Unsupervised Category Modeling, Recognition and Segmentation in Images

Sinisa Todorovic and Narendra Ahuja
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, U.S.A.
{sintod, ahuja}@vision.ai.uiuc.edu

Abstract

Suppose a set of arbitrary (unlabeled) images contains frequent occurrences of 2D objects from an unknown category. This paper is aimed at simultaneously solving the following related problems: (1) unsupervised identification of photometric, geometric, and topological properties of multiscale regions comprising instances of the 2D category; (2) learning a region-based structural model of the category in terms of these properties; and (3) detection, recognition and segmentation of objects from the category in new images. To this end, each image is represented by a tree that captures a multiscale image segmentation. The trees are matched to extract the maximally matching subtrees across the set, which are taken as instances of the target category. The extracted subtrees are then fused into a tree-union that represents the canonical category model. Detection, recognition, and segmentation of objects from the learned category are achieved simultaneously by finding matches of the category model with the segmentation tree of a new image. Experimental validation on benchmark datasets demonstrates the robustness and high accuracy of the learned category models, when only a few training examples are used for learning without any human supervision.

I. Introduction

Suppose we are given a set of arbitrary images which contain frequent occurrences of 2D objects belonging to an unknown visual category, defined here as a collection of subimages that share similar geometric and photometric properties, and occur in similar spatial configurations. Whether, and where, any objects from the category occur in a specific image is not known. We are interested in extracting instances of the category from the image set, and in obtaining a compact model of the extracted 2D objects. A model derived from such training can then be used to determine whether a new test image contains objects from the learned category, and when it does, to segment all instances of the category.

We define a category model in terms of the structure of image regions (or segments) comprising the 2D category instances. Specifically, the category model we use captures the canonical properties of regions: (i) geometric properties, such as area and shape; (ii) photometric properties, such as gray-level contrast with the surround; and (iii) topological properties, such as the layout and recursive embedding of segments. Thus, two critical ideas lie at the foundation of our approach. First, we use regions as features for deriving the category model, since they are rich descriptors, usually stable to small illumination and viewpoint changes, robust to common (e.g., additive) noise, facilitate simultaneous object detection and segmentation, and they naturally capture the recursive definitions of object parts. Second, we exploit the ubiquitous structural properties of objects – specifically, the spatial layout and recursive containment of their parts. This leads to a representation of category instances consisting of a finitely deep recursion of regions. The depth is finite because the region size is upper bounded by the object size that can occur in a given size image, and lower bounded by the pixel size. The resulting finite-size hierarchy model facilitates learning of objects as a whole by learning category-specific parts that exhibit smaller intra-category variations compared to whole objects.

Our approach consists of the following major steps. (1) Segment the images to identify all homogeneous-intensity regions at all degrees of homogeneity present. (2) Match the training images to identify frequently occurring subimages that have similar geometric, photometric and topological properties. Interpret the maximally matching, recurring subimages as evidence and instances of some category. (3) From these category instances, obtain a hierarchical model of region properties defining the category. (4) Use the category model to detect, recognize, and segment all instances of the category in a new unseen image, by delineating all defining regions of each instance.

As our literature review in the next section indicates, most prior work requires human supervision, to provide a label of the object category that the training images contain. To the best of our knowledge, this paper presents the first attempt at completely unsupervised learning of an unknown visual category that frequently occurs in an arbitrary (unlabeled) image set. The need for human input to specify a category is eliminated by defining a category as a set of subimages sharing similar geometric, photometric and topological properties of their constituent regions. As we demonstrate in the sequel, this definition is adequate for addressing a wide range of real-world, rigid and articulated, object categories, including faces, cars, horses, cows, etc.

A. Relationship to Prior Work

In general, object recognition approaches consist of four major stages: (i) feature extraction, (ii) object representation, (iii) training, and (iv) recognition. This section reviews prior work and points out the differences with our approach with regard to each of these stages. Other related work will be discussed in the subsequent sections.

The first stage – feature extraction – uses image regions, interest points, curve fragments, image-filter responses, or a combination of these as image features. Since our focus is on region features obtained via low-level segmentation, we will omit here a review of the work that uses other types of features, for brevity. Region-based feature extraction has been used for object representation for a long time [1]–[9]. Regions are higher-dimensional features, and thus, in general, richer descriptors, more discriminative, and more noise-tolerant than interest points and curve fragments. Regions offer many advantages over point and edge features for the same problems. For example, region boundaries coincide with the boundaries of objects and their subparts, allowing for simultaneous object detection and segmentation. Also, regions make various constraints, frequently used in object recognition, such as those dealing with contiguity, smoothness, containment and adjacency, implicit and easier to incorporate than other types of lower-dimensional features (e.g., keypoints).

For the second stage – object representation – most approaches partition extracted features into clusters, called “parts.” They represent the objects as either planar or hierarchical graphs, whose nodes usually encode intrinsic appearance properties of these “parts,” and whose edges capture the spatial relationships among the “parts.” For example, the pictorial structures [10], [11] and constellation models [12] are planar graphs with a user-specified number of “parts,” configured in a pre-specified model structure. Hierarchical models are typically derived by hierarchical clustering of features [13]–[28]. This hierarchical clustering can be performed with respect to a statistical dependence that exists among subsets of features, or simply the spatial containment relationships between a large feature cluster (e.g., large region) and its constituent subclusters (e.g., embedded subregions). These two bases of clustering lead to ascendant-descendant connections between nodes in a hierarchical model. In some models, nodes may be shared by multiple parent nodes (e.g., [14], [21]–[23]). The model structure is typically controlled by a pre-specified hierarchy depth or branching factor, or by minimizing model complexity via the minimum description length principle. In contrast, our hierarchical model allows a priori unknown hierarchy depth, and an arbitrary number of nodes forming arbitrary spatial configurations, all of which are learned from training images.

Our goal to derive the canonical model of a visual category from a given set of 2D examples has been pursued by many researchers. Early work is characterized by restricted problem domains and heuristic algorithms that make use of the domain knowledge (e.g., example images show only one object from a given class on a uniform background without real-world problems, such as occlusion, and illumination and viewpoint changes). For example, the seminal work of Winston [29] considers addition and subtraction of features from an evolving model as successive positive and negative exemplars are presented, each designed to add precisely one relevant feature to the model. In [30], a hierarchical object shape representation is learned from exemplars, where a supervised decomposition of the curvature primal sketch of an example into subparts is followed by augmenting the hierarchical model with these subparts so that the matching subparts are consolidated into a single instance in the model. Another approach to automatic construction of object shape models recursively merges pairs of primitive curve elements that satisfy a set of user-specified generalization criteria [31]. In [32], a hierarchical category model is incrementally refined through matching the segmentation trees of a given set of images with the model, where matching is done top-down, in a greedy manner, only between regions at the same tree level, such that a bad match between two regions penalizes attempts to match their respective descendants. In [33], a tree model of an object shown in a given input image is learned by matching the input image to a sequence of templates provided by the user. There have also been efforts to generate a prototypical graph from a set of examples represented as graphs. For example, a heuristic, genetic search algorithm is proposed in [34] to learn a median graph from a given set of graphs. The related problem of graph clustering using a spectral embedding of graphs is explored in [35]. It is important to note that these graph-theoretic approaches do not accommodate many-to-many node correspondences, as required when dealing with real-world exemplars characterized by large structural variations. These problems have been recently addressed by a number of approaches. For example, in [8], an object shape model, which represents a planar region-adjacency graph, is learned by searching for plausible region groupings. Also, in [36], a hierarchical shape model is learned by many-to-many matching of graphs representing image blobs and their proximity relations. Our approach differs from prior work in that we perform many-to-many matching among example segmentation trees and fuse the matches to learn their tree-union as the canonical model of a visual category. As we will demonstrate in this paper, these attributes advance the state of the art, e.g., in terms of handling more challenging real-world images containing partial occlusion, clutter, and common variations in imaging conditions.

With respect to training, in the third stage, different approaches involve different degrees of supervision in learning the aforementioned object representations. Most early work requires that training images be diligently selected to ensure that they contain a single occurrence of the object class of interest preselected by the user, where each occurrence is manually segmented from the rest of the image. Recently, a number of semi-supervised approaches have been proposed [12], [37]–[43], where learning broader object classes, called categories, in more challenging images with clutter and occlusion is addressed, and where manual segmentation of object examples is not required. However, these approaches still involve a significant amount of human labor to label training images with respect to a pre-specified category they contain. Also, a careful preparation of images containing a "background" category is required. This is because "background" is treated as an additional object category, although it is not defined in any intrinsic way, but as the absence of all prespecified object categories. Thus, selection of "background" training images becomes a difficult problem, which is solved by the user choosing a training dataset that is sufficiently distinctive from the images of target object categories. This degree of supervision is sometimes reduced, so that each training image may remain unlabeled, by using alternate constraints, e.g., specifying the total number of user-defined categories present in the training set and the number of their occurrences in each training image as input parameters [44], [45]. In contrast, we attempt learning an unknown visual category in a completely unsupervised manner. The absence of supervision here means that it is not known whether and where any objects from the category appear in a specific image from the set. Thus, some training images may not contain any example of the frequently occurring (target) category, while others may contain multiple instances of multiple categories. Also, unlike some approaches, aimed at learning a discriminant object classification function (e.g., [38]), we do not require the training set to be large. In addition, we do not need to model the background as a category by itself, and, hence, do not require a careful preparation of the background training dataset.

Finally, object recognition, in stage four, is typically evaluated only through image classification in terms of whether the learned object class/category is present or absent [12], [27], [38],

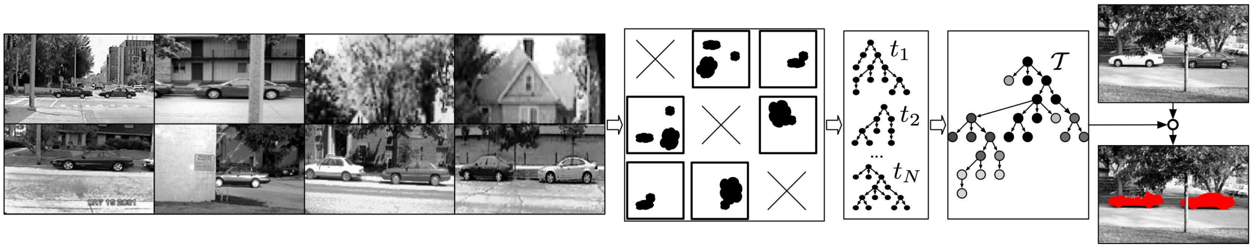
[42]–[44]. There are also approaches that attempt object localization by placing a bounding box around a detected object, or by thresholding a probabilistic map that a pixel belongs to the object given the detected features [37], [40], [41]. These estimates are imprecise (bounding box) or non-deterministic (probability map), to begin with, and are further worsened by the fact that both locations of detected features and thresholds for object localization are image dependent. To overcome these issues, some methods hypothesize the total number of target objects present in the image [37]. Few approaches [45], like ours, delineate the boundaries of all instances of the learned categories appearing in the image, i.e., simultaneously conduct object detection, recognition and segmentation.

B. Overview of Our Approach

In this section, we present an overview of the main steps of our approach and point out their motivation and contributions. (1) We begin with the detection of image regions which are the basic features of our models. An image is represented by a segmentation tree [46]–[48] which captures the low-level, spatial and photometric, image structure in a hierarchical manner. Nodes at upper levels correspond to larger, more salient segments, while their children nodes capture embedded, less salient details (e.g., segments with smaller gray-level contrasts with the surround). Each node is associated with the geometric and photometric properties of the corresponding segment, while the tree structure captures the mutual containment (topological) properties of segments. Therefore, the segmentation tree serves as a rich description of the image. (2) Given an image set that contains frequent occurrences of an unknown category, we expect that subimages with category specific values of the above properties will be abundant in the set. Each such subimage will correspond to one or more subtrees in the segmentation tree, thus leading to frequent occurrences of subtrees with similar properties. The category subtrees can be detected by a tree matching algorithm that searches for the common subtrees of the given image trees having a large similarity measure. This similarity measure is defined in terms of the tree structure, as well as the geometric and photometric properties associated with tree nodes. The result is a set of subtrees from each image that have cross-image similarity measures above a chosen level. The tree matching algorithm identifies exactly which region properties are shared by the matching subtrees. These subtrees are interpreted as instances of the target category whose intercategory variability depends on the chosen level of the similarity measure. (3) The extracted subtrees may represent complete object occurrences or their parts. Extraction of only object parts occurs when they remain unaltered, while the region properties of other parts, and hence of entire objects, are changed due to, e.g., partial occlusions, or illumination, viewpoint, or scale variations across the images. Therefore, the extracted similar subtrees provide for many observations of entire objects or their parts in the category, thus allowing robust estimation of the entire, characteristic region structure of the category. All of these subtrees can be fused (i.e., partially matched and registered) within a canonical graph, which we call the tree-union. Hence, the tree-union subsumes all extracted category instances, and thus represents the learned category model. The tree-union specifies: how segmented regions are recursively laid out to comprise an object from the category, and what their geometric and photometric properties are. (4) When a new image is encountered, any matches between its segmentation tree and the category model will denote the presence of the category, and simultaneously specify the exact boundaries of the recognized objects and their constituent image regions. The block-diagram of our approach is given in Fig. 1.

As a result of these basic steps, the performance of our approach has desirable invariance characteristics with respect to: (i) Translation, in-plane Rotation and Object-Articulation (changes in relative orientations of object parts): because the segmentation tree itself is invariant to these changes; (ii) Scale: because subtree matching is based on relative properties of nodes, not absolute values; (iii) Occlusion in the training set: because subtrees are registered and stitched together within the tree-union encoding the entire (unoccluded) category structure; (iv) Occlusion in the test set: because subtrees corresponding to visible object parts can still be matched with the model; (v) Small Appearance Changes (e.g. due to noise): because changed regions may still be the best matches; (vi) Region Shape Deformations (e.g., due to minor depth rotations of objects): because changes in geometric/topological properties of regions (e.g., splits/mergers) are accounted for during matching; and (vii) Clutter: because clutter regions, being non-category subimages, are not repetitive and therefore frequent.

The preliminary version of our approach is presented in [48]. This paper contributes the following major extensions to [48]: (i) additional region properties are used; (ii) similarity between two trees is estimated using a new measure; (iii) while in [48] all region properties are equally weighted for recognition, we here present an algorithm for finding the optimal weights of region properties; and (iv) a more extensive experimental evaluation of the proposed approach



(a) (b) (c) (d) (e)

Fig. 1. Block-diagram of our approach: (a) A set of input images contains frequent occurrences of a car category. A specific image in the set may not contain cars, or may show more than one car. Also, cars may appear at different scales, and may be partially occluded. (b) Pairwise image matching; black regions indicate maximally matching subimages. (c) Extracted subtrees representing maximally matching subimages shown in (b). (d) Tree-union represents a model of the car category learned from the extracted similar subtrees shown in (c). The relative significance to recognition of model nodes is marked with different shades of gray. (e) Simultaneous object detection, recognition, and segmentation in a new image.

addressing both rigid and non-rigid object categories is presented.

This paper is organized as follows. The segmentation tree, and region properties selected for modeling a category are defined in Sec. II. Sec. III discusses the tree matching algorithm. Learning the category model is presented in Sec. IV. Optimal weighting of region properties used to learn the model is discussed in Sec. V. Experimental validation is presented in Sec. VI.

II. Segmentation Trees and Region Properties for Category Modeling

An input image is represented by a segmentation tree, obtained using a multiscale segmentation algorithm, presented in [46], [47], [49]. The segmentation algorithm partitions an image into homogeneous regions of a priori unknown shape, size, gray-level contrast, and topological context. Here, a region is considered to be homogeneous if variations in intensity within the region are smaller than intensity change across its boundary, regardless of its absolute degree of variability. Consequently, image segmentation is performed at a range of homogeneity values, i.e., intensity contrasts. As the intensity-contrast sensitivity parameter increases, regions with smaller contrasts than the current parameter value strictly merge. A sweep of the parameter values thus results in the extraction of all the segments present in the image. The segmentation tree is derived by organizing the segmented regions into a tree structure, where the root represents the whole image, nodes closer to the root represent large regions, while their children nodes capture smaller embedded details, as depicted in Fig. 2. The number of nodes (typically 50–

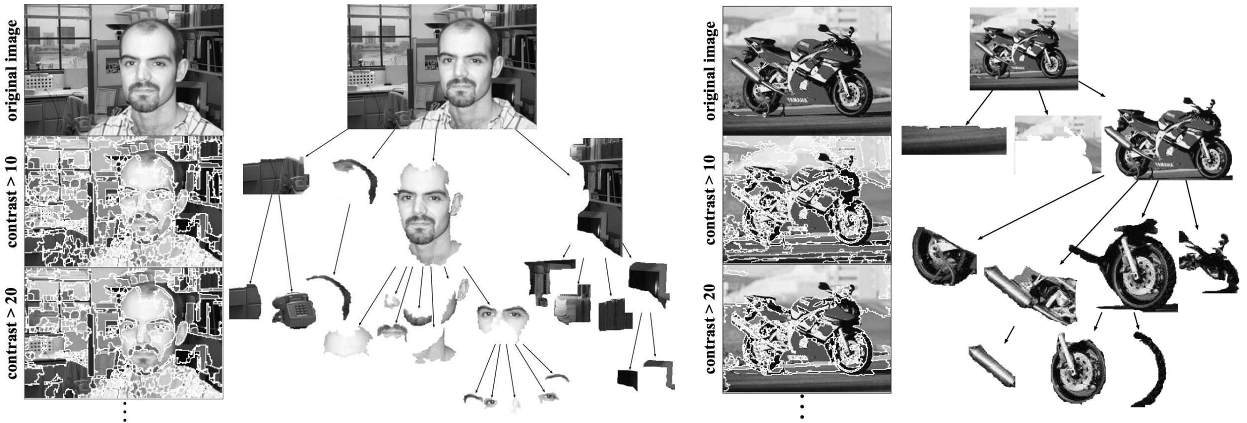


Fig. 2. Segmentation trees of sample Caltech-101 images [42]: (left) segmentations obtained for two sample intensity contrast values from the exhaustive range [1,255]; (right) sample nodes of the corresponding segmentation tree, where the root represents the whole image, nodes closer to the root represent large regions, while their children nodes capture smaller embedded details. The number of nodes (typically 50–100), branching factor (typically 0–10), and the number of levels (typically 7–10) in different parts of the segmentation tree are image dependent, and automatically determined.

100), branching factor (typically 0–10), and the number of levels (typically 7–10) in different parts of the tree are image dependent.

Each node v is characterized by a vector of properties of the corresponding region, denoted as $\boldsymbol{\psi}_v$. We use intrinsic photometric and geometric properties of the region, as well as relative inter-region properties describing the spatial layout of the region and its neighbors. In this way, $\boldsymbol{\psi}_v$ encodes the spatial layout of regions, while the tree structure itself captures their recursive containment. The properties are defined to allow scale and rotation-in-plane recognition invariance. In particular, elements of $\boldsymbol{\psi}_v$ are defined relative to the corresponding properties of v's parent-node u, and thus ultimately relative to the entire image.

Let w, v, and u denote regions forming a child-parent-grandparent triple. Then, the properties of each region v we use are as follows: (1) normalized gray-level contrast g_v, defined as a function of the mean region intensity G, $g_v \triangleq \frac{|G_u - G_v|}{|G_v - G_w|}$; (2) normalized area $a_v \triangleq A_v/A_u$, where A_v and A_u are the areas of v and u; (3) area dispersion AD_v of v over its children $w \in C(v)$, $\text{AD}_v \triangleq \frac{1}{|C(v)|}\sum_{w\in C(v)}(a_w - \overline{a_{C(v)}})^2$, where $\overline{a_{C(v)}}$ is the mean of the normalized areas of v's children; (4) the first central moment μ_v^{11}; (5) squared perimeter over area $\text{PA}_v \triangleq \frac{\text{perimeter}(v)^2}{A_v}$; (6) angle γ_v between the principal axes of v and u; the principal axis of a region is estimated as the eigenvector of matrix $\frac{1}{\mu^{00}}\begin{bmatrix}\mu^{20} & \mu^{11}\\ \mu^{11} & \mu^{02}\end{bmatrix}$ associated with the larger eigenvalue, where the μ's are the standard central moments; (7) normalized displacement $\overrightarrow{\Delta}_v \triangleq \frac{1}{\sqrt{A_u}}\overrightarrow{d}_v$, where $|\overrightarrow{d}_v|$ is the

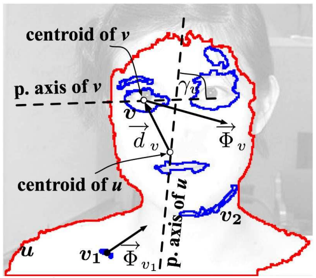


Fig. 3.

Fig. 3. Properties of a region associated with the corresponding node in the segmentation tree: Region u (marked red) contains a number of embedded regions $v, v_1, v_2, \ldots$ (marked blue). The principal axes of u and v subtend angle γ_v, the displacement vector $\boldsymbol{d}_v$ connects the centroids of u and v, while the context vector $\boldsymbol{\Phi}_v$ records the general direction in which the siblings $v_1, v_2, \ldots$ of v are spatially distributed.

distance between the centroids of u and v, and $\measuredangle \overrightarrow{d}_v$ is measured relative to the principle axis of parent node u, as illustrated in Fig. 3; $\sqrt{A_u}$ represents an estimate of the diameter of parent region u; and (8) context vector $\overrightarrow{\Phi}_v \triangleq \sum_{s \in S(v)} \frac{A_s}{|\overrightarrow{d}_{vs}|^3} \overrightarrow{d}_{vs}$, where $S(v)$ is the set of v's sibling regions s, and $|\overrightarrow{d}_{vw}|$ is the distance between the centroids of v and s, and $\measuredangle \overrightarrow{d}_{vs}$ is measured relative to the principle axis of their parent node u; as illustrated in Fig. 3, the context vector records the general direction v sees its sibling regions and disallows matching of scrambled layouts of regions at a specific tree level. In summary, the vector of region properties associated with node v is $\boldsymbol{\psi}_v = [g_v, a_v, \mathrm{AD}_v, \mu_v^{11}, \mathrm{PA}_v, \gamma_v, \overrightarrow{\Delta}_v, \overrightarrow{\Phi}_v]^{\mathrm{T}}$. Each element of $\boldsymbol{\psi}_v$ is normalized over all multiscale regions of all training images to take a value in the interval $[0, 1]$. This list of useful region properties, can be easily modified to reflect the needs of different applications.

III. Extracting Category Instances

To extract recurring similar subimages from the given image set $\mathbb{T} = \{t_1, t_2, \ldots t_M\}$, all pairs of segmentation trees $(t, t') \in \mathbb{T} \times \mathbb{T}$ are matched to identify those pairs that have a similarity measure above a chosen threshold (see Fig. 1). Prior work mostly uses only the intrinsic geometry and appearance of regions for their matching. We extend the matching criteria to include the information about the mutual containment of regions, which is expected to improve the robustness of cross-image region matching. Thus, given two segmentation trees, our matching algorithm pairs those nodes whose associated region properties match, and recursively the same holds for their descendant nodes. As another means of making extraction of category instances more robust, our matching algorithm explicitly accounts for the fact that certain image regions are less likely to be preserved across the images than others. For example, low contrast regions may split or merge with bordering regions due to slight changes in the directions of lighting, viewing and object orientation. This in turn changes the segmentation tree structure, and thus requires matching to explicitly account for these uncertainties. To accomplish these objectives, we resort to the well-known framework of edit-distance graph matching [50]–[55].

While there are many diverse techniques for matching image graph representations used in computer vision, we briefly review only the two most common approaches to focus our presentation. The structural properties of graphs can be captured by the eigenvectors of the associated adjacency matrix [26], [35], [56], [57]. However, the spectral approaches to graph matching encounter the major difficulty that structurally different graphs may have the same spectrum. Another group of approaches involves transforming the two graphs by applying basic edit operations on nodes and edges – such as insertion, deletion, merging, splitting and relabeling – until the transformed graphs become isomorphic. The goal of these methods is to minimize the cost of modifications needed in the two graphs to match them, referred to as edit-distance. One great advantage of edit-distance matching over the spectral approaches is that edits can be naturally interpreted in the image domain, allowing one to appropriately define edit costs, while in general this is not the case for algebraic manipulations of spectral graph representations. However, traditionally, the edit-distance methods are based on the assumption that there exist only one-to-one node correspondences in matching [50]–[54], which is usually too restrictive for our case, as stated above. This problem can be addressed by considering many-to-many matching. For example, in [58], a subset of graph nodes are merged into a single node (merger) when the difference between their attributes is smaller than a chosen threshold, after which this combined node is matched to a node or merger in the other graph, thereby conducting many-to-many matching. However, since the magnitude of node attribute disparities is a priori unknown, this method is very sensitive to threshold selection. In [55], many-to-many matching is considered within the edit-distance framework. This approach, however, has a large bias toward favoring one-to-one node correspondences over one-to-many, since the heuristically defined cost of matching a single node with many is higher than the cost of matching two single nodes. Spectral-based approaches also present promising solutions to many-to-many matching [59], [60]; however, it is not clear how to use these methods to explicitly account for splits and mergers between *bordering* regions in our segmentation trees.

In this paper, we use our edit-distance matching algorithm presented in [48], [61]. For completeness, below, we briefly review its main characteristics, and point out the major improvements made here. Our algorithm extends Torsello and Hancock's approach [54] by searching for correspondences between individual regions, as well as between groups of contiguous regions in two given segmentation trees. This amounts to considering one-to-one, one-to-many, and many-to-many region correspondences, all at the same time, unlike in [54] where only one-to-one matching is allowed. Specifically, the segmentation trees are first modified by inserting and appropriately connecting new nodes (i.e., regions), representing mergers, as illustrated in Fig. 4. Each merger is the union of a few neighboring sibling nodes under a parent node. It instantiates the hypothesis that the children are formed due to an incorrect split, caused by, e.g., lighting changes etc., and therefore they should be restored together as a single node. To cover all possibilities under a given node, mergers are made corresponding to all members of the power set of the node's children sharing the same boundary. Mergers do not eliminate their source nodes in the tree. Instead, each merger is inserted as a parent of the merged nodes, which converts the trees into directed acyclic graphs (DAGs), as depicted in Fig. 4. Second, for each DAG thus obtained, we construct its transitive closure by adding new edges between all ancestor-descendant node pairs in the DAG (Fig. 4). The reason for constructing transitive closures is that their matching is more flexible than matching DAGs and trees, allowing matches of all descendants, instead of only children, under a visited node. Thus, we will formulate segmentation-tree matching as a search for the maximum subtree isomorphism between the transitive closures of segmentation DAGs. This search will be constrained, because the resulting maximum similarity common subtree must respect ascendant-descendant relationships of the initial trees. These consistency constraints will disallow many node-pairs from being candidates for matching, and thus improve the overall matching efficiency. Below, we present our matching algorithm.

A. *Formulation of the Matching Algorithm*

Given two transitive closures of the segmentation DAGs, obtained from the segmentation trees as explained above, our edit-distance matching algorithm identifies two legal, minimum-cost sequences of basic edit-operations applied to the two DAGs,[1] respectively, which produce their common subtrees, and preserve the original node adjacency and ascendant-descendant relationships. The edit-operations considered here consist of only node removals and matches. A candidate node v when paired with another node v', is either considered matchable, with an

[1] Note that the transitive closure of a DAG is also a DAG.

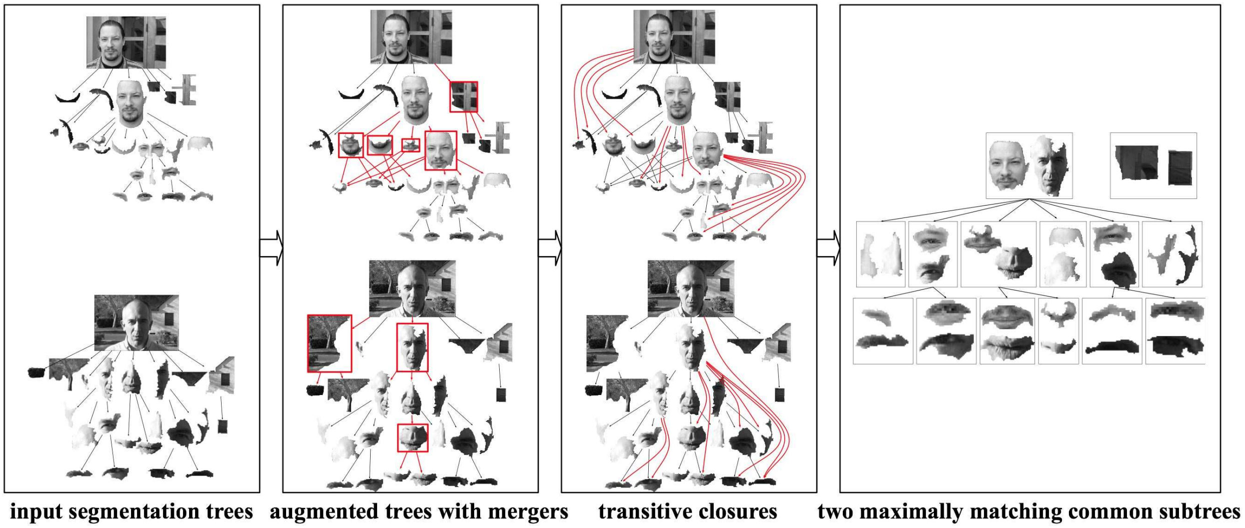

input segmentation trees
augmented trees with mergers
transitive closures
two maximally matching common subtrees

Fig. 4. Matching segmentation trees: Input tree is first converted into a DAG by inserting mergers (only a few sample mergers are marked red for clarity), which represent the union of a few neighboring sibling nodes under a parent node. Mergers correspond to all members of the power set of children sharing the same boundary under each node. Then, the transitive closure of the DAG is constructed by adding new edges between all ancestor-descendant node pairs in the DAG (only a few sample edges are marked red for clarity). Matching segmentation trees amounts to a search for the maximum subtree isomorphism between the two transitive closures of the DAGs.

edit-cost $m_{vv'}$, or considered unmatchable and "removed," with a cost proportional to its salience r_v. The total cost associated with the sequence of edit operations represents the edit-distance, i.e., a measure of similarity between the two DAGs. It can be shown that finding the maximum similarity edit-sequence between two DAGs, consisting of only node removals and matches, is equivalent to finding the maximum similarity subtree isomorphism [53], [54]. Therefore, the goal of our matching algorithm can also be interpreted as finding maximum subtree isomorphism. To specify the matching algorithm, we use the following definitions.

Definition 1. (Topological consistency) Let t and t' be two transitive closures of the segmentation DAGs. Node pair (v, v'), where $v{\in}t$ and $v'{\in}t'$, is said to be topologically consistent with (u, u'), where $u{\in}t$ and $u'{\in}t'$, if the topological relation between v and u (i.e., presence/absence of ascendant-descendant relationship) is the same as the topological relation between v' and u'. Topologically consistent node pairs are denoted as $(v, v'){\sim}(u, u')$,

Definition 2. (Consistent bijection) Let $f : U{\rightarrow}U'$ be a bijection between two subsets of nodes U and U' in two DAGs. f is consistent if $\forall(v, u){\in}U$, $(v, u){\sim}(f(v), f(u))$.

Definition 3. (Matching algorithm) Given two transitive closures of the segmentation DAGs

$t{=}(V,E,\Psi)$ and $t'{=}(V',E',\Psi')$, where V and E are the sets of nodes and edges, and Ψ is a function that assigns a vector of region properties $\boldsymbol{\psi}_v$ to each node $v\in V$, the matching algorithm finds a consistent bijection (i.e., subtree isomorphism) $f{:}U{\rightarrow}U'$, where $U{\subseteq}V$ and $U'{\subseteq}V'$, which maximizes their similarity measure $\mathcal{S}_{tt'}$ defined as

$$\mathcal{S}_{tt'} \triangleq \max_{f\subset V_t\times V_{t'}} \sum_{(v,v')\in f} \left[\min(r_v,r_{v'}) - m_{vv'}\right] \ . \tag{1}$$

From (1), the algorithm seeks consistent matches $(v,u){\sim}(f(v),f(u))$ among nodes $v\in V$ and $v'\in V'$ whose saliencies r_v and $r_{v'}$ are high, but cost $m_{vv'}$ is low. Therefore, by selecting highly salient nodes in the matching result, the algorithm minimizes the total penalty for removing the other nodes from the two graphs while finding their common subgraph. The literature reports different strategies for defining the edit-costs r_v and $m_{vv'}$, ranging from heuristic to information-theoretic definitions [51], [52], [54]. In this paper, the node saliency r_v and the cost of node matching $m_{vv'}$ are defined in terms of region properties $\boldsymbol{\psi}$ as

$$r_v \triangleq \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\psi}_v, \quad \text{and} \quad m_{vv'} \triangleq |r_v - r_{v'}| = \max(r_v,r_{v'}) - \min(r_v,r_{v'}) \ , \tag{2}$$

where $\boldsymbol{\xi}$ is a vector of coefficients weighting the relative significance to recognition of the corresponding region properties in $\boldsymbol{\psi}_v$, and whose $L2$-norm is $\|\boldsymbol{\xi}\|{=}1$, and $\boldsymbol{\xi}{\geq}0$. In [48], [61], region properties are equally weighted, i.e., $\boldsymbol{\xi} = 1/|\boldsymbol{\psi}_v|$, where $|\boldsymbol{\psi}_v|$ is the number of region properties used. In this paper, we examine their relative contributions to recognition, thus obtaining an optimal weighting of region properties, as discussed in Sec. V. Note that $r_v, m_{vv'}\in[0,1]$. From (1) and (2) we have

$$\mathcal{S}_{tt'} = \max_{f\subset V_t\times V_{t'}} \textstyle\sum_{(v,v')\in f} \left[2\min(r_v,r_{v'}) - \max(r_v,r_{v'}) + 1\right] \ , \tag{3}$$

where 1 is added to make the expression in the brackets non-negative, which does not change the solution f. Thus, we formulate matching as an optimization problem given by (3). The result of matching t and t' is the set of nodes paired by a consistent f and comprising the two maximum similarity common subtrees of t and t', respectively.

In our preliminary work [48], we used a different definition of similarity measure $\mathcal{S}^{\text{old}}_{tt'} = \max_{f\subset V_t\times V_{t'}} \sum_{(v,v')\in f} [r_v + r_{v'} - |r_v - r_{v'}|] = \max_{f\subset V_t\times V_{t'}} \sum_{(v,v')\in f} 2\min(r_v,r_{v'})$. Hence, to maximize $\mathcal{S}^{\text{old}}_{tt'}$, the matching algorithm pairs only those nodes whose saliencies are large, but does not explicitly verify the discrepancy between their associated region properties. As demonstrated in Sec. VI, the new definition of similarity measure given by (3) improves performance.

The optimization problem of (3) can be solved recursively, bottom-up, starting from leaf nodes. Suppose that at any stage during matching, for all descendants v of u in t, and for all descendants v' of u' in t' we have previously computed $\mathcal{S}_{vv'}$. Then, our goal is to find the optimal set of consistent descendant pairs $(v,v')\in\mathcal{C}_{uu'}$, while maximizing $\mathcal{S}_{uu'}$. From (3), we have

$$\mathcal{S}_{uu'} = 2\min(r_u, r_{u'}) - \max(r_u, r_{u'}) + 1 + \textstyle\sum_{(v,v')\in\mathcal{C}_{uu'}} \mathcal{S}_{vv'}. \tag{4}$$

As shown in [52], [54], the optimal $\mathcal{C}_{uu'}$ can be found as the maximum weight clique of the association graph $\mathscr{A}_{uu'}{=}(V_{\mathscr{A}}, E_{\mathscr{A}}, \mathcal{S})$ characterizing the directed acyclic subgraphs rooted at u and u'. In particular, $V_{\mathscr{A}}$ is the set of all possible matches $\{(v,v')\}$, where the v and v' are all descendants underneath u in t and u' in t', respectively. $\mathcal{S}$ is a function that assigns a weight equal to the similarity measure $\mathcal{S}_{vv'}$ to every node (v,v'). $E_{\mathscr{A}}$ is the set of undirected edges that connect only consistent nodes $(v,v')\in V_{\mathscr{A}}$. Thus, imposing the structural constraints in finding a consistent subtree isomorphism is done in a simple manner during the construction of the association graph $\mathscr{A}_{uu'}$. To solve the maximum weight clique problem, we use the well-known game (replicator) dynamics approach thoroughly discussed in [62]. This algorithm uses the Motzkin-Straus theorem to transform the maximum clique problem, known to be NP-hard, into a continuous quadratic programming problem with complexity $O(|V_{\mathscr{A}}|^2)$ in the number of nodes in $\mathscr{A}_{uu'}$.

From (4), $\mathcal{S}_{uu'}$ is directly proportional to both the quality of match between the region properties associated with the node pair (u,u') and the size of matched subtree structure underneath them. Once computed, $\mathcal{S}_{uu'}$ is used to recursively find the similarity measure of subgraphs rooted at the ancestors of u and u'. In this vein, $\mathcal{S}_{vv'}$ values of all node pairs $(v,v')\in t{\times}t'$ are obtained.

B. Unsupervised Selection of Maximally Matching Subtrees

The matching algorithm presented in the previous section is used to extract similar subtrees from the given set of segmentation trees $\mathbb{T}{=}\{t_1, t_2, \dots t_M\}$. Thanks to relatively small training sets considered in this paper, a total of $M(M{-}1)$ tree pairs are matched to identify their common subtrees, whose similarity measures $\mathcal{S}$ are above a chosen threshold. The appropriate selection of this threshold in unsupervised settings is a challenging research topic beyond the scope of this paper. A straight forward strategy that we use here is based on the frequency histogram of all $\mathcal{S}_{vv'}$ values observed over all node pairs (v,v') across all $M(M{-}1)$ image-tree pairs, denoted as

$\mathcal{H}(\mathcal{S})$. Note that $\mathcal{S}$ accounts for all the properties we have chosen to define a category – namely, photometric, geometric and topological properties of regions. Small variations in $\mathcal{S}_{vv'}$ values across subtrees representing category instances are to be expected, as they reflect intra-category inter-instance variations. It therefore follows that the frequency histogram $\mathcal{H}(\mathcal{S})$ will be in general characterized by a number of modes, each corresponding to frequent occurrences of instances from a different category present in training images. Since our objective is to identify the most frequently occurring similar subimages that correspond to a single most frequent category in the training set, we extract all those similar subtrees whose $\mathcal{S}_{vv'}$ values are high (i.e., belong to a category) and fall in the largest mode in the histogram (i.e., most frequently occur).

For detection of the histogram modes, we use the well-known relaxation labeling algorithm of [63], which uses the contextual information of neighboring histogram bins to reduce local ambiguities in the histogram values, and yields reliable results after only a few iterations. After detecting histogram modes, we identify the interval of similarity measure values that contains the mode with the aforementioned category properties. Formally, we compute $[\mathcal{S}_{\text{min}}, \mathcal{S}_{\text{max}}] = \arg\max_{\text{modes}} \sum_{\mathcal{S}\in\text{modes}} \mathcal{S}\cdot\mathcal{H}(\mathcal{S})$. All subtrees in $\mathbb{T}$ with $\mathcal{S}_{vv'}$ values in the interval $[\mathcal{S}_{\text{min}}, \mathcal{S}_{\text{max}}]$ are identified as category instances.

C. Computational Complexity

Two major steps contribute to the computational complexity of discovering category instances: augmentation of given trees with merger nodes, and actual matching of the resulting DAGs. Let s denote the average number of sibling regions that share a portion of their boundary under a node. Note that s is considerably smaller than the average number of a node's children (typically $0 \leq s \leq 3$), and thus the size of the power set of contiguous siblings is typically not very large. Then, given a segmentation tree with $|V|$ nodes, the complexity of transforming the tree into a DAG by inserting merger nodes is $O(2^s|V|)$.

In the next step, we solve $2^s|V| \times 2^s|V|$ maximum weight clique problems, as explained in Sec. III-A. The replicator dynamics algorithm used for this purpose converges for such problems after only a few iterations. Each iteration involves $O(|\mathscr{A}|^2)$ multiplications, where $|\mathscr{A}|$ is the total number of nodes in the association graph whose maximum weight clique is computed. Thus, the complexity of tree matching is $O([2^s|V|]^4)$, which typically amounts to $O(10^{10})$ computations, performed in approximately 20-30 seconds on a 2.8GHz, 2GB RAM PC, for images used in experimental evaluation discussed in Sec. VI. In comparison with the standard edit-distance tree matching approaches (e.g., [54]), typically used for matching binary images with silhouettes of objects, ours increases computational complexity $O(16^s){\approx}O(10^3)$ times. This increase is justified by significant improvements in matching performance as a result of simultaneously accounting for many-to-many, one-to-many and one-to-one node correspondences, which in turn allows us to address more complex, real images with clutter and occlusion.

To extract category instances, we conduct pairwise matching of M image trees, after which the relaxation labeling algorithm is used for finding the optimal mode of the frequency histogram of similarity measures, as explained in Sec III-B. The complexity of relaxation labeling is $O(n\ell^2)$, where $n{=}4$ is the number of histogram bins within the sliding window used in the algorithm, and $\ell{=}2$ is the number of classes (mode, valley) we consider.

Overall, if each segmentation tree in $\mathbb{T}$ has no more than $|V|$ nodes, then the complexity of extracting similar subtrees from $\mathbb{T}$ is $O(M^2 16^s |V|^4)$.

IV. Learning the Category Model

The set of extracted similar subtrees, $\mathbb{D} = \{t_1, t_2, ..., t_N\}$, in the sequel simply referred to as trees, may represent fully or partially visible objects of the discovered category, as well as some outlier objects that do not belong to the category. We are interested in obtaining a compact, canonical model of the target category from $\mathbb{D}$. In this section, we explain how to integrate the information from all visible category parts by fusing the trees of $\mathbb{D}$ into a tree-union, and thus derive the category model.

Tree-unions are well studied graph structures, the detailed treatment of which can be found, for example, in [53], [64]–[68]. The tree-union $\mathcal{T}$ is the smallest directed acyclic graph (DAG), which contains every tree in $\mathbb{D}$. Ideally, $\mathcal{T}$ should be constructed by first finding the maximum common subtree of $\mathbb{D}$, and then by adding to the common subtree, and appropriately connecting, the remaining nodes from $\mathbb{D}$. However, finding this maximum common subtree would entail factorial complexity $O(N!)$ in the number of trees N in $\mathbb{D}$ which can be arbitrarily large.[2] Since such an algorithm is computationally infeasible for real training sets $\mathbb{D}$ which are usually very

[2] Typically, in our experiments, the number of training images M is much smaller than the number of extracted similar subtrees N, since multiple instances of the same category may co-occur in a single image.

large, we resort to a suboptimal sequential approach. In each iteration $\mathcal{T}$ is extended by adding a new tree t from $\mathbb{D}$ until every tree from $\mathbb{D}$ has been added to the tree-union, as illustrated in Fig. 5. As can be seen, the selected t is first matched against the current estimate $\mathcal{T}^{(n)}$, which results in their common subtree τ, and then the unmatched nodes from t are added and appropriately connected to τ in order to form $\mathcal{T}^{(n+1)}$.

For matching t and $\mathcal{T}^{(n)}$, we use the same algorithm presented in Sec. III-A. After adding the unmatched nodes, the result is a DAG with multiple directed paths between nodes, which preserve the node ascendant-descendant relationships from $\mathbb{D}$. As detailed in [66], [68], the matching algorithm of Sec. III-A can be used for matching trees and DAGs under the condition that a given path in the tree can match only one path in the DAG. Imposing the same three consistency constraints as used in matching, namely: (1) preserve node connectivity, (2) preserve ancestor-descendant relationships, and (3) disallow multiple paths between nodes, is done in a simple manner during the construction of the association graph, $\mathcal{A}_{uu'}$, for each visited node pair $(u, u')\in t\times\mathcal{T}$, as explained in Sec. III-A. To define the saliency r_v and the cost of node matching $m_{vv'}$ of nodes $v \in \mathcal{T}^{(n)}$, which are used for computing the similarity measure in (3), we record the region properties $\boldsymbol{\psi}_{v'}$ associated with all nodes $v'\in\mathbb{D}$ that got matched with v in the previous τ iterations. Then, a region-property vector $\boldsymbol{\psi}_v$ associated with $v \in \mathcal{T}^{(n)}$ can be defined in terms of the statistics of these recorded vectors $\{\boldsymbol{\psi}_{v'}\}$. In this paper, $\boldsymbol{\psi}_v$ is computed as the median vector of the matched regions' properties, $\boldsymbol{\psi}_v = \text{median}\{\boldsymbol{\psi}_{v'}\}$. Other statistics, e.g., the mean vector, may also be used. In our experiments, using the median yields slightly better performance over the mean. Finally, similar to the definitions in (2), we have $\forall v \in \mathcal{T}^{(n)}$, $r_v \triangleq \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\psi}_v$, and $m_{vv'} \triangleq |r_v - r_{v'}|$, where $\boldsymbol{\xi}$ specifies the relative significance to recognition of the region properties in $\boldsymbol{\psi}_v$.

Due to sequential matching of trees in $\mathbb{D}$, their different orderings may result in different tree-unions. This problem is addressed in [66] by merging first those trees with the maximum similarity measure. This strategy, however, does not account for possible outliers in $\mathbb{D}$. Outliers may be present in $\mathbb{D}$ because of unsupervised extraction of similar subtrees. Therefore, for our purposes, it is necessary to use an algorithm that finds the best approximation of the tree-union, while at the same time accounting for outliers in $\mathbb{D}$. Our basic assumptions is that $\mathbb{D}$ contains trees with similar structure and node properties, so that each node in $\mathcal{T}$ should have approximately the same frequency of matching with nodes in $\mathbb{D}$. Nodes in $\mathcal{T}$ that come from outliers are likely

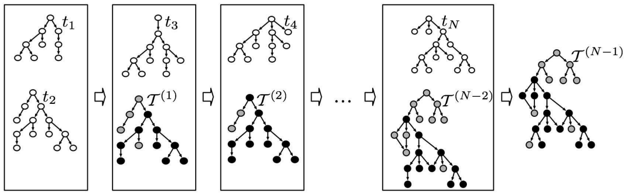


Fig. 5. Construction of tree-union $\mathcal{T}$ from the extracted set of similar trees $\mathbb{D}$=$\{t_1, t_2, \ldots, t_N\}$: In each iteration, a selected tree t from $\mathbb{D}$ is first matched against the current estimate $\mathcal{T}^{(n)}$, which yields their maximum common subtree τ (marked black). Then the unmatched nodes from t are added and appropriately connected (marked gray), to form $\mathcal{T}^{(n+1)}$. The result is a directed acyclic graph (DAG).

```
Algorithm 1: Learning Tree-Union $\mathcal{T}$
Input : $\mathbb{D} = \{t_1, t_2, \ldots, t_N\}$
for $i = 1 : N-1$ do
    Find random permutation of the input set $\mathcal{P}_i(\mathbb{D})$;
    Find maximum similarity, consistent, common subtree $\tau$
    of $t_1$ and $t_2$ in $\mathcal{P}_i(\mathbb{D})$ (Section III-A) ;
    Add and appropriately connect the unmatched nodes from
    $t_1$ and $t_2$ to $\tau$, to form $\mathcal{T}^{(2)}$;
    for $n = 3, N$ do
        Find $\tau$ of $\mathcal{T}^{(n-1)}$ and $t_n$ ;
        Add and appropriately connect the unmatched nodes
        from $t_n$ and $\mathcal{T}^{(n-1)}$ to $\tau$, to form $\mathcal{T}^{(n)}$;
    end
    $\mathcal{T}(\mathcal{P}_i) = \mathcal{T}^{(n)}$;
    Compute node frequencies $\forall v \in \mathcal{T}(\mathcal{P}_i)$, $\varphi_v = \frac{\text{\# of matches}}{\text{\# of nodes in } \mathbb{D}}$;
    Find entropy $H(\mathcal{P}_i) = -\sum_{v \in \mathcal{T}(\mathcal{P}_i)} \varphi_v \log \varphi_v$;
end
Output: $\mathcal{T} = \mathcal{T}(\hat{\mathcal{P}})$, where $\hat{\mathcal{P}} = \arg\min_{\mathcal{P}_i} H(\mathcal{P}_i)$
```

to have a relatively lower frequency of matching with nodes in $\mathbb{D}$. These frequencies can be conveniently described by their entropy. Since majority of trees in $\mathbb{D}$ are likely to represent category instances, node frequencies of $\mathcal{T}$ will be characterized by a small entropy. Therefore, to learn the category model, we obtain a set of tree-unions $\{\mathcal{T}(\mathcal{P}_1), \ldots, \mathcal{T}(\mathcal{P}_R)\}$ for various permutations $\mathcal{P}_i$ of $\mathbb{D}$. Then, we compute for each node $v \in \mathcal{T}(\mathcal{P}_i)$ the frequency of its matches with nodes in $\mathbb{D}$, $\varphi_v = \frac{\text{\# of matches}}{\text{\# of nodes in } \mathbb{D}}$. The best approximation of the tree-union is selected based on entropy $H(\mathcal{P}_i) = -\sum_{v \in \mathcal{T}(\mathcal{P}_i)} \varphi_v \log \varphi_v$, which achieves a minimum for the sets containing all isomorphic trees. Thus, the permutation $\hat{\mathcal{P}}$ for which H is minimum over all $\mathcal{P}_1, \ldots, \mathcal{P}_R$ is selected to compute $\mathcal{T}(\mathcal{P}_i)$ as the best approximation of the tree-union. In the case of multiple solutions, $\mathcal{T}(\mathcal{P}_i)$ with the smallest number of nodes is selected.

Algorithm 1 summarizes our learning of the tree-union. The choice of the number of permutations R is subject to the trade off between accuracy and computational complexity. Suppose no tree in $\mathbb{D}$ has more than $|V|$ (typically about 20) nodes. Note that $|V|$ is much smaller than the typical number of nodes in segmentation trees. Then, similar to the complexity of matching explained in Sec. III-C, the complexity of learning the tree-union is $O(RN16^s|V|^4)$. In our experiments, we set $R = N - 1$.

The segmentation tree of a previously unseen image is matched with the learned category model for identifying similar subtrees, representing category instances. For this matching, one may use the aforementioned definition of node saliency r_v of nodes $v \in \mathcal{T}$. Alternatively, r_v may also be defined so as to account for φ_v. Thus, in our experiments, we use two definitions of r_v: (i) $r_v \triangleq \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\psi}_v$, and (ii) $r_v \triangleq \varphi_v\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\psi}_v$. As demonstrated in Sec. VI, the latter definition improves recognition performance, since it forces the matching algorithm to remove from the subtree isomorphism f all those nodes in $\mathcal{T}$ with low frequency of occurrence in $\mathbb{D}$, which are likely to come from outliers in $\mathbb{D}$.

V. Learning the Optimal Weights of Region Properties

We have assumed that the relative significance to recognition of the various region properties included in $\boldsymbol{\psi}$ can be expressed by the perceptual weight vector $\boldsymbol{\xi}$ in (2). Estimation of the weights $\boldsymbol{\xi}$ is ideally done in a supervised psychophysical setting. In general, there is very limited past work on determining the perceptually valid weights of region properties without human supervision. In this paper, we approximate these ideal weights by a vector $\hat{\boldsymbol{\xi}}$ which maximizes the similarity measures of those image regions that are likely to belong to a category. To this end, we first select a subset of image regions from the given set of images having small differences in their properties (i.e., which are similar and thus candidates to represent a frequently occurring category), and then optimize $\hat{\boldsymbol{\xi}}$ over the selected subset, as detailed below.

For each node pair $(v, v') \in t \times t'$, $\forall t, t' \in \mathbb{T}$, we compute the empirical distribution of node-property differences $\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|$, where $\|\cdot\|$ denotes the vector 2-norm. If a category occurs in the given image set, the distribution of these differences may be expected to form two main modes. One mode would correspond to pairs of regions comprising the category subimages, having small $\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|$ values. The other mode would consist of arbitrary region pairs with larger $\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|$ values. Since there are more dissimilar than similar regions, the latter mode would have considerably larger distribution values. Of course, each mode would also contain contributions from chance similarities and differences.

The frequency histogram thus obtained is modeled as the two-component Gaussian mixture density, $P(\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|) = \pi_1 \mathscr{G}_1(\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|) + \pi_2 \mathscr{G}_2(\|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}\|)$. The means, variances of the Gaussian distributions $\mathscr{G}_1$ and $\mathscr{G}_2$, and the mixing coefficients π_1 and π_2 are computed via the standard EM algorithm [69]. Then, all of the node pairs $(v, v') \in \mathbb{T} \times \mathbb{T}$ are partitioned into two mutually exclusive subsets 1 and 2 corresponding to the two components of the Gaussian mixture density $\mathscr{G}_1$ and $\mathscr{G}_2$. Thus, a given node pair (v, v') is included in subset 1 if $\pi_1\mathscr{G}_1(\|\boldsymbol{\psi}_v{-}\boldsymbol{\psi}_{v'}\|) > \pi_2\mathscr{G}_2(\|\boldsymbol{\psi}_v{-}\boldsymbol{\psi}_{v'}\|)$. The subset of regions, $\mathbb{G}{\subset}\mathbb{T}{\times}\mathbb{T}$, corresponding to the Gaussian-mixture component with the smaller mean is taken to represent similar regions.

Next, we optimize $\boldsymbol{\xi}$ so that the sum of $\mathcal{S}$ values over $\mathbb{G}$ is maximum. From (1) and (2), we have $\min(r_v, r_{v'}) - |r_v - r_{v'}| = \frac{r_v+r_{v'}-|r_v-r_{v'}|}{2} - |r_v - r_{v'}| = \boldsymbol{\xi}^{\mathrm{T}}\frac{1}{2}(\boldsymbol{\psi}_v + \boldsymbol{\psi}_{v'} - 3|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}|)$. Let $\boldsymbol{\eta}_{vv'} \triangleq \frac{1}{2}(\boldsymbol{\psi}_v + \boldsymbol{\psi}_{v'} - 3|\boldsymbol{\psi}_v - \boldsymbol{\psi}_{v'}|)$. Then, from (4), maximizing similarity measures over $\mathbb{G}$ is computed as

$$\max_{\boldsymbol{\xi}} \sum_{(u,u')\in\mathbb{G}} \mathcal{S}_{uu'}(\boldsymbol{\xi}){=}\max_{\boldsymbol{\xi}} \textstyle\sum_{(u,u')\in\mathbb{G}} \left[\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\eta}_{uu'}{+}\sum_{(v,v')\in\mathcal{C}_{uu'}} \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\eta}_{vv'}\right], \tag{5}$$

$$={(|\mathbb{G}|{+}1)} \max_{\boldsymbol{\xi}} \boldsymbol{\xi}^{\mathrm{T}} \textstyle\sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'} {-} \sum_{(u,u')\in\mathbb{G}} \displaystyle\min_{\boldsymbol{\xi}} \boldsymbol{\xi}^{\mathrm{T}} \textstyle\sum_{(v,v')\in\mathbb{G}\setminus\mathcal{C}_{uu'}} \boldsymbol{\eta}_{vv'}, \tag{6}$$

$$\geq{(|\mathbb{G}|{+}1)} \max_{\boldsymbol{\xi}} \boldsymbol{\xi}^{\mathrm{T}} \textstyle\sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'} {-} \sum_{(u,u')\in\mathbb{G}} \hat{\boldsymbol{\xi}}^{\mathrm{T}} \sum_{(v,v')\in\mathbb{G}\setminus\mathcal{C}_{uu'}} \boldsymbol{\eta}_{vv'}, \tag{7}$$

where the optimal $\hat{\boldsymbol{\xi}}$ is computed by maximizing the lower bound in (7) as

$$\max_{\boldsymbol{\xi}} \boldsymbol{\xi}^{\mathrm{T}} \textstyle\sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'}, \quad \text{s.t. } \|\boldsymbol{\xi}\|{=}1,\ \boldsymbol{\xi}{\geq}0 \quad \Rightarrow \quad \hat{\boldsymbol{\xi}}{=}\frac{\left(\sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'}\right)_+}{\left\|\left(\sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'}\right)_+\right\|}, \tag{8}$$

where $(x)_+ \triangleq \max(0, x)$. The detailed derivation of the last step in (8) is given in Appendix . Eq. (8) simply enforces that the differences in the properties of a matching region pair should not, on an average, exceed their sum. The optimal $\hat{\boldsymbol{\xi}}$ thus obtained is used for computing the node saliencies r_v for our matching algorithm (Sec. III-A).

This concludes the description of our algorithms. The entire procedure of discovering category instances and learning the category model is summarized in Alg. 2. In the next section, we present the experimental evaluation of our approach.

```
Algorithm 2: Discovering Category Occurrences and Learning the Category Model
  Input : Set of training images $\mathbb{T}$ containing frequent occurrences of an object category, but not necessarily in every image.
  Represent the training images by segmentation trees $\mathbb{T}{=}\{t_1, \ldots, t_M\}$, using the algorithm of [46], [47], [49] (Sec. II);
  $\forall(t, t')\in\mathbb{T}\times\mathbb{T}$, and $\forall(v, v')\in t\times t'$, estimate the relative significance to recognition of region properties $\boldsymbol{\xi}$ (Sec. V);
  $\forall(t, t')\in\mathbb{T}\times\mathbb{T}$, and $\forall(v, v')\in t\times t'$, compute $\mathcal{S}_{vv'}$, given by (4). Compute the histogram $\mathcal{H}(\mathcal{S})$ (Sec. III-A) ;
  Detect the modes of $\mathcal{H}(\mathcal{S})$, using the algorithm of [63], and identify as category instances $\mathbb{D}\{t_1, \ldots t_N\}$ all subtree pairs rooted at nodes $(v, v')\in\mathbb{T}\times\mathbb{T}$ whose $\mathcal{S}_{vv'}$ values fall in $[\mathcal{S}_{\text{min}}, \mathcal{S}_{\text{max}}] = \arg\max_{\text{modes}} \sum_{\mathcal{S}\in\text{modes}} \mathcal{S}\cdot\mathcal{H}(\mathcal{S})$ (Sec III-B);
  Construct tree-unions $\mathcal{T}(\mathscr{P})$ of different permutations $\mathscr{P}$ of $\mathbb{D}$. Select the smallest entropy $\mathcal{T}(\mathscr{P})$ as the category model (Sec. IV) ;
```

VI. Results

This section presents a two-pronged empirical validation of our approach: (i) qualitative evaluation of tree-union models learned on arbitrary image sets, and (ii) quantitative evaluation of simultaneous detection, recognition and segmentation of all instances of a learned category present in a test image. To this end, we use the following benchmark datasets: (1) Caltech-101 faces (435 images), motorbikes (800 images), and airplanes (800 images) [42]; (2) Caltech rear-view cars (526 images) [12]; (3) UIUC multi-scale side-view cars (108 images); (4) Weizmann side-view horses (328 images) [70]; and TUD side-view cows (111 images) [37]. In the sequel, we will refer to faces, motorbikes, airplanes, cars, horses, and cows as target categories, since their instances will most frequently occur in our training sets as compared to some other categories (e.g., grass, trees, bookshelves, etc.). The Caltech-101 images are captured under varying illumination conditions, and contain a single, prominently featured object from the category amidst clutter. The Caltech cars and the UIUC cars increase complexity, since the images contain multiple cars, which appear at different resolutions, have low contrast with the textured background, and may be partially occluded. The other two datasets contain sideviews of walking/galloping horses and cows in their natural (cluttered) habitat. They help evaluate our algorithm's capability to handle articulated, non-rigid objects. We also use a total of 100 background images from Caltech-101 which do not contain the target categories. These background images will be referred to as negative examples, while images showing objects from the target category will be referred to as positive examples.

We use three different strategies to form training and test sets, which leads to three types of experiments. In Experiment 1, one half of the training set consists of positive images, while the other half consists of negative examples. The training images are not labeled positive or negative to ensure unsupervised training, i.e., it is not a priori known whether any specific training image contains objects from the category. The test set is formed from the remaining positive and negative examples. In Experiment 2, the training and test sets are selected as in Experiment 1, but the test images are randomly rotated, to evaluate rotation-in-plane recognition invariance. The image size is preserved by "filling out" the background, vacated by rotation, with a randomly selected negative example. Finally, Experiment 3 is aimed at testing the effect of varying the numbers of positive M_p and negative M_n examples in the training set. Two cases are considered: (1) M_p is fixed, while M_n increases, and (2) M_n is fixed, while M_p increases. The test set differs from that used in Experiments 1 and 2, in that it contains the remaining positive examples of all target categories, but not negative examples. Each experiment is repeated 10 times, and the average performance is reported.

Object detection, recognition and segmentation are conducted jointly by matching the learned tree-union model with the test-image trees. Those common subtrees whose similarity measure is larger than a specified threshold are adjudged as detected objects. The threshold is varied to plot a recall-precision curve, as a preferred measure of performance with respect to object detection and segmentation, compared to those used by classification-based techniques (e.g., ROC curve, and equal error rate) [38]. The results presented in this section are obtained for the similarity-measure threshold that yields the highest F-measure, $F\triangleq 2\cdot\text{Precision}\cdot\text{Recall}/(\text{Precision}+\text{Recall})$. To obtain the ground truth, we manually delineated the outer contours of cars in Caltech and UIUC images. Manually annotated target objects (i.e., ground truth) for the Caltech-101 faces, motorbikes, airplanes, Weizmann horses and TUD cows are publicly available. Let A_d denote the area of a detected object in the test image, and let A_g denote the ground-truth area of an object in the test image. Then, a detected object is said to be false positive (FP) if $\frac{A_d\cap A_g}{A_d\cup A_g}<0.5$, where $\cup$ denotes union, and $\cap$ denotes intersection. The remaining cases are declared true positives (TP). Segmentation error is defined only for TP's as $\frac{\text{XOR}(A_d,A_g)}{A_d\cup A_g}$. Average segmentation error is defined as the mean of segmentation errors on all TP's. We observe that these detection and segmentation performance criteria appear to agree with our own subjective judgement. We also define measures of recognition performance, evaluated in Experiment 3. Let n_p denote the number of TP detections whose ground-truth category (verified by visual inspection) is the same as that identified by our algorithm. Then, precision of recognition is defined as the ratio of n_p and the total number of TP detections. Also, recall of recognition is defined as the ratio of n_p and the total number of target objects in the test set. Note the difference between the notions of precision and recall of detection, and precision and recall of recognition that we use in this paper. To distinguish between the two sets of measures, in the sequel, we will use terms precision and recall to denote measures of detection performance, and recognition precision and recognition recall to denote measures of recognition performance.

In Experiments 1–3, we test the following variants of our approach. Ours0 corresponds to our preliminary work, presented in [48], where region properties are equally weighted to compute the node saliency (i.e., $\boldsymbol{\xi}=\mathbf{1}$), and where the similarity measure characterizing subtree isomorphism f between two trees t and t' is computed as $\mathcal{S}_{tt'}^{\text{old}}=\sum_{(v,v')\in f}2\min(r_v,r_{v'})$. In Ours1, instead of $\mathcal{S}_{tt'}^{\text{old}}$, we use the new similarity measure defined by (3), while $\boldsymbol{\xi}=\mathbf{1}$. In addition to the new similarity measure, Ours2 also uses the optimal weights of region properties given by (8). Finally, we also evaluate Ours2 when the saliency of nodes v in the tree-union are not weighted by their frequencies φ_v (end of Sec. IV), referred to as Ours2$^-$.

Regarding the comparison with prior work, there is very limited past work on segmenting (i.e., delineating the boundaries of) category instances in test images. Though the datasets used in our experiments are very popular benchmarks, at the time of initial submission of this work no quantitative segmentation results have been reported for Caltech-101, UIUC, and TUD cow images. For Weizmann horses, the best segmentation results are presented in [41], [70]. While the approach of [41] is semi-supervised requiring training images to contain only horses, the approach of [70] requires additional human supervision in terms of manually segmenting horses in training images. Thus, except for the segmentation results on Weizmann horses, our comparison with prior work is mostly in terms of detection accuracy. To this end, we consider the semi-supervised methods of [12], [37], [38], [40], [41], [43], which require training images to be labeled with respect to the category they contain. Note that our evaluation of detection error is also more rigorous than that of the referred methods. We consider precise extent (segmentation) of objects in the images, whereas in [12], [37], [43] bounding boxes around detections and true objects are used, in [38] correct detection is required to lie within an ellipse of a certain size centered at the true object's centroid, and in [40] correct detection is marked when a detected object's centroid lies within 25 pixels of the true centroid. We use the method of [37] without the post-processing step of pruning the false positives. Therefore, for fair comparison, we report two sets of our results, one obtained using the aforementioned more demanding evaluation criteria, and the other using the same experimental procedures as those of the corresponding baseline methods.

Experiment 1 – Qualitative Evaluation of Category Models: Fig. 6 illustrates two tree-unions $\mathcal{T}$ learned in Experiment 1 by Ours2 on two training sets $\mathbb{T}$ which contain four and six positive examples of Caltech-101 faces and Weizmann horses, respectively. The figure also shows the extracted similar subtrees $\mathbb{D}$ from the Caltech-101 training set. Nodes of $\mathcal{T}$ are depicted as rectangles that contain those regions in $\mathbb{D}$ that got matched with the corresponding node in $\mathcal{T}$ during learning. As can be seen, the structure of $\mathcal{T}$ correctly captures the recursive containment and spatial layout of regions that comprise the category instances appearing in the training set. For example, in the face tree-union, nodes "left-eye," "nose," and "right-eye" are found to be children of the node representing a larger "eyes-and-nose" region, which in turn is correctly identified as a child of the "face" node. Also, since context vector associated with "left-eye" points toward

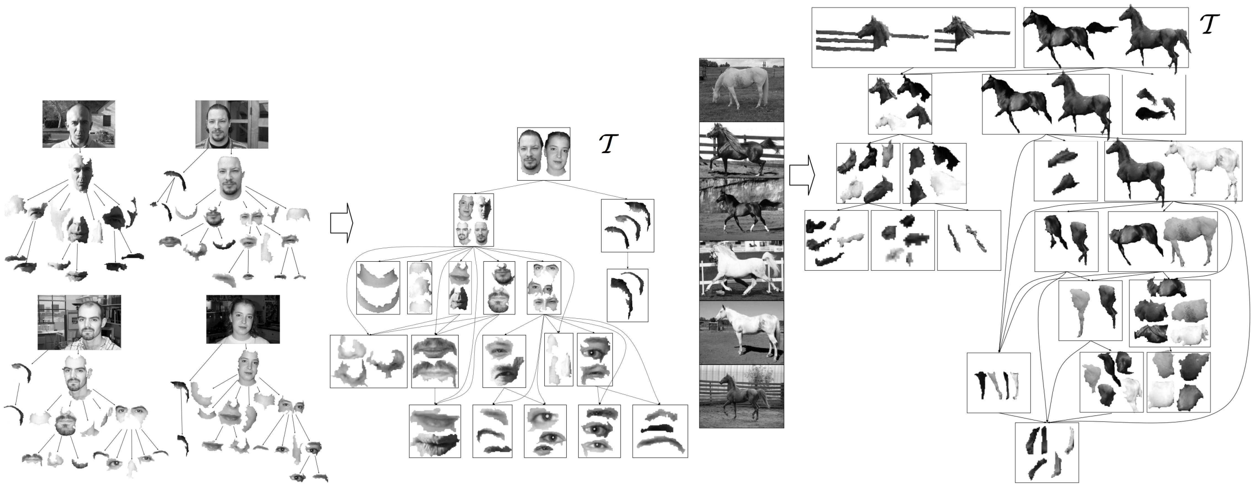


Fig. 6. An example of two tree-unions $\mathcal{T}$ (bottom) constructed from two training sets $\mathbb{T}$ (top) consisting of four and six positive images of Caltech-101 faces and Weizmann horses, respectively. Negative images are not shown. A subset of corresponding regions from $\mathbb{D}$ (middle row for Caltech-101 faces) that define a node in $\mathcal{T}$ are enclosed in the corresponding rectangle. The structure of $\mathcal{T}$ correctly captures the recursive containment and spatial layout of these regions.

the locations of “nose” and “right eye,” the tree-union encodes that “left-eye” is positioned to the left from “nose” and “right eye.” Similarly, “nostrils” are found to be above “mouth.” Note that none of the extracted similar subtrees in $\mathbb{D}$ of Caltech-101 faces has a node that corresponds to “face-and-hair,” which is the root of the tree-union. This root is obtained during augmenting similar subtrees with merger nodes for the purposes of many-to-many matching. The tree-union of horses contains two roots one of which represents “head-and-fence.” This root is assigned a relatively low frequency of occurrence in $\mathbb{D}$ ($\varphi_v = 2/137$), as compared to the other tree-union nodes, which indicates that it may represent an outlier.

Experiments 1 and 2 – Qualitative Evaluation of Detection and Segmentation: Figs. 7–10 illustrate simultaneous object detection and segmentation. As can be seen, all occurrences of the target categories in the images are detected without hypothesizing the number of category instances appearing in a specific image, as done in prior work (e.g., in [37]). Also, object detection and segmentation are accurate for relatively small training sets, despite background clutter and occlusions. Performance is good even in cases when: (1) the object edges are jagged and blurred (e.g., motorbikes in Fig. 8a); (2) the object parts are thin regions with low intensity contrast (e.g., airplanes in Fig. 8b); (3) the target objects appear at different scales in the test images (e.g., Caltech cars in Fig. 9a); (4) the category instances are partially occluded (e.g., UIUC cars

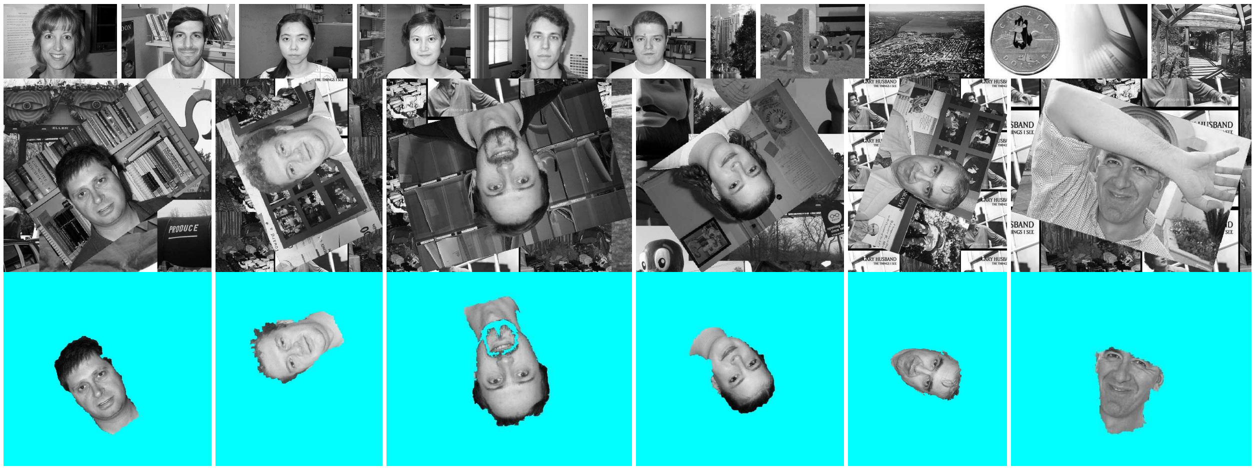


Fig. 7. Experiment 2 on the Caltech-101 faces: (top row) Sample training set consisting of six positive and negative examples. People appearing in the training set do not have beard. (middle row) Sample test images randomly rotated in the image plane showing people not seen during training. (bottom row) Detection and segmentation using Ours2.

in Fig. 9b); and (5) the target objects are randomly rotated in the image plane (e.g., UIUC cars in Fig. 9c and Caltech-101 faces in Fig. 7). Category instances that are not detected, for the most part, have low intensity contrasts with the surround, and thus their corresponding subtrees in test-image trees do not appear similar to the learned model structure. Some partially occluded Caltech and UIUC cars are not detected, since their matches with the model have lower similarity measures than the threshold, determined by the highest F-measure. Also, huge variations in the appearance of car windows, due to the reflections of surround, lead to the appearance of spurious regions in varying locations, not consistently present in training images, which do not become part of the learned model, and, therefore, are not matched with the model (Fig. 9a). Typically, the aforementioned effects are large enough to penalize the corresponding matched subimage from being interpreted as a true positive, but localized enough for the subimage to be evaluated as a false positive.

Experiments 1–3 – Quantitative Evaluation: Table I presents the average recall, precision, and segmentation errors obtained using Ours2 in Experiment 1, for the highest F-measure. The training set contains $M{=}100$ images out of which only $M_p{=}50$ are positive. The last two rows show the recall reported in [40] and [43]. As mentioned before, these state-of-the-art methods require training images to be labeled with respect to the category they contain, and for training use 50 images drawn from only positive examples. Also, their evaluation criteria are less

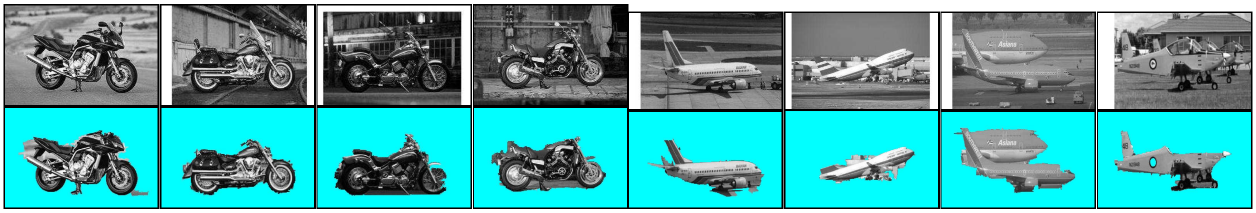
TABLE I

AVERAGE RESULTS USING OURS2 IN EXPERIMENT 1 FOR 50 POSITIVE TRAINING EXAMPLES (IN %)

	Faces	Motorbikes	Airplanes	Cars rear	Cars side	Horses	Cows
Recall	89.3±1.1	91.2±4.3	84.5±2.1	84.7±6.7	89.2±1.5	78.6±7.6	86.3±2.2
Precision	86.1±1.5	81.2±4.3	89.9±2.5	80.3±10.1	89.8±2.3	81.5±7.3	84.5±1.2
Seg. error	7.2±4.8	10.2±6.9	12.4±6.3	13.1±2.5	8.3±3.2	14.1±6.4	12.5±3.2
Recall using setup of [43]	98.2±0.6	94.3±1.1	94.1±0.8	99.2±0.6	99.2±0.7	96.6±1.2	100±0
Recall in [40]	94	92.4	NA	NA	92.8	92.1	NA
Recall in [43]	96.4	95.6	92.6	97.7	NA	NA	100

rigorous than ours, since they use bounding boxes or object's centroid estimates instead of object segmentation, and report results obtained for equal-error rate. The top three rows of Table I show the price we pay for: reducing the degree of supervision, using random negative examples in the training set, whose total number is the same as positive examples, and conducting a more demanding evaluation. Since prior work uses a different experimental setup, for fair comparison, we have also run our algorithms using their experimental procedures – specifically, discarding negative examples in training, and using the same numbers of training and test images, and the evaluation criteria for object detection as those used in [43]. The resulting equal-error-rate recall of Ours2 is reported in the fourth row of Table I. In this case, Ours2 outperforms the approaches of [40], [43] for almost all categories, except for the category motorbikes, with the loss of only 1.3% with respect to [43]. Also, for the purposes of comparison with the approach of [41] on the category Weizmann horses, we have used their setup: 20 positive training examples, 200 test images, and flipping all horses in test and train images to face a consistent direction, for which we have obtained the segmentation error of 4.3%, compared to theirs of 7%.

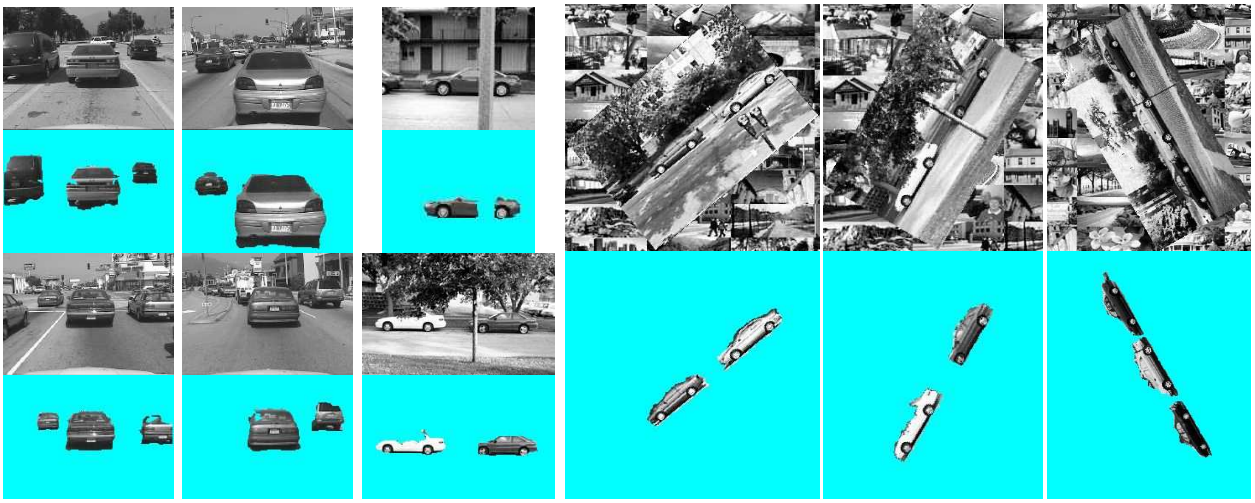
In Experiment 2, we obtained similar results to those in Experiment 1 (Table I). The corre-



(a) Caltech-101 motorbikes

(b) Caltech-101airplanes

Fig. 8. Experiment 1: Detection and segmentation on the Caltech-101 images showing motorbikes and airplanes using Ours2. The training set of each target category consists of 10 positive and 10 negative examples that are not labeled as positive or negative.



(a) Caltech cars rear-view (b) UIUC cars (c) Rotated UIUC cars

Fig. 9. Experiments 1 and 2 : Detection and segmentation on the Caltech and UIUC car images using Ours2. The training set consists of 10 positive and 10 negative examples. Ours2 successfully handles variations in scale (a), partial occlusions (b), and is invariant to rotation-in-plane (c).

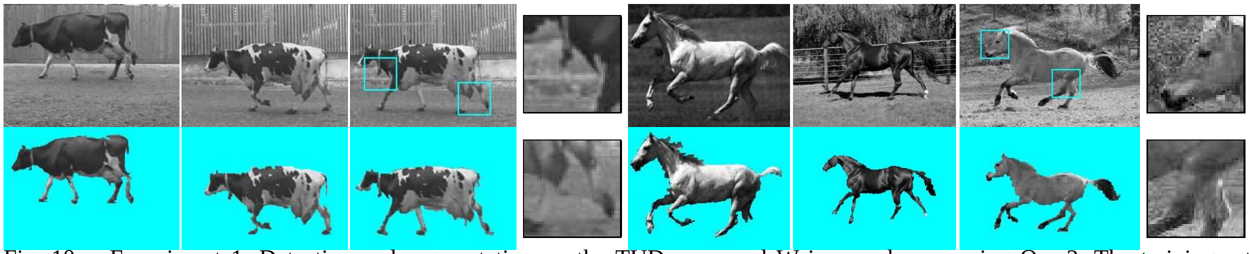


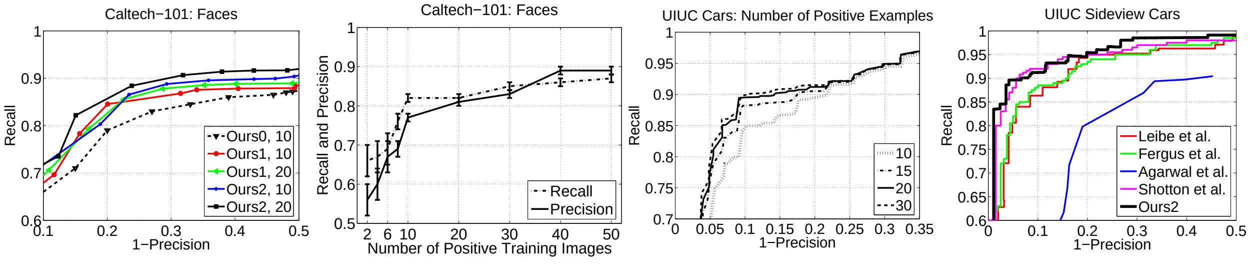
Fig. 10. Experiment 1: Detection and segmentation on the TUD cows and Weizmann horses using Ours2. The training set consists of 10 positive and 10 negative examples. The small images represent zoomed-in details (enclosed by the rectangles) of the larger image. Object segmentation fails on those object parts (e.g., zoomed-in details) that have low intensity contrasts with the surround, and thus do not form category-characteristic subtrees in the segmentation tree which can be matched with the category model.

sponding recall, precision and segmentation errors of Experiments 1 and 2 differ in less than one half of standard deviation on all the seven datasets. This small difference (in part due to the quantization error accompanying rotation with arbitrary digital rotation angles) demonstrates that our approach is invariant to rotations in the image plane.

Fig. 11 presents recall-precision curves (RPCs) obtained using Ours0, Ours1, and Ours2 on the Caltech-101 faces and UIUC cars in Experiment 1. As expected, increase in the number of positive training examples improves performance. The figure also compares the RPC of Ours2 against those of [12], [37], [38], [40]. For this comparison, we have adopted the same experimental procedure as described in these methods – specifically, we have used 50 training images randomly selected only from positive examples, and detection is measured with respect to an ellipse around the true object. As can be seen, Ours2 yields a slightly better performance than the competing methods under the same experimental conditions. For example, increase in the area under the RPCs of Ours2 vs. that of [40] is 2.3%.

Table II shows increase in the area under the RPCs of Ours2 as the number of training images becomes larger for the Caltech-101 faces, UIUC cars, and Weizmann horses. This increase is expressed as a percentage of the RPC area obtained for the smaller training set. Interestingly, for larger training sets we get only modest improvements. This suggests that our learning algorithm saturates after reaching a certain size of the training set (e.g., >40 positive examples for the Caltech-101 faces). Thus, for example, Table II details that increase from 10 to 20 positive examples enlarges the area under RPC of Ours2 by 2.1% and 1.7% for the Caltech-101 faces and UIUC cars, respectively. The corresponding performance measures for the same datasets are only 1% and 0.8% when the number of positive training images increases from 20 to 30. When more than 50 positive examples are used for training (see also Fig. 12), performance of any of Ours0, Ours1, and Ours2 does not downgrade, which suggests that our learning algorithm does not suffer from overfitting. Similar results are observed for the other datasets. Figs. 11 and Table II also demonstrate accuracy gains of Ours1 and Ours2 over Ours0, measured as increase in the area under RPCs. This increase is expressed as a percentage of the area for Ours0. Thus, for example, the new similarity measure used in Ours1 yields 7.3% area increase over Ours0 for the UIUC cars. Also, we get 3.6% area increase of Ours2 over Ours1 for the Caltech-101 faces. This result demonstrates the value of using perceptually motivated weights of region properties obtained by the algorithm discussed in Sec. V. In addition, Table II shows the gain in detection performance of Ours2 versus Ours2^{-} where outlier nodes are not accounted for in the tree-union. For example, for UIUC cars this gain is reflected in 4.4% increase in the area under RPC.

Fig. 12 and Table III show recognition performance of Ours2, evaluated in Experiment 3. Recognition recall and recognition precision are averaged over the seven target categories. As can be seen, small increase in negative examples M_n does not downgrade performance. As M_n becomes larger, it so happens that in our training set objects belonging to other categories start appearing more frequently. Therefore, by our basic definition, these objects become the target category. As a result, the algorithm now correctly learns the new category instances, as



(a) (b) (c) (d)

Fig. 11. Recall-precision curves: (a) Results on Caltech-101 faces obtained using Ours0, Ours1 and Ours2 for 10 and 20 positive training images in Experiment 1. (b) Performance of Ours2 on Caltech-101 faces for the highest F-measure improves as the number of positive training examples increases. (c) RPCs of Ours2 on UIUC cars as the number of positive training examples increases. (d) Comparison of Ours2 with [12], [37], [38], [40] on UIUC cars (multiscale), using the setup of the cited work: 50 positive training images, and detection is measured with respect to an ellipse around the true object.

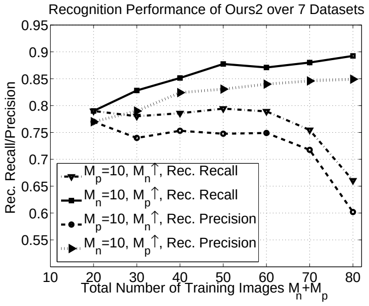


Fig. 12. Experiment 3: Recognition recall and precision of Ours2 for the highest F-measure of detection, averaged over the seven target categories. M_p and M_n are the numbers of positive and negative training examples, respectively. We consider two cases: (1) M_p=10 is fixed, while M_n increases, and (2) M_n=10 is fixed, while M_p increases.

TABLE II

INCREASE IN THE AREA UNDER RPC IN EXPERIMENT 1

Algorithm	Ours1 vs. Ours0	Ours2 vs. Ours0	Ours2 vs. Ours2^{-}	Ours2	Ours2
# positive imgs	10	10	10	20vs.10	30vs.10
UIUC cars	7.3%	9.2%	4.4%	2.1%	3.1%
Faces	4.9%	8.5%	3.4%	1.7%	2.5%
Horses	6.1%	9.5%	4.1%	2.8%	2.9%

TABLE III

RECOG. RECALL AND PRECISION OF OURS2 IN EXPERIMENT 3

	M_p=20 M_n=10		M_p=30 M_n=10	
	Recall	Precision	Recall	Precision
Faces	87.3%	81.2%	88.2%	84.5 %
Motorbikes	87.4%	78.9%	88.6%	77.8%
Airplanes	79.4%	79.9%	81.5%	88.5 %
Cars rear	82.7%	73.8%	83.2%	79.8%
Cars side	86.3%	79.2%	87.5%	88.1%
Horses	77.2%	75.9%	79.0%	80.2%
Cows	83.5%	78.2%	84.1%	82.2%

expected. Thus, with increase of M_n, the training set becomes inappropriate. Increasing the number of positive training examples yields higher recognition recall and precision.

VII. CONCLUSION

In this paper, we have formulated a new problem, that of completely unsupervised extraction and learning of a visual category frequently occurring in a given arbitrary image set, and presented its solution. The visual category is defined as a set of subimages characterized by similar geometric, photometric, and topological properties. Unsupervised means that the target category is not defined by the user, and whether and where any instances of the category appear in a specific image is not known. To discover category occurrences in the unlabeled image set, we have proposed to use a many-to-many matching algorithm that finds matching subimages within every pair of images. We have defined a new similarity measure between matching subimages which is recursively computed in terms of differences in geometric, photometric, and topological properties of subregions embedded within the subimages. This similarity measure fuses the information of similarities of the embedded subimages, where the similarities are weighted with respect to their relative significance to recognition. We have presented an algorithm for estimating these weights, without using any supervision. We have also proposed to compute a union of all matching subimages in the image set, interpreted as category instances, and thus obtain the category model. The category model registers all (partial) views of category occurrences in the image set, yielding a representation of the complete (unoccluded) object. Empirical validation on seven benchmark datasets, which present challenges such as object articulation, occlusion, and significant background clutter, demonstrates high recall and precision of category detection and recognition, as well as high accuracy of segmentation of category occurrences, in completely unsupervised settings. In weakly supervised settings, using the same experimental procedures as those presented in prior work, our approach outperforms existing baseline methods in object detection and segmentation on almost all categories tested, with one exception where our performance is slightly inferior within standard deviation. Our qualitative empirical evaluation demonstrates that the learned category model correctly captures the recursive containment and spatial layout of regions comprising the category instances in the image set.

APPENDIX

DERIVATION OF THE OPTIMAL WEIGHTS OF REGION PROPERTIES

In this section we derive the optimal weights of region properties $\hat{\boldsymbol{\xi}}$ as a solution of the optimization problem stated in (8). Recall that $\boldsymbol{\eta}_{uu'}$ is a function of region properties of those node pairs (u, u') that belong to the set of similar regions $\mathbb{G}$, as explained in Sec. V. Specifically, we have $\boldsymbol{\eta}_{uu'} = \frac{1}{2}(\boldsymbol{\psi}_u + \boldsymbol{\psi}_{u'} - 3|\boldsymbol{\psi}_u - \boldsymbol{\psi}_{u'}|)$. Let $\boldsymbol{\eta} \triangleq \sum_{(u,u')\in\mathbb{G}} \boldsymbol{\eta}_{uu'}$. Then, $\hat{\boldsymbol{\xi}}$ can be found by solving the following problem:

$$\max_{\boldsymbol{\xi}} \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\eta}, \quad \text{s.t. } \|\boldsymbol{\xi}\|^2 = 1, \ \boldsymbol{\xi} \geq 0 \ . \tag{9}$$

The Lagrangian of (9) reads: $L = -\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\eta} + \lambda(\|\boldsymbol{\xi}\|^2-1) + \sum_i \zeta_i(-\xi_i)$, where λ and $\boldsymbol{\zeta}\geq 0$ are the Lagrangian multipliers. Taking the derivative of L with respect to $\boldsymbol{\xi}$, and setting it to zero gives

$$\partial L/\partial\boldsymbol{\xi} = -\boldsymbol{\eta} + 2\lambda\boldsymbol{\xi} - \boldsymbol{\zeta} = 0 \;\Rightarrow\; \boldsymbol{\xi} = \frac{\boldsymbol{\eta}+\boldsymbol{\zeta}}{2\lambda}. \tag{10}$$

To derive a closed-form solution of (9), we make the weak assumption that there exists one region property i for which the corresponding element η_i of $\boldsymbol{\eta}$ is positive. This assumption is very weak, since from the definition of $\boldsymbol{\eta}$, the converse (i.e., $\boldsymbol{\eta}<0$) would mean that there are on average more node pairs in $\mathbb{G}$ whose differences of region properties are larger than their sums. This in turn is very unlikely, because nodes considered for estimating $\hat{\boldsymbol{\xi}}$ belong to $\mathbb{G}$, which is a *large* set of similar regions with very *small* differences in their properties.

By making use of the above assumption, we prove that $\lambda>0$. Suppose the converse, i.e., $\lambda<0$. Since there exists $\eta_i>0$, then $\eta_i+\zeta_i>0$. It follows from (10) that $\xi_i<0$, which contradicts the constraint $\boldsymbol{\xi}\geq 0$. From the Karush-Kuhn-Tucker condition [71], namely $\sum_v \zeta_i\xi_i=0$, it follows:

1) If $\eta_i=0 \;\Rightarrow\; \zeta_i=0 \;\Rightarrow\; \xi_i=0$;
2) If $\eta_i<0 \;\Rightarrow\; \zeta_i>0 \;\Rightarrow\; \xi_i=0$;
3) If $\eta_i>0 \;\Rightarrow\; \eta_i+\zeta_i>0 \;\Rightarrow\; \xi_i>0 \;\Rightarrow\; \zeta_i=0 \;\Rightarrow\; \xi_i=\dfrac{\eta_i}{2\lambda}$.

It immediately follows that the optimal $\hat{\boldsymbol{\xi}} = \frac{(\boldsymbol{\eta})_+}{\|(\boldsymbol{\eta})_+\|}$, where $(x)_+ \triangleq \max(0,x)$.

References

[1] J. Feldman and Y. Yakimovsky, "Decision theory and artificial intelligence: A semantics based region analyzer," *AI*, vol. 5, no. 4, pp. 349–371, 1974.

[2] A. Hanson and E. Riseman, "VISIONS: A computer system for interpreting scenes," in *Computer Vision Systems*, Hanson and Riseman, Eds. Academic Press, New York, 1978, pp. 303–333.

[3] T. J. Fan, G. Medioni, and R. Nevatia, "Recognizing 3D objects using surface descriptors," *IEEE Trans. PAMI*, vol. 11, no. 11, pp. 1140–1157, 1989.

[4] D. T. Clemens, "Region-based feature interpretation for recognizing 3D models in 2D images," MIT, Tech. Rep. AITR-1307, 1991.

[5] R. Basri and D. Jacobs, "Recognition using region correspondences," *IJCV*, vol. 25, no. 2, pp. 145–166, 1997.

[6] A. R. Ahmadyfard and J. V. Kittler, "Using relaxation technique for region-based object recognition," *Image and Vision Computing*, vol. 20, no. 11, pp. 769–781, 2002.

[7] R. Zhang and Z. Zhang, "Hidden semantic concept discovery in region based image retrieval," in *CVPR*, vol. 2, 2004, pp. 996–1001.

[8] Y. Keselman and S. Dickinson, "Generic model abstraction from examples," *IEEE TPAMI*, vol. 27, no. 7, pp. 1141–1156, 2005.

[9] I. Weiss and M. Ray, “Recognizing articulated objects using a region-based invariant transform,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 27, no. 10, pp. 1660–1665, 2005.

[10] M. A. Fischler and R. A. Elschlager, “The representation and matching of pictorial structures,” *IEEE Trans. Computers*, vol. C-22, no. 1, pp. 67–92, 1973.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[12] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *CVPR*, vol. 2, 2003, pp. 264–271.

[13] J. L. Crowley and A. C. Sanderson, “Multiple resolution representation and probabilistic matching of 2-D gray-scale shape,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 1, pp. 113–121, 1987.

[14] J. Utans, “Learning in compositional hierarchies: Inducing the structure of objects from data,” in *Advances in Neural Information Processing Systems*, vol. 6, 1994, pp. 285–292.

[15] L. Bretzner and T. Lindeberg, “Qualitative multi-scale feature hierarchies for object tracking,” in *SCALE-SPACE ’99, LNCS Vol. 1682*, 1999, pp. 117–128.

[16] C. A. Bouman and M. Shapiro, “A multiscale random field model for Bayesian image segmentation,” *IEEE Trans. Image Processing*, vol. 3, no. 2, pp. 162–177, 1994.

[17] A. Shokoufandeh, I. Marsic, and S. Dickinson, “View-based object recognition using saliency maps,” *Image and Vision Computing*, vol. 17, no. 5-6, pp. 445–460, 1999.

[18] H. Cheng and C. A. Bouman, “Multiscale Bayesian segmentation using a trainable context model,” *IEEE Trans. Image Processing*, vol. 10, no. 4, pp. 511–525, 2001.

[19] S. Krempp, D. Geman, and Y. Amit, “Sequential learning of reusable parts for object detection,” CS Johns Hopkins, Tech. Rep., 2002.

[20] A. J. Storkey and C. K. I. Williams, “Image modeling with position-encoding dynamic trees,” *IEEE TPAMI*, vol. 25, no. 7, pp. 859–871, 2003.

[21] S. Todorovic and M. C. Nechyba, “Dynamic trees for unsupervised segmentation and matching of image regions,” *IEEE TPAMI*, vol. 27, no. 11, pp. 1762–1777, 2005.

[22] ——, “Interpretation of complex scenes using dynamic tree-structure Bayesian networks,” *Computer Vision and Image Understanding*, vol. In Press, Available online, 2006.

[23] Y. Jin and S. Geman, “Context and hierarchy in a probabilistic image model,” in *CVPR*, vol. 2, 2006, pp. 2145–2152.

[24] S. Fidler, G. Berginc, and A. Leonardis, “Hierarchical statistical learning of generic parts of object structure,” in *CVPR*, vol. 1, 2006, pp. 182–189.

[25] B. Ommer, M. Sauter, and J. M. Buhmann, “Learning top-down grouping of compositional hierarchies for recognition,” in *CVPR, Workshop Percep. Org. Comp. Vision*, 2006, p. 194.

[26] A. Shokoufandeh, L. Bretzner, D. Macrini, M. F. Demirici, C. J´onsson, and S. Dickinson, “The representation and matching of categorical shape,” *Computer Vision and Image Understanding*, vol. 103, no. 2, pp. 139–154, 2006.

[27] W. Wang, I. Pollak, T. S. Wong, C. A. Bouman, M. P. Harper, and J. M. Siskind, “Hierarchical stochastic image grammars for classification and segmentation,” *IEEE Trans. Image Processing*, vol. 15, no. 10, pp. 3033–3052, 2006.

[28] J. M. Siskind, J. J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman, “Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape,” *IEEE Trans. PAMI*, vol. 29, no. 9, pp. 1504–1519, 2007.

[29] P. H. Winston, “Learning structural descriptions from examples,” in *Psychology of computer vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975, ch. 5, pp. 157–209.

[30] G. J. Ettinger, “Large hierarchical object recognition using libraries of parameterized model sub-parts,” in *CVPR*, 1988, pp. 32–41.

[31] H. Nishida and S. Mori, “An algebraic approach to automatic construction of structural models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1298–1311, 1993.

[32] B. Perrin, N. Ahuja, and N. Srinivasa, “Learning multiscale image models of 2D object classes,” in *ACCV, Springer LNCS*, vol. 1352, 1998, pp. 323–331.

[33] Y. Xu, E. Saber, and A. M. Tekalp, “Dynamic learning from multiple examples for semantic object segmentation and search,” *Comp. Vision Image Underst.*, vol. 95, pp. 334–353, 2005.

[34] X. Jiang, A. Munger, and H. Bunke, “On median graphs: properties, algorithms, and applications,” *IEEE TPAMI*, vol. 23, no. 10, pp. 1144–1151, 2001.

[35] B. Luo, R. C. Wilson, and E. R. Hancock, “Spectral embedding of graphs,” *Pattern Recognition*, vol. 36, no. 18, pp. 2213–2230, 2003.

[36] A. Levinshtein, C. Sminchisescu, and S. Dickinson, “Learning hierarchical shape models from examples,” in *EMMCVPR, Springer LNCS*, vol. 3757, 2005, pp. 251–267.

[37] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model,” in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 17–32.

[38] S. Agarwal, A. Awan, and D. Roth, “Learning to detect objects in images via a sparse, part-based representation,” *IEEE TPAMI*, vol. 26, no. 11, pp. 1475–1490, 2004.

[39] J. Winn, A. Criminisi, and T. Minka, “Object categorization by learned universal visual dictionary,” in *ICCV*, vol. 2, 2005, pp. 1800–1807.

[40] J. Shotton, A. Blake, and R. Cipolla, “Contour-based learning for object detection,” in *ICCV*, vol. 1, 2005, pp. 503–510.

[41] J. Winn and N. Jojic, “Locus: learning object classes with unsupervised segmentation,” in *ICCV*, vol. 1, 2005, pp. 756–763.

[42] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE TPAMI*, vol. 28, no. 4, pp. 594– 611, 2006.

[43] A. Opelt, A. Pinz, and A. Zisserman, “Incremental learning of object detectors using a visual shape alphabet,” in *CVPR*, vol. 1, 2006, pp. 3–10.

[44] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, “Discovering object categories in image collections,” in *ICCV*, 2005.

[45] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, “Using multiple segmentations to discover objects and their extent in image collections,” in *CVPR*, vol. 2, 2006, pp. 1605–1604.

[46] N. Ahuja, “A transform for multiscale image segmentation by integrated edge and region detection,” *IEEE TPAMI*, vol. 18, no. 12, pp. 1211–1235, 1996.

[47] H. Arora and N. Ahuja, “Analysis of ramp discontinuity model for multiscale image segmentation,” in *ICPR*, 2006.

[48] S. Todorovic and N. Ahuja, “Extracting subimages of an unknown category from a set of images,” in *CVPR*, vol. 1, 2006, pp. 927–934.

[49] M. Tabb and N. Ahuja, “Multiscale image segmentation by integrated edge and region detection,” *IEEE Trans. Image Processing*, vol. 6, no. 5, pp. 642–655, 1997.

[50] H. Bunke and G. Allermann, “Inexact graph matching for structural pattern recognition,” *Pattern Rec. Letters*, vol. 1, no. 4, pp. 245–253, 1983.

[51] M. A. Eshera and K. S. Fu, “An image understanding system using attributed symbolic representation and inexact graph-matching,” *IEEE TPAMI*, vol. 8, no. 5, pp. 604–618, 1986.

[52] M. Pelillo, K. Siddiqi, and S. W. Zucker, “Matching hierarchical structures using association graphs,” *IEEE TPAMI*, vol. 21, no. 11, pp. 1105–1120, 1999.

[53] H. Bunke and A. Kandel, “Mean and maximum common subgraph of two graphs,” *Pattern Rec. Letters*, vol. 21, no. 2, pp. 163 – 168, 2000.

[54] A. Torsello and E. R. Hancock, “Computing approximate tree edit distance using relaxation labeling,” *Pattern Recogn. Lett.*, vol. 24, no. 8, pp. 1089–1097, 2003.

[55] T. B. Sebastian, P. N. Klein, and B. B. Kimia, “Recognition of shapes by editing their shock graphs,” *IEEE Trans. PAMI*, vol. 26, no. 5, pp. 550–571, 2004.

[56] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, “Shock graphs and shape matching,” *Int. J. Comput. Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[57] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, and S. W. Zucker, “Indexing hierarchical structures using graph spectra,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1125–1140, 2005.

[58] M. Pelillo, K. Siddiqi, and S. W. Zucker, “Many-to-many matching of attributed trees using association graphs and game dynamics,” in *Int. Workshop Visual Form, Springer LNCS*, vol. 2059, 2001, pp. 583–593.

[59] M. F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. J. Dickinson, “Object recognition as many-to-many feature matching,” *Int. J. Computer Vision*, vol. 69, no. 2, pp. 203–222, 2006.

[60] T. Caelli and S. Kosinov, “An eigenspace projection clustering method for inexact graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 515–519, 2004.

[61] S. Todorovic and N. Ahuja, “Region-based hierarchical image matching,” *Int. J. Computer Vision*, to appear.

[62] M. Pelillo, “Replicator equations, maximal cliques, and graph isomorphism,” *Neural Computation*, vol. 11, no. 9, pp. 1935 – 1955, 1999.

[63] A. Touzani and J. G. Postaire, “Mode detection by relaxation,” *IEEE Trans. PAMI*, vol. 10, no. 6, pp. 970–978, 1988.

[64] A. Gupta and N. Nishimura, “Finding largest subtrees and smallest supertrees,” *Algorithmica*, vol. 21, no. 2, pp. 183–210, 1998.

[65] H. Bunke, X. Jiang, and A. Kandel, “On the minimum common supergraph of two graphs,” *Computing*, vol. 65, no. 1, pp. 13–25, 2000.

[66] A. Torsello and E. R. Hancock, “Matching and embedding through edit-union of trees,” in *ECCV*, vol. 3, 2002, pp. 822–836.

[67] H. Bunke, P. Foggia, C. Guidobaldi, and M. Vento, “Graph clustering using the weighted minimum common supergraph,” in *IAPR Workshop GbRPR, Springer LNCS*, vol. 2726, 2003, pp. 235–246.

[68] A. Torsello and E. R. Hancock, “Learning shape-classes using a mixture of tree-unions,” *IEEE Trans. PAMI*, vol. 28, no. 6, pp. 954–967, 2006.

[69] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Stat. Soc.*, vol. B-39, no. 1, pp. 1–38, 1977.

[70] E. Borenstein and S. Ullman, “Class-specific, top-down segmentation,” in *ECCV*, vol. 2, 2002, pp. 109–124.

[71] E. K. P. Chong and S. H. Zak, *An introduction to optimization*, 2nd ed. New York: John Wiley & Sons, Inc., 2001.