Object Discovery: Soft Attributed Graph Mining

Quanshi Zhang, Xuan Song, Xiaowei Shao, Huijing Zhao, and Ryosuke Shibasaki

Abstract—We categorize this research in terms of its contribution to both graph theory and computer vision. From the theoretical perspective, this study can be considered as the first attempt to formulate the idea of mining maximal frequent subgraphs in the challenging domain of messy visual data, and as a conceptual extension to the unsupervised learning of graph matching. We define a soft attributed pattern (SAP) to represent the common subgraph pattern among a set of attributed relational graphs (ARGs), considering both their structure and attributes. Regarding the differences between ARGs with fuzzy attributes and conventional labeled graphs, we propose a new mining strategy that directly extracts the SAP with the maximal graph size without applying node enumeration. Given an initial graph template and a number of ARGs, we develop an unsupervised method to modify the graph template into the maximal-size SAP. From a practical perspective, this research develops a general platform for learning the category model (i.e., the SAP) from cluttered visual data (i.e., the ARGs) without labeling "what is where," thereby opening the possibility for a series of applications in the era of big visual data. Experiments demonstrate the superior performance of the proposed method on RGB/RGB-D images and videos.

Index Terms—Graph mining, graph matching, big visual data, attributed relational graphs, ubiquitous learning

1 Introduction

In the current era of big data, is it still necessary to label a set of object samples for each category and train category models individually? In this section, we address this question by identifying our research in terms of both graph matching and graph mining, and introduce its contributions to ubiquitous learning from big visual data.

1.1 Views of Graph Matching & Task Introduction

In the field of computer vision, attributed relational graphs (ARGs) are widely used. As shown in Fig. 1, ARGs can represent either scenes or objects, using the local and pairwise attributes to describe parts features and the spatial relationship between the parts, respectively. The goal attributed graph matching is to estimate node correspondences between two ARGs based on the similarity of local and pairwise attributes, e.g. mapping a small ARG template of an object to a large ARG of an image.

In the general case,[1] the graph matching of ARGs is typically formulated as a quadratic assignment problem (QAP), which requires global optimization.

Recently, a number of approaches for "learning graph matching" have been proposed. These train models or matching parameters, and their superior performance in terms of improving matching accuracy has been demonstrated. Indeed, the concept of learning graph matching has been extended. Generally speaking, we can categorize these approaches as supervised methods [2], [3], [4], [5], [6] and unsupervised methods [5], [7]. Unsupervised methods do not require the matching correspondences of target objects in the ARGs to be manually labeled for training, whereas supervised methods require such labeling. In this study, we focus on unsupervised approaches, which are analogous to automatic category modeling using big visual data.

In this paper, we propose a new concept of learning graph matching that focuses on the discovery of missing[2] graph parts (nodes) of the common subgraph pattern. As shown in Fig. 2, 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. In other words, this technique provides a general solution to recovering full-size graphical patterns from object fragments, as shown in Fig. 3. Obviously, this is orthogonal to conventional unsupervised approaches that learn attribute weights [5] and refine template structures [7], [8].

1.2 Views of Graph Mining & the Proposed Method

From another perspective, the proposed method mines maximal-size[3] subgraph patterns, which is one of the core branches of graph mining. Many related techniques have

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

- *Q. Zhang, X. Song, X. Shao, and R. Shibasaki are with the Center for Spatial Information Science University of Tokyo, Tokyo, Japan. E-mail: {zqs1022, songxuan, shaoxw, shiba}@csis.u-tokyo.ac.jp.*
- *H. Zhao is with the Key Laboratory of Machine Perception (MoE) Peking University, Beijing, China. E-mail: zhaohj@cis.pku.edu.cn.*

Manuscript received 1 July 2014; revised 12 June 2015; accepted 2 July 2015. Date of publication 14 July 2015; date of current version 10 Feb. 2016. Recommended for acceptance by V. Ferrari.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2456892

2. The discovery of missing parts is the key to growing the current fragmentary subgraph pattern to the maximal size. The nodes in different ARGs, which correspond to the missing parts, should have similar unary and pairwise attributes, so as to maintain the fuzziness of the subgraph pattern.

3. The word "maximal" indicates that the target subgraph pattern should be grown until the graph size of the pattern reaches a maximum.

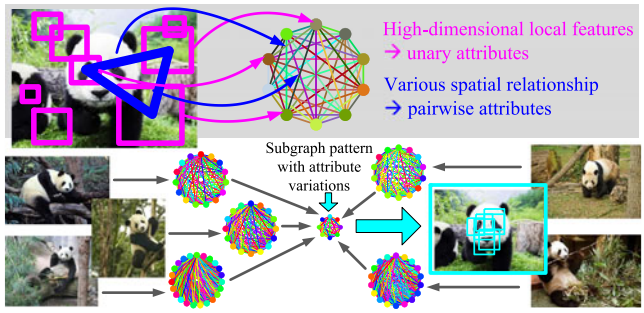


Fig. 1. If we represent an image using an ARG, the subgraph pattern (with attribute variations) corresponds to the model of the common objects.

been extensively investigated and developed, including maximal frequent subgraph (MFS) extraction and maximal clique mining.

However, the mining of maximal subgraph patterns encounters a bottleneck in the strict constraints of the target graphs. Pioneering studies have mainly considered "labeled graphs" (those that have distinct node labels or edge labels) and graphs with a list of pre-determined potential node correspondence candidates. Such graphs are usually generated from tabular data and have distinguishing structures.

By contrast, when we extend this topic to the messy visual data collected from real-world situations, both the definition of the subgraph pattern and the mining method become much more challenging. As shown in Fig. 2, people use "fuzzier[1]" ARGs with varied attributes to model scenes/objects with great intra-category variations. Without distinguishing structures or node/edge labels, conventional judgments of a graph isomorphism between the pattern and these target subgraphs can no longer be applied to such ARGs. Alternatively, we redefine the subgraph pattern as a "soft" attributed pattern (SAP), which 1) uses the graph matching technique to compute the correspondences between the

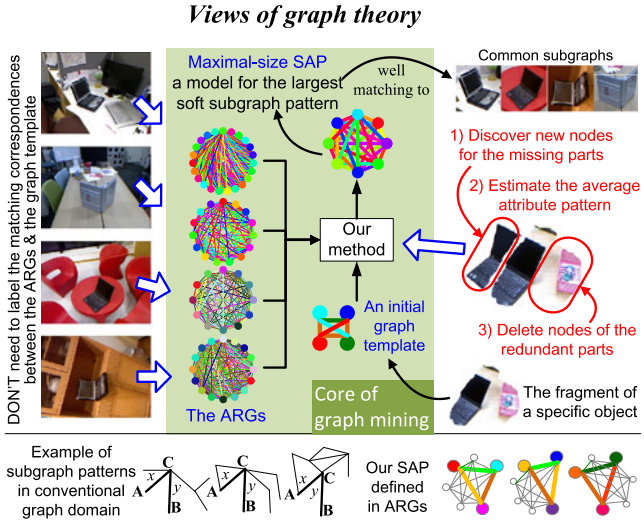


Fig. 2. Overview from the perspective of graph theory. We define and extract the soft attributed pattern (SAP) from ARGs, and maximize the size of the SAP. This study overcomes a key challenge in graph mining, as we formulate the idea of mining maximal-size common subgraphs in the challenging graph domain of ARGs. This method also extends the concept of unsupervised 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.

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

pattern and the target subgraphs embedded in the ARGs, and 2) uses a threshold to limit its attribute differences from the target subgraphs, thereby maintaining the pattern's significance. Consequently, the mining process is to extract the SAP with the maximal graph size among the ARGs.

1.2.1 Chicken-and-Egg Problem

Conventional approaches oriented to labeled graphs mainly use node enumeration (or node search) strategies to discover new nodes for the pattern. However, these approaches are all hampered[4] in the graph domain of ARGs. Therefore, in this paper, we design a new mining methodology that directly discovers new pattern nodes from ARGs without any node enumeration.

Node discovery from ARGs is a chicken-and-egg problem. The two interdependent terms are: 1) the estimation of the unary and pairwise attributes related to the new node, and 2) the determination of the new node's matching assignments to different ARGs. The conventional idea is that we should first discover the target nodes hidden in different ARGs that share similar unary and pairwise attributes. Then, we can set the new pattern node to represent this attribute pattern. However, the node correspondences between the target subgraphs in different ARGs can only be computed by graph matching between the SAP and ARGs, which requires the prior knowledge of the attributes of the new node. In fact, these two terms are reflected in the definition of the SAP (i.e., in Definitions 2a, 2b), which will be explained later.

In this paper, we demonstrate that when we use the typical squared differences to define the dissimilarity between the SAP and its target subgraphs, we can obtain an approximate closed-form expression of node attributes *w.r.t.* node matching assignments. This enables us to simultaneously determine the attributes and matches of the new pattern node.

1.3 General Platform for Model Mining from Big Visual Data

As shown in Fig. 4, we consider this study to describe a method of "model mining from big visual data," rather

4. The conventional strategy of node enumeration cannot ensure the algorithm's stability, because without distinct node labels, the enumerated nodes in each specific ARG may have heavily biased attributes. Moreover, owing to the existence of the dummy matching choice ("ϵ"), we cannot limit the node enumeration within any single ARG to reduce the computational load.

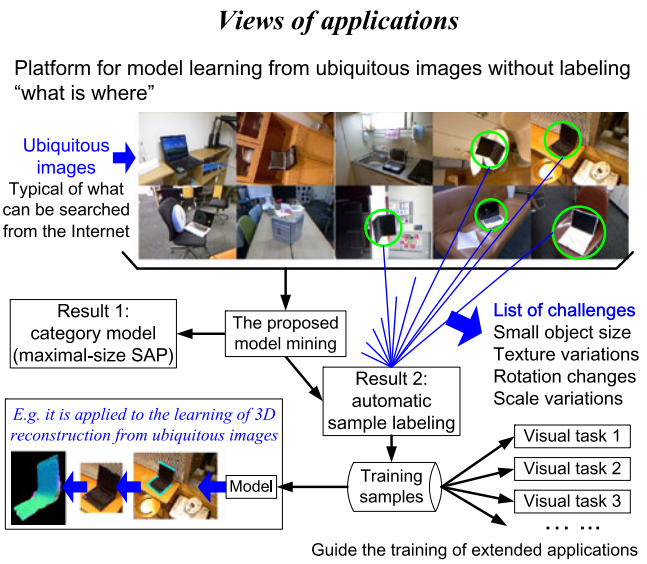


Fig. 4. Overview from the perspective of applicability. We propose a platform for model learning and sample labeling using big visual data, which has a wide range of potential applications.

than the conventional "model training." When faced with big visual data, the main bottleneck is the difficulty of model learning from ubiquitous images. If it is necessary to manually prepare a set of training images for each category, the considerable cost of human labeling may hamper the ultimate goal of building an all-embracing base to provide an object-level visual knowledge. Therefore, an efficient way of mining category models from ubiquitous images without labeling "what is where" would be valuable.

However, messy visual data collected from complex real-world situations or the internet involves almost all the typical challenges in computer vision. Target objects are usually small and randomly located in cluttered scenes, with considerable intra-category variations in texture, pose, rotation, and scale.

Therefore, we propose this graph-mining method as a general platform for learning from ubiquitous visual data. Visual data, such as RGB and RGB-D images, 3D point clouds, and video frames, can all be represented by ARGs; thus, objects in the target category within these images correspond to the common subgraphs embedded in the ARGs. The intra-category variations can be formulated as attribute variations, and the maximal-size SAP can be considered as the category model. In this case, the mining of the maximal-size SAP can be regarded as an elegant way of discovering common objects from unlabeled visual data.

In addition to mining category models, our method also simultaneously detects object parts. Such part-level object detection can be regarded as the automatic labeling of training samples in ubiquitous images, and is thereby able to guide the training of many visual tasks, such as object recognition, tracking, and segmentation. For example, based on this method, we can use part-level labeling to learn the knowledge for single-view 3D reconstruction from informally collected RGB-D images, as reported in [9].

1.4 Contributions

The contributions of this paper can be summarized as follows. First, we redefine the concept of unsupervised learning for graph matching in order to idealize the spirit of training graphical structures. To the best of our knowledge, this study is the first attempt to encode the discovery of missing parts into the learning of graph matching. Second, in terms of graph mining, this research extends the mining of maximal-size subgraph patterns to challenging visual data. We demonstrate the existence of a direct solution to graph mining that does not require computationally intensive node enumeration. Both the pattern definition and mining strategy for visual data are totally different from pioneering approaches. Third, the proposed technique can be understood as a platform for model learning from big visual data, which automatically labels common objects in ubiquitous images. This can be used to guide many extended visual tasks.

The rest of this paper is organized as follows. The following section discusses some related work. Section 3 defines the target problem of this research, and Section 4 presents the detailed algorithm. In Section 5, we describe the design and results of experiments to evaluate the algorithm. Finally, the overall study is summarized in Section 6.

A preliminary version of this paper appeared in [10].

2 Related Work

2.1 Views of Graph Matching

Given a graph template and a number of ARGs, conventional algorithms for learning graph matching [3], [4], [5], [6] mainly train the matching parameters, and Cho et al. [2] proposed to learn a model for matching. Most of these are supervised methods that require the target subgraphs in ARGs to be labeled for training. Leordeanu et al. [5] proposed the first unsupervised method that did not require such manual labeling. In [7], the structural refinement was considered as part of the unsupervised learning for graph matching. A similar idea was utilized to mine spatial patterns from ARGs [11]. Cho and Lee [8] matched two ARGs and simultaneously learned the node linkage of the two matched subgraphs, which can also be regarded as structural refinement. Essentially, structural refinement deletes "bad" nodes from the graph template, rather than recovering the prototype graphical 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.

2.2 Views of Graph Mining

In the field of graph mining (reviewed in [12]), the concept of mining maximal subgraph patterns has been realized by MFS extraction [13] and maximal clique mining [14], [15]. As shown in Fig. 2(bottom), MFS extraction [1], [13], [16] is based on graph isomorphisms, and usually requires 1) the distinguishing (topological) structure of the subgraph pattern, and 2) pre-defined distinct node/edge labels or potential inter-graph node correspondences determined by local consistency. Thus, node enumeration 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.

TABLE 1
Simultaneous Modeling of All Typical Challenges Using Attributed Graph Mining

Challenges	Modeled as
Small-size objects are unaligned in large images.	Automatic object detection in large images is modeled as a graph-matching problem.
Intra-category texture variations Intra-category rotation variations Deformable structure of objects Illumination changes Scale variations	Local features of object parts are modeled as unary attributes. Scale-independent or rotation-independent measurements of the spatial relationship between each pair of parts are modeled as pairwise attributes. These intra-category variations can be modeled as the attribute variations.

Similarly, maximal clique mining [14], [15], [17], [18] mainly extracts a dense graph clique to maintain geometric consistency during matching. Some studies [19], [20] were proposed to formulate the softness of the clique pattern. However, for the task of mining from fuzzy visual data collected from real-world situations, clique mining methods did not provide a solid way to ensure robustness to attribute variations and still required distinguishing local features to pre-determine local matching correspondence candidates among the graphs.

In contrast, fuzzily defined ARGs usually have neither distinguishing structures nor distinct node labels. In many applications, nodes are connected in a uniform style. The ARGs may even contain only pairwise attributes without local attributes. Thus, we require 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. Thus, we redefine the common subgraph pattern as the SAP based on the attributes' consistency, rather than a graph isomorphism *w.r.t.* the structure and labels.

Furthermore, if we do not strictly limit our discussions to graph theory, [21], [22] can be considered as pioneering works that discovered common structural patterns within images by estimating common graph structures. In [23], [24], [25], [26], [27], [28], common objects were extracted from images based on techniques related to maximal clique mining [14], [15]. However, these studies were mainly designed with some data-driven techniques oriented to their own applications. They thereby require target objects to have 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 has a much wider range of computer-vision applications, as shown in the four experiments.

2.3 Views of Visual Mining from Big Data

The concept of visual mining is extensive. For example, deep learning [29] has been applied to unsupervised feature learning for image recognition and produced a clear improvement in performance. However, in this research, we limit our discussion to the concept of "object-level" visual mining, i.e., mining knowledge of small objects from large and cluttered images.

Some pioneering studies, including those on object discovery [30] and co-segmentation [31], [32], [33], can be thought of as a kind of object-level visual mining. However, these have some pre-requisites for the target objects (mainly based on texture information). It is difficult for them to encode detailed structural knowledge. Thus, they are all sensitive to texture variations. Our previous work [34], [35] mined object patterns from 3D point clouds. [36], [37] extracted object structural knowledge from unlabeled images, which used relatively reliable structural information in RGB-D images to guide the model learning and applied the learned model to ordinary RGB images. In contrast, in this paper, we focus on a more general case. We define the problem of object-level visual mining and list its challenges in Table 1. Our method can be regarded as a general solution to these challenges.

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 complete graph. G contains N_P types of local attributes for each node and N_Q types of pairwise attributes for each edge. $\mathbf{F}_V = \{\mathcal{F}_i^s | s \in V, i = 1, 2, \ldots, N_P\}$ and $\mathbf{F}_{V\times V} = \{\mathcal{F}_j^{st} | s, t \in V, s \neq t, j = 1, 2, \ldots, N_Q\}$ denote the local and pairwise attribute sets, respectively. Each attribute corresponds to a feature vector.*

Actually, in this definition, the ARG is defined as a fully connected graph, but it can be extended to encode knowledge of incomplete graphs with the form $G^* = (V, E, \mathbf{F}_V, \mathbf{F}_E)$. We can transform G^* to our fully connected 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 $GS = \{G'_k | k = 1, 2, \ldots, N\}$, $G'_k = (V_k, \mathbf{F}_{V_k}, \mathbf{F}_{V_k\times V_k})$, the graph template $G = (V, \mathbf{F}_V, \mathbf{F}_{V\times V})$ represents an attribute pattern among the ARGs in GS. Note that G 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 $x_s^k \in V_k \cup \{\epsilon\}$ maps node s in G to either a node in G'_k or a dummy choice ϵ. ϵ is used when some nodes in G do not exist in G'_k. The graph matching is formulated as a typical QAP with the following energy function:

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

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 [38]. In this study, matching penalties are defined using squared differences,

$$P_s\big(x_s^k|G,G'_k\big)=\begin{cases}\sum_{i=1}^{N_P} w_i^P\|\mathcal{F}_i^s-\mathcal{F}_i^{x_s^k}\|^2, & x_s^k\in V_k\\ P_\epsilon, & x_s^k=\epsilon\end{cases}\tag{2a}$$

$$Q_{st}\big(x_s^k,x_t^k|G,G'_k\big)=\begin{cases}\frac{\sum_{j=1}^{N_Q} w_j^Q\|\mathcal{F}_j^{st}-\mathcal{F}_j^{x_s^k x_t^k}\|^2}{\|V\|-1}, & x_s^k\neq x_t^k\in V_k\\ +\infty, & x_s^k=x_t^k\in V_k\\ \frac{Q_\epsilon}{\|V\|-1}, & x_s^k \text{ or } x_t^k=\epsilon,\end{cases}\tag{2b}$$

where P_ϵ and Q_ϵ are relatively large constant penalties for matching to the dummy node ϵ in the case of occlusions. $\|\cdot\|$ 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[5] by $(\|V\|-1)$. Penalties P_ϵ and Q_ϵ and attribute weights $\{w_i^P\}$ and $\{w_j^Q\}$ can be manually set or automatically mined, which will be introduced in Section 4.3.

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

(a) $\hat{\mathbf{x}}^k=\arg\min_{\mathbf{x}^k}\mathcal{E}(\mathbf{x}^k|G,G'_k)$; *we set* $\hat{\mathbf{x}}^k=\{x_s^k|s\in V,k=1,2,\ldots,N\}$;

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

(c) $\forall s\in V, E_s(\{\hat{\mathbf{x}}^k\}|G,GS)\le\tau$;

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

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

Maximal SAP. The definition of the SAP can be visualized in Fig. 5, and we introduce the physical meaning of each item in Definition 2, as follows.

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

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.*

5. Node insertion (or delete) will increase (or decrease) the overall weights for pairwise attributes in both the graph matching (1) and the calculation of $E_s(\{\hat{\mathbf{x}}^k\}|G,GS)$, causing an unstable performance. To prevent such effects, we normalize $Q_{st}(x_s,x_t|G,G'_k)$.

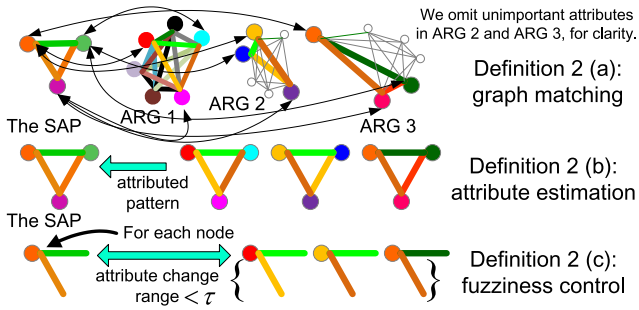


Fig. 5. Visualization of the SAP in Definition 2. Colors in ARGs denote different local and pairwise attributes. Note that in graph matching, we use pairwise attributes (edge colors), rather than simply geometric distance between nodes (although such distances can be used as one of the N_Q types of pairwise attributes).

4 Proposed Algorithm

To extract a maximal SAP, the initial graph template G is modified in the following expectation-maximization (EM) framework. In each iteration, we use the current G to estimate the matching assignments in the ARGs in GS, $\{\hat{\mathbf{x}}^k\}$, and then use $\{\hat{\mathbf{x}}^k\}$ 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).

4.1 Attribute Estimation

First, we use Definition 2a to obtain matching assignments $\{\hat{\mathbf{x}}^k\}$ for the current G. Then, given $\{\hat{\mathbf{x}}^k\}$, we can directly solve Definition 2b by estimating G's attributes as the arithmetic mean of the attributes of all corresponding subgraphs in the ARGs, which minimizes the squared Euclidean distance in $\sum_{k=1}^N\mathcal{E}(\hat{\mathbf{x}}^k|G,G'_k)$:

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

where $\delta(\cdot)$ indicates whether a node in G is matched to ϵ. If $\hat{x}_s^k=\epsilon$, $\delta(\hat{x}_s^k)$ is set to 0; otherwise 1.

4.2 Structure Modification

We grow the initial G into the maximal SAP using a greedy strategy (see Fig. 6). 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 2c. The worst node is determined as $\hat{s}=\arg\max_{s\in V}E_s(\{\hat{\mathbf{x}}^k\}|G,GS)$ in G. If $E_{\hat{s}}(\{\hat{\mathbf{x}}^k\}|\,G,GS)>\tau$, we delete $\hat{s}$ from G; otherwise, this node is retained.

Actually, the core task of structure modification is to discover new nodes for the SAP. It requires us to determine both the attributes related to the new pattern node and its matching assignments to all the ARGs, which is a chicken-and-egg problem, as analyzed in Section 1.2. Thus, we have developed an efficient solution that simultaneously determines the attributes and matching assignments of the

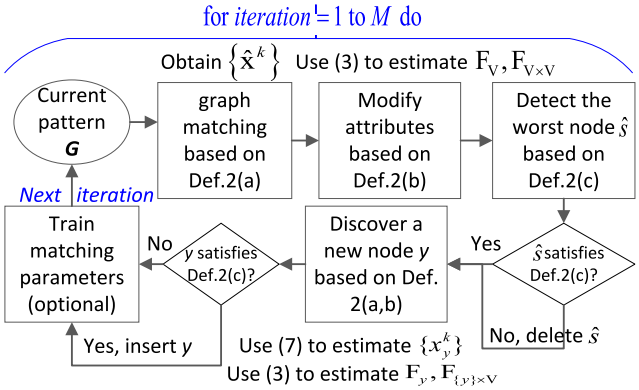


Fig. 6. Algorithm flowchart.

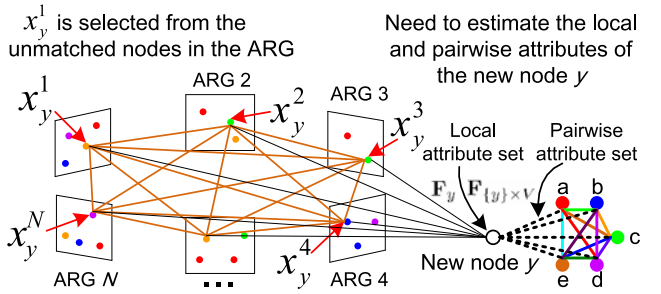


Fig. 7. Discovery of the missing node y in G. We have demonstrated a direct solution to the determination of y's matching assignments $\{x_y^k\}$ in the N ARGs that minimize $E_y(\{\mathbf{x}_{new}^k\}|G^{new}, GS)$, without requiring any prior knowledge of y's attributes. The ARGs are connected to each other to construct a Markov random field that solves this problem.

missing node. Let y be the missing node of G, and let $\mathbf{F}_y = \{\mathcal{F}_i^y | 1 \le i \le N_P\}$ and $\mathbf{F}_{\{y\}\times V} = \{\mathcal{F}_j^{yt}, \mathcal{F}_j^{ty} | t \in V, 1 \le j \le N_Q\}$ 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 $\{x_y^k | k = 1, 2, .., N\}$.

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: $V^{new} = V \cup \{y\}$, $\mathbf{F}_{V^{new}} = \mathbf{F}_V \cup \mathbf{F}_y$, $\mathbf{F}_{V^{new}\times V^{new}} = \mathbf{F}_{V\times V} \cup \mathbf{F}_{\{y\}\times V}$, $\mathbf{x}_{new}^k = \hat{\mathbf{x}}^k \cup \{x_y^k\}$. Thus, the local matching penalty of y is transformed to

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

The goal of node insertion is transformed to

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

$$\arg\min_{\{x_y^k\}} E_y(\{\mathbf{x}_{new}^k\}|G^{new}, GS). \tag{5b}$$

These two equations correspond to Definitions 2a, 2b. We should simultaneously estimate the matching assignments ($\{x_y^k\}$) and attributes ($\mathbf{F}_y, \mathbf{F}_{\{y\}\times V}$) of y that minimize the overall matching energy. Because the new node y should be well matched to most of the ARGs, we tentatively ignore the possibility of matching y to ϵ, so as to simplify the calculation (inaccuracy caused by this approximation can be corrected in further iterations):

$$\forall k = 1, 2, \ldots, N, \quad \delta(x_y^k) = 1. \tag{6}$$

Thus, as in (3), we use the arithmetic mean to minimize the squared Euclidean distance in the matching energy, as the solution to (5a). Considering $\delta(x_y^k) = 1$, attributes in $\mathbf{F}_y$ and $\mathbf{F}_{\{y\}\times V}$ can be represented as $\mathcal{F}_i^y = \sum_{k=1}^N \mathcal{F}_i^{x_y^k}/N$, $\mathcal{F}_i^{yt} = \text{mean}_{k:\delta(\hat{x}_t^k)=1} \mathcal{F}_i^{x_y^k \hat{x}_t^k}$, and $\mathcal{F}_i^{ty} = \text{mean}_{k:\delta(\hat{x}_t^k)=1} \mathcal{F}_i^{\hat{x}_t^k x_y^k}$.

We substitute $\mathcal{F}_i^y$ and $\mathcal{F}_i^{yt}$ into $\mathbf{P}_y$ and $\mathbf{Q}_{yt}$ in (4). Considering the identity $\sum_{u=1}^N \|a_u - \frac{1}{N}\sum_{v=1}^N a_v\|^2 = \frac{1}{2N}\sum_{1\le u,v\le N} \|a_u - a_v\|^2$, we demonstrate that (5b) can be transformed as

$$\begin{aligned} &\arg\min_{\{x_y^k\}} E_y(\{\mathbf{x}_{new}^k\}|G^{new}, GS) = \arg\min_{\{x_y^k\}} \{\mathbf{P}_y + \sum\nolimits_{t\in V} \mathbf{Q}_{yt}\} \\ &= \arg\min_{\{x_y^k\}} \sum\nolimits_{1\le k,l\le N} M_{kl}(x_y^k, x_y^l), \\ &\text{where } M_{kl}(x_y^k, x_y^l) = \frac{1}{2N^2}\sum_{i=1}^{N_P} w_i^P \|\mathcal{F}_i^{x_y^k} - \mathcal{F}_i^{x_y^l}\|^2 \\ &+ \sum\nolimits_{t\in V:\delta(\hat{x}_t^k)\delta(\hat{x}_t^l)=1} \frac{\sum_{i=1}^{N_Q} w_i^Q \|\mathcal{F}_i^{x_y^k \hat{x}_t^k} - \mathcal{F}_i^{x_y^l \hat{x}_t^l}\|^2}{2\|V\| N \sum_j \delta(\hat{x}_t^j)}. \end{aligned} \tag{7}$$

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. 7, the ARGs are connected to each other to construct the MRF and determine y's matching assignments $\{x_y^k\}$. In this study, we use TRW-S [38] to solve the energy minimization of the MRF. We then compute y's attributes $\mathbf{F}_y$ and $\mathbf{F}_{\{y\}\times V}$ by substituting $\{x_y^k\}$ into (3). If $E_y(\{\mathbf{x}_{new}^k\}|G^{new}, GS) \le \tau$, we replace G by the enlarged graph template G^{new}.

4.3 Matching Parameter Estimation

In addition to graph mining, we also propose a method to train matching parameters in G, including attribute weights (i.e., $\{w_i^P\}$ and $\{w_j^Q\}$) and penalties for ϵ (i.e.,P_ϵ *and* Q_ϵ). Under the maximum entropy principle, given the optimal matching assignments $\hat{\mathbf{x}}^k$ mapping G to G'_k, we can formulate the probability of the matched subgraph as

$$P(\hat{\mathbf{x}}^k|W) = \frac{1}{Z_k(W)} \exp\big[-\mathcal{E}(\hat{\mathbf{x}}^k|G, G'_k)\big], \tag{8}$$

where $W = \{w_i^P | i = 1, 2, \ldots, N_P\} \cup \{w_i^Q | j = 1, 2, \ldots, N_Q\}$, and $Z_k(W) = \sum_{\mathbf{x}^k} \exp[-\mathcal{E}(\mathbf{x}^k|G, G'_k)/\|V\|]$. To simply the calculation, we assume that good matches can only be found in the neighborhood of the optimal one $\hat{\mathbf{x}}^k$, and approximate $Z_k(W)$ as

$$Z_k(W) \approx \sum_{s\in V} \sum_{x_s^k \in V_k} \exp\big[-\mathcal{E}(\widetilde{\mathbf{x}}^k|G, G'_k)\big]\Big|_{\widetilde{x}_s^k = x_s^k; \forall t \ne s, \widetilde{x}_t^k = \hat{x}_t^k}. \tag{9}$$

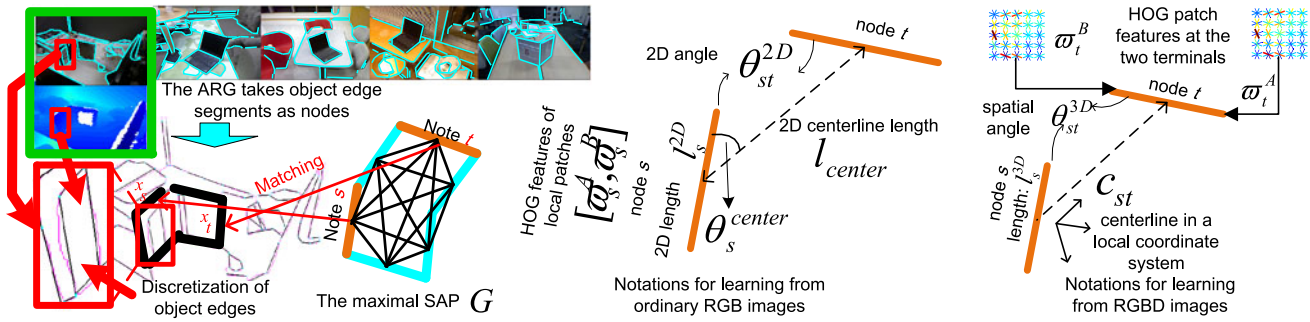


Fig. 8. Notation for the ARGs based on line segments of object edges in RGB and RGB-D images [7]. Please see [7], [36] for more details of attribute settings.

The objective of parameter estimation is to maximize the probability of all the matches to the N ARGs,

$$\arg\max_{W} P(\{\hat{\mathbf{x}}^k\}|W),\ P(\{\hat{\mathbf{x}}^k\}|W) = \prod_{k=1}^{N} P(\hat{\mathbf{x}}^k|W). \quad (10)$$

We use the steepest descent method to solve this equation, i.e., $W \leftarrow W + \eta \nabla_W \log P(\{\hat{\mathbf{x}}^k\}|W)$. When W has been trained, we normalize it as $W \leftarrow W/\|W\|_1$, which will not affect the graph matching in (1). We can further estimate P_ϵ and Q_ϵ as

$$\begin{aligned} &P_\epsilon \leftarrow \bar{P}^+ + \alpha(\bar{P}^- - \bar{P}^+),\ Q_\epsilon \leftarrow \bar{Q}^+ + \alpha(\bar{Q}^- - \bar{Q}^+);\\ &\bar{P}^+ = \text{mean}_{1\le k\le N, s\in V} P_s(\hat{x}_s^k|G, G'_k),\\ &\bar{Q}^+ = \text{mean}_{1\le k\le N, s\in V} \sum_{t\in V, t\neq s} Q_{st}(\hat{x}_s^k, \hat{x}_t^k|G, G'_k)\\ &\bar{P}^- = \text{mean}_{1\le k\le N, s\in V, x_s^k\in V_k} P_s(x_s^k|G, G'_k),\\ &\bar{Q}^- = \text{mean}_{1\le k\le N, s\in V, x_s^k\in V_k} \sum_{t\in V, t\neq s} Q_{st}(x_s^k, \hat{x}_t^k|G, G'_k), \end{aligned} \quad (11)$$

where $\bar{P}^+$, $\bar{Q}^+$ denote the average unary and pairwise matching penalties in the optimal matches, respectively, while $\bar{P}^-$, $\bar{Q}^-$ correspond to the penalties for other matching choices; $\alpha > 0$ (we set $\alpha = 0.5$).

5 Experiments

The proposed method is applicable 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 use different experiments to evaluate the proposed method in different visual tasks.

In Experiments 1 and 2, we test our method for mining category models (i.e., maximal-size SAPs) from cluttered RGB-D and RGB images, respectively. The category model is constructed using object edge segments as graph nodes, and thus performs well in detecting objects with clear edges. Then, in Experiment 3, we extend the model-mining task to more general images, i.e., those collected from the internet by search engines. Therefore, we propose another ARG model that takes scale-invariant feature transform (SIFT) feature points as graph nodes to describe objects in general images. Finally, in Experiment 4, we further extend the application to the learning of deformable object models from videos.

We compare the proposed method with unsupervised approaches that learn graph matching, although the discovery of missing nodes in the proposed method is orthogonal to the conventional learning of attribute weights.

5.1 Experiment 1: Mining Edge-Based Models from Cluttered RGB-D Images

5.1.1 Dataset

We use the category dataset of Kinect RGB-D images [36], [37], [39], published as a standard RGB-D object dataset[6] for graph-matching-based model learning. This dataset was applied in [36] and the competing method [7]. The seven largest categories—*notebook PC*, *drink box*, *basket*, *bucket*, *sprayer*, *dustpan*, and *bicycle*—in this dataset contain a sufficient number of RGB-D objects, and are chosen for training. These images depict cluttered scenes containing objects with different textures and rotations.

5.1.2 ARG-Based Category Models

As illustrated in Fig. 8, in our previous studies [7], [36], [37], we designed ARGs to represent objects in RGB-D images. The category model is mined as the SAP among these ARGs. The ARGs are designed as follows. We first use an edge extraction method [40] to extract object edges from images, and then discretize continuous edges into line segments as the graph nodes. The technical details of edge segmentation were introduced in [36]. We connect each pair of graph nodes to construct a complete graph. Two local features ($N_P = 2$) and three pairwise attributes ($N_Q = 3$) are designed to describe local features and the spatial relationship between local parts in images.

The first unary attribute is the histogram of oriented gradients (HoG) feature [41] of two local patches collected at the line segment terminals of node s, denoted by $\mathcal{F}_1^s = [\varpi_s^A, \varpi_s^B]$[7]. The HoG features [41] are extracted using

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

7. Actually, this attribute is affected by the order of the two terminals, but there is no good way to pre-define this order without considering the terminal order of other nodes. Therefore, we slightly modify the distance measurement of $\mathcal{F}_1^s$ in Equation (2) from $\|\mathcal{F}_1^s - \mathcal{F}_1^{\hat{x}_s^k}\|^2$ to $\min\{\|\mathcal{F}_1^s - [\varpi_{x_s^k}^A, \varpi_{x_s^k}^B]\|^2, \|\mathcal{F}_1^s - [\varpi_{x_s^k}^B, \varpi_{x_s^k}^A]\|^2\}$. The final terminal order of each node $\hat{x}_s^k$ in ARG G'_k is determined as that which best matches node s in G.

5×5 cells, each of which covers half of its neighboring cells. We use four orientation bins (from 0 to 180 degree) to compute the gradient histogram in each cell. Because the patch is locally collected without significant illumination changes, we normalize all of the cells within a single block. The second unary attribute is given by $\mathcal{F}_2^s = \log l_s^{3D}$, where l_s^{3D} is the spatial length of the line segment of node s. The first of the three pairwise attributes, $\mathcal{F}_1^{st} = \theta_{st}^{3D}$, denotes the spatial angle between the line segments of nodes s and t in the 3D space. For each edge (s,t), we define the *centerline* as the line connecting the centers of the line segments of s and t. We 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 $\mathbf{c}_{st}$. Based on this, the second and third pairwise attributes, i.e., $\mathcal{F}_2^{st} = \|\mathbf{c}_{st}\|$ and $\mathcal{F}_3^{st} = \mathbf{c}_{st}/\|\mathbf{c}_{st}\|$, represent the length and local orientation of the centerline, respectively. We set the attribute weights as $w_1^P = 0.2$, $w_2^P = 0.1$, $w_{j=1,2,3}^Q = 1$, and set P_ϵ and Q_ϵ as 0.4 and 0.2, respectively.

5.1.3 Technical Details

We set the maximum iteration number as $M = 10$. We apply different thresholds τ to the mining processes in order to mine the category model (i.e., the maximal-size SAP) with different loose constraints. Larger values of τ indicate a fuzzier level of the maximal-size SAP, and lead to larger SAPs.

For each setting of τ, we perform a series of cross validations, as in [7], [36]. We pick each of the RGB-D images from a category to start an individual mining process, thus obtaining a set of maximal-size SAPs. To extract each maximal-size SAP, the target object in the selected image is labeled as the initial graph template. We then randomly select $2/3$ and $1/3$ of the remaining images for training and testing, respectively. We design different evaluation metrics, which will be introduced in Section 5.5.2. Based on these metrics, the proposed method is evaluated using the average performance among all the mined SAPs.

5.2 Experiment 2: Mining Edge-Based Models from Cluttered RGB Images

This experiment is similar to Experiment 1. We use the same dataset of Kinect RGB-D images, but consider only the RGB channels in this experiment. We design a new type of ARGs for object representation in the RGB images. We test the model-mining performance under different values of τ. The performance is evaluated via cross validation as in Experiment 1.

5.2.1 ARG-Based Category Models

The model uses edge segments as graph nodes, and we use one unary attribute ($n^U = 1$) and three pairwise attributes ($n^P = 3$) for the model. The notation is illustrated in Fig. 8. The only unary attribute is the HoG feature collected at the terminals of line segments, as for the first unary attribute for RGB-D images. The first of the three pairwise attributes between nodes s and t is the angle between their line segments, denoted by $\mathcal{F}_1^{st} = \theta_{st}^{2D}$. The second pairwise attribute describes the angles between the centerline and the node line segments, denoted by $\mathcal{F}_2^{st} = [\theta_s^{center}, \theta_t^{center}]$, where θ_s^{center} is the angle between the line segment of s and the centerline. The third pairwise attribute represents relative segment lengths, and is denoted by $\mathcal{F}_3^{st} = \frac{1}{l_{center}}[l_s^{2D}, l_t^{2D}]$, where l_s^{2D} and l_{center} are the lengths of the line segment of s and the centerline, respectively. The attribute weights are simply set to $w_1^P = 0.2$ and $w_{j=1,2,3}^Q = 1$. P_ϵ and Q_ϵ are set to 0.4 and 0.2, respectively, as in the model for RGB-D images. These settings are uniformly used for the model mining of all categories.

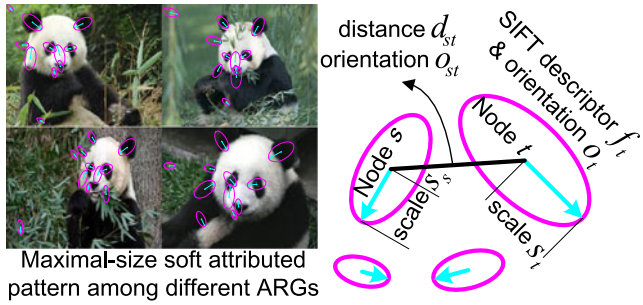


Fig. 9. Notation for the ARGs that take the SIFT feature points as graph nodes.

5.3 Experiment 3: Mining General Models from Web Images

5.3.1 Web Images

In this experiment, we focus on a more common case of ubiquitous images, i.e., those collected from the internet by search engines. We use 10 keywords—"bag," "boot," "camera," "coca cola," "glasses," "hamster," "iphone," "panda," "sailboat," "spider"—to collect images for 10 categories. Each category contains 200 images.

5.3.2 ARG-Based Model for General Images

Generally speaking, the images collected from the internet are fuzzier than those in our Kinect RGB-D images dataset [36], [39]. Therefore, we design a new type of ARGs for image representation that take SIFT points, rather than edge segments, as graph nodes. These SIFT-based ARGs are oriented to more general images, especially those that do not contain clear edges. The SIFT points in the images are detected using different scales. We connect each pair of points in an image to construct an ARG. Two local features ($N_P = 2$) and five pairwise attributes ($N_Q = 5$) are designed to describe local features and the spatial relationship between local parts in the images. The notation is illustrated in Fig. 9. The two unary attributes are the 128-dimensional descriptor and the orientation of the SIFT feature, denoted by $\mathcal{F}_1^s = f_s$ and $\mathcal{F}_2^s = o_s$. The first of the three pairwise attributes, $\mathcal{F}_1^{st} = \theta_{st}$, denotes the spatial angle between the SIFT orientation of nodes s and t. For each edge (s,t), we use d_{st} and o_{st} to represent its length and orientation. We directly set the second pairwise attribute as $\mathcal{F}_2^{st} = o_{st}$. The third pairwise attribute, $\mathcal{F}_3^{st} = [angle(o_{st}, o_s), angle(o_{st}, o_t)]^T$, is defined as the angle between edge (s,t) and each SIFT orientation of nodes s and t. Let s_s and s_t denote the SIFT scales of nodes s and t. We set the fourth and fifth attributes as

$\mathcal{F}_4^{st} = \log(s_s/s_t)$ and $\mathcal{F}_5^{st} = [\log(s_s/d_{st}), \log(s_t/d_{st})]^T$, respectively. The attribute weights are set to $w_1^P = 3$, $w_2^P = 1$, $w_{j=1,2,4}^Q = 0.5$, and $w_{j=3,5}^Q = 1$. P_ϵ and Q_ϵ are set to 3 and 5, respectively. These settings are uniformly used for the model mining of all categories.

5.3.3 Technical Details

We set the maximum iteration number to $M = 20$, and test the mining performance with different threshold values of τ. Furthermore, considering the fuzzy cases of web images, we cannot ensure that each image contains an object in the target category. Therefore, not all the images can be used in each model mining iteration, and we set two criteria to control the image quality. The first criterion is that, when we match the model to the image, the ratio of model nodes matched to ϵ should be greater than 40 percent. The second criterion states that, in each iteration, no more than 20 web images should be used for model mining. Because we cannot ensure that all web images contain the target objects, we simply select the 20 top-ranked images from those collected, i.e., the 20 images whose ARGs can match the current model (graph template) with the lowest energy. All of these settings are uniformly applied to model mining for all 10 categories.

5.4 Experiment 4: Mining Deformable Object Models from Videos

We collect three video sequences (containing a cheetah, swimming girls, and a frog) from the internet, and use our method to mine models for deformable objects from these videos. We consider each video frame as an ARG, and the deformable model as the mSAP among the ARGs. For each video, we only label three nodes to construct an initial graph template. The design of the ARGs is the same as in Experiment 3. Matching parameters are initialized as $w_{i=1,2}^P = w_{j=1,2,3}^Q = 1$ and $P_\epsilon = Q_\epsilon = +\infty$. We set $\tau = 1.5$ and apply the graph mining procedure for $M = 20$ iterations. In each iteration, we simultaneously mine the pattern and train matching parameters.

5.5 Quantitative Analysis and Evaluations

5.5.1 Competing Methods

We compare the proposed method with nine approaches from the fields of graph matching and graph mining, although as mentioned previously, our graph-mining method is orthogonal to the learning concepts in the competing methods. To enable a fair comparison, each method considers the same scenario of "learning a graphical model (or target pattern) from a set of ARGs with a single labeled graph template." All the competing methods use the same initial graph templates and the same training and testing ARGs to start each learning process in a cross validation.

First, we focus on the approaches of "graph (or image) matching" and "unsupervised learning for graph (or image) matching." We take graph/image matching methods without training as the baseline. Actually, there are two typical paradigms for graph (image) matching, i.e., the minimization of matching energy and the maximization of matching compatibility. Matching compatibility is usually formulated as $\arg\max_{\mathbf{x}} \mathcal{C}(\mathbf{x}) = \sum_{s,t} e^{-P_s(x_s) - P_t(x_t) - Q_{st}(x_s, x_t)}$, where $P_s(\cdot)$ and $Q_{st}(\cdot,\cdot)$ are defined using absolute differences. Thus, we design three competing methods, i.e., *MA*, *MS*, and *MT*, which represent the two image matching paradigms. *MA* uses TRW-S [38] to minimize the matching energy in (1) for image matching, while *MS* and *MT* maximize the overall matching compatibility. The compatibility of matching objects was proposed by [42], and *MS* and *MT* use spectral techniques [42] and TRW-S [38], respectively, to solve the matching optimization $\arg\max_{\mathbf{x}} \mathcal{C}(\mathbf{x})$.

Then, let us consider the unsupervised approaches for learning graph matching. They mainly train matching parameters to improve the model's matching accuracy, do not require matching assignments to be labeled, and do not involve the learning of graphical structures. As the benchmark unsupervised method for learning graph matching, we use the method proposed by Leordeanu and Hebert [5]. 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 $Q_{st}(x_s, x_t)$. Thus, based on [5], the two competing approaches of *LS* and *LT* are obtained by applying [42] and [38] for matching optimization, respectively. Note that the original version of [5] applies a uniform initialization for w_i^P and w_j^Q, but can suffer from biased learning (which we discuss later). To enable a fair comparison and ease the problem of bias, *LS* and *LT* are further modified to have the same weight initialization as our method[8], denoted by *LS-O* and *LT-O*. Finally, we use the part of our method that iteratively estimates model attributes, according to Definitions 2a, 2b, as another baseline, denoted as *SM*.

Finally, we compare our method with graph-mining approaches. To enable a fair comparison, the competing methods must be oriented to the graph domain of ARGs, but should not employ certain techniques, such as using the similarities between unary attributes to pre-determine a set of matching assignment candidates. Thus, we compare our approach to structural refinement [7], denoted by *SR*. This method is on the boundary between graph mining and learning graph matching, as it only trains matching parameters and attributes, and deletes "bad" nodes to refine the model's structure, rather than mining new nodes from ARGs. From this perspective, the mining of a maximal-size SAP proposed in this study is more close to the spirit of graph mining. Note that our method can mine iSAPs with different graph sizes by setting different threshold values τ. Therefore, given an iSAP that is mined with a certain value of τ, to enable a fair comparison, *SR* must modify the initial graph template to a model containing the same number of nodes as this iSAP.

5.5.2 Evaluation Metrics & Results

Fig. 11 illustrates the object detection performance of the models mined in Experiments 1, 3, and 4.

Pattern growth: Generally speaking, we can use the change in graph size to illustrate the performance of our

8. As a tradeoff, we apply a raw setting for the weights of just one or two attributes to ease the bias learning problem.

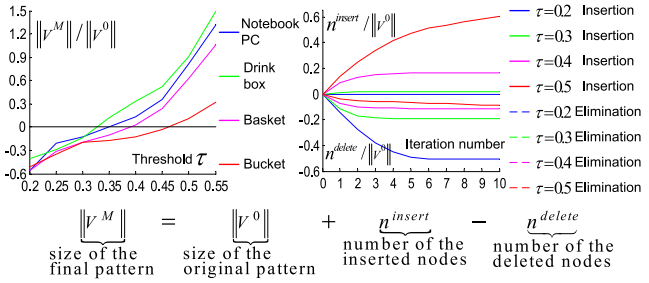


Fig. 10. Rate of change in size of the SAPs mined with different thresholds (τ) (left) and the rate of node insertion (solid curves) and elimination (dotted curves) in different iterations (right).

method for different values of τ. Fig. 10 shows the growth of the SAP with an increase in the threshold τ in Experiment 1.

Average matching rate: We use the average matching rate (AMR) to evaluate the matching performance. AMR is widely used in the evaluation of learning graph matching [4], [5], [7]. The AMR is measured across all matching results produced by the extracted maximal SAP in the cross validation. Table 2 lists the quantitative results for comparison, where the threshold τ is set to 0.25 for the learning of all categories for both the RGB and RGB-D images. With the exception of *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, in turn, increases the 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 discovers missing parts, thereby exhibiting better performance.

Selection rate & error rate: Our method mines the pattern of common subgraphs in ARGs as the category model. Obviously, as shown in Fig. 11, the subgraph pattern cannot cover all the feature points extracted from the objects.

Some feature points on the target object represent object-specific textures, rather than the common category pattern. We call these redundant feature points. Our method automatically identifies redundant feature points, and only selects points from common object parts to form the category model during the graph-mining process. However, there is no convincing way to label the ground truth of whether a feature point is redundant. Therefore, the goal of category modeling from big data is to mine a maximal number of feature points from target objects and a minimal number of feature points from the background.

Consequently, in this paper, we propose the selection rate and error rate of the mined model as metrics for evaluating the relative pattern size of the model and the errors in model mining, respectively. When we match a model to an ARG (i.e., an image), the selection rate of the model is defined as the proportion of feature points on the target object that are selected to construct the model, i.e., $\|V^M \bigcap V^O\|/\|V^O\|$, where V^O and V^M denote the subset of nodes on the target object in the ARG and the subset of nodes that are matched by the model, respectively. The error rate is defined as the proportion of nodes matched by the model that are located on the background, i.e., $\|V^M \setminus V^O\|/\|V^M\|$.

Fig. 11. Object detection performance on the maximal SAPs trained from RGB-D images in Experiment 1 (upper left), web images in Experiment 3 (right; τ is set to 2.5), and videos in Experiment 4 (lower left; τ is set to 3.0). Results for the "iphone" and "coca cola" categories are not shown due to the copyright problem.

TABLE 2
Comparison of Average Matching Rates

Category	From RGB images							From RGB-D images						
Method	NP	DB	BA	BU	SP	DU	BI	NP	DB	BA	BU	SP	DU	BI
MA [38]	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
MS [42]	55.11	40.31	45.56	56.39	41.02	50.70	75.55	63.17	44.98	52.46	67.98	69.04	49.74	90.63
MT [38]	56.05	56.15	55.28	58.02	none	none	none	62.02	59.11	60.88	61.13	none	none	none
LS [5], [42]	46.40	49.46	50.25	54.74	none	none	none	57.40	55.66	56.99	59.15	none	none	none
LS-O [5], [42]	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 [5], [38]	48.83	51.24	50.97	56.39	none	none	none	61.94	57.91	59.59	60.51	none	none	none
LT-O [5], [38]	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 [7]	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**

NP, DB, BA, BU, SP, DU, and BI indicate the *notebook PC, drink box, basket, bucket, sprayer, dustpan,* and *bicycle categories.*

In Experiment 3, we set different values of τ to mine category models with different graph sizes. Fig. 12 illustrates the change in selection rate and error rate for different values of τ. As mentioned in Section 5.3.3, we cannot ensure that all web images contain objects with the model's pattern.[9] We simply select the 20, 40, 60, and 80 top-ranked images to compute the average selection rate and error rate of a model. The histograms in Fig. 12 show the average performance for all the mined models in the 10 categories.

Considering the existence of redundant feature points, the average selection rate rarely reaches 100 percent. Therefore, the objective is to mine a category model with a certain average selection rate and a low average error rate. In Fig. 12, a larger value of τ corresponds to a higher selection rate. However, the relationship between the value of τ and the error rate is not significant, although in general, larger values of τ lead to larger error rates. In addition, setting a large number of web images (e.g. the 80 top-ranked images) for evaluation usually leads to a high error rate. When the number of images is large, we cannot ensure their quality during the testing stage.

Training of matching parameters: We further combine the training of matching parameters into the iterative flowchart of graph mining, and apply this technique to the categories of the notebook PC (in Experiment 1), panda (in Experiment 3), cheetah, swimming girls, and frog (in Experiment 4). The matching parameters are incrementally modified at the end of each iteration. Fig. 13 visualizes the attribute weights that are estimated for the static object in Experiment 3 and the deformable animal in Experiment 4. For each pair of nodes (s,t), the unary SIFT features ($\mathcal{F}_1^s$ and $\mathcal{F}_1^t$) and the third pairwise feature ($\mathcal{F}_3^{st}$ that measures the angle between the centerline and each of the nodes s and t) have more impacts on graph matching. Compared to the SAP of a dynamic frog, the SAP of the static panda head has higher weights for deformation-sensitive attributes, i.e., $\mathcal{F}_2^s$ and $\mathcal{F}_{j=1,2,3,5}^{st}$.

Then, we focus on the quantitative evaluation of the parameter training module. When we match a model (pattern) to a set of positive ARGs (the ARGs containing the target pattern) and a set of negative ARGs (those representing the background), the ratio of the average energy of the positive matches to that of the negative matches can be regarded as a metric to evaluate the distinguishing capacity of the model. This metric is just like the "eigengap" for evaluating the spectral graph matching in [5]. In Experiments 1 and 4, we produce different sets of models with different sizes by applying different values of τ via a series of cross validations. Thus, Figs. 14a and 14b shows how the average energy ratio changes with the average model size among different sets of models. The method of *Ours+UniformWeight* uniformly sets the attribute weights to 1, while the *Ours+LearnWeight* method automatically learns the attribute weights. Both *Ours +UniformWeight* and *Ours+LearnWeight* train P_ϵ and Q_ϵ to enable a fair comparison. *Ours+LearnWeight* exhibits superior performance to the other competing methods. Figs. 14c and 14d illustrates the mining performance using different values of τ. Considering that the mining process may drift if the pattern is modified to contain too few or too many nodes, the comparison of the models with the same sizes in Figs. 14a and 14b is more reasonable than that in Figs. 14c and 14d.

Computation time: Fig. 15 shows the average time required to mine each category model in Experiment 3. The algorithm is implemented in Matlab, and we compute the time cost using eight hyper threads on an Intel Xeon CPU X5560 @2.80 GHz. Large values of τ usually lead to large graph sizes in the mined models, and require additional computation time. Then, we analyze the computational cost of different competing methods. Actually, the main computational cost of these methods is in the energy minimization of the QAPs during the matching/mining processes. Let $\mathcal{V}^0$

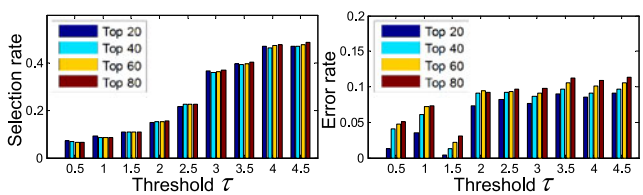


Fig. 12. Selection rate and error rate. We selected the top 20, 40, 60, and 80 object samples for each model to calculate the selection rate and error rate.

9. This is similar to the idea that people use different deformable part models to represent different object sub-patterns (*e.g.* shape patterns from different viewpoints) within a whole category.

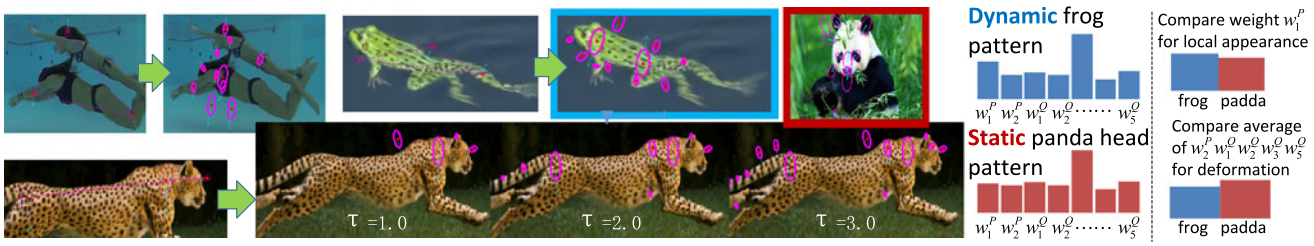


Fig. 13. Initial graph templates and final models that are mined from videos. We also compare the weights for the static pattern of the panda head mined from web images and the dynamic frog pattern mined from videos.

denote the node set of the initial graph template and V_k denote the node set of the kth ARG G'_k. First, *MA*, *MS*, and *MT* directly match the graph template to ARGs without learning. Thus, their computation cost for learning is zero. Second, the *LS*,*LS-O*,*LT*,*LT-O*, and *SM* methods modify the graph template in M iterations, but do not change the graph size of the model. In each iteration, they match the graph template to all the ARGs. The matches to G'_k can be computed as a QAP that assigns each of the $\|\mathcal{V}^0\|$ fully connected nodes in the template to one of the $\|V_k\|$ labels[10] as its matching assignment. We denote this computational cost as $c(\|\mathcal{V}^0\|, \|V_k\|)$. Note that there are various graph matching optimization techniques that can solve this QAP, and each of them has a different accuracy and $c(\|\mathcal{V}^0\|, \|V_k\|)$ (please see [43] for a comparison between them). Therefore, their computation cost for model learning can be summarized as $M\sum_{k=1}^{N} c(\|\mathcal{V}^0\|, \|V_k\|)$. Third, for the proposed method, let $\mathcal{V}^1, \mathcal{V}^2, \ldots, \mathcal{V}^M$ denote the node sets of the model after $1, 2, \ldots, M$ iterations. Thus, the computational cost of graph mining is $\sum_{m=0}^{M-1}\sum_{k=1}^{N} c(\|\mathcal{V}^m\|, \|V_k\|)$. Moreover, in each iteration, we propose a candidate for the new node, which is a QAP that assigns each of the N fully connected ARGs to one of $[\max_k\|V_k\| - \|\mathcal{V}^m\|]$ labels with computational cost $c(N, \max_k\|V_k\| - \|\mathcal{V}^m\|)$. Therefore, the overall computation is $\sum_{m=0}^{M-1}[c(N, \max_k\|V_k\| - \|\mathcal{V}^m\|) + \sum_{k=1}^{N} c(\|\mathcal{V}^m\|, \|V_k\|)]$. Finally, considering that *SR* cannot add new nodes to the model, we assume that the size of the model is $\min\{\|\mathcal{V}^1\|, \mathcal{V}^0\}, \ldots, \min\{\|\mathcal{V}^M\|, \mathcal{V}^0\}$ after $1, \ldots, M$ iterations. Hence, its time cost is $\sum_{m=0}^{M-1}\sum_{k=1}^{N} c(\min\{\|\mathcal{V}^m\|, \|\mathcal{V}^0\|\}, \|V_k\|)$. In summary, image matching methods, i.e., *MA*, *MS*, and *MT*, spend no computation on model learning. If we ignore the computation of node discovery, the computational cost of the proposed method is comparable to that of the other competing methods. In particular, if we delete more redundant nodes and add fewer new nodes during the mining process (by setting a small value of τ), the graph matching cost would approximately be that of *SR* and be less than those of *LS*,*LS-O*,*LT*,*LT-O* and *SM*.

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 corresponding to the maximal graph (object) deformability in the ARGs. In real applications, we can apply the following method to determine the value of τ. We can first extract a number of SAPs with different values of τ, and then select the SAP with the minimum ratio of matching energies between positive and negative ARGs (see Figs. 14c and 14d).

The maximum iteration number M can be empirically set to a large number that is far greater than the operation times of node insertion and node elimination. Given τ, the target model can finally converge to a pattern with a certain size and attributes. The value of M has little effect on the mining performance, as long as M is sufficiently large. Instead of directly setting M, we can use a stopping criteria for the mining procedure, i.e., the pattern size does changes in the latest three iterations, which would yield the same performance.

In terms of graph mining, this study can also be understood as the mining of maximal-size subgraph patterns. We proposed the SAP as the subgraph pattern of fuzzy ARGs, and demonstrated a plausible method of mining the maximal-size subgraph pattern in the challenging 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 a graph template. This is because the matching between ARGs is

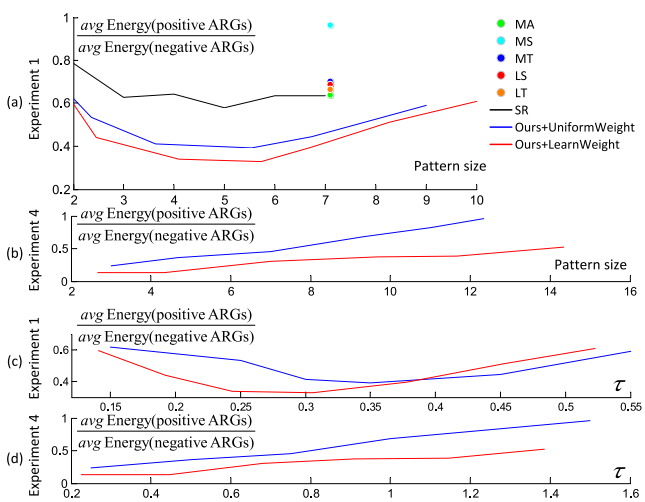


Fig. 14. Ratio of the energies of positive matches to those of negative matches. (a,b) We apply different parameters that control the final pattern size to the competing methods to enable a fair comparison. (c,d) The ratio changes along with τ.

10. We ignore matching choices of ϵ.

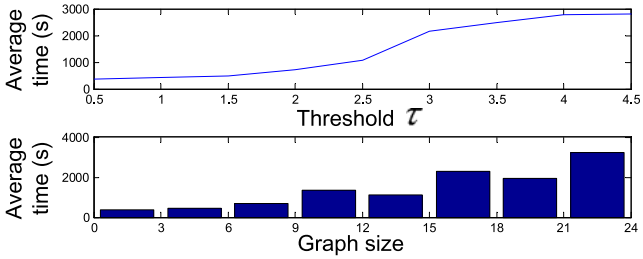


Fig. 15. Average graph-mining time.

formulated as a QAP, meaning that this graph matching can only be reliably achieved when an approximate area of interest has been provided for the subgraph pattern.

REFERENCES

[1] H. Jiang and C.-W. Ngo, "Image mining using inexact maximal common subgraph of multiple args," in *Proc. 9th Int. Conf. Vis. Inf. Syst.*, 2003, pp. 446–449.

[2] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Proc. IEEE Int. Conf. Comput. Vis.* 2013, pp. 25–32.

[3] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.

[4] M. Leordeanu and M. Hebert, "Smoothing-based optimization," in *Proc. Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[5] M. Leordeanu and M. Hebert, "Unsupervised learning for graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1–8.

[6] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. 10th Eur. Conf. Comput. Vision: Part II.*, 2008, pp. 596–609.

[7] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Learning graph matching for category modeling from cluttered scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1329–1336.

[8] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 398–405.

[9] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "When 3d reconstruction meets ubiquitous rgb-d images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 700–707.

[10] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Attributed graph mining and matching: An attempt to define and extract soft attributed patterns," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 23–28.

[11] P. Hong and T. Huang, "Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs," *Discrete Appl. Math.*, vol. 139, pp. 113–135, 2004.

[12] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowl. Eng. Rev.*, vol. 28, no. 1, pp. 1–31, 2012.

[13] L. Thomas, S. Valluri, and K. Karlapalem, "Margin: Maximal frequent subgraph mining," in *Proc. 6th IEEE Int. Conf. Data Mining*, vol. 4, no. 3, pp. 1097–1101, 2010.

[14] J. Wang, Z. Zeng, and L. Zhou, "Clan: An algorithm for mining closed cliques from large dense graph databases," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, pp. 73–82.

[15] Z. Zeng, J. Wang, L. Zhou, and G. Karypis, "Coherent closed quasi-clique discovery from large dense graph databases," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 797–802.

[16] J. Huan, W. Wang, J. Prins, and J. Yang, "Spin: Mining maximal frequent subgraphs from graph databases," in *Proc. 10th ACM Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 581–586.

[17] H. Xie, K. Gao, Y. Zhang, J. Li, and H. Ren, "Common visual pattern discovery via graph matching," in *Proc. 19th ACM Int. Conf. Multimedia*, 2012, pp. 1385–1388.

[18] H. Liu and S. Yan, "Common visual pattern discovery via spatially coherent correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1609–1616.

[19] N. Quadrianto, C. Chen, and C. H. Lampert, "The most persistent soft-clique in a set of sampled graphs," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 583–590.

[20] M. Brunato, H. H. Hoos, and R. Battiti, "On effectively finding maximal quasi-cliques in graphs," in *Proc. Learn. Intell. Optim. Conf.*, 2008, vol. 5313, pp. 41–55.

[21] M. Leordeanu, M. Hebert, and R. Sukthankar, "Beyond local appearance: Category recognition from pairwise interactions of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[22] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 778–785.

[23] G. Kim, C. Faloutsos, and M. Hebert, "Unsupervised modeling of object categories using link analysis techniques," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[24] H.-K. Tan and C.-W. Ngo, "Localized matching using earth movers distance towards discovery of common patterns from small image samples," *Image Vis. Comput.*, vol. 27, pp. 1470–1483, 2009.

[25] J. Yuan, G. Zhao, Y. Fu, Z. Li, A. Katsaggelos, and Y. Wu, "Discovering thematic objects in image collections and videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 2207–2219, Apr. 2012.

[26] G. Zhao and J. Yuan, "Mining and cropping common objects from images," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 975–978.

[27] M. Cho, Y. M. Shin, and K. M. Lee, "Unsupervised detection and segmentation of identical objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1617–1624.

[28] D. Parikh, C. Zitnick, and T. Chen, "Unsupervised learning of hierarchical spatial structures in images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2743–2750.

[29] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 81–88.

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

[31] L. Mukherjee, V. Singh, J. Xu, and M. Collins, "Analyzing the subspace structure of related images: Concurrent segmentation of image sets," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 128–142.

[32] G. Kim and E. Xing, "On multiple foreground cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 837–844.

[33] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 542–549.

[34] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Unsupervised 3d category discovery and point labeling from a large urban environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2685–2692.

[35] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Start from minimum labeling: Learning of 3d object models and point labeling from a large and complex environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3082–3089.

[36] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 193–200.

[37] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "From rgb-d images to rgb images: Single labeling for mining visual models," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, pp. 16:1–16:29, 2015.

[38] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.

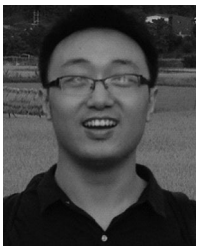
[39] Category dataset of kinect RGBD images [Online]. Available: http://sites.google.com/site/quanshizhang

[40] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

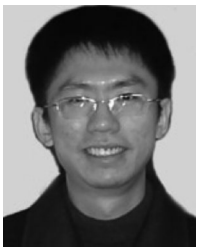
[42] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1482–1489.

[43] 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 *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 16–29.

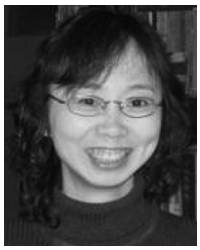


Quanshi Zhang received the BS degree in machine intelligence from Peking University, China, in 2009, and MS and PhD degrees in center for spatial information science from the University of Tokyo, Japan, in 2011 and 2014, respectively. In 2014, he joined the Center for Vision, Cognition, Learning, and Art, at the University of California, Los Angeles, as a post-doctoral researcher. His research is mainly in computer vision, robotics, and graph theory.

Xuan Song received the BS degree in information engineering from the Jilin University, China, in 2005 and the PhD degree in signal and information processing from Peking University, China, in 2010. From 2010 to 2012, he joined the Center for Spatial Information Science, The University of Tokyo as a post-doctoral researcher. In 2012 and 2015, he was promoted to a project assistant professor and project associate professor at the same university. His research is mainly in artificial intelligence, computer vision, and robotics.



Xiaowei Shao received the BE and PhD degree in electronic engineering and information science from the University of Science and Technology of China in 1999 and 2006, respectively. Since 2006, he worked in the Center for Spatial Information Science, University of Tokyo, Japan. Now, he has been a project associate professor at the same University. His research interests include computer vision and pattern recognition.



Huijing Zhao received the BS degree in computer science from Peking University, Beijing, China, in 1991, and the ME and PhD degrees in civil engineering from the University of Tokyo, Tokyo, Japan, in 1996 and 1999, respectively. In 2003, she became a visiting associate professor with the Center for Spatial Information Science. In 2007, she joined Peking University as a professor with the Key Laboratory of Machine Perception (MOE), and the School of Electronics Engineering and Computer Science. Her research interests include machine perception, intelligent vehicles, and spatial data handling.

Ryosuke Shibasaki received the MS and PhD degrees in civil engineering from the University of Tokyo in 1982 and 1987, respectively. From 1982 to 1988, he was with the Public Works Research Institute, Ministry of Construction. From 1988 to 1991, he was an associate professor in the Civil Engineering Department, University of Tokyo. In 1991, he joined the Institute of Industrial Science, University of Tokyo. In 1998, he was promoted to a professor in the Center for Spatial Information Science, University of Tokyo. His research interest covers three-dimensional data acquisition for GIS, conceptual modeling for spatial objects, and agent-based microsimulation in a GIS environment.