Mining And-Or Graphs for Graph Matching and Object Discovery

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu
University of California, Los Angeles
{zhangqs,ywu,sczhu}@ucla.edu

Abstract

This paper reformulates the theory of graph mining on the technical basis of graph matching, and extends its scope of applications to computer vision. Given a set of attributed relational graphs (ARGs), we propose to use a hierarchical And-Or Graph (AoG) to model the pattern of maximal-size common subgraphs embedded in the ARGs, and we develop a general method to mine the AoG model from the unlabeled ARGs. This method provides a general solution to the problem of mining hierarchical models from unannotated visual data without exhaustive search of objects. We apply our method to RGB/RGB-D images and videos to demonstrate its generality and the wide range of applicability. The code will be available at https://sites.google.com/site/quanshizhang/mining-and-or-graphs.

1. Introduction

In this paper, we redefine the concept of graph mining, which was originally developed for discovering frequent entity relations from tabular data, and reformulate its theory in order to solve problems in computer vision. Our theory aims to discover a hierarchical pattern to represent common subgraphs embedded in a number of unannotated Attributed Relational Graphs (ARGs). A theoretical solution to this graph-mining problem can have broad applications in different vision tasks, such as object discovery and recognition, scene understanding, etc.

A variety of visual data (*e.g.* RGB/RGB-D images and videos) can be represented as ARGs. Each ARG node may contain different unary attributes to represent different high-dimensional local features. Pairwise attributes on edges measure spatial relationships between object parts.

We define a hierarchical And-Or Graph (AoG) model to represent the common subgraphs in a set of ARGs, as shown in Fig. 1. The top AND node is a composition of a set of OR nodes. Each OR node describes a local part and has several alternative terminal nodes as local pattern candidates.

Task: In this study, we aim to mine (learn) the AoG (the model) from a number of ARGs (unlabeled data). Given the ARGs and an initial graph template, our method gradually modifies this template into the target AoG with the maximal number of OR nodes. During the graph-mining process, we discover new OR nodes and terminal nodes, delete redundant nodes, update attributes, and train matching parameters for the AoG. The mined AoG has stable graph-matching performance, and can be used for object inference in previously unseen ARGs (images or videos).

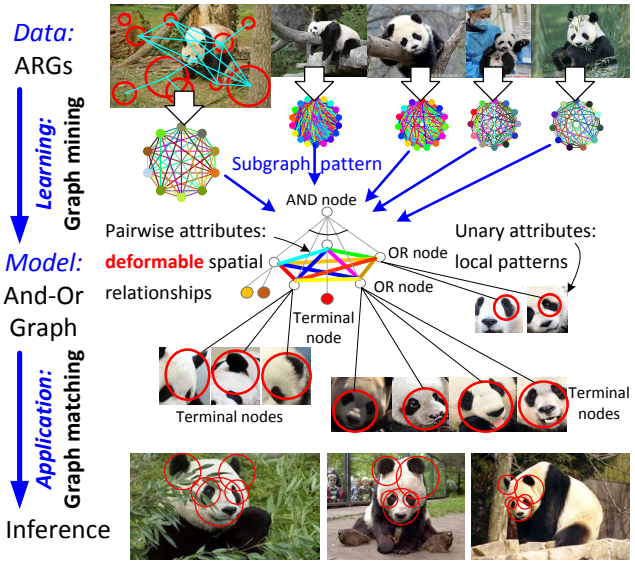

Figure 1. And-Or graph (AoG). The AoG models the regularities and variabilities of the common ***subgraphs*** embedded in the ARGs. The top AND node has several OR nodes as different object parts. Each OR node has several terminals to represent alternative local patterns. The AoG model can represent a variety of visual data, and we present an example for RGB images. Node/edge colors denote high-dimensional unary/pairwise attributes.

This study extends the concepts from both the fields of graph matching/mining and unsupervised learning, which are outlined below.

Graph matching & mining: As shown in Fig. 1, graph matching can be regarded as object inference. It maps nodes in a small graph template (a model) to a target ARG (*e.g.* an image) via a global optimization.

In contrast, graph mining is presented from the opposite perspective: When we use ARGs to represent images, com-

mon subgraphs in the ARGs correspond to the common objects inside. Graph mining is developed as a theoretical solution to the problem of discovering the common subgraphs and modeling their prototype pattern for good matching performance. Therefore, this can be regarded as a general method of learning deformable models from unlabeled data.

Unlike conventional single-layer patterns, in this study, we define a hierarchical AoG for modeling the regularities and variabilities of the subgraph pattern. Each OR node contains a number of alternative terminal nodes as local pattern candidates. In addition, we also incorporate the concept of negative (background) ARGs in the scope of graph mining. We discriminatively learn the AoG model to be exclusively contained inside positive ARGs.

Advantages: generality, applicability, & efficiency In terms of knowledge discovery, our method has the following advantages. Generality: unlike techniques oriented towards some particular types of data, our method can be applied to various kinds of visual data (*e.g.* RGB/RGB-D images and videos). People can use their own ARG attributes (features) to represent these visual data.

Applicability: it is necessary to develop a technique to simultaneously deal with all types of visual variations for broad applicability, which presents great challenges to the state-of-the-art *unsupervised* approaches. In our method, object occlusions are considered, and intra-category variations in texture, *roll* rotation, scale, and pose can all be automatically mined and formulated as attribute variations among ARGs.

Efficiency: in the era of big data, the mining efficiency has gradually become a new bottleneck. We propose an approximate but efficient method to directly discover the AoG model without enumerating AoG nodes from ARGs.

The contributions can be summarized as follows: 1) We reformulate the graph-mining theory for the field of computer vision. In particular, we propose a hierarchical AoG for modeling the subgraph pattern. 2) We require the pattern to be exclusively contained by positive ARGs to ensure its distinguishing capability. 3) For the purpose of visual knowledge discovery, we propose a general graph-mining method that can automatically extract the maximal-size AoG without applying an enumeration strategy. 4) Our method can be applied to various types of visual data.

2. Related work

Graph mining: Originally, the concept of "mining the maximal-size subgraph patterns" was developed in the domain of "labeled graphs," such as maximal frequent subgraph (MFS) extraction [29] and maximal clique mining [33, 36]. However, these methods require people to manually assign each node/edge in the graph with a certain label or determine some inter-graph node correspondence candidates based on local consistency. In addition, they usually apply time-consuming node enumeration for graph mining. Therefore, these methods are oriented towards tabular data.

However, for the ARGs representing visual data, the mining theory should be reformulated on the technical basis of graph matching[1]. The visual ARGs usually have large variations in attributes, and cannot provide node labels/correspondences in a local manner. Instead, the matching between two ARGs should be computed as a quadratic assignment problem (QAP) via a global optimization. [19, 1] estimated common objects or graph structures from images, but they did not provide a general solution in terms of graph theory. Zhang *et al.* [39, 41] did a pioneering study of graph mining for visual ARGs. In contrast, our method is more suitable for visual data. Unlike the single-layer pattern in [39, 41], we define a hierarchical AoG to represent object knowledge. Moreover, we introduce the concept of negative ARGs to ensure the pattern's discriminative capability.

Learning graph matching: Graph/image matching theories have been developed for decades [24, 5, 6, 35, 12, 28, 4]. Given a graph template, methods for learning graph matching usually train parameters or refine the template to achieve stable matching performance. Most of them [3, 2, 18, 30] are supervised methods, *i.e.* the matching assignments must be labeled for training. In contrast, unsupervised methods are more related to graph mining. Leordeanu *et al.* [20] trained attribute weights, and Zhang *et al.* [40] refined the template structure in an unsupervised fashion. These methods cannot discover new nodes from unlabeled graphs. Yan *et al.* [35] discovered common node correspondences between graphs without learning a model.

Weakly supervised learning, co-segmentation, & object discovery: Two issues in theories of weakly supervised model learning have long been ignored. First, the two interdependent terms—object similarity and part similarity—usually cannot be simultaneously measured in object discovery process. [22, 7] first exhaustively searched object candidates from unlabeled images, and then trained part-based models using a cluster of object-level similar candidates. [27] first extracted frequently appearing parts from images in a local manner, and then use them to define common objects. In contrast, our theory discovers objects in a more convincing way, *i.e.* simultaneously considering both part similarities and global object structure.

Second, other methods usually simplify the visual-mining problem by selectively modeling some visual variations and neglecting others. For example, co-segmentation [14] and object discovery [32] are mainly based on texture information and find it difficult to encode structural knowledge appropriately, whereas contour mod-

[1] Unlike graph mining oriented to labeled graphs, mining techniques for visual ARGs (as in [39, 41] and ours) usually label an inaccurate fragmentary pattern as an initial graph template to start the mining process.

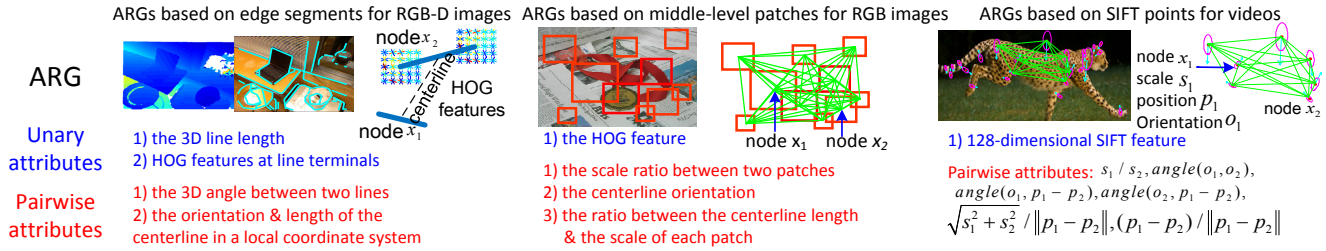

Figure 2. Representation of ARG attributes for a variety of visual data. For clarity, we show sub-ARGs that may not contain all the nodes.

els [16, 34] focuses on geometric shapes and ignores textures. In contrast, our hierarchical AoG model can encode different variations in visual data.

And-Or graph: Compared deformable part models (DPMs), the hierarchical AoG model has more expressive power and more transparency in its description of objects. Pioneering studies [25, 31] all trained the AoG in a supervised fashion. In contrast, we propose to directly mine such AoGs without labeling object bounding boxes.

3. And-or graph representation

ARG: As shown in Fig. 3, an ARG $\mathcal{G}$ is a three-tuple $\mathcal{G} = (\mathcal{V}, \mathbf{F}_\mathcal{V}, \mathbf{F}_\mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the node and edge set, respectively. All pairs of nodes are fully connected to form a complete graph. $\mathbf{F}_\mathcal{V} = \{\mathcal{F}_i^x | x \in \mathcal{V}, i = 1, 2, ..., N_u\}$ denotes the set of unary attributes. Each node x has N_u unary attributes. The set of pairwise attributes $\mathbf{F}_\mathcal{E} = \{\mathcal{F}_j^{x_1 x_2} | x_1, x_2 \in \mathcal{V}, x_1 \neq x_2, j = 1, 2, ..., N_p\}$ assign each edge $\langle x_1, x_2 \rangle$ with N_p pairwise attributes. Each attribute corresponds to a feature vector.

Attributes (features): For different visual data, we can use different unary and pairwise attributes to represent local features and spatial relationships.

For example, in our experiments, we use four types of ARGs to represent three kinds of visual data, including RGB-D images, RGB images, and videos. We illustrate the attribute design for three types of ARGs in Fig. 2. In these ARGs, we use 1) line segments of object edges, 2) automatically learned middle-level patches, and 3) SIFT points as graph nodes. Thus, their unary attributes can be defined as 1) a combination of HoG features extracted at line terminals and line lengths, 2) HoG features of middle-level patches, and 3) 128-dimensional SIFT features, respectively.

Accordingly, the pairwise attributes between each pair of nodes x_1 and x_2 can be defined as a combination of 1) the angle between the own orientations of x_1 and x_2, 2) the orientation of the line connecting x_1 and x_2 (namely, the *centerline* of x_1 and x_2), 3) the angle between the centerline and each of x_1 and x_2, 4) the ratio between the scales of x_1 and x_2, and 5) the ratio between the centerline length and the scale of each node. In addition, people can design their own attributes for each specific task.

AoG: Similarly, a hierarchical AoG is defined as a five-tuple $G = (V, \mathbf{\Omega}, \mathbf{F}_V, \mathbf{F}_E, \mathbf{W})$. The AoG has three layers. The top AND node has a set of OR nodes $s \in V$. Each OR node s has a set of terminal nodes $\psi_s \in \Omega_s$. Each terminal node ψ_s has N_u unary attributes $\mathbf{F}_V = \{F_i^{\psi_s} | s \in V, \psi_s \in \Omega_s, i = 1, 2, ..., N_u\}$. $\mathbf{\Omega} = \bigcup_{s \in V} \Omega_s$ denotes the overall set of terminal nodes. We connect all pairs of OR nodes to form a complete graph. Each edge $\langle s, t \rangle \in E$ contains N_p pairwise attributes $\mathbf{F}_E = \{F_j^{st} | s \neq t \in V, j = 1, 2, ..., N_p\}$ ($\langle s, t \rangle$ and $\langle t, s \rangle$ denote two different edges). $\mathbf{W}$ is a set of matching parameters, which will be introduced later.

4. Inference: Graph matching for the AoG

The matching between an AoG G and an ARG $\mathcal{G}$ is denoted by $G \mapsto \mathcal{G}|_{\Psi, \mathbf{x}}$. It selects a terminal node $\psi_s \in \Omega_s$ under each OR node $s \in V$ and maps ψ_s to an ARG node $x_s \in \mathcal{V}$, which is given as $s \mapsto \mathcal{G}|_{\psi_s x_s}$. All mapping assignments are represented by $\mathbf{x} = \{x_s | s \in V\}$ and $\Psi = \{\psi_s | s \in V\}$. The matching process can be formulated as the maximization/minimization of the following probability/energy function *w.r.t.* Ψ and $\mathbf{x}$:

$$\begin{aligned} &P(G \mapsto \mathcal{G}) \propto \exp\big[-\mathcal{E}(G \mapsto \mathcal{G})\big] \\ &\mathcal{E}(G \mapsto \mathcal{G}) = \sum\nolimits_{s \in V} \mathcal{E}(s \mapsto \mathcal{G}) + \sum\nolimits_{\langle s,t \rangle \in E} \mathcal{E}(\langle s, t \rangle \mapsto \mathcal{G}) \end{aligned} \tag{1}$$

where $P(G \mapsto \mathcal{G})$ and $\mathcal{E}(G \mapsto \mathcal{G})$ denote the matching probability/energy of $G \mapsto \mathcal{G}$, respectively. The overall matching energy consists of unary and pairwise matching energies, *i.e.* $\mathcal{E}(s \mapsto \mathcal{G})$ and $\mathcal{E}(\langle s, t \rangle \mapsto \mathcal{G})$. To simplify notations, we use $G \mapsto \mathcal{G}$, $s \mapsto \mathcal{G}$, and $\langle s, t \rangle \mapsto \mathcal{G}$ to represent $G \mapsto \mathcal{G}|_{\Psi, \mathbf{x}}$, $s \mapsto \mathcal{G}|_{\psi_s x_s}$, and $\langle s, t \rangle \mapsto \mathcal{G}|_{\psi_s \psi_t x_s x_t}$ without ambiguity. The matching energies can be defined using squared attribute differences:

$$\mathcal{E}(s \mapsto \mathcal{G}) = \begin{cases} \sum_{i=1}^{N_u} w_i^u \|F_i^{\psi_s} - \mathcal{F}_i^{x_s}\|^2, & x_s \in \mathcal{V} \\ u_{none}, & x_s = none \end{cases} \tag{2}$$

$$\mathcal{E}(\langle s, t \rangle \mapsto \mathcal{G}) = \begin{cases} \sum_{j=1}^{N_p} \frac{w_j^p \|F_j^{st} - \mathcal{F}_j^{x_s x_t}\|^2}{|V|-1}, & x_s \neq x_t \in \mathcal{V} \\ +\infty, & x_s = x_t \in \mathcal{V} \\ \frac{1}{|V|-1} p_{none}, & x_s \text{ or } x_t = none \end{cases}$$

where $\|\cdot\|$ measures the Euclidean distance. We design a dummy node "*none*" for **occlusion**. ψ_s is assigned to *none*, when its corresponding node in $\mathcal{G}$ is occluded. u_{none} and p_{none} denote the energies of matching to *none*. $w_i^u, w_j^p > 0$ represent weights for attributes $\mathcal{F}_i^{\psi_s}$ and $\mathcal{F}_j^{st}$, respectively. The matching parameters are set as $\mathbf{W} = \{w_i^u | i = 1, ..., N_u\} \cup \{w_j^p | j = 1, ..., N_p\} \cup \{u_{none}, p_{none}\}$.

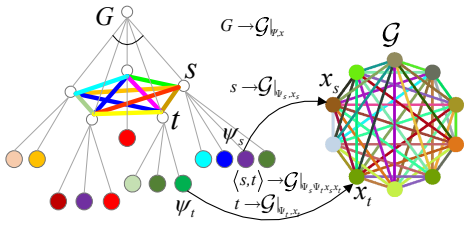

Figure 3. Notations for graph matching

We use the marginal matching energy of OR node s, denoted by $\mathcal{E}_s^{\mathcal{G}}$ to evaluate the local matching quality of s.

$$\mathcal{E}_s^{\mathcal{G}} = \mathcal{E}(s \mapsto \mathcal{G}) + \sum_{t:\langle s,t\rangle \in E} \mathcal{E}(\langle s,t\rangle \mapsto \mathcal{G}) \;\Rightarrow\; \mathcal{E}(G \mapsto \mathcal{G}) = \sum_{s \in V} \mathcal{E}_s^{\mathcal{G}} \quad (3)$$

For each OR node s, we apply a standard inference strategy for OR nodes [25, 31], *i.e.* matching its best terminal $\psi_s \in \Omega_s$ to x_s that minimizes its unary matching energy, $\text{argmin}_{\psi_s} \mathcal{E}(s \mapsto \mathcal{G})$.

Therefore, node correspondences $(\mathbf{x}, \Psi)$ are computed by $\min_{\mathbf{x},\Psi} \mathcal{E}(G \mapsto \mathcal{G}) = \min_{\mathbf{x}} \{\sum_{s \in V} \min_{\psi_s} \mathcal{E}(s \mapsto \mathcal{G}) + \sum_{\langle s,t\rangle \in E} \mathcal{E}(\langle s,t\rangle \mapsto \mathcal{G})$, which is a typical QAP. The optimal terminals Ψ are selected in a local manner. The matching assignments $\mathbf{x}$ can be estimated via energy minimization of an MRF *w.r.t* $\{x_s\}$. We use the TRW-S [15] for energy minimization.

Average marginal energy: Given a set of ARGs $\Lambda = \{\mathcal{G}_k\}$ and an AoG G, we match G to each $\mathcal{G}_k$, and obtain its matching assignments $\hat{\mathbf{x}}^k = \{\hat{x}_s^k | s \in V\}$, $\hat{\Psi}^k = \{\hat{\psi}_s^k | s \in V\}$ via $\text{argmin}_{\mathbf{x}^k, \Psi^k} \mathcal{E}(G \mapsto \mathcal{G}_k)$. The average matching quality for each OR node s in G can be measured by its average marginal energy:

$$\mathcal{E}_s = \text{mean}_{\mathcal{G}_k \in \Lambda} \mathcal{E}_s^{\mathcal{G}_k} \quad (4)$$

5. Learning: Graph mining

5.1. Objective

In this subsection, we define the objective of graph mining, which describes the optimal states of the AoG and is used to guide all the mining operations. Given a set of positive ARGs $\Lambda^+ = \{\mathcal{G}_k^+ | k = 1, ..., N^+\}$ and a set of negative ARGs $\Lambda^- = \{\mathcal{G}_l^- | l = 1, ..., N^-\}$, the objective is defined as

$$G^* = \underset{G}{\text{argmax}} \left\{ \frac{P(\Lambda^+|G)}{P(\Lambda^-|G)} \cdot e^{-\lambda \text{Complexity}(G)} \right\} \quad (5)$$

We aim to maximize the gap between the probability of positive matches $P(\Lambda^+|G)$ and that of negative matches $P(\Lambda^-|G)$. Meanwhile, we also consider the complexity of the AoG to avoid overfitting. The probability of positive matches can be given as

$$P(\Lambda^+|G) \propto \prod_{k=1}^{N^+} P(G \mapsto \mathcal{G}_k^+)^{\frac{1}{N^+}} \quad (6)$$

Because substantial terminal divisions for OR nodes will increase the risk of overfitting, we use the graph size to define the AoG complexity, which is similar to [31].

$$\text{Complexity}(G) = |\boldsymbol{\Omega}| + \beta|V| = \sum_{s \in V} (|\Omega_s| + \beta) \quad (7)$$

Therefore, we can re-write the objective in (5) as

$$G^* = \underset{G}{\text{argmax}} \log \left\{ \frac{P(\Lambda^+|G)}{P(\Lambda^-|G)} \cdot e^{-\lambda \text{Complexity}(G)} \right\} \quad (8)$$
$$= \underset{G}{\text{argmin}} \{ \underbrace{\sum_{s \in V} (\mathcal{E}_s^+ - \mathcal{E}_s^-)}_{\text{generative loss; we hope } \varepsilon_s^+ \ll \varepsilon_s^-.} + \underbrace{\sum_{s \in V} \lambda(|\Omega_s| + \beta)}_{\text{complexity loss}} \}$$

where for each OR node s, $\mathcal{E}_s^+$ and $\mathcal{E}_s^-$ denote its marginal energies among positive and negative matches, respectively. The generative loss represents the gap between marginal energies, which describes the discriminative power of G.

The comprehensive mining of the AoG includes 1) determination of the corresponding subgraphs in ARGs, 2) discovery of new OR nodes and terminal nodes, 3) elimination of redundant nodes, 4) attribute estimation, and 5) training of matching parameters. Therefore, we propose the following sub-objectives, namely **Objs.(a–d)**, to train each of these terms. **Objs.(a–d)** alternatively maximize the above logarithmic form of the objective in (5)[2].

First, **Obj.(a)** presents an object-inference procedure using the current pattern. It estimates the most probably object in each positive/negative image, *i.e.* computing the best matching assignments to each positive/negative ARG $\mathcal{G}_k^+/\mathcal{G}_l^-$ via graph matching. Marginal energies $\{\mathcal{E}_s^+\}$ and $\{\mathcal{E}_s^-\}$ are defined using these matching assignments.

Obj.(a): $\text{argmin}_{\hat{\mathbf{x}}^k, \hat{\Psi}^k} \mathcal{E}(G \mapsto \mathcal{G}_k^+)$, $\text{argmin}_{\hat{\mathbf{x}}^l, \hat{\Psi}^l} \mathcal{E}(G \mapsto \mathcal{G}_l^-)$

Second, for each OR node s, we use the following equation to uniformly optimize its unary and pairwise attributes and the division of its terminal nodes.

Obj.(b): $\text{argmin}_{\boldsymbol{\Omega}, \mathbf{F}_V, \mathbf{F}_E} \sum_{s \in V} (\mathcal{E}_s^+ - \mathcal{E}_s^- + \lambda|\Omega_s|)$

This is similar to sparse representation. First, we minimize the generative loss by enlarging the matching-energy gap between positive and negative matches, so that the AoG attributes can represent the pattern that is exclusively contained by positive ARGs. Second, we use the complexity loss to limit the terminal number. We simply set $\lambda = 1.0$ for all the categories in all the experiments.

Third, we need to grow the current pattern to the maximal size, *i.e.* extracting an AoG with the maximal number of OR nodes $|V|$ by discovering new OR nodes and deleting redundant OR nodes. Thus, we apply a threshold τ to control both the node discovery and elimination.

Obj.(c): $\text{argmax}|V| \quad \text{s.t. } \forall s \in V, \mathcal{E}_s^+ - \mathcal{E}_s^- + \lambda|\Omega_s| \leq \tau$

Finally, we use a linear SVM to train the matching parameters $\mathbf{W}$, which is an approximate solution to the minimization of the generative loss in (9).

Obj.(d): $\min_{\mathbf{W}} \|\mathbf{w}\|^2 + \frac{C}{N^+} \sum_{k=1}^{N^+} \xi_k^+ + \frac{C}{N^-} \sum_{l=1}^{N^-} \xi_l^-$,
$\forall k = 1, 2, ..., N^+, -[\mathcal{E}(G \mapsto \mathcal{G}_k^+) + b] \geq 1 - \xi_k^+$,
$\forall l = 1, 2, ..., N^-, \mathcal{E}(G \mapsto \mathcal{G}_l^-) + b \geq 1 - \xi_l^-$

[2]Please see the supplementary materials for proofs.

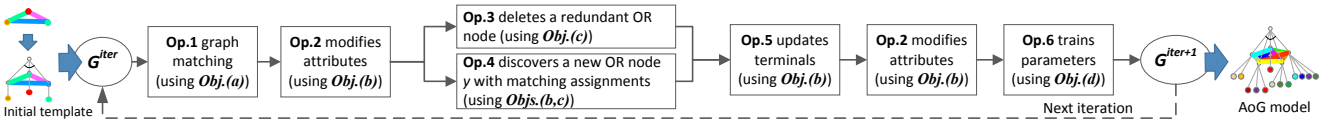

Figure 4. Flowchart of an approximate solution to graph mining

5.2. Flowchart

As shown in Fig. 4, we design an EM flowchart to mine the optimal AoG defined in (5). In the beginning, we need to label an initial graph template G^0 that roughly corresponds to an object fragment. Then, the algorithm recursively optimizes the graph pattern and grows the number of OR nodes to the maximum $G^0 \rightarrow G^1 \rightarrow ... \rightarrow G^n = \text{AoG}$. On the basis of **Objs.**(a–d), we define a total of six operations to construct the EM flowchart. We can demonstrate[2] that these operations lead to an approximate but efficient solution to (5).

Initialization: We label the object in a positive ARG to initialize the graph template G^0. G^0 is a special AoG, in which each OR node contains a single terminal node. Even bad labeling, *e.g.* an object fragment mixed with background area, is acceptable. We then initialize parameters in $\mathbf{W}$ as $u_{none} = p_{none} = +\infty$ and $w^u_{i=1,...,N_u} = w^p_{j=1,...,N_p} = 1$.

Operation 1, graph matching: We use **Obj.**(a) to match G to each of the positive and negative ARGs.

Operation 2, attribute estimation: This is based on **Obj.**(b). We approximate[2] the unary/pairwise attribute on each terminal/edge as the average attribute among its corresponding nodes/edges in the positive ARGs.

Operation 3, node elimination: Given the matching results, we use **Obj.**(c) to identify the worst OR node as $S_{worst} = \text{argmax}_s(\mathcal{E}^+_s - \mathcal{E}^-_s + \lambda|\Omega_s|)$. If $\mathcal{E}^+_{S_{worst}} - \mathcal{E}^-_{S_{worst}} + \lambda|\Omega_{S_{worst}}| > \tau$, node S_{worst} and all its terminals will be deleted from G; otherwise, they are not.

Operation 4, node discovery: Node discovery aims to discover a new OR node y for G. We use **Objs.**(b,c) to simultaneously 1) determine y's terminal nodes Ω_y, 2) estimate its attributes $(\{F^{\psi_y}_i\}, \{F^{yt}_j\})$, and 3) compute its matching assignments $(\{\hat{x}^k_y\}, \{\hat{\psi}^l_y\})$. As the new node y should be well matched to most of the positive ARGs in Λ^+, to simplify the calculation, we ignore the small possibility of y being matched to *none*. Based on this assumption, we have proved[2] an approximate solution that 1) provides a rough estimation of y's positive matching assignments $\{\hat{x}^k_y\}$ and 2) minimizes uncertainty of other parameters:

$$\min_{\{\hat{x}^k_y\}} \left\{ \sum_{k=1}^{N^+} \Phi(\hat{x}^k_y) + \sum_{k_1=1}^{N^+} \sum_{k_2=1}^{N^+} \Phi(\hat{x}^{k_1}_y, \hat{x}^{k_2}_y) \right\} \quad (9)$$

$$\Phi(\hat{x}^k_y) = -\sum_{l=1}^{N^-} \min_{\check{x}^l_y} \left[\sum_{t \in V} \frac{\mathbf{1}(\hat{x}^k_t)\mathcal{E}(\langle \hat{x}^k_y, \hat{x}^k_t \rangle \mapsto \langle \check{x}^l_y, \check{x}^l_t \rangle)}{|V|(N^-)\sum_j \mathbf{1}(\hat{x}^j_t)} + \frac{\mathcal{E}(\hat{x}^k_y \mapsto \check{x}^l_y)}{(N^-)(N^+)} \right]$$

$$\Phi(\hat{x}^{k_1}_y, \hat{x}^{k_2}_y) = \frac{\mathcal{E}(\hat{x}^{k_1}_y \mapsto \hat{x}^{k_2}_y)}{(N^+)^2} + \sum_{t \in V} \left(\frac{[1 - \mathbf{1}(\hat{x}^{k_1}_t)\mathbf{1}(\hat{x}^{k_2}_t)]p_{none}}{|V|(N^+)[(N^+) + \sum_j \mathbf{1}(\hat{x}^j_t)]} \right.$$
$$\left. + \frac{\mathbf{1}(\hat{x}^{k_1}_t)\mathbf{1}(\hat{x}^{k_2}_t)(\sum_j \mathbf{1}(\hat{x}^j_t) + N^+)}{2|V|(N^+)[\sum_j \mathbf{1}(\hat{x}^j_t)]^2} \mathcal{E}(\langle \hat{x}^{k_1}_y, \hat{x}^{k_1}_t \rangle \mapsto \langle \hat{x}^{k_2}_y, \hat{x}^{k_2}_t \rangle) \right)$$

where $\mathbf{1}(x)$ returns 0, if $x = none$; otherwise, 1.$\mathcal{E}(x_1 \mapsto x_2) = \sum_{i=1}^{N_u} w^u_i \|\mathcal{F}^{x_1}_i - \mathcal{F}^{x_2}_i\|^2$; $\mathcal{E}(\langle x_{11}, x_{12} \rangle \mapsto \langle x_{21}, x_{22} \rangle) = \sum_{j=1}^{N_p} w^p_j \|\mathcal{F}^{x_{11}x_{12}}_j - \mathcal{F}^{x_{21}x_{22}}_j\|^2 / |V| - 1$. Thus, $\{\hat{x}^k_y\}$ can be directly estimated via energy minimization of an MRF [15]. Then, given $\{\hat{x}^k_y\}$, other parameters ($\{\hat{\psi}^k_y\}$, Ω_y, $\{F^{\psi_y}_i\}$, and $\{F^{yt}_j\}$) can be consequently determined and iteratively refined (see the flowchart in Fig. 4).

Operation 5, terminal determination: Given matching assignments $\{\hat{x}^k_s\}$ and $\{\check{x}^l_s\}$ of each OR node s, this operation uses **Obj.**(b) to modify the terminal number of s, $|\Omega_s|$, and meanwhile refine terminal assignments $\{\hat{\psi}^k_s\}$. This operation can be approximated[2] as a hierarchical clustering: For each node s, we use a set of feature points $\{\mathbf{f}^{\hat{x}^k_s} | 1 \le k \le N^+, \mathbf{1}(\hat{x}^k_s) = 1\}$ to represent its corresponding nodes in positive ARGs $\{\hat{x}^k_s\}$, each as $\mathbf{f}^{\hat{x}^k_s} = [\sqrt{w^u_1}(\mathcal{F}^{\hat{x}^k_s}_1)^T, \ldots, \sqrt{w^u_{N_u}}(\mathcal{F}^{\hat{x}^k_s}_{N_u})^T]^T$. We group these points to several clusters via a hierarchical clustering. In this way, we can construct the terminal set Ω_s and use each terminal node $\psi_s \in \Omega_s$ to represent a cluster, *i.e.* each ARG node $\hat{x}^k_s$ in this cluster is matched to ψ_s, $\hat{\psi}^k_s = \psi_s$. Unary attributes of ψ_s, $\{F^{\psi_s}_i\}$, correspond to the cluster center. We need to keep merging the nearest clusters, until the following energy is minimized[2].

$$\min_{\Omega_s, \{\hat{\psi}^k_s\}} \left\{ C \sum_{\psi_s \in \Omega_s} \sum_{\substack{1 \le k_1 \le N^+: \\ \hat{\psi}^{k_1}_s = \psi_s, \mathbf{1}(\hat{x}^{k_1}_s) = 1}} \|\mathbf{f}^{\hat{x}^{k_1}_s} - \text{mean}_{\substack{1 \le k_2 \le N^+: \\ \hat{\psi}^{k_2}_s = \psi_s, \mathbf{1}(\hat{x}^{k_2}_s) = 1}} \mathbf{f}^{\hat{x}^{k_2}_s}\|^2 + \lambda|\Omega_s| \right\} \quad (10)$$

where $C = 1/N^+ + \sum_j \mathbf{1}(\check{x}^j_s)/[(N^-)\sum_j \mathbf{1}(\hat{x}^j_s)]$.

Operation 6, parameter training: This is defined in **Obj.**(d). Given the current matching assignments, we apply the technique in [43] to train parameters in $\mathbf{W}$.

6. Experiments

Our method provides a visual-mining solution that can be applied to a variety of visual data. Therefore, to test the generality of the algorithm, we applied our method to three kinds of visual data (*i.e.* RGB-D images, RGB images, and videos) in four experiments. In these experiments, we mined object models (*i.e.* AoGs) for 38 categories from these visual data. The mined models can be used to 1) collect object samples from the unlabeled training data and 2) match objects in perviously unseen video frames/images. Thus, we evaluate our method in both terms of graph matching and object discovery.

Because our method extends concepts ranging across the fields of graph matching, graph mining, and weakly-supervised learning, we comprehensively compared our method with a total of 13 competing methods. These methods include image/graph matching approaches, unsupervised learning for graph matching, the pioneering method

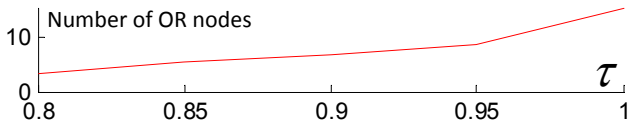

Figure 5. The average pattern size of the mined AoGs monotonically increases with increasing values of τ in Experiment 1.

of mining from visual ARGs, object discovery, and co-segmentation methods.

6.1. Settings for the four experiments

The inputs are the unlabeled RGB-D/RGB images and videos. In the four experiments, we used four types of ARGs, respectively, to represent these data. Each RGB-D/RGB image or video frame is represented as an ARG. Then, we mined AoGs as category models from these ARGs. In Experiments 1, 2, and 4, the AoGs were used to match objects in testing images and videos, and in Experiment 3, we focused on the object-discovery performance during the mining process. Detailed experimental settings are introduced as follows.

Experiment 1, Mining from RGB-D images: We applied our method to five categories in the Kinect RGB-D image database [40]—*notebook PC*, *drink box*, *basket*, *bucket*, and *dustpan*, which is widely used [40, 39, 41, 42] as a benchmark dataset to evaluate graph matching performance. As shown in Fig. 2, the ARGs for RGB-D images were designed by [40], which were robust to rotation and scale variations in graph matching. [40] extracted object edges from images, and divided them into line segments. These line segments were used as nodes of the ARGs.

We tested the graph-mining performance of our method under different settings of parameter τ. As in [41, 40], given each value of τ, we followed the process of leave-one-out cross-validation to evaluate the mining performance: [41] has labeled a set of initial graph templates G^0 for each category, and we performed an individual model-mining process to produce an AoG using each of these templates. The overall graph-mining performance was evaluated by the average performance among all the mined AoGs.

Experiment 2, Mining from unlabeled RGB images: We use the second type of ARGs introduced in [40] to represent RGB images. Just like the ARGs for RGB-D images, the ARGs for RGB images also take edge segments as nodes. We used the ETHZ Shape Class dataset [11], which contains five classes, *i.e.* *apple logos*, *bottles*, *giraffes*, *mugs*, and *swans*. We randomly select $2/3$ of the images for training and leave the other $1/3$ for testing.

Experiment 3, Mining from ARGs based on middle-level patches: We mined models in 25 categories from the SIVAL dataset [23] and 12 categories from the PASCAL07-6×2 (training) dataset [8]. We applied the ARGs designed by [43] for more general RGB images. We used [26] to extract middle-level patches as ARG nodes[3] (see Fig. 2).

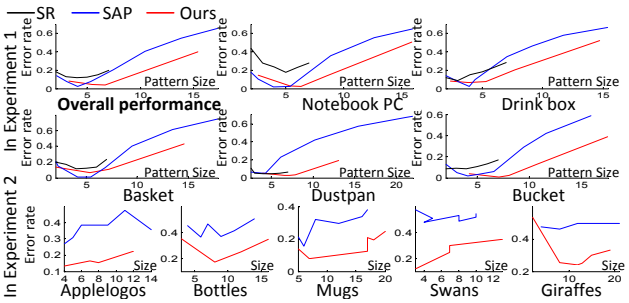

Figure 6. Average error matching rates. In Experiments 1 and 2, we mine edge-based subgraph patterns from RGB-D and RGB images. Our method has lower error rates than other baselines.

In this experiment, we learned a mixture model with three AoG components for each category. Thus, we labeled three initial templates to start the mining process. In addition, we modified the *Operation 1* to separate the ARGs into the three AoGs in each mining iteration, which assigned each ARG with its best matched AoG.

Experiment 4, Mining from video sequences: We collected three video sequences (containing a cheetah, swimming girls, and a frog) from the Internet, and used our method to mine AoGs for deformable objects from these videos. In this experiment, each video frame was represented as an ARG. We applied the ARGs designed by [43] to represent the video frame (see Fig. 2). These ARGs take SIFT feature points as nodes.

6.2. Baselines

We used a total of thirteen competing methods. In terms of graph matching and mining, seven methods followed the scenario of "learning models or matching objects with a single template." The other six methods were state-of-the-art object discovery and co-segmentation approaches.

Graph matching & mining: All the seven methods were provided with the same initial graph templates, as well as the same sets of training ARGs and testing ARGs to ensure a fair comparison.

First, we compared our method to three image/graph-matching methods. These methods directly matched the graph template to the testing ARGs without training. In general, there are two typical paradigms for image matching. The competing method *MA* was designed to represent the first paradigm, *i.e.* the minimization of matching energy. *MA* used TRW-S [15] to minimize the matching energy in (1)[4]. Then, we used competing methods *MS* and *MT* to represent the second paradigm, *i.e.* the maximization of matching compatibility. As in [17,

[3]For images from the PASCAL07 dataset, [26] selected middle-level patches from a number of region candidates that were provided by [21].

[4]This matches two single-layer graphs without hierarchical structures, which is a special case for the energy in (1) where $\Psi \equiv \mathbf{1}$.

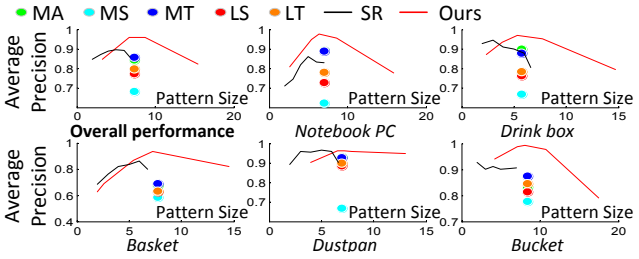

Figure 7. APs of object detection in Experiment 1. Our method performed better in mining models from RGB-D images.

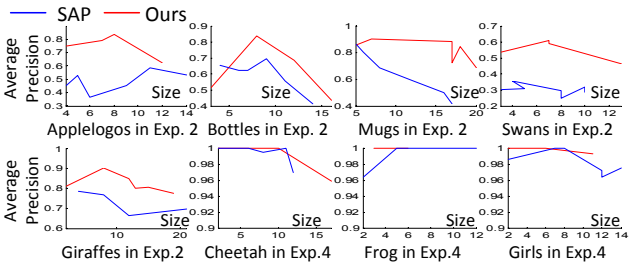

Figure 8. APs of object detection in Experiments 2 and 4. Our method mined better models from RGB images and videos.

20], the matching compatibility was defined as $\mathcal{C}(\mathbf{x}) = \sum_{s,t} e^{-\mathcal{E}(s\mapsto\mathcal{G})-\mathcal{E}(t\mapsto\mathcal{G})-\mathcal{E}(\langle s,t\rangle\mapsto\mathcal{G})}$, where they used absolute attribute differences to define matching energies. *MS* and *MT* used spectral techniques [17] and TRW-S [15], respectively, to compute $\text{argmax}_{\mathbf{x}}\,\mathcal{C}(\mathbf{x})$.

Second, we compared our method to the benchmark of unsupervised learning for graph matching proposed by Leordeanu *et al.* [20]. Two competing methods, *i.e. LS* and *LT*, were implemented to learn attribute weights based on [20]. *LS* and *LT* used [17] and [15], respectively, to solve the matching optimization during the learning procedure.

Third, we compared our method to [40], denoted by *SR*. *SR* refines the structure of the graph template by deleting "bad" nodes, but does not involve the key component for graph mining, *i.e.* the discovery of new pattern nodes.

Finally, we compared the proposed method to the earlier method that mines the soft attributed patterns (SAP) from visual ARGs [41]. We initialized u_{none}, p_{none}, and $\mathbf{w}$ for [41] as our initializations to enable a fair comparison.

Object discovery & co-segmentation: *Operation 1* in each model-mining iteration can be regarded as a process of object discovery; therefore, we compared the object discovery performance between the mined AoG and six recent methods for object discovery and co-segmentation: saliency-guided multiple class learning (*bMCL*) [44]; a state-of-the-art object-discovery approach [13] (*UnSL*) that achieves top performance (approximately 98% measured in purity) on a subset of Caltech-101 [9]; a foreground co-segmentation method [14] (*MFC*); two multiple instance clustering approaches, *BAMIC* [38] (with the best distance metric) and *M^{3}IC* [37]; and a K-means clustering of the most "salient" window obtained with [10] in each image (called *SD* and implemented by [44]).

6.3. Evaluation metrics, results, & analysis

We used the mined AoGs to match target objects in the four experiments. Fig. 9 illustrates the matching results. The parameter τ controls the pattern size[5]. Fig. 5 shows how the pattern size changes with the value of τ in Experiment 1. A greater value of τ would produce a larger AoG.

[5]For single-layer graph models used in baselines, this is the total node number. For our hierarchical AoG, this is the number of OR nodes.

Comparisons in terms of image/graph matching and mining: We used the following two metrics to evaluate the graph-matching performance of the mined AoGs. First, as in [42], the error matching rate (EMR) was used to measure the matching accuracy. When we matched an AoG to a positive ARG, its error rate was defined as the proportion of the ARG nodes that were matched by the AoG but located in the background, *i.e.* $|\mathcal{V}^M \setminus \mathcal{V}^O|/|\mathcal{V}^M|$, where $\mathcal{V}^M$ denotes the set of ARG nodes that were matched by the AoG, and $\mathcal{V}^O \subseteq \mathcal{V}^M$ represents the subset of nodes that were correctly localized on the target object. Thus, given an AoG, its EMR was computed across all its positive matches.

Second, we used the AoG to detect (match) target objects among a number of previously unseen positive and negative ARGs. We used the simplest means of identifying the true and false detections: given a threshold, if the matching energy is less than the threshold[6], we consider this to be a true detection; otherwise, it is a false detection. Thus, we can draw an object detection precision-recall curve by choosing different thresholds. The average precision (AP) of the precision-recall curve was used as a metric to evaluate the matching-based object detection.

Fig. 6 compares error rates of our method with seven competing methods in the first two experiments. Note that the matching performance is a function of the pattern size[5]. A pattern with too few nodes may lack sufficient information for reliable matching, while a too large pattern may contain unreliable nodes, which decreases the performance. The *SR* method cannot produce large patterns, because it simply deletes redundant nodes without the capability for discovering new nodes. In general, our method exhibits lower matching rates than competing methods. Figs. 7 and 8 show the APs of the mined patterns in Experiments 1, 2, and 4. Except *SR*, *SAP*, and our method, other competing methods cannot change the pattern size. As mentioned in Experiment 1, we used different initial templates to produce a number of models for each category. Thus, in Figs. 7, both the pattern size and the ER were computed via cross-validation. Our method shows superior performance to the competing methods.

[6]For competing methods *MS*, *MT*, *LS*, *LT*, and *SR*, a true detection is identified if the matching compatibility is greater than the threshold.

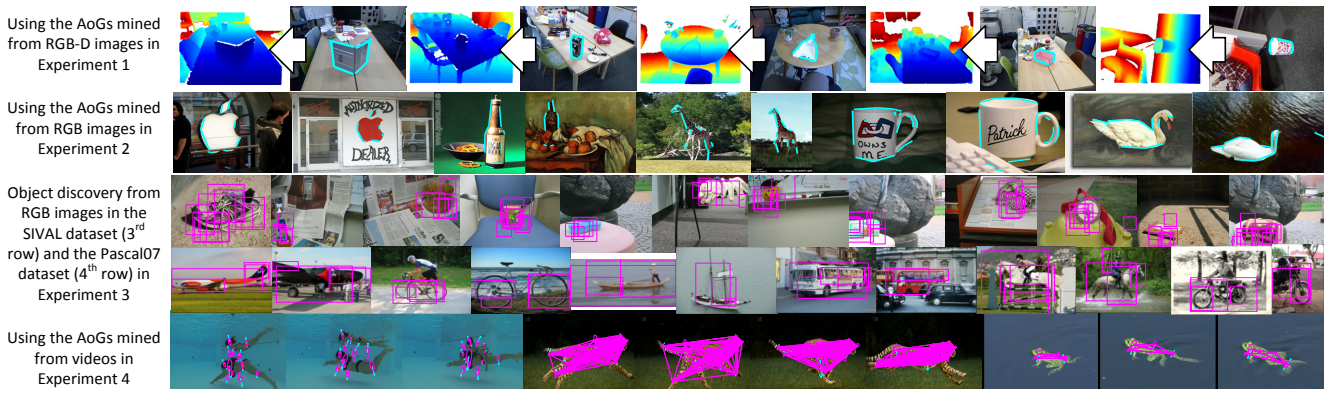

Figure 9. Matching results based on AoGs. We draw edge connections of the frog and cheetah models to show their structure deformation.

	Ours	bMCL	SD	M^{3}IC	BAMIC	UnSL	MFC
SIVAL1	89.0	**95.3**	80.4	39.3	38.0	27.0	45.0
SIVAL2	**93.2**	84.0	71.7	40.0	33.3	35.3	33.3
SIVAL3	**88.4**	74.7	62.7	37.3	38.7	26.7	41.3
SIVAL4	87.8	**94.0**	86.0	33.0	37.7	27.3	53.0
SIVAL5	**92.7**	75.3	70.3	35.3	37.7	25.0	48.3
Average	**90.2**	84.7	74.2	37.0	37.1	28.3	44.2

Table 1. Average purity of object discovery in the SIVAL dataset

Pose		aero.	bicy.	boat	bus	horse	moto.	**Avg.**
Left	MA	16.1	18.5	10.6	42.9	11.3	27.7	21.3
	Ours	73.2	64.6	29.8	71.4	58.1	80.9	63.2
Right	MA	13.5	10.7	17.3	57.9	8.20	20.5	
	Ours	75.0	66.1	42.3	73.7	45.9	77.3	

Table 2. Average localization rate in PASCAL07-6×2 dataset

Comparisons in terms of co-segmentation & object discovery: First, we compared our method with six object discovery and co-segmentation approaches in Experiment 3. The automatically mined category models can be used to collect object samples from the SVIAL dataset, which can be considered object discovery. As in object-discovery work [16, 44], such an object collection was understood as a clustering of unannotated objects, and the clustering purity was chosen as the evaluation metric. We took the models in different categories as cluster centers. For each image in the dataset, we used all the models to match target objects in this image. We regarded the object with the lowest matching energy as a true detection, and assigned it to its corresponding cluster. The clustering purity was computed for each cluster (background objects were considered as incorrect samples). Note that [44] partitioned the 25 categories in the dataset into 5 sub-groups, respectively; namely, *SIVAL1* to *SIVAL5*. We measured the average clustering purity within each of these sub-groups for evaluation. Please see Table 1 for comparison. Our method exhibits better performance.

Second, object discovery performance in the PASCAL07-6 × 2 training dataset was evaluated using the average localization rate. Just like the "$IOU > 50\%$" criterion in [22, 7], we regarded an object as being correctly localized, if more than 50% of the detected patches had their centers within the true object bounding box. Table 2 shows the average localization rate. Our method performed better than *MA* (*MA* had same parameters $\mathbf{W}$ as ours).

7. Discussion and conclusions

In this paper, we extend the graph-mining theory by defining a hierarchical AoG model in a general form, which represents the common subgraphs that are exclusively embedded in positive ARGs. We develop an iterative framework to mine such AoG models without node enumeration. Our method discovers new OR nodes and terminal nodes, deletes redundant nodes, estimates attributes, and trains matching parameters. The generality and broad applicability of our method are demonstrated in a series of experiments.

From the view of model learning, our graph-mining theory has the following three advantages: 1) The AoG represents a hierarchical deformable template that has strong expressive power in modeling objects. 2) The AoG can be mined without labeling object positions. 3) We do not use sliding windows to enumerate object candidates for model mining.

In this paper, we seek to explore a general theoretical solution to this new type of graph mining, without a sophisticated design for specific visual tasks. However, task-specific techniques can be further added to this mining platform to improve its performance. For example, we can design a root template for the AoG and combine it with a non-linear SVM. Just like in [27], we can extract CNN features for local parts as unary ARG attributes. Tracking information can be used to guide the mining from videos.

8. Acknowledgement

This work was supported by 1) DARPA Award N66001-15-C-4035, 2) MURI grant N000141010933, and 3) NSF IIS 1423305.

References

[1] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. *In* ICCV, 2011. 2

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

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

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

[5] M. Cho, J. Sun, O. Duchenne, and J. Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. *In* CVPR, 2014. 2

[6] T. Collins, P. Mesejo, and A. Bartoli. An analysis of errors in graph-based keypoint matching and proposed solutions. *In* ECCV, 2014. 2

[7] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. *In* ECCV, 2010. 2, 8

[8] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. 6

[9] L. Fei-Fei, R. Fergus, S. Member, and P. Perona. One-shot learning of object categories. *In* PAMI, 2006. 7

[10] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. *In* ICCV, 2011. 7

[11] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *In* IJCV, 2009. 6

[12] N. Hu, R. Rustamov, and L. Guibas. Stable and informative spectral signatures for graph matching. *In* CVPR, 2014. 2

[13] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. *In* CVPR, 2008. 7

[14] G. Kim and E. P. Xing. On multiple foreground cosegmentation. *In* CVPR, 2012. 2, 7

[15] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *In* PAMI, 28(10):1568–1583, 2006. 4, 5, 6, 7

[16] Y. J. Lee and K. Grauman. Shape discovery from unlabeled image collections. In *CVPR*, 2009. 3, 8

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

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

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

[20] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *In* IJCV, 2012. 2, 7

[21] S. Manen, M. Guillaumin, and L. V. Gool Prime object proposals with randomized prim's algorithm *In*ICCV, 2013 6

[22] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. *In* ICCV, 2011. 2, 8

[23] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts. Localized content based image retrieval. *In* PAMI, 2008. 6

[24] A. Shaji, A. Varol, L. Torresani, and P. Fua. Simultaneous point matching and 3d deformable surface reconstruction. *In* CVPR, 2010. 2

[25] Z. Si and S.-C. Zhu. Learning and-or templates for object recognition and detection. *In* PAMI, 2013. 3, 4

[26] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. *In* ECCV, 2012. 6

[27] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. *In* ICML, 2014. 2, 8

[28] Y. Suh, M. Cho, and K. M. Lee. Graph matching via sequential monte carlo. *In* ECCV, 2012. 2

[29] L. Thomas. Margin: maximal frequent subgraph mining. *In* TKDD, 2010. 2

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

[31] K. Tu, M. Pavlovskaia, and S.-C. Zhu. Unsupervised structure learning of stochastic and-or grammars. *In* NIPS, 2013. 3, 4

[32] H. Wang, G. Zhao, and J. Yuan. Visual pattern discovery in image and video data: a brief survey. *In* Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery, 2014. 2

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

[34] Y. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *In*IJCV, 2010. 3

[35] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. M. Chu. Graduated consistency-regularized optimization for multi-graph matching. *In* ECCV, 2014. 2

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

[37] D. Zhang, F. Wang, L. Si, and T. Li. M^3ic: Maximum margin multiple instance clustering. *In* IJCAI, 2009. 7

[38] M.-L. Zhang and Z.-H. Zhou. Multi-instance clustering with applications to multi-instance prediction. *In* Applied Intelligence, 2009. 7

[39] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Object discovery: Soft attributed graph mining. *In*PAMI *in press DOI: 10.1109/TPAMI.2015.2456892*. 2, 6

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

[41] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Attributed graph mining and matching: An attempt to define and extract soft attributed patterns. *In* CVPR, 2014. 2, 6, 7

[42] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. From rgb-d images to rgb images: Single labeling for mining visual models. *In* ACM TIST, 2015. 6, 7

[43] Q. Zhang, X. Song, and R. Shibasaki. Visual graph mining. *In* Technical report. 5, 6

[44] J.-Y. Zhu, J. Wu, Y. Wei, E. Chang, and Z. Tu. Unsupervised object class discovery via saliency-guided multiple class learning. In *CVPR*, 2012. 7, 8