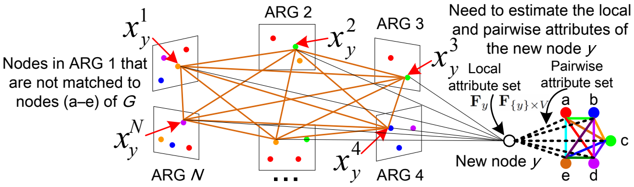
x_y^1
ARG 2
x_y^2
ARG 3
Need to estimate the local and pairwise attributes of the new node y
Nodes in ARG 1 that are not matched to nodes (a–e) of G
x_y^3
Local attribute set
Pairwise attribute set
$\mathbf{F}_y$
$\mathbf{F}_{\{y\}\times V}$
a
b
c
x_y^N
New node y
e
d
ARG N
...
x_y^4
ARG 4

Figure 4. Discovery of the missing node y in G. We have demonstrated a direct solution to the determination of y's matching assignments $\mathbf{X}_y=\{x_y^k\}$ in the N ARGs that minimize $E_y(\mathbf{X}^{new}|G^{new},GS)$, without requiring any prior knowledge of y's attributes. The ARGs are connected to each other to construct a Markov random field that solves this problem.

where $\delta(x_y^k)=1, k=1,2,...,N$. This is because, as the new node y should be well matched to most of the ARGs, we approximate $\mathcal{F}_i^y$, $\mathcal{F}_i^{yt}$, and $\mathcal{F}_i^{ty}$ by ignoring the possibility of matching y to *none*, so as to simplify the calculation. We substitute $\mathcal{F}_i^y$ and $\mathcal{F}_i^{yt}$ into $\mathbf{P}_y$ and $\mathbf{Q}_{yt}$ in (4), and it is easy to demonstrate that

$$\begin{aligned}
\mathbf{P}_y&=\frac{1}{2N^2}\sum_{i=1}^{N_P}w_i^P\sum_{1\leq k,l\leq N}\|\mathcal{F}_i^{x_y^k}-\mathcal{F}_i^{x_y^l}\|^2 \qquad (7)\\
\mathbf{Q}_{yt}&=\frac{\sum_{i=1}^{N_Q}w_i^Q\sum_{k,l:\delta(\hat{x}_t^k)\delta(\hat{x}_t^l)=1}\|\mathcal{F}_i^{x_y^k\hat{x}_t^k}-\mathcal{F}_i^{x_y^l\hat{x}_t^l}\|^2}{2\|V\|N\sum_j\delta(\hat{x}_t^j)}+C_t
\end{aligned}$$

where $C_t=\sum_{k:\delta(\hat{x}_t^k)=0}\frac{Q_{none}}{\|V\|N}=\frac{Q_{none}\sum_k(1-\delta(\hat{x}_t^k))}{\|V\|N}$ is a constant *w.r.t* x_y^k, given $\hat{\mathbf{X}}$. Because C_t is a constant *w.r.t* $\mathbf{X}_y$, we insert $\mathbf{P}_y$ and $\mathbf{Q}_{yt}$ into (4) and (5b), and obtain

$$\begin{aligned}
&\arg\min_{\mathbf{X}_y}E_y(\mathbf{X}^{new}|G^{new},GS)=\arg\min_{\mathbf{X}_y}\sum_{1\leq k,l\leq N}M_{kl}(x_y^k,x_y^l),\\
&\text{where, }M_{kl}(x_y^k,x_y^l)=\frac{1}{2N^2}\sum_{i=1}^{N_P}w_i^P\|\mathcal{F}_i^{x_y^k}-\mathcal{F}_i^{x_y^l}\|^2 \qquad (8)\\
&\qquad+\sum_{t\in V:\delta(\hat{x}_t^k)\delta(\hat{x}_t^l)=1}\frac{\sum_{i=1}^{N_Q}w_i^Q\|\mathcal{F}_i^{x_y^k\hat{x}_t^k}-\mathcal{F}_i^{x_y^l\hat{x}_t^l}\|^2}{2\|V\|N\sum_j\delta(\hat{x}_t^j)}
\end{aligned}$$

Thus, the problem of (5b) is transformed to a QAP, which can be directly solved using a Markov random field (MRF). As shown in Fig. 4, the ARGs are connected to each other to construct the MRF and determine y's matching assignments $\mathbf{X}_y$. In this study, we use TRW-S [14] to solve the energy minimization of the MRF. y's attributes $\mathbf{F}_y$ and $\mathbf{F}_{\{y\}\times V}$ are computed by (6). If $E_y(\mathbf{X}^{new}|G^{new},GS)\leq\tau$, we replace G by the enlarged graph template G^{new}.

5. Experiments

The proposed method is meaningful in the field of computer vision, enabling the discovery of a general category model for image matching when the target objects are randomly placed in large and cluttered scenes. In particular, our technique satisfies the condition of relatively weak local attributes for matching. We have designed two experi-

sure the algorithm's stability, as the attributes of the enumerated nodes may be greatly biased. Moreover, owing to the existence of the dummy matching choice ("none"), we cannot limit the node enumeration within any single ARG to reduce computation.

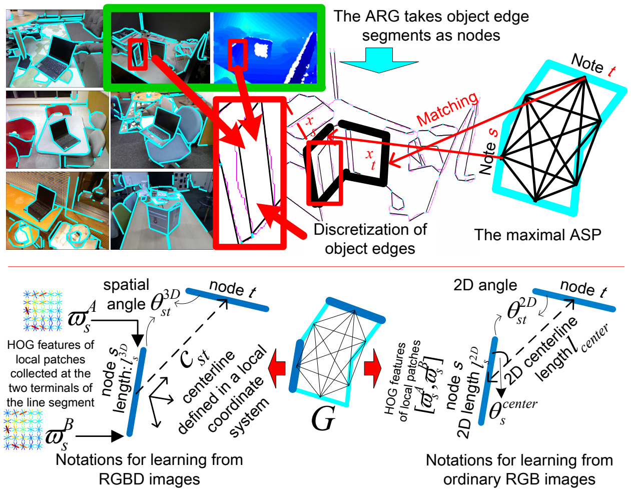
The ARG takes object edge segments as nodes
Note t
Matching
Note s
Discretization of object edges
The maximal ASP
spatial angle θ_{st}^{3D}
node t
ϖ_s^A
HOG features of local patches collected at the two terminals of the line segment
ϖ_s^B
node s length: l_s^{3D}
$\mathbf{c}_{st}$ centerline defined in a local coordinate system
G
HOG features of local patches $[\varpi_s^A, \varpi_s^B]$
node s 2D length l_s^{2D}
2D angle
θ_{st}^{2D}
2D centerline length l^{center}
θ_s^{center}
Notations for learning from RGBD images
Notations for learning from ordinary RGB images

Figure 5. Notation for the ARGs based on line segments of object edges in RGB and RGB-D images [30]. Please see [29, 30] for more details of attribute settings.

ments to validate the proposed method on the ARGs generated from RGB and RGB-D images. We compare the proposed method with unsupervised approaches to learning graph matching, although the discovery of missing nodes is orthogonal to conventional learning of attribute weights.

We use the category dataset of Kinect RGB-D images [1, 29], published as a standard RGB-D object dataset[7] for graph-matching-based model learning. This dataset have been applied with [29] and the competing method [30]. The seven largest categories—*notebook PC*, *drink box*, *basket*, *bucket*, *sprayer*, *dustpan*, and *bicycle*—in this dataset contain enough RGB-D objects, and are chosen for training. These images depict cluttered scenes containing objects with different textures and rotations, and both experiments were performed on these scenes.

5.1. Learning from RGB & RGB-D images

We apply the two types of ARGs proposed in [29, 30], each of which uses [2] to extract object edges from images and then discretizes continuous edges into line segments as the graph nodes (see Fig. 5). The two models use different attributes to represent objects in RGB and RGB-D images. Here, we briefly introduce these attributes (please refer to [29, 30] for more details of attribute settings).

Experiment 1: For ARGs generated from RGB images, one local attribute ($N_P = 1$) and three pairwise attributes ($N_Q = 3$) are designed, as illustrated in Fig. 5. We use the HOG features [8] of two local patches collected at line segment terminals of s ($[\varpi_s^A, \varpi_s^B]$) as the only local attribute. The three pairwise attributes consist of 1) the angle between each pair of lines s and t (θ_{st}^{2D}), 2) the angle between the "centerline" and each of s and t ($[\theta_s^{center}, \theta_t^{center}]$), and 3) the relative segment lengths of s and t *w.r.t* the "centerline" ($\frac{1}{l_{center}}[l_s^{2D}, l_t^{2D}]$). The attribute weights are simply set to $w_1^P = 0.2$ and $w_{j=1,2,3}^Q = 1$.

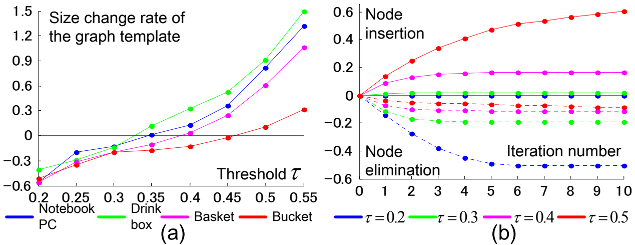

Figure 6. Size change rates of the graph template with different thresholds (τ) (a) and the rates of node insertion and elimination in different iterations (b). Vertical axes in both (a) and (b) indicate the percentage of the graph size changes, relative to the initial graph template. Solid and dotted lines in (b) indicate the effects of node insertion and elimination, respectively.

Experiment 2: For ARGs generated from RGB-D images, two local attributes ($N_P = 2$) and three pairwise attributes ($N_Q = 3$) are designed, as illustrated in Fig. 5. The local attributes describe the HOG feature (as mentioned above) and the spatial length of the line segment s (l_s^{3D}). The three pairwise attributes are the spatial angle between s and t (θ_{st}^{3D}), and the length and orientation of the relative spatial translation between s and t ($\mathbf{c}_{st}$). The setting of attribute weights for the RGB-D images is similar to that for RGB images, *i.e.* $w_1^P = 0.2$, $w_2^P = 0.1$, and $w_{j=1,2,3}^Q = 1$.

5.2. Quantitative analysis and evaluations

In the experiments, we set P_{none} and Q_{none} as 0.4 and 0.2, respectively. We use different thresholds τ for learning to extract maximal SAPs with different loose constraints. Larger threshold values τ indicate a fuzzier level of the maximal SAP, and lead to a larger graph size. Fig. 6 shows the size growth of the SAP with an increase in the threshold.

We compare our method with seven competing approaches for learning graph matching. First, we take graph matching without training, denoted by *MA*, as the baseline. *MA* uses TRW-S [14] to match the initial graph template to the target objects in images. As the benchmark method in unsupervised learning for graph matching, [17], proposed by Leordeanu *et al.*, is also used for comparison. This iteratively trains the attribute weights for matching, *i.e.* w_i^P and w_j^Q in the matching penalties $P_s(x_s)$ and $Q_{st}(x_s, x_t)$. Different from our energy minimization in (1), their graph-matching assignments are computed based on another typical form, *i.e.* compatibility maximization $\text{argmax}_{\mathbf{x}} \mathcal{C}(\mathbf{x}) = \sum_{s,t} e^{-P_s(x_s) - P_t(x_t) - Q_{st}(x_s,x_t)}$, where $P_s(\cdot)$ and $Q_{st}(\cdot,\cdot)$ are defined using absolute differences. Thus, based on [17], the two competing approaches of *LS* and *LT* are obtained by applying spectral techniques [15] and TRW-S [14], respectively, to solve the matching optimization $\text{argmax}_{\mathbf{x}} \mathcal{C}(\mathbf{x})$. Note that the original version of [17] applies a uniform initialization for w_i^P and w_j^Q, but risks biased learning (which will be discussed later). To enable a fair comparison and

[7] This is one of the largest RGB-D object datasets, and fits the requirements of learning graph matching.

Category	From RGB images							From RGB-D images						
Method	NP	DB	BA	BU	SP	DU	BI	NP	DB	BA	BU	SP	DU	BI
MA	44.30	69.93	45.44	68.71	66.88	58.43	58.94	69.00	75.25	57.97	75.33	72.65	83.96	77.78
LS	46.40	49.46	50.25	54.74	none	none	none	57.40	55.66	56.99	59.15	none	none	none
LS-O	53.62	69.52	55.21	70.74	71.31	73.24	81.44	64.63	74.18	60.75	77.99	76.48	84.53	87.61
LT	48.83	51.24	50.97	56.39	none	none	none	61.94	57.91	59.59	60.51	none	none	none
LT-O	56.57	73.01	57.35	73.66	72.84	80.15	82.60	69.04	75.09	65.37	80.44	77.31	85.47	89.33
SR	60.31	79.38	79.59	85.92	91.76	93.43	84.15	72.23	85.84	88.65	86.91	84.69	95.47	91.05
SM	72.02	85.90	72.16	83.56	79.87	83.91	71.75	89.05	85.97	75.61	84.95	90.85	95.31	94.82
Ours	**72.91**	**96.18**	**91.49**	**91.18**	**94.45**	**99.01**	**88.99**	**99.06**	**98.74**	**98.57**	**96.76**	**93.62**	**96.65**	**97.69**

Table 1. Comparison of average matching rates. *NP*, *DB*, *BA*, *BU*, *SP*, *DU*, and *BI* indicate the *notebook PC*, *drink box*, *basket*, *bucket*, *sprayer*, *dustpan*, and *bicycle* categories.

ease the bias-learning problem, *LS* and *LT* are further modified to perform with the same weight initialization as our method[8], denoted by *LS-O* and *LT-O*. The structural refinement [30] is also used for comparison, denoted by *SR*. This modifies the graph template by deleting "bad" nodes for learning. Finally, we design a competing framework that iteratively estimates model attributes, according to Definition 2(a,b) without structure modification, namely *SM*.

As in [30, 29], the evaluation is achieved via cross validation. We pick each RGB or RGB-D image in a category to start an individual learning process, thus obtaining a set of maximal SAPs. To extract each maximal SAP, the target object in the picked image is labeled as the initial graph template, and we randomly select $2/3$ and $1/3$ of the remaining images for training and testing, respectively. Note that, given the same divisions of training and testing images, *SR* modifies the graph template with the same number of nodes as our method, for a fair comparison.

We use the average matching rate (AMR) to evaluate the matching performance. This is widely used for the evaluation of learning graph matching [30, 17, 16]. The AMR is measured across all matching results produced by the extracted maximal SAP in the cross validation.

Fig. 7 illustrates the object detection performance of the maximal SAPs extracted using RGB-D images. Table 1 lists the quantitative results for comparison, where the threshold τ is set to 0.25 for the learning of all the categories in both RGB images and RGB-D images. Except *SR*, the competing methods do not have the ability to refine the topological structure of the graph template. Thus, they are sensitive to the bias in the initial graph template, including biased attributes, occluded nodes, and redundant nodes. The biased graph template may produce a biased matching, and this would, in turn, increase learning bias, thus propagating into a significant bias. In contrast, our method modifies the biased structure in early iterations to reduce the prevalence of biased matching in further iterations. Besides the elimination of "bad" parts as in *SR*, our approach also discover missing parts, thereby exhibiting better performance.

6. Discussion and conclusions

In this paper, we redefined the unsupervised learning of graph matching to model the discovery of missing parts, and thus idealize the spirit of structural learning.

The proposed method corrects errors in the topological structure of the initial graph template. As the threshold τ controls the fuzziness of the maximal SAP, it should be set up corresponding to the maximal graph (object) deformability in the ARGs. Given a suitable setting of τ, our method exhibits very good performance.

In terms of graph mining, this study can also be understood as the mining of maximal graph patterns. We proposed the SAP as the graph pattern of fuzzy ARGs, and demonstrated a plausible method of achieving the idea of mining the maximal graph pattern in the challenging graph domain of ARGs. We provided an approximate solution for maximal SAP extraction that does not require node enumeration. Another difference between conventional graph mining methods and our approach lies in the need for the graph template. This is because the matching between ARGs is formulated as a QAP, meaning that this graph matching can only be reliably achieved when an approximate area of interest for the graph pattern has been provided.

References

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

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *In* PAMI, 33(5):898–916, 2011. 6

[3] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. *In* ICCV, 2011. 3

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

[5] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. *In* ICCV, 2013. 2

[6] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. *In* CVPR, 2012. 1, 2

[7] M. Cho, Y. M. Shin, and K. M. Lee. Unsupervised detection and segmentation of identical objects. In *CVPR*, 2010. 3

[8] As a tradeoff, we apply a raw setting for the weights of just one or two attributes (see Section 5.1) to ease the bias learning problem.

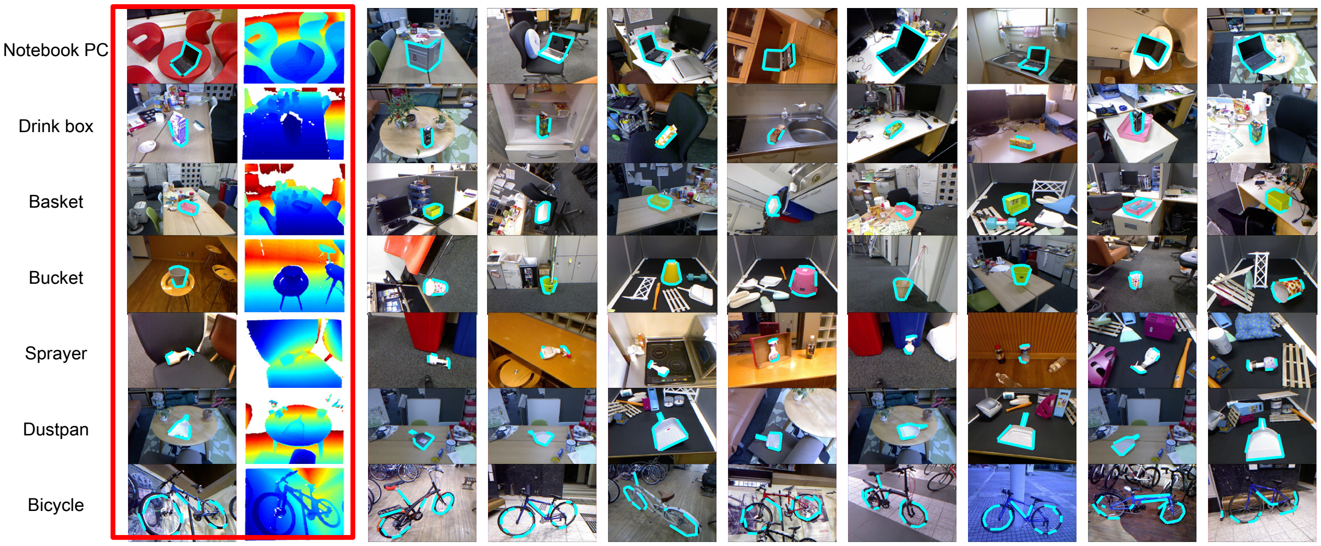

Figure 7. Object detection performance on the maximal SAPs trained from RGB-D images. The second column shows the depth images corresponding to the images in the first column.

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

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

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

[11] C. Jiang, F. Coenen, and M. Zito. A survey of frequent subgraph mining algorithms. *In* The Knowledge Engineering Review, pages 1–31, 2012. 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. 1, 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. 3

[14] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *In* IEEE PAMI, 28(10):1568–1583, 2006. 3, 5, 6

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

[16] M. Leordeanu and M. Hebert. Smoothing-based optimization. *In* CVPR, 2008. 2, 7

[17] M. Leordeanu and M. Hebert. Unsupervised learning for graph matching. *In* CVPR, 2009. 1, 2, 6, 7

[18] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: category recognition from pairwise interactions of simple features. *In* CVPR, 2007. 3

[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. 3

[21] 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. 3

[22] L. Thomas. Maximal frequent subgraph mining. *International Institute of Information Technology, Hyderabad, India*, 2010. 2

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

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

[25] J. Wang, Z. Zeng, and L. Zhou. Clan: An algorithm for mining closed cliques from large dense graph databases. *In* ACM SIGKDD, 2006. 2, 3

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

[27] 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. 3

[28] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Coherent closed quasi-clique discovery from large dense graph databases. *In* ACM SIGKDD, 2006. 2, 3

[29] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Category modeling from just a single labeling: Use depth information to guide the learning of 2d models. *In* CVPR, 2013. 6, 7

[30] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Learning graph matching for category modeling from cluttered scenes. *In* ICCV, 2013. 1, 2, 6, 7

[31] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. When 3d reconstruction meets ubiquitous rgb-d images. *In* CVPR, 2014. 2

[32] G. Zhao and J. Yuan. Mining and cropping common objects from images. *In* ACM MM, 2010. 3