Connected Segmentation Tree – A Joint Representation of Region Layout and Hierarchy

Narendra Ahuja and Sinisa Todorovic
Beckman Institute, University of Illinois at Urbana-Champaign
{n-ahuja, sintod}@uiuc.edu

Abstract

This paper proposes a new object representation, called Connected Segmentation Tree (CST), which captures canonical characteristics of the object in terms of the photometric, geometric, and spatial adjacency and containment properties of its constituent image regions. CST is obtained by augmenting the object's segmentation tree (ST) with inter-region neighbor links, in addition to their recursive embedding structure already present in ST. This makes CST a hierarchy of region adjacency graphs. A region's neighbors are computed using an extension to regions of the Voronoi diagram for point patterns. Unsupervised learning of the CST model of a category is formulated as matching the CST graph representations of unlabeled training images, and fusing their maximally matching subgraphs. A new learning algorithm is proposed that optimizes the model structure by simultaneously *searching for both the most salient nodes (regions) and the most salient edges (containment and neighbor relationships of regions) across the image graphs. Matching of the category model to the CST of a new image results in simultaneous detection, segmentation and recognition of all occurrences of the category, and a semantic explanation of these results.*

1. Introduction

Physical objects in 3D world are finite and cohesive, having characteristic photometric and geometric properties, such as contrast, size, and shape. They also possess characteristic visual structure which may be hierarchical, reflecting the containment and spatial layout of structure of the matter comprising them. Finally, they occupy distinct positions in space. Real world images are 2D projections of real world objects, giving rise to 2D objects in images. The images also exhibit a structure that mimics the real world structure: (1) The 2D regions occur in a certain spatial configuration, or spatial layout. (2) The hierarchical structure of 3D, physical objects appears as recursive embedding of subregions within the object region. (3) 2D regions comprising a subimage occupied by an object have certain photometric and geometric properties. Most prior work on 2D image/object representation uses only (3), or a combination of either (1)+(3) or (2)+(3). This paper proposes an object representation that simultaneously captures all three aspects of image structure, namely, (1)+(2)+(3), and demonstrates the advantages of this more comprehensive object representation over the existing approaches in category modeling and recognition.

Specifically, we extend the segmentation tree (ST) representation, used previously in [19, 3], which models (2)+(3) of regions that occur in a multiscale segmentation of images, by representing regions as nodes and their embedded regions as the node's children. Like other strictly hierarchical representations, ST can only help one infer some aspects of (1) from the information explicitly stored in it via (2, 3), e.g., the centroid locations and orientations of subregions. However, ST cannot distinguish many different ways in which the same set of subregions may be spatially distributed within the parent region, giving rise to significantly different visual appearances (Fig. 1a), while their properties (2,3) remain fixed. Consequently, STs for many visually distinct objects are identical. The extended model we propose in this paper addresses this problem by including new information about (1) – namely, information about 2D spatial adjacency among the regions – while retaining the information about their recursive embedding structure already present in ST. The new model augments ST with region adjacency graphs, one for the children of each ST node. A neighbor edge is added between two sibling nodes in ST if the corresponding two regions are neighbors in the image. This transforms ST into a graph, consisting of two distinct sets of edges – one representing the original, parent-child hierarchy, and the other, consisting of lateral links, representing the newly added neighbor relationships (Fig. 1). The neighbor relationships between any nonsibling nodes in CST can be easily retrieved by examining the neighbor relations of their ancestor nodes. To highlight the presence of the complementary, neighbor informa-

tion modifying the segmentation tree, the new representation is referred to as connected segmentation tree (CST), even though it is strictly a graph. Both nodes and edges of CST have attributes, i.e., they are weighted, where the node (edge) weight is defined in terms of properties of the corresponding region (spatial relationship between regions). Thus, CST generalizes ST to represent images as a hierarchy of region adjacency graphs. As multiscale regions may be viewed as a basic vocabulary of object categories, the CST may be seen as a basis for defining general purpose image syntax, which can serve as an intermediate stage to isolate and simplify inference of image semantics.

Since different spatial distributions of the same set of regions result in significantly different 2D objects, modeling the region adjacency distribution captured in property (1) above is important. However, formalizing this distribution is difficult, in part, because there is not even a clear intuitive notion of neighbors among regions. For example, it is not clear which of the many compact regions in Fig. 1a should be called neighbors. Most prior work considers only contiguous regions that share borders as neighbors. As the second major contribution of this paper, we propose an approach to defining a region's neighbors, as well as the strength of their neighborliness. Specifically, we generalize the Voronoi diagram, conventionally used for point patterns, to define region neighbors. Our generalized Voronoi diagram partitions an image into polygons, each containing a region, representing its area of influence around it. Regions having neighboring polygons define simple neighbors, and the polygon properties determine their neighborliness. This definition has yielded perceptually valid neighbors in most cases in our informal evaluation (not presented in this paper) on a large collection of image regions. The neighbors and the strengths of their neighborliness are encoded by lateral links and associated weights in CST.

As the third major contribution of this paper, we propose a new algorithm for learning CSTs. The max-clique based algorithm for learning STs [19, 3] cannot be used for CSTs, because a clique is defined only for graphs with unweighted edges. We resolve this by treating the weighted edges in CST as a new set of weighted vertices, disjoint from the set of nodes representing regions. Given the CST representations of training images, our new algorithm dynamically optimizes the model structure by simultaneously searching for nodes (regions) and edges (neighbor relationships) with the highest weights across the images. The resulting model may, at one extreme, degenerate into a planar graph, encoding only the region adjacency, or at another extreme, into a strict tree, encoding only the recursive embedding of regions. Consequently, CSTs, due to their richer coverage of object structure, are expected to more accurately model a broader variety of 2D categories than the existing approaches based on capturing either (2)+(3) or (1)+(3). While the literature discusses whether (1)+(3) or (2)+(3) is more important for modeling objects, we present the first empirical evaluation of advantages of jointly modeling (1)+(2)+(3) vs. modeling either (1)+(3) or (2)+(3).

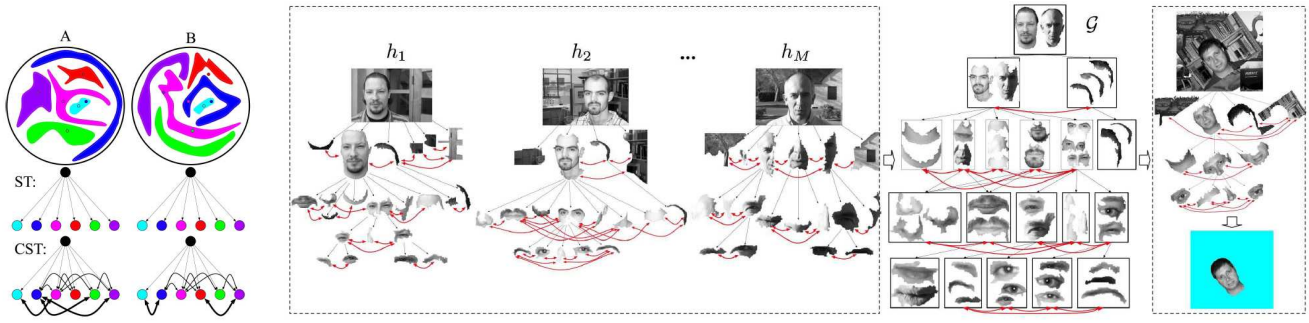
By adding (1) to aspects (2) and (3) of the image structure already captured by ST, CSTs either retain or strengthen the following desired characteristics of category recognition based on them: (I) Efficient training under varying degrees of supervision, including unsupervised settings, and on training sets of sizes very small to arbitrarily large; (II) Providing for both object recognition and segmentation that is invariant to translation, in-plane rotation, object articulation, partial occlusion, background clutter, and a certain degree of scale changes; and (III) Providing for a semantic explanation of object recognition in terms of the learned object structure captured in the representation.

Given a set of training images, the three main steps of the CST based approach to object learning and recognition are illustrated in Fig. 1b. Step 1: A CST is obtained for each image. Step 2: The training images need not all contain examples of the unspecified category(ies) contained in the training set which we want to learn. The category occurrences are discovered by searching for subimages within the training images that are more similar to each other with respect to (1)+(2)+(3) than to any other objects. This is done by matching CSTs, and finding their common subgraphs. Each set of matched subgraphs represents all occurrences of one discovered category. The subgraphs within each such set are then fused into a single graph-union which constitutes the canonical model of the category. Step 3: Given the CST of a new image, it is matched with the learned model to simultaneously detect, recognize and segment all category occurrences in the image. This matching also identifies object parts along with their containment and neighbor relationships present, which can be used as an explanation of why each object is recognized. These steps parallel the corresponding steps that would be followed if an ST were used as in [19]; however, CST based processing involves cyclic graphs instead of trees which significantly changes the nature and complexity of the associated algorithms.

Section 2 reviews prior work; Sections 3 and 4 describe Steps 1 and 2; and Sec. 5 presents experimental evaluation.

2. Prior Work and Our Contributions

There is a wide agreement in the literature that modeling the spatial information along with (3) (i.e., photometric and geometric properties) of regions is beneficial. However, different approaches advocate different representations of this spatial information. Methods that account only for (1) (i.e., object's spatial layout) and (3) simplify the object's interior structure to a flat layout of regions, and compensate for the missing information about (2) (i.e., containment or compositionality) by developing complex models of (3). This



(a) STs vs. CSTs (b) Block-diagram of our approach using the actual results of our algorithms

Figure 1. **(a)** Segmentation trees (STs), like other strict hierarchical models, do not directly encode the spatial layout of parts, but have to infer this from the intrinsic part properties explicitly stored in STs, e.g., part orientation and centroid location relative to the object. Parts of objects A and B have the same centroid locations and orientations. Therefore, the structure of two STs representing objects A and B is identical. In contrast, the connected segmentation tree (CST) adds lateral edges to ST that link neighboring parts, thus significantly reducing the modeling ambiguity about their spatial layout. Indeed, the two CSTs representing objects A and B differ in neighbor links marked bold. **(b)** Our approach: Training images containing faces are represented by CSTs which capture the recursive containment (black edges) and neighbor relationships (red edges) of regions. Similar common subgraphs of the CSTs (faces), are registered and fused into the category model $\mathcal{G}$. CST of a new image is matched with $\mathcal{G}$ to simultaneously detect, recognize, segment, and explain face occurrences. "Explanation" refers to the ability to recursively backtrace the results of recognition to the recognition of constituent facial parts (which objects in their own right) and their spatial relationships.

usually leads to infeasible learning and inference, forcing these approaches to resort to restrictive assumptions about (1). For example, the constellation [10] and k-fan [8] models have a pre-specified, small number of parts, configured in a pre-specified planar-graph structure. The hierarchical models of, e.g., [4, 13, 11, 18, 15, 9] capture only (1)+(3), by allowing an object part to appear alternatively as a set of subparts. The hierarchy underlying these models is usually constrained for tractability, e.g., by assuming a fixed number of object parts, depth, or branching factor. The (1)+(3) based model of [6], and (2)+(3) based model of [19, 3] relax the restrictive assumptions of their peer models; however, either approach misses to jointly encode (1) and (2), i.e., the complete spatial information about object's structure.

Only a few approaches recursively decompose an object into parts while retaining lateral relations among the parts themselves. For example, the And-Or graph [7] specifies a context sensitive grammar that uses both Markov tree and Markov random fields to arrange user-specified templates. Also, the model used in [14] captures the object-characteristic blobs and their containment, and subsequently their pairwise contiguity relationships. In contrast, our definition of neighbors allows even non-contiguous regions as neighbors, and we *simultaneously* identify object-characteristic regions, and their containment and neighbor relationships.

CSTs inherit a number of attractive properties from STs [19, 3]. Since region boundaries coincide with object contours, the use of CST results in simultaneous object recognition and segmentation. The use of regions and their two types of relationships by CST helps efficiently model the natural properties of real-world objects, such as spatial cohesiveness and relative locations. The exact learning of CSTs is computationally feasible. The requirement for unsupervised learning is that the training images should contain frequent occurrences of a category, although not necessarily in every training image.

3. From Image to CST

This section presents Step 1 of our approach (Sec. 1). The CST is derived from the ST of the image. ST captures the recursive embedding of smaller regions within larger ones, obtained from the multiscale segmentation algorithm of [2]. In ST, the total number of nodes ($\approx$50), branching factor, and depth ($\approx$10) are all automatically determined by the image at hand. The ST is transformed into CST by introducing lateral edges connecting neighboring sibling nodes under every node in ST. Below, we first present our algorithm for the computation of region neighbors and their strengths, and then review the region properties associated with nodes in CST which completes the representation.

3.1. Neighbors of a Region

While many approaches have been proposed to define neighbors of points in a point pattern [1], the definition of neighbors for nonpoint objects has received little attention in the literature. To define a region's neighbors as well as the strength of their neighborliness, we generalize the Voronoi diagram for point patterns to regions. The Voronoi diagram of a point pattern $\mathscr{S}$ associates with each point $P \in \mathscr{S}$ a cell $\mathcal{V}_P$ which is the re-

gion closer to P than to any other point $Q \in \mathscr{S}$, $\mathcal{V}_P = \{T : T \in \mathbb{R}^n, \forall Q \in \mathscr{S}, d(Q,T) > d(P,T)\}$. Thus, for any non-degenerate distribution of points, the Voronoi diagram tessellates the space into a set of cells. For the 2D case here, the cells belonging to the interior of $\mathscr{S}$ are convex polygons, each containing exactly one of the points in the pattern. The points at the boundary of $\mathscr{S}$ have incomplete cells, extending to infinity in the directions of no neighbors.

The intuition underlying our extension of the Voronoi diagram to regions is that regions are exposed to each other through their nearby boundary segments. If parts of the borders of two regions are visible to and near each other, and are sufficiently far from other region boundaries, then the two regions are called neighbors. Thus, the exposure of one region to another here means more than just line-of-sight connectivity. Neighbors are derived from the Voronoi relationships among the individual pixels along the region boundaries. Given a set of regions, we first compute the point based Voronoi tessellation for all pixels along the regions' boundaries (Fig. 2). Then, for each region v, we find the union of the Voronoi polygons of pixels along its boundary, thus obtaining a generalized Voronoi polygon of the region $\mathcal{V}_v$ that defines the area of influence of v in the image. Generalized here means that the line segments connecting the vertices of $\mathcal{V}_v$ are a sequence of short line segments, in general, not aligned with each other, thus resulting in a jagged edge between the vertices. Any such sequence of line segments between two vertices of $\mathcal{V}_v$ represents a shared border with another adjacent polygon, e.g., $V_{v'}$ belonging to region v', which means that v and v' are neighbors (Figs. 2). The relative degrees of exposure of a region to all its neighbors are used as measures of the strengths of its neighborliness to these neighbors. The neighborliness is in general asymmetric, by definition. Specifically, given that v and v' are neighbors, the neighborliness seen by v is defined as the length of their shared Voronoi edge divided by the perimeter of $\mathcal{V}_v$. A value closer to 1 indicates a stronger neighborhood relationship than that closer to 0. The Voronoi diagram can be computed very efficiently (for n points, complexity is $O(n \log n)$).

3.2. Characterization of Nodes and Edges in CST

As for STs in [3], a vector of region properties $\boldsymbol{\psi}_v$, e.g., contrast, area, perimeter, etc., is associated with each node v in CST, where the properties are specified relative to v's parent, to allow scale and rotation-in-plane invariance. Thus, images are represented by CSTs, $h = (V, E, \psi, \phi)$, where V and E are the sets of nodes and edges, and ψ and ϕ are functions that assign $\boldsymbol{\psi}_v$ to $v \in V$, and weights ϕ_e to $e \in E$. For ascendant-descendant edges, $\phi_e \in \{0, 1\}$ indicates the absence/presence of region embedding. For a directed lateral edge from node v to node v', $\phi_e \in [0, 1]$ equals the strength of their neighbor relationship seen by v.

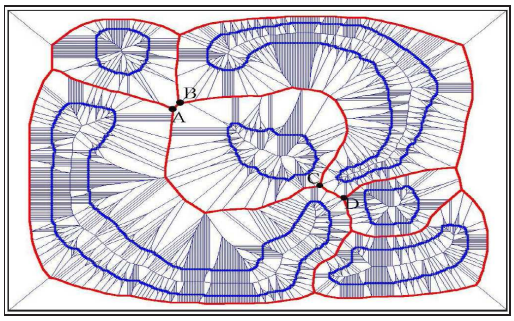


Figure 2. A generalized Voronoi polygon (red) is the union of Voronoi polygons of all pixels along the region's boundary (blue); regions are called neighbors if their generalized Voronoi polygons touch. It is correctly captured that the two relatively close regions that can "see" each other only through a narrow gap C-D are not neighbors, and that the two more distant regions are neighbors since they share the Voronoi segment AB. If the two small regions that are not neighbors come closer to each other along segment CD, the neighborliness of the two elongated regions decreases.

4. Learning Object Categories

This section presents Step 2 of our approach mentioned in Sec. 1 that discovers category occurrences in the training set $\mathbb{H} = \{h_1, \ldots h_M\}$, and then learns their models. To this end, the common subgraphs of all pairs of CSTs $(h, h') \in \mathbb{H} \times \mathbb{H}$ are found as described below.

4.1. Matching CSTs

We here present a new matching algorithm that generalizes the max-clique approaches [19, 12, 20, 16]. Our algorithm achieves robustness by pairing regions whose properties (1)+(2)+(3) (defined in Sec. 1) match, and the same holds for their neighbors, and these two conditions recursively hold for their embedded subregions. In view of the lateral connections to neighbors, our matching is context sensitive, unlike is the case in [19] which involves context free matching. Like [19], we also account for splits within regions, or, the opposite, mergers between low-contrast, contiguous regions, both of which may occur due to changes, e.g., in illumination, viewpoint, and object orientation as images are being acquired. This may cause a node in one CST to split into multiple nodes at multiple levels. These potential structural changes of CSTs across the images are addressed by considering matches of all descendants under a node, even when its direct children cannot find a good match. Fig. 3 illustrates our matching algorithm. Given two CSTs, they are first transformed into unweighted CSTs, and then matched. Before we can present the algorithm, we need the following five definitions.

Def. 1. *Unweighted CST*, $\tilde{h}$, is obtained from CST, h, by inserting between any two connected nodes $v_1, v_2 \in h$ a new node w, deleting the original edge $e = (v_1, v_2)$, and associating weight ϕ_e with w, $\psi_w = \phi_e$. $\tilde{h}$ preserves the original con-

nectivity among nodes in h, replacing the weighted edges (v_1, v_2) in h with unweighted paths (v_1, w, v_2) (Fig. 3).

Def. 2. *Saliency* r_v of node v in $\tilde{h}$ is defined as follows. If v is a node inserted according to Def. 1, then $r_v \triangleq \psi_v$, which is the weight of the edge between the corresponding regions in h. If v is the original node (i.e., region) from h then $r_v \triangleq \|\boldsymbol{\psi}_v\|_1$.

Def. 3. (*Consistency* "$\sim$") Let $\tilde{h}$ and $\tilde{h}'$ be unweighted CSTs, and nodes $v_1, v_2 \in \tilde{h}$ and $v_1', v_2' \in \tilde{h}'$. (v_1, v_2) is consistent with (v_1', v_2'), $(v_1, v_2) \sim (v_1', v_2')$, if: (i) v_1 and v_1' are exclusively regions, or containment relationship, or neighbor relationship, in the original CSTs h and h', and the same holds for v_2 and v_2'; AND (ii) there is a directed path between v_1 and v_2, and the same holds for v_1' and v_2'.

Def. 4. (*Consistent subgraph isomorphism*) Let $\tilde{h}=(\tilde{V}, \tilde{E}, \tilde{\psi})$ and $\tilde{h}'=(\tilde{V}', \tilde{E}', \tilde{\psi}')$ be unweighted CSTs, and $f: \tilde{U} \rightarrow \tilde{U}'$ be a bijection between two subsets of nodes $\tilde{U} \subseteq \tilde{V}$ and $\tilde{U}' \subseteq \tilde{V}'$ in $\tilde{h}$ and $\tilde{h}'$. f is consistent subgraph isomorphism if for any $(v_1, v_2) \in \tilde{U}$ connected with a directed path holds $(v_1, v_2) \sim (f(v_1), f(v_2))$.

Def. 5. (*Matching algorithm*) Given two unweighted CSTs, $\tilde{h}$ and $\tilde{h}'$, the matching algorithm finds a consistent subgraph isomorphism f, which maximizes their similarity measure $S_{\tilde{h}\tilde{h}'}$, defined as

$$S_{\tilde{h}\tilde{h}'} \triangleq \max_{f} \sum_{(v,v') \in f} (2\min(r_v, r_{v'}) - \max(r_v, r_{v'}) + 1). \quad (1)$$

From (1), the algorithm seeks consistent matches among both regions and their relationships whose saliencies are high, and whose cost of matching (differences in saliency) is small. To satisfy the consistency constraints (Def. 3), the algorithm matches regions with regions, and separately region relationships with corresponding relationships, while preserving the original topology of h and h'. This is done by constructing an association graph $A=(V_A, E_A, s)$, where $V_A = \tilde{V} \times \tilde{V}'$ is the set of node pairs (v, v') from $\tilde{h}$ and $\tilde{h}'$, representing all possible region matches or relationship matches. E_A is the set of edges connecting only consistent vertices $E_A = \{(a, b) : a \neq b \in V_A, a \sim b\}$. Note that while constructing A, we account for structural changes in CSTs, since E_A connects all descendants under a visited node, and thus allows their matching. s assigns weight $s_{vv'} = 2\min(r_v, r_{v'}) - \max(r_v, r_{v'}) + 1$ to each $(v, v') \in V_A$. Given A, the next theorem fully specifies the algorithm.

Theorem 1. The maximum similarity, consistent, subgraph isomorphism f between $\tilde{h}$ and $\tilde{h}'$ is equivalent to the maximum weight clique of A.

Proof: Follows directly from the construction of A. ■

To compute the maximum weight clique of A, we use the well-known replicator dynamics approach of [16]. The resulting maximally matching subgraphs $\tilde{g} \subset \tilde{h}$ and $\tilde{g}' \subset \tilde{h}'$ can be easily transformed into the corresponding weighted subgraphs $g \subset h$ and $g' \subset h'$, by replacing the previously inserted nodes with weighted, directed edges. Complexity of matching is $O((|V|+|E|)^2)$, and a MATLAB implementation takes about 10s on a 2.8GHz, 2GB RAM PC for two CSTs with approximately 50 nodes each.

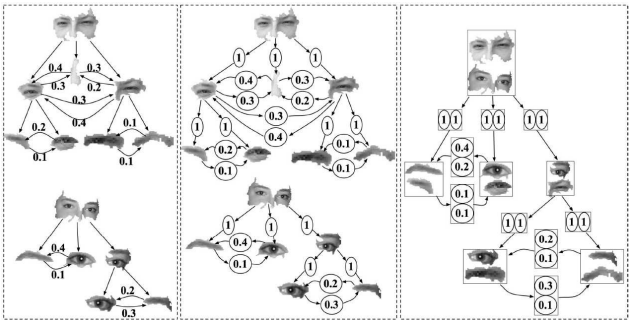


Figure 3. Matching algorithm: edges of CST are represented by new nodes in the resulting unweighted CST, and then regions and their relations that preserve the original topology are matched.

4.2. From Category Instances to their Model

To extract category occurrences from the training set $\mathbb{H}$, we match all pairs of CSTs $(h, h') \in \mathbb{H} \times \mathbb{H}$ using the algorithm of Sec. 4.1. Specifically, we match all subgraphs h_v and h_v' rooted at regions $(v, v') \in h \times h'$, and thus compute the similarity measure $S_{vv'}$, given by (1), of every region pair $(v, v') \in \mathbb{H} \times \mathbb{H}$. Since $\mathcal{S}$ measures the similarity of regions in terms of (1)+(2)+(3), the $\mathcal{S}$ values of matches belonging to a category are expected to be more similar than the $\mathcal{S}$ values of matches belonging to different categories. Therefore, categories and their occurrences can be discovered by clustering region pairs $(v, v') \in \mathbb{H} \times \mathbb{H}$ with respect to their associated $S_{vv'}$ values. The choice of a suitable clustering algorithm for this purpose depends on the degree of supervision available in training. In our experiments, we use the standard N-cuts clustering algorithm, since the total number of categories present in $\mathbb{H}$ is known. In case this information is unknown, one can use any other algorithm that does not require the number of clusters as an input parameter, but requires the level of sensitivity to inter-cluster (i.e., inter-category) differences. Each cluster of matching subgraphs of CSTs, $\mathbb{G} = \{g_1, \ldots, g_N\}$, represents a discovered category, defined by the cluster properties.

The cluster $\mathbb{G}$ may contain complete views of category instances, but it may also contain partial views, because: some parts of the category are occluded, or because some of the regions split or merge due to segmentation instabilities, causing structural changes in CSTs (e.g., due to splits or mergers of low-contrast regions under different lighting or viewing conditions). A minimum-size model, that represents the entire category and with which both entire and partial object views can be registered, is the union of graphs in $\mathbb{G}$. To find the union of $\mathbb{G}$, $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}}, \psi, \phi)$, and thus

derive the category model, we use an approach similar to that presented in [19]. The main difference is that their algorithm learns an unweighted, acyclic tree-union, whereas our graph-union is cyclic and contains weighted edges capturing the strength of both containment and neighbor relationships among nodes in $\mathbb{G}$. We construct $\mathcal{G}$ sequentially. Namely, in each iteration τ, $\mathcal{G}^{(\tau)}$ is constructed by matching $\mathcal{G}^{(\tau-1)}$ with a new graph $g \in \mathbb{G}$, and then by adding and appropriately connecting the unmatched nodes to $\mathcal{G}^{(\tau-1)}$. For matching g and $\mathcal{G}^{(\tau-1)}$, we use the algorithm of Sec. 4.1. In the resulting $\mathcal{G}^{(\tau)}$, multiple parent nodes may share the same children, as illustrated in Fig. 1.

As in [3], the vectors $\boldsymbol{\psi}_v$ associated with nodes $v \in \mathcal{G}$ are defined as $\boldsymbol{\psi}_v \triangleq \text{median}\{\boldsymbol{\psi}_{v'}\}$ of all nodes $v' \in \mathbb{G}$ that got matched with $v \in \mathcal{G}$. Similarly, for all edges e in $\mathcal{G}$, we define $\phi_e \triangleq \text{median}\{\phi_{e'}\}$ of all edges $e' \in \mathbb{G}$ that got matched with e. The result of learning are graph-union models that capture the canonical properties (1)+(2)+(3) of regions defining each category present in the training set.

5. Results

Experimental evaluation presented in this section demonstrates that the proposed CST model possesses the desired characteristics (I), (II), and (III), stated in Sec. 1, and quantifies the performance gains of modeling (1)+(2)+(3) vs. (2)+(3) and (1)+(3) for the tasks of object recognition and segmentation. We consider 14 categories from four datasets: 435 faces, 800 motorbikes, 800 airplanes, 526 cars (rear) from Caltech-101; 328 Weizmann horses; 1554 images queried from LabelMe to contain cars, trees, and buildings together; and 200 images with 715 occurrences of cows, horses, sheep, goats, camels, and deer from UIUC Hoofed Animals dataset. Caltech-101 images contain only a single, prominently featured object from the category, except for images of cars (rear) containing multiple, partially occluded cars appearing at different scales, with low contrast against textured background. The Weizmann dataset contains sideviews of walking/galloping horses of different breeds, colors and textures, with different object articulations in their natural (cluttered) habitat. LabelMe is a more difficult collection of real-world images which contain many other object categories along with the queried ones, captured under different lighting conditions, and at varying scales. The Hoofed Animals dataset presents the mentioned challenges, and has higher complexity as it contains multiple instances of multiple very similar animal categories per image, requiring high inter-category resolvability.

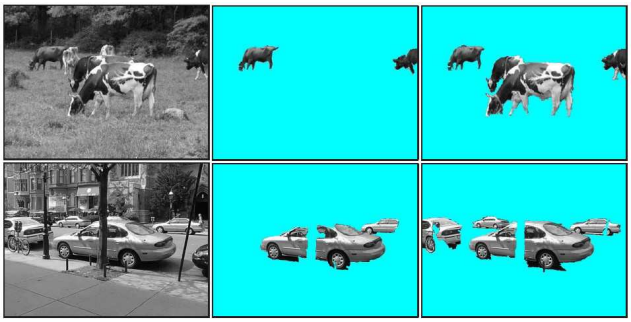
The Caltech-101 and Weizmann categories are learned one category at a time on the training set that consists of M_p randomly selected examples showing the category, and $M_n \geq 0$ images from the background category in Caltech-101 ($M = M_p + M_n$). The LabelMe and Hoofed Animals categories are all learned together by randomly selecting

Figure 4. Samples from Hoofed Animals (left) and LabelMe (right). Segmentation results of CST are overlaid on the original. Different colors denote recognized categories. CST successfully resolves small differences between the categories sheep and goats.

M images from the corresponding dataset. To recognize and segment any category occurrences in a test image, the learned category model is matched with CST of the image. The matched subtrees (i.e., detections) whose similarity measure is larger than a threshold are adjudged as detected objects. Results shown in tables and figures are obtained for the threshold that yields equal error rate. We use the following definitions of detection (DE), and segmentation (SE) errors. Let D denote the area that a detection covers in the test image, and G denote the ground-truth object area. Then, $\text{DE} \triangleq \frac{D \cap G}{D \cup G}$, and $\text{SE} \triangleq \frac{\text{XOR}(D,G)}{D \cup G}$. A detection is a false positive if $\text{DE} < 0.5$, otherwise it is a true positive (TP). Recognition is evaluated only on TP's by visual inspection.

Qualitative evaluation – Segmentation: Figs. 4–5 demonstrate high accuracy of simultaneous object detection and segmentation in images from LabelMe and Hoofed Animals datasets, using $M = 50$ training images. Each TP in the figures is correctly recognized. CSTs outperform STs in both object detection and segmentation, especially in cases of partial occlusion (e.g., cars and cows in Fig. 5), and for objects defined rather as a region spatial layout than containment (e.g., spotted cows in Fig. 5). In these cases, modeling of the region adjacency by CSTs proves advantageous. Segmentation is good even in cases when object boundaries are jagged and blurred (e.g., trees in Fig. 4), and when objects from the same category occlude each other, forming a complex region topology with low-intensity contrasts (e.g., cars in Fig. 4). Objects that are not detected, for the most part, have low intensity contrasts with the surround, and thus do not form category-characteristic subgraphs within CSTs that can be matched with the category model.

Qualitative evaluation – Model: Fig. 6 illustrates the model $\mathcal{G}$ obtained for the category horses, learned on six, randomly selected images $\mathbb{D}$ from the Weizmann dataset. Nodes v in $\mathcal{G}$, depicted as rectangles, contain regions from $\mathbb{D}$ that got matched with v during learning. As can be seen, the structure of $\mathcal{G}$ correctly captures the recursive containment and neighbor relations of regions occupied by the horses in $\mathbb{D}$. For example, nodes *head*, *neck*, and *mane* are found to be children of node *head&neck*, and they are all



(a) original image (b) STs (c) CSTs

Figure 5. CSTs outperform STs in both detection and segmentation on samples from Hoofed Animals (top) and LabelMe (bottom). Undetected image parts are masked out.

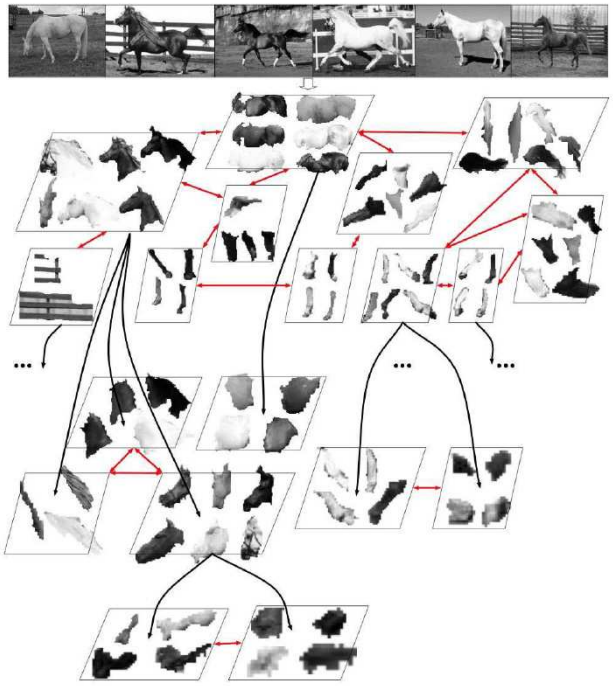


Figure 6. CST-based model of Weizmann horses.

identified as neighbors. Also, it is correct that *head&neck* and *tail* are not neighbors. Similar background regions that co-occur with horses in $\mathbb{D}$ may also be included in the model (e.g., nodes corresponding to *fence*). Typically, the percentage of background nodes out of the total number of model nodes is small (3-5%).

Quantitative evaluation: Fig. 7 (left) presents the recall-precision curves (RPC) of detection for the Caltech-101 categories using CSTs and STs. Detection performance in the presence of occlusion is tested by masking out a randomly selected rectangular area in the image, and replacing this area with a patch from the background category of Caltech-101. CST increases the area under the RPC of ST by $6.5 \pm 0.3\%$, and by $3.1 \pm 0.2\%$ in the presence of the occluding patch covering 20% of the image. Invariance to in-plane rotation is tested by randomly rotating test images, as illustrated in Fig. 1b. Performance on these rotated images is the same as the one presented in Fig. 7. Measuring the strength of neighborliness using the generalized Voronoi diagram improves performance over the case when the weights of links in CST are set to take only values 1 or 0, referred to as CST-unweight. CST increases the area under the RPC of CST-unweight by $2.3 \pm 0.3\%$. Fig. 7 (right) shows recognition accuracy of CST and ST. A small increase in M_n does not downgrade the accuracy. As M_n becomes larger, objects belonging to other categories start appearing more frequently, and thus get learned, making the training set inappropriate. Increasing M_p yields smaller recognition error. CST outperforms ST in recognition, and longer maintains high accuracy with the increase of M_n. In general, the number of nodes in the model quickly reaches saturation as new positive examples are added to the training set, and continues to very slowly increase, in part, due to chance repetitions of background regions.

Table 1 summarizes detection recall, and segmentation and recognition errors obtained for the equal error rates on LabelMe and Hoofed Animals datasets. For Hoofed Animals, CST outperforms ST in detection recall by 7.5%, segmentation by 10.7%, and recognition by 8.6%. For comparison, we obtained SE=6.5% on a relatively simple UIUC (multiscale) car dataset, using the same set-up as in [11], while their result is SE=7.9%. The other hierarchical approaches cited here use non-benchmark datasets, or report a single retrieval result for the entire Caltech-101, beyond the focus of this paper. Non-hierarchical approaches that model objects using image segments obtained at only one pre-selected scale, report the following state-of-the-art results: [17] – SE=47% for buildings, and SE=79% for cars of LabelMe; [21] – SE=7% for Weizmann horses; and [5] – SE=18.2% for Weizmann horses. In comparison with these approaches, Table 1 indicates that the CSTs yield better, or, in only a few cases, very similar performance. Regarding recognition accuracy, Fig. 7 shows that we outperform by $1.8 \pm 0.3\%$ the recognition rate of 94.6% of [5] on the four Caltech-101 categories. Other approaches cited here use a different, less demanding recognition evaluation based on classifying either the entire images or bounding boxes around objects.

The presented results demonstrate that our approach is invariant with respect to: (i) translation, in-plane rotation and object articulation, since CST itself is invariant to these changes; (ii) certain degree of scale changes, since matching is based on relative properties of regions; (iii) occlusion in the training and test sets, since graph-union registers the entire (unoccluded) category structure from partial views of occurrences in the training set, while subgraphs of visible object parts in the CST of a test image can still be matched

	LabelMe Trees	LabelMe Buildings	LabelMe Cars	Weizmann Horses	Horses	Cows	Deer	Sheep	Goats	Camels
Recall	47.6±6.9	92.6±6.9	67.6±6.9	91.9±5.2	81.2±10.3	78.4±4.2	88.1±6.9	81.2±5.3	78.2±8.6	89.9±7.2
Seg. error	41.6±7.9	34.6±13.4	32.5±8.2	7.2±2.5	15.9±5.3	17.1±4.6	11.1±8.4	24.8±7.2	20.1±8.1	11.5±5.1
Rec. error	19.7±3.8	11.6±2.9	12.9±4.8	7.9±4.1	7.8±4.2	6.5±6.2	7.7±3.4	7.8±4.1	12.2±5.4	3.2±3.9

Table 1. Detection recall, segmentation and recognition errors (in %) using the same number of training and test images as in [17, 21, 5].

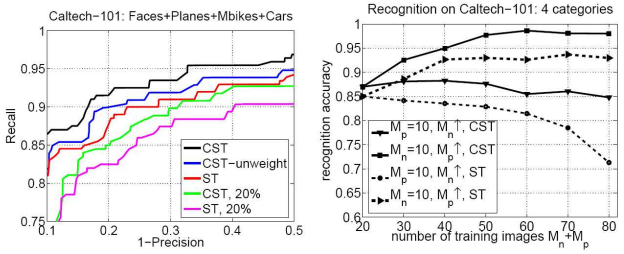


Figure 7. **(left)** Detection recall-precision curves: "CST-unweight" means that edges in CST are not weighted. 20% is the size of a rectangular occlusion w.r.t. the image size. M_p=10, M_n=10. ST is the method of [19]. **(right)** Recognition accuracy of CST and ST for the varying ratio of M_p and M_n in the training set.

with the model; (iv) minor depth rotations of objects causing their shape deformations, because structural instability of CSTs (e.g., due to region splits/mergers) is accounted for during matching; and (v) clutter, since clutter regions are not frequent and thus not learned.

6. Conclusion

We have presented what we believe is the first attempt to *jointly* learn a canonical model of an object in terms of photometric and geometric properties, and containment and neighbor relationships of its parts. As other fundamental contributions, the paper proposes: (1) A generalized Voronoi diagram over regions, which is used for finding a region's neighbors, and measuring the strength of region neighborliness; and (2) A new max-clique based algorithm for matching graphs with weighted edges and nodes.[1]

References

[1] N. Ahuja. Dot pattern processing using Voronoi neighborhoods. *PAMI*, 4(3):336–343, 1982.

[2] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE TPAMI*, 18(12):1211–1235, 1996.

[3] N. Ahuja and S. Todorovic. Learning the taxonomy and models of categories present in arbitrary images. In *ICCV*, 2007.

[4] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, pages 710–715, 2005.

[1] **Acknowledgement:** The support of the National Science Foundation under grant NSF IIS 07-43014 is gratefully acknowledged.

[5] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *ICCV*, 2007.

[6] G. Carneiro and D. Lowe. Sparse flexible models of local features. In *ECCV*, pages III: 29–43, 2006.

[7] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. In *CVPR*, volume 1, pages 943–950, 2006.

[8] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, volume 1, pages 10–17, 2005.

[9] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *CVPR*, 2007.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, 2003.

[11] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007.

[12] R. Glantz, M. Pelillo, and W. G. Kropatsch. Matching segmentation hierarchies. *Int. J. Pattern Rec. Artificial Intelligence*, 18(3):397–424, 2004.

[13] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, volume 2, pages 2145–2152, 2006.

[14] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Learning hierarchical shape models from examples. In *EMMCVPR, Springer LNCS*, volume 3757, pages 251–267, 2005.

[15] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual objects. In *CVPR*, 2007.

[16] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE TPAMI*, 21(11):1105–1120, 1999.

[17] 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*, volume 2, pages 1605–1604, 2006.

[18] A. Shokoufandeh, L. Bretzner, D. Macrini, M. Demirci, C. Jonsson, and S. Dickinson. The representation and matching of categorical shape. *Computer Vision Image Understanding*, 103(2):139–154, 2006.

[19] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, volume 1, pages 927–934, 2006.

[20] A. Torsello and E. R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recogn. Lett.*, 24(8):1089–1097, 2003.

[21] J. Winn and N. Jojic. Locus: learning object classes with unsupervised segmentation. In *ICCV*, pages 756–763, 2005.