in Proc. 11th European Conference on Computer Vision, Crete, Greece, 2010

From a Set of Shapes to Object Discovery

Nadia Payet and Sinisa Todorovic

Oregon State University,
Kelley Engineering Center, Corvallis, OR 97331, USA
payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu

Abstract. This paper presents an approach to object discovery in a given unlabeled image set, based on mining repetitive spatial configurations of image contours. Contours that similarly deform from one image to another are viewed as collaborating, or, otherwise, conflicting. This is captured by a graph over all pairs of matching contours, whose maximum a posteriori multicoloring assignment is taken to represent the shapes of discovered objects. Multicoloring is conducted by our new Coordinate Ascent Swendsen-Wang cut (CASW). CASW uses the Metropolis-Hastings (MH) reversible jumps to probabilistically sample graph edges, and color nodes. CASW extends SW cut by introducing a regularization in the posterior of multicoloring assignments that prevents the MH jumps to arrive at trivial solutions. Also, CASW seeks to learn parameters of the posterior via maximizing a lower bound of the MH acceptance rate. This speeds up multicoloring iterations, and facilitates MH jumps from local minima. On benchmark datasets, we outperform all existing approaches to unsupervised object discovery.

1 Introduction

This paper explores a long-standing question in computer vision, that of the role of shape in representing and recognizing objects from certain categories occurring in images. In psychophysics, it is widely recognized that shape is one of the most categorical object properties [1]. Nevertheless, most recognition systems rather resort to appearance features (e.g., color, textured patches). Recent work combines shape with appearance features [2,3], but the relative significance of each feature type, and their optimal fusion for recognition still remains unclear.

Toward answering this fundamental question, we here focus on the problem of discovering and segmenting instances of frequently occurring object categories in arbitrary image sets. For object discovery, we use only the geometric properties of contour layouts in the images, deliberately disregarding appearance features. In this manner, our objective is to show that shape, on its own, without photometric features, is expressive and discriminative enough to provide robust detection and segmentation of common objects (e.g, faces, bikes, giraffes, etc.) in the midst of background clutter. To this end, we develop an approach to mining repetitive spatial configurations of contours across a given set of unlabeled images. As demonstrated in this paper, our shape mining indeed results in extracting (i.e., simultaneously detecting and segmenting) semantically meaningful objects recurring in the image set.

To our knowledge, this paper presents the first approach to extracting frequently occurring object contours from a clutter of image contours without any supervision,

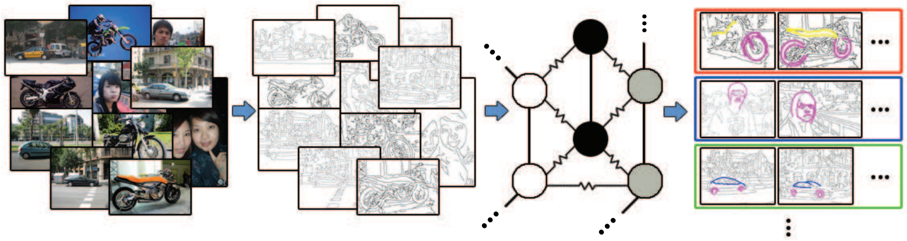


Fig. 1. Overview: Given a set of unlabeled images (left), we extract their contours (middle left), and then build a graph of pairs of matching contours. Contour pairs that similarly deform from one image to another are viewed as collaborating (straight graph edges), or conflicting (zigzag graph edges), otherwise. Such coupling of contour pairs facilitates their clustering, conducted by our new algorithm, called Coordinate Ascent Swendsen-Wang cut (CASW). The resulting clusters represent shapes of discovered objects (right). (best viewed in color)

and without any help from appearance features. Existing work that uses only shape cues for recognition in real-world images requires either a manually specified shape template [4, 5], or manually segmented training images to learn the object shape [6]. Also, all previous work on unsupervised object-category discovery exploits the photometric properties of segments [7, 8], textured patches [9], and patches along image contours [10]. In our experiments, we outperform all these appearance-based, unsupervised approaches in both object detection and segmentation on benchmark datasets.

Approach: Our approach consists of three major steps, illustrated in Fig. 1. **Step 1:** Given a set of unlabeled images, we detect their contours by the minimum-cover algorithm of [11]. Each contour is characterized as a sequence of beam-angle descriptors, which are beam-angle histograms at points sampled along the contour. Similarity between two contours is estimated by the standard dynamic time warping (DTW) of the corresponding sequences of beam-angle descriptors. **Step 2** builds a weighted graph of matching contours, aimed at facilitating the separation of background from object shapes in Step 3. We expect that there will be many similarly shaped curves, belonging to the background in the images. Since the backgrounds vary, by definition, similar background curves will most likely have different spatial layouts across the image set. In contrast, object contours (e.g., curves delineating a giraffe's neck) are more likely to preserve both shape and layout similarity in the set. Therefore, for object discovery, it is critical that we capture similar configurations of contours. To this end, in our graph, nodes correspond to pairs of matching contours, and graph edges capture spatial layouts of quadruples of contours. All graph edges can be both *positive* and *negative*, where their polarity is probabilistically sampled during clustering of image contours, performed in the next step. Positive edges support, and negative edges hinder the grouping of the corresponding contour pairs within the same cluster, if the contours *jointly* undergo similar (different) geometric transformation from one image to another. This provides stronger coupling of nodes than the common case of graph edges being only strongly or weakly "positive", and thus leads to faster convergence to more accurate object discovery. **Step 3** conducts a probabilistic, iterative multicoloring of the graph,

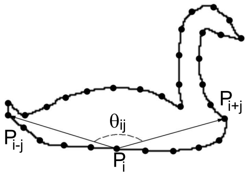


Fig. 2. BAH is a weighted histogram of beam angles θ_{ij} at contour points P_i, i=$1, 2, ...$

Contour detectors	**BAH**	BAH-U	[14]	[15]	[2]
Canny	**0.23±0.01**	0.21	0.18	0.15	0.21
	0.59±0.02	0.57	0.48	0.48	0.52
[3]	**0.32±0.03**	0.30	0.25	0.18	0.29
	0.78±0.03	0.75	0.62	0.61	0.72
gPb+ [11]	**0.37±0.02**	0.34	0.26	0.20	0.34
	0.81±0.03	0.78	0.63	0.61	0.74

Table 1. Contour matching on the ETHZ image dataset [3]. Top is $Precision$, bottom is $Recall$. The rightmost column shows matching results of Oriented Chamfer Distance [2], and other columns show DTW results. Descriptors (left to right): our BAH, unweighted BAH, Shape Context [14], and SIFT [15].

by our new algorithm, called Coordinate-Ascent Swendsen-Wang (CASW) cut. In each iteration, CASW cut probabilistically samples graph edges, and then assigns colors to the resulting groups of connected nodes. The assignments are accepted by the standard Metropolis-Hastings (MH) mechanism. To enable MH jumps to better solutions with higher posterior distributions, we estimate parameters of the posterior by maximizing a lower bound of the MH acceptance rate. After convergence, the resulting clusters represent shapes of objects discovered, and simultaneously segmented, in the image set.

Contributions: Related to ours is the image matching approach of [12]. They build a similar graph of contours extracted from only two images, and then conduct multicoloring by the standard SW cut [12, 13]. They pre-specify the polarity of graph edges, which remains fixed during multicoloring. Also, they hand-pick parameters of the posterior governing multicoloring assignments. In contrast, our graph is designed to accommodate transitive matches of many images, and we allow our graph edges to probabilistically change their polarity, in every MH iteration. We introduce a new regularization term in the posterior, which provides a better control of the probabilistic sampling of graph edges during MH jumps. Finally, we seek to *learn* parameters of our posterior via maximizing a lower bound of the MH acceptance rate. Our experiments show that this learning speeds up MH iterations, and allows jumps to solutions with higher posteriors.

Sec. 2 specifies our new shape descriptor. Sec. 3 describes how to build the graph from all pairs of image contours. Sec. 4 presents our new CASW cut for multicoloring of the graph. Sec. 5–6 present experimental evaluation, and our concluding remarks.

2 Image Representation Using Shapes and Shape Description

This section presents Step 1 of our approach. In each image, we extract relatively long, open contours using the minimum-cover algorithm of [11], referred to as gPb+ [11]. Similarity between two contours is estimated by aligning their sequences of points by the standard Dynamic Time Warping (DTW). Each contour point is characterized by our new descriptor, called weighted Beam Angle Histogram (BAH). BAH is a weighted version of the standard unweighted BAH, aimed at mitigating the uncertainty in contour extraction. BAH down-weights the interaction of distant shape parts, as they are more likely to belong to different objects in the scene.

The beam angles, θ_{ij}, at contour points P_i, $i = 1, 2, \ldots$, are subtended by lines (P_{i-j}, P_i) and (P_i, P_{i+j}), as illustrated in Fig. 2. P_{i-j} and P_{i+j} are two neighboring points equally distant by j points along the contour from P_i, $j = 1, 2, \ldots$. BAH is a weighted histogram, where the weight of angle θ_{ij} is computed as $\exp(-\kappa j)$, $j = 1, 2, \ldots$ ($\kappa = 0.01$). BAH is invariant to translation, in-plane rotation, and scale. Experimentally, we find that BAH with 12 bins gives optimal and stable results.

Table 1 compares BAH with other popular shape descriptors on the task of contour matching. We match contours from all pairs of images belonging to the same class in the ETHZ dataset [3], and select the top 5% best matches. True positives (false positives) are pixels of the matched contour that fall in (outside of) the bounding box of the target object. The ground truth is determined from pixels of the initial set of detected contours that fall inside the bounding box. For matching, we use DTW, and Oriented Chamfer Distance [2]. Tab. 1 shows that our BAH descriptor gives the best performance with all contour detectors, and the highest accuracy with gPb+ [11]. Also, DTW with our BAH outperforms Oriented Chamfer Distance.

3 Constructing the Graph of Pairs of Image Contours

This section presents Step 2 which constructs a weighted graph, $G = (V, E, \rho)$, from contours extracted from the image set. Nodes of G represent candidate matches of contours, $(u, u')\in V$, where u and u' belong to different images. Similarity of two contours is estimated by DTW. We keep only the best 5% of contour matches as nodes of G.

Edges of G, $e = ((u, u'), (v, v')) \in E$, capture spatial relations of corresponding image contours. If contours u and v in image 1, and their matches u' and v' in image 2 have similar spatial layout, then they are less likely to belong to the background clutter. All such contour pairs will have a high probability to become positively coupled in G. Otherwise, matches (u, u') and (v, v') will have a high probability to become negatively coupled in G, so that CASW could place them in different clusters. This probabilistic coupling of nodes in G is encoded by edge weights, ρ_e, defined as the likelihood $\rho_e^+ \propto \exp(-w_\delta^+ \delta_e)$, given the positive polarity of e, and $\rho_e^- \propto \exp(-w_\delta^- (1-\delta_e))$, given the negative polarity of e. w_δ^+ and w_δ^- are the parameters of the exponential distribution, and $\delta_e \in [0, 1]$ measures a difference in spatial layouts of u and v in image 1, and their matches u' and v' in image 2. We specify δ_e for the following two cases. In Cases 1 and 2, there are at least two contours that lie in the same image. This allows establishing geometric transforms between $((u, u'), (v, v'))$. Note that this would be impossible, in a more general case, where $((u, u'), (v, v'))$ come from four distinct images.

Case 1: (u, u') and (v, v') come from *two* images, where u and v are in image 1, and u' and v' are in image 2, as illustrated in Fig. 3a. We estimate δ_e in terms of affine homographies between the matching contours, denoted as $H_{uu'}$, and $H_{vv'}$. Note that if u, v in image 1 preserve that same spatial layout in image 2, then $H_{vv'}{=}H_{vu}H_{uu'}$. Since the estimation of H_{vu} between arbitrary, non-similar contours u and v in image 1 is difficult, we use the following strategy. From the DTW alignment of points along u and u', we estimate their affine homography $H_{uu'}$. Similarly, for v and v', we estimate $H_{vv'}$. Then, we project u' to image 1, as $u''{=}H_{vv'}u'$, and, similarly, project v' to image 1 as $v''{=}H_{uu'}v'$ (Fig. 3a right). Next, in image 1, we measure distances between

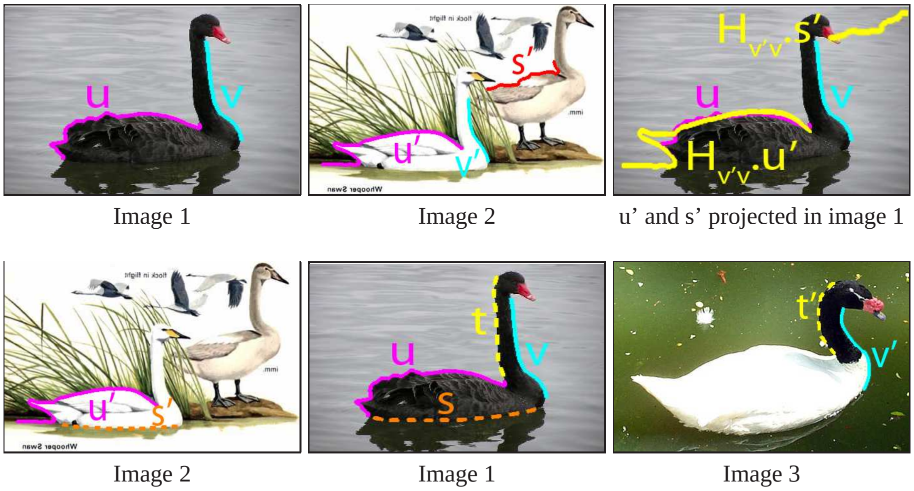


Fig. 3. (a) Case 1: Estimating $\delta_{(u,u',v,v')}$ when contours u and v are in image 1, and their matches u' and v' are in image 2. We use the affine-homography projection of u' and v' to image 1, $u'' = H_{vv'}u'$ and $v'' = H_{uu'}v'$, and compute δ as the average distance between u and u'', and v and v''. As can be seen, pairs (u, s') and (v, v') do not have similar layouts in image 1 and image 2. (b) Case 2: Estimating $\delta_{(u,u',v,v')}$ when u and v are in image 1, and their matches u' and v' are in image 2 and image 3. We use multiple affine-homography projections of u' and v' to image 1 via auxiliary, context contours s' and t' in a vicinity of u' and v'.

corresponding points of u and u'', where the point correspondence is obtained from DTW of u and u'. Similarly, we measure distances between corresponding points of v and v''. δ_e is defined as the average point distance between u and u'', and v and v''.

Case 2: (u, u') and (v, v') come from *three* images, where u and v belong to image 1, u' is in image 2, and v' is in image 3. In this case, we can neither use $H_{vv'}$ to project u' from image 2 to image 1, nor $H_{uu'}$ to project v' from image 3 to image 1. Instead, we resort to context provided by auxiliary contours s' in a vicinity of u', and auxiliary contours t' in a vicinity of v'. For every neighbor s' of u' in image 2, we find its best DTW match s in image 1, and compute homography $H_{ss'}$. Similarly, for every neighbor t' of v' in image 3, we find its best DTW match t in image 1, and compute homography $H_{tt'}$. Then, we use all these homographies to project u' to image 1, multiple times, as $u''_s{=}H_{ss'}u'$, and, similarly, project v'' to image 1, multiple times, as $v''_t{=}H_{tt'}v'$. Next, as in Case 1, we measure distances between corresponding points of all u and u''_s pairs, and all v and v''_t pairs. δ_e is defined as the average point distance.

4 Coordinate-Ascent Swendsen-Wang Cut

This section presents Step 3. Given the graph $G = (V, E, \rho)$, specified in the previous section, our goal is to perform multicoloring of G, which will partition G into two subgraphs. One subgraph will represent a composite cluster of nodes, consisting of a number of connected components (CCPs), receiving distinct colors, as illustrated in

Fig. 4. This composite cluster contains contours of the discovered object categories. Nodes outside of the composite cluster are interpreted as the background. All edges $e \in E$ can be negative and positive. A negative edge indicates that the nodes are conflicting, and thus should not be assigned the same color. A positive edge indicates that the nodes are collaborative, and thus should be favored to get the same color. If nodes are connected by positive edges, they form a CCP, and receive the same color (Fig. 4). A CCP cannot contain a negative edge. CCPs connected by negative edges form a composite cluster. The amount of conflict and collaboration between two nodes is defined by the likelihood ρ, defined in Sec. 3.

For multicoloring of G, we formulate a new Coordinate Ascent Swendsen-Wang cut (CASW) that uses the iterative Metropolis-Hastings algorithm. CASW iterates the following three steps: (1) Sample a composite cluster from G, by probabilistically cutting and sampling positive and negative edges between nodes of G. This results in splitting and merging nodes into a new configuration of CCPs. (2) Assign new colors to the resulting CCPs within the selected composite cluster, and use the Metropolis-Hastings (MH) algorithm to estimate whether to accept this new multicoloring assignment of G, or to keep the previous state. (3) If the new state is accepted, go to step (1); otherwise, it the algorithm converged, re-estimate parameters of the pdf's controlling the MH iterations, and go to step (1), until the pdf re-estimation does not affect convergence.

CASW is characterized by large MH moves, involving many strongly-coupled graph nodes. This typically helps avoid local minima, and allows fast convergence, unlike other related MCMC methods. In comparison with [12], our three key contributions include: (a) the on-line learning of parameters of pdf's governing MH jumps; (b) enforcing stronger node coupling by allowing the polarity of edges to be dynamically estimated during the MH iterations; and (c) regularizing the posterior of multicoloring assignments to help MH jumps escape from trivial solutions. In the following, we present our Bayesian formulation of CASW, inference, and learning.

Bayesian Formulation: Multi-coloring of G amounts to associating labels l_i to nodes in V, $i{=}1,\dots,|V|$, where $l_i \in \{0,1,\dots,K\}$. K denotes the total number of target objects, which is a priori unknown, and $(K+1)$th label is the background. The multicoloring solution can be formalized as $\mathcal{M}{=}(K, \{l_i\}_{i=1,\dots,|V|})$. To find $\mathcal{M}$, we maximize the posterior distribution $p(\mathcal{M}|G)$, as

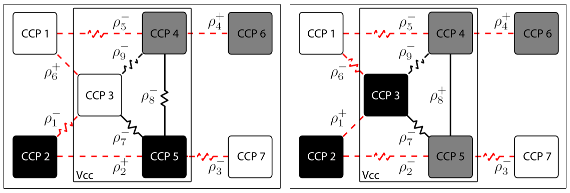
$$\mathcal{M}^* = \arg\max_{\mathcal{M}} p(\mathcal{M}|G) = \arg\max_{\mathcal{M}} p(\mathcal{M})p(G|\mathcal{M}). \quad (1)$$

Let N denote the number of nodes that are labeled as background $l_i = 0$. Also, let binary functions $\mathbb{1}_{l_i \neq l_j}$ and $\mathbb{1}_{l_i = l_j}$ indicate whether node labels l_i and l_j are different, and the same. Then, we define the prior $p(\mathcal{M})$ and likelihood $p(G|\mathcal{M})$ as

$$p(\mathcal{M}) \propto e^{-w_K K} e^{-w_N N} \ , \quad (2)$$

$$p(G|\mathcal{M}) \propto \textstyle\prod_{e\in\mathbb{E}^+} \rho_e^+ \prod_{e\in\mathbb{E}^-} \rho_e^- \prod_{e\in\mathbb{E}^0} (1-\rho_e^+) \mathbb{1}_{l_i \neq l_j} \cdot (1-\rho_e^-) \mathbb{1}_{l_i = l_j} \ , \quad (3)$$

where $p(\mathcal{M})$ penalizes large K and N. w_K and w_N are the parameters of the exponential distribution. $\mathbb{E}^+$ and $\mathbb{E}^-$ denote positive and negative edges present in the composite cluster, and $\mathbb{E}^0$ denotes edges that are probabilistically cut (i.e., not present in the solution). Our $p(G|\mathcal{M})$, defined in (3), differs from the likelihood defined in [12]. In [12],



(a) State A

(b) State B

$$p(G|\mathcal{M}{=}A){=}(\rho_7^-\,\rho_8^-)(1{-}\rho_9^+)$$

$$p(G|\mathcal{M}{=}B){=}(\rho_7^-\,\rho_8^+)(1{-}\rho_9^+)$$

$$q(V_{cc}|A){=}(1{-}\rho_1^-)(1-\rho_2^+)(1{-}\rho_3^-)$$
$$(1{-}\rho_4^+)(1{-}\rho_5^-)(1{-}\rho_6^+)$$

$$q(V_{cc}|B){=}(1{-}\rho_1^+)(1-\rho_2^-)(1{-}\rho_3^-)$$
$$(1{-}\rho_4^+)(1{-}\rho_5^-)(1{-}\rho_6^-)$$

(c) Probabilities for this example

Fig. 4. (a) In state A, probabilistically sampled positive (straight bold) and negative (zigzag bold) edges define composite cluster V_{cc}=$\{CCP3, CCP4, CCP5\}$ (cut edges are dashed). The cut is a set of edges (red) that have not been probabilistically sampled, which would otherwise connect Vcc to external CCPs. (b) The coloring of CCPs within V_{cc} is randomly changed, resulting in new state B. This also changes the type of edges $\rho_1, \rho_2, \rho_6, \rho_8$, since the positive (negative) edge may link only two CCPs with the same (different) label(s). (c) Probabilities in states A and B.

nodes can be connected by only one type of edges. They pre-select a threshold on edge weights, which splits the edges into positive and negative, and thus define the likelihood as $p(G|\mathcal{M}) \propto \prod_{e\in\mathbb{E}^+} \rho_e^+ \prod_{e\in\mathbb{E}^-} \rho_e^-$. Since we allow both types of edges to connect every pair of nodes, where the right edge type gets probabilistically sampled in every MH iteration, we enforce a stronger coupling of nodes. As shown in Sec. 5, this advanced feature of our approach yields faster convergence and better clustering performance. This is because our formulation maximizes the likelihood $p(G|\mathcal{M})$ when every two nodes with the same label are (i) connected by a strong positive edge ($e \in E^+$, and ρ_e^+ large), or (ii) remain unconnected, but the likelihood that these nodes should not have the same label is very low ($e \in E^0$, and ρ_e^- small). Similarly, our likelihood $p(G|\mathcal{M})$ is maximized when every two nodes with different labels are (i) connected by a strong negative edge ($e \in \mathbb{E}^-$, and ρ_e^- large), or (ii) remain unconnected, but the likelihood that these nodes should have the same label is very low ($e \in E^0$, and ρ_e^+ small).

Inference: We here explain the aforementioned iterative steps (1) and (2) of our CASW cut. Fig. 4 shows an illustrative example. In step (1), edges of G are probabilistically sampled. If two nodes have the same label, their positive edge is sampled, with likelihood ρ_e^+. Otherwise, if the nodes have different labels, their negative edge is sampled, with likelihood ρ_e^-. This re-connects all nodes into new connected components (CCPs). The negative edges that are sampled will connect CCPs into a number of composite clusters, denoted by V_{cc}. This configuration is referred to state A. In step (2), we choose at random one composite cluster, V_{cc}, and probabilistically reassign new colors to the CCPs within V_{cc}, resulting in a new state B. Note that all nodes within one CCP receive the same label, which allows large moves in the search space.

The CASW accepts the new state B as follows. Let $q(A \rightarrow B)$ be the proposal probability for moving from state A to B, and let $q(B \rightarrow A)$ denote the reverse. The acceptance rate, $\alpha(A{\rightarrow}B)$, of the move from A to B is defined as

$$\alpha(A \rightarrow B) = \min\left(1, \frac{q(B \rightarrow A)p(\mathcal{M} = \mathcal{B}|\mathcal{G})}{q(A \rightarrow B)p(\mathcal{M} = \mathcal{A}|\mathcal{G})}\right). \quad (4)$$

Note that complexity of each move is relatively low, since computing $\frac{q(B\rightarrow A)}{q(A\rightarrow B)}$ involves only those edges that are probabilistically cut around V_{cc} in states A and B — not all edges. Also, $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})}$ accounts only for the recolored CCPs in V_{cc} — not the entire graph G. Below, we derive $\frac{q(B\rightarrow A)}{q(A\rightarrow B)}$ and $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})}$, and present a toy example (Fig. 4).

$q(A \rightarrow B)$ is defined as a product of two probabilities: (i) the probability of generating V_{cc} in state A, $q(V_{cc}|A)$; and (ii) the probability of recoloring the CCPs within V_{cc} in state B, where V_{cc} is obtained in state A, $q(B(V_{cc})|V_{cc}, A)$. Thus, we have $\frac{q(B\rightarrow A)}{q(A\rightarrow B)}=\frac{q(V_{cc}|B)q(A(V_{cc})|V_{cc},B)}{q(V_{cc}|A)q(B(V_{cc})|V_{cc},A)}$. The ratio $\frac{q(A(V_{cc})|V_{cc},B)}{q(B(V_{cc})|V_{cc},A)}$ can be canceled out, because the CCPs within V_{cc} are assigned colors under the uniform distribution. Let Cut_A^+ and Cut_A^- (Cut_B^+ and Cut_B^-) denote positive and negative edges which are probabilistically "cut" around V_{cc} in state A (state B). Since the probabilities of cutting the positive and negative edges are $(1-\rho_e^+)$ and $(1-\rho_e^-)$, we have

$$\frac{q(B\rightarrow A)}{q(A\rightarrow B)} = \frac{q(V_{cc}|B)}{q(V_{cc}|A)} = \frac{\prod_{e\in \text{Cut}_B^+}(1-\rho_e^+)\prod_{e\in \text{Cut}_B^-}(1-\rho_e^-)}{\prod_{e\in \text{Cut}_A^+}(1-\rho_e^+)\prod_{e\in \text{Cut}_A^-}(1-\rho_e^-)} \ . \tag{5}$$

For the example shown in Figure 4, we compute $\frac{q(B\rightarrow A)}{q(A\rightarrow B)} = \frac{(1-\rho_1^+)(1-\rho_2^-)(1-\rho_6^-)}{(1-\rho_1^-)(1-\rho_2^+)(1-\rho_6^+)}$.

Also, $\frac{p(\mathcal{M}=\mathcal{B}|\mathcal{G})}{p(\mathcal{M}=\mathcal{A}|\mathcal{G})} = \frac{p(\mathcal{M}=\mathcal{B})p(G|\mathcal{M}=B)}{p(\mathcal{M}=\mathcal{A})p(G|\mathcal{M}=A)}$ can be efficiently computed. $p(\mathcal{M} = \mathcal{B})$ can be directly computed from the new coloring in state B, and $\frac{p(G|\mathcal{M}=B)}{p(G|\mathcal{M}=A)}$ depends only on those edges that have changed their polarity. For the example shown in Fig.4, we compute $\frac{p(\mathcal{M}=B|G)}{p(\mathcal{M}=A|G)}=\frac{\rho_8^+}{\rho_8^-}$.

When $\alpha(A \rightarrow B)$ has a low value, and new state B cannot be accepted by MH, CD-SW remains in state A. In the next iteration, CD-SW either probabilistically selects a different V_{cc}, or proposes a different coloring scheme for the same V_{cc}.

Learning: Our Bayesian model is characterized by a number of parameters that we seek to learn from data. We specify that learning occurs at a standstill moment when MH stops accepting new states (we wait for 100 iterations). In that moment, the previous state A is likely to have the largest pdf in this part of the search space. By learning new model parameters, our goal is to allow for larger MH moves, and thus facilitate exploring other parts of the search space characterized by higher posterior distributions $p(\mathcal{M}|G)$. Since the moves are controlled by $\alpha(A\rightarrow B)$, given by (4), we learn the parameters by maximizing a lower bound of $\alpha(A\rightarrow B)$. If this learning still does not result in accepting new states, we conclude that the algorithm has converged.

From (3) and (4), and the definitions of edge likelihoods ρ_e^+ and ρ_e^- given in Sec. 3, we derive a lower bound of $\log(\alpha(A \rightarrow B))$ as

$$\log(\alpha(A \rightarrow B)) \geq \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{w} \ , \tag{6}$$

where $\boldsymbol{w} = \left[w_K, w_N, w_\delta^+, w_\delta^-\right]^{\mathrm{T}}$, and $\boldsymbol{\phi} = [\phi_1, \phi_2, \phi_3, \phi_4]^{\mathrm{T}}$ is the vector of observed features, defined as $\phi_1 = K_A-K_B$, $\phi_2 = N_A-N_B$, $\phi_3 = \sum_{e\in\mathbb{E}_A^+}\delta_e-\sum_{e\in\tilde{\mathbb{E}}_B^+}\delta_e$, and $\phi_4 = \sum_{e\in\mathbb{E}_A^-}(1-\delta_e)-\sum_{e\in\tilde{\mathbb{E}}_B^-}(1-\delta_e)$. $\tilde{\mathbb{E}}_B^+$ denotes all edges in state B whose likelihood is $\rho+$, $\tilde{\mathbb{E}}_B^+ = \mathbb{E}_B^+ \cup \text{Cut}_B^+ \cup \mathbb{E}_B^{0-}$, and $\tilde{\mathbb{E}}_B^-$ denotes all edges in state B whose likelihood is $\rho-$, $\tilde{\mathbb{E}}_B^- = \mathbb{E}_B^- \cup \mathrm{Cut}_B^- \cup \mathbb{E}_B^{0+}$. From (6), we formulate learning as the following linear program

$$\max_{\boldsymbol{w}} \; \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{w} \; , \quad \text{s.t.} \quad \|\boldsymbol{w}\|_2 = 1 \; , \tag{7}$$

which has a closed-form solution [16], $\boldsymbol{w} = \frac{1}{\|\boldsymbol{\phi}_+\|} \boldsymbol{\phi}_+$, where $(\boldsymbol{\phi})_+ = \max(0, \boldsymbol{\phi})$.

5 Results

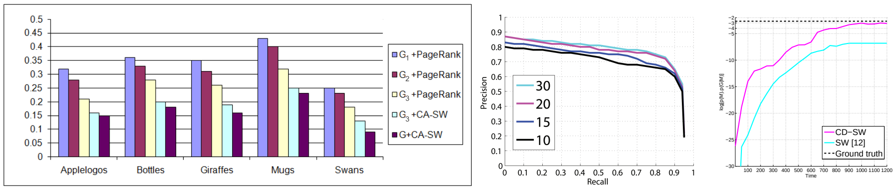
Given a set of images, we perform object discovery in two stages, as in [9, 10, 17]. We first coarsely cluster images based on their contours using CASW cut, and then again use CASW to cluster contours from only those images that belong to the same coarse cluster. The first stage serves to discover different object categories in the image set, whereas the second, fine-resolution stage serves to separate object contours from background clutter, and also extract characteristic parts of each discovered object category.

We use the following benchmark datasets: Caltech-101 [18], ETHZ [3], LabelMe [19], and Weizmann Horses [20]. In the experiments on Caltech-101, we use all Caltech images showing the same categories as those used in [9]. Evaluation on ETHZ and Weizmann Horses uses the entire datasets. For LabelMe, we keep the 15 first images retrieved by keywords *car side*, *car rear*, *face*, *airplane* and *motorbike*. ETHZ and LabelMe increase complexity over Caltech-101, since their images contain multiple object instances, which may: (a) appear at different resolutions, (b) have low contrasts with textured background, and (c) be partially occluded. The Weizmann Horses are suitable to evaluate performance on articulated, non-rigid objects.

We study two settings S1 and S2. In S1, we use only ETHZ to generate the input image set. The set consists of positive and negative examples, where positive images show a unique category, and negative ones show objects from other categories in ETHZ. In S2, the image set contains examples of all object categories from the considered dataset. S1 is used for evaluating particular contributions of our approach, and S2 is used for evaluating our overall performance.

In the first stage of object discovery, CASW finds clusters of images. This is evaluated by *purity*. Purity measures the extent to which a cluster contains images of a single dominant object category. When running CASW in the second stage, on each of these image clusters, we use *Bounding Box Hit Rate* (BBHR) to verify whether contours detected by CASW fall within the true foreground regions. The ground truth is defined as all pixels of the extracted image contours that fall in the bounding boxes or segments of target objects. A contour detected by CASW is counted as "hit" whenever the contour covers 50% or more of the ground-truth pixels. Since we discard contours that are less than 50 pixels, this means that at least 25 ground-truth pixels need to be detected within the bounding box. Our accuracy in the second clustering stage depends on the initial set of pairs of matching contours (i.e., nodes of graph G) input to CASW. This is evaluated by plotting the ROC curve, parameterized by a threshold on the minimum DTW similarity between pairs of matching contours which are included in G.

Evaluation in S1: We present three experiments in S1. *Experiment 1 in S1:* We evaluate the merit of: (a) using pairs of contours as nodes of G, and (b) accounting for



(a) (b) (c)

Fig. 5. Evaluation in S1 on the ETHZ dataset. (a): We evaluate five distinct formulations of object discovery, explained in the text, by computing False Positive Rate (FPR) at Bounding Box Hit Rate BBHR=0.5. Our approach G+CASW gives the best performance. (b): $Precision$ and $Recall$ as a function of the number of positive examples in the input image set. Performance increases with more positive examples, until about 20 positive images. (c): Evolution of $\log(p(\mathcal{M})p(G|\mathcal{M})$ estimated by our CASW (magenta), and standard SW [12] (cyan) on all positive examples of class *Giraffes*, and the same number of negative examples from ETHZ.

spatial configuration of contours as edge weights of G, against the more common use of individual contours as graph nodes, and contour similarities as edge weights. To this end, we build three weighted graphs G_1, G_2 and G_3 of contours extracted only from all positive examples of a single object category in the ETHZ dataset (i.e., the set of negative examples is empty). Nodes of G_1 are individual contours, edges connect candidate matches (u, u'), and edge weights $s_{uu'}$ represent the DTW similarity of contours u and u'. In G_2 and G_3, nodes are instead pairs of contours (u, u'). In G_2, each edge $((u, u'), (v, v'))$ receives weight $(s_{uu'}+s_{vv'})/2$. In G_3, edges can only be positive and receive weights ρ_e^+, defined in Sec. 3. For all three graphs, we apply the standard PageRank algorithm, also used in [9, 10, 17], to identify the most relevant contours, which are then interpreted as object contours. False Positive Rate (FPR) is computed for BBHR = 0.5, and averaged across all categories in the ETHZ dataset. Fig. 5(a) shows that G_2+PageRank decreases the FPR of G_1+PageRank by 3.2%. However, G_2+PageRank still yields a relatively high value of FPR, which suggests that accounting only for shape similarity and ignoring the spatial layout of contours may not be sufficient to handle the very difficult problem of object discovery. Using G_3+PageRank significantly decreases FPR, which motivates our approach. We also run our CASW on graph G_3, and on G, specified in Sec. 3. In comparison with G_3+CASW, our approach G+CASW additionally allows the negative polarity of graph edges. Fig. 5(a) shows that G_3+CASW outperforms G_3+PageRank, and that G+CASW gives the best results.

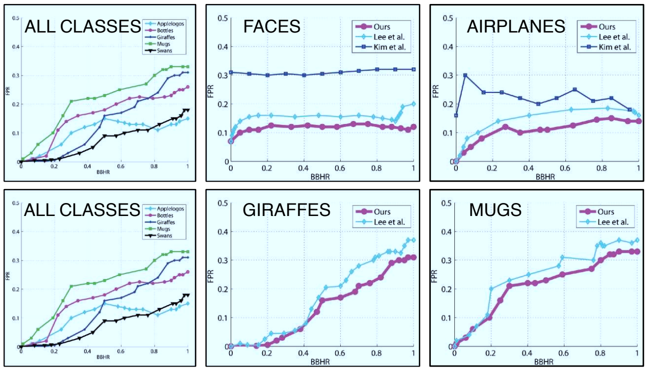
Experiment 2 in S1: We test performance in object detection as a function of the number of positive examples in the input image set. The total number of images $M = 32$ is set to the number of images of the "smallest" class in the ETHZ dataset. In Fig.5(b), we plot the ROC curves when the number of positive images increases, while the number of negative ones proportionally decreases. As expected, performance improves with the increase of positive examples, until reaching a certain number (on average about 20 for the ETHZ dataset).

Experiment 3 in S1: Finally, we test our learning of pdf parameters. Fig.5(c) shows the evolution of $\log(p(\mathcal{M})p(G|\mathcal{M}))$ in the first stage of object discovery in the image set consisting of all positive examples of class *Giraffes*, and the same number of neg-

Caltech categories	**Our method**	[10]	[9]	[17]
A,C,F,M	**98.62±0.51**	98.03	98.55	88.82
A,C,F,M,W	**97.57±0.46**	96.92	97.30	N/A
A,C,F,M,W,K	**97.13±0.42**	96.15	95.42	N/A

ETHZ categories	**Our method**	[10]
A,B,G,M,S (bbox)	**96.16±0.41**	95.85
A,B,G,M,S (expanded)	**87.35±0.37**	76.47
A,B,G,M,S (entire image)	**85.49±0.33**	N/A

Table 2. Mean purity of category discovery for Caltech-101 (A:Airplanes, C: Cars, F: Faces, M: Motorbikes, W: Watches, K: Ketches), and ETHZ dataset (A:Applelogos, B: Bottles, G: Giraffes, M: Mugs, S: Swans).



	CASW	[9]	[10]
A	**0.11±0.01**	0.21	0.17
F	**0.12±0.01**	0.30	0.15
K	**0.06±0.003**	0.19	0.08
M	**0.04±0.002**	0.11	0.07
W	**0.02±0.003**	0.08	0.03

	CASW	[9]	[10]
A	**0.15±0.02**	N/A	0.18
B	**0.18±0.01**	N/A	0.20
G	**0.16±0.01**	0.32	0.18
M	**0.23±0.04**	N/A	0.27
S	**0.09±0.002**	N/A	0.11

Fig. 6. Bounding Box Hit Rates (BBHR) vs False Positive Rates (FPR). Top is Caltech-101, bottom is ETHZ. Left column is our CASW on all classes, and middle and right columns show a comparison with [9, 10] on a specific class (lower curves are better). The tables show FPR at BBHR=0.5. Caltech-101: A: Airplanes, F: Faces, K: Ketches, M: Motorbikes, W: Watches. ETHZ: A: Applelogs, B: Bottles, G: Giraffes, M: Mugs, S: Swans. (best viewed in color)

ative examples showing other object categories from the ETHZ dataset. We compare our CASW with the standard SW of [12], where the pdf parameters are not learned, but pre-specified. Since these parameters are unknown, to compute both the ground-truth value and the value produced by [12] of $\log(p(\mathcal{M})p(G|\mathcal{M}))$, we use the pdf parameters learned by our approach after CASW converged. As CASW and SW make progress through iterative clustering of the images, Fig. 5(c) shows that CASW yields a steeper increase in $\log(p(\mathcal{M})p(G|\mathcal{M}))$ to higher values, closer to the ground-truth. Notice that CASW avoids local minima and converges after only few iterations.

Evaluation in S2: We evaluate the first and second stages of object discovery in S2. *First Stage in S2:* We build a graph whose nodes represent entire images. Edges between images in the graph are characterized by weights, defined as an average of DTW similarities of contour matches from the corresponding pair of images. A similar characterization of graph edges is used in [9, 10]. For object discovery, we apply CASW to the graph, resulting in image clusters. Each cluster is taken to consist of images showing a unique object category. Unlike [9, 10], we do not have to specify the number of categories present in the image set, as an input parameter, since it is automatically inferred by CASW. Evaluation is done on Caltech-101 and the ETHZ dataset. Table 2 shows that our mean purity is superior to that of [9, 10, 17]. On Caltech-101, CASW successively finds $K = 4, 5, 6$ clusters of images, as we gradually increase the true number of categories from 4 to 6. This demonstrates that we are able to automatically find the number of categories present, with no supervision. On ETHZ, CASW again correctly finds $K = 5$ categories. As in [10], we evaluate purity when similarity between the images (i.e., weights of edges in the graph) is estimated based on contours falling within: (a) the bounding boxes of target objects, (b) twice the size of the original bounding boxes (called expanded in Table 2), and (c) the entire images. On ETHZ, CASW does not suffer a major performance degradation when moving from the bounding boxes, to the challenging case of using all contours from the entire images. Overall, our purity rates are high, which enables accurate clustering of contours in the second stage.

Second Stage in S2: We use contours from all images grouped within one cluster in the first stage to build our graph G, and then conduct CASW. This is repeated for all image clusters. The clustering of contours by CASW amounts to foreground detection, since the identified contour clusters are taken to represent parts of the discovered object category. We evaluate BBHR and FPR on Caltech-101, ETHZ, LabelMe, and Weizmann Horses. Fig.6 shows that our BBHR and FPR values are higher than those of [9, 10] on the Caltech and ETHZ. CASW finds $K = 1$ for *Airplanes, Cars Rear, Faces, Ketches, Watches* in Caltech-101, *Apples, Bottles, Mugs* in ETHZ, and *Car rear, Face, Airplane* in LabelMe. These objects do not have articulated parts that move independently, hence, only one contour cluster is found. On the other hand, it finds $K = 2$ for *Giraffes, Swans* in ETHZ, *Cars side, Motorbikes* in Caltech and LabelMe, and $K = 3$ for Weizmann Horses. In Fig.7, we highlight contours from different clusters with distinct colors. Fig.7 demonstrates that CASW is capable not only to discover foreground objects, but also to detect their characteristic parts, e.g., wheels and roof for *Cars side*, wheels and seat for *Motorbikes*, head and legs for *Giraffes*, etc. The plot in Fig.7 evaluates our object detection on LabelMe and Weizmann Horses. Detection accuracy is estimated as the standard ratio of intersection over union of ground-truth and detection bounding boxes, $(BB_{gt} \cap BB_d)/(BB_{gt} \cup BB_d)$, where BB_d is the smallest bounding box that encloses detected contours in the image. The average detection accuracy for each category is: [Face(F): 0.52, Airplane(A): 0.45, Motorbike(M): 0.42, Car Rear(C): 0.34], whereas [10] achieves only [(F): 0.48, (A): 0.43, (M): 0.38, (C): 0.31]. For Weizmann Horses, we obtain $Precision$ and $Recall$ of 84.9%±0.68% and 82.4%±0.51%, whereas [8] achieves only 81.5% and 78.6%.

Remark: The probability of contour patterns that repeat in the background increases with the number of images. On large datasets, CASW is likely to extract clusters of those background patterns. However, the number of contours in these clusters is relatively small, as compared to clusters that contain true object contours, because the frequency of such patterns is, by definition, smaller than that of foreground objects. Therefore, these spurious clusters can be easily identified, and interpreted as background. For example, in setting S1, when the input image set consists of only positive, 100 images of Weizmann Horses, we obtain $K = 3$ very large clusters (Fig.7), and 9 additional clusters with only 5 to 10 background contours.

Implementation. The C-implementation of our CASW runs in less than 2 minutes on any dataset of less than 100 images, on a 2.40GHz PC with 3.48GB RAM.

6 Conclusion

We have shown that shape alone is sufficiently discriminative and expressive to provide robust and efficient object discovery in unlabeled images, without using any photometric features. This is done by clustering image contours based on their intrinsic geometric properties, and spatial layouts. We have also made contributions to the popular research topic in vision, that of probabilistic multicoloring of a graph, including: (a) the on-line learning of pdf parameters governing multicoloring assignments; (b) enforcing stronger positive and negative coupling nodes in the graph, by allowing the polarity of graph edges to dynamically vary during the Metropolis-Hestings (MH) jumps; and (c) regularizing the posterior of multicoloring assignments to help MH jumps escape from trivial solutions. These extensions lead to faster convergence to higher values of the graph's posterior distribution than the well-known SW cut.

References

1. Biederman, I.: Surface versus edge-based determinants of visual recognition. Cognitive Psychology **20** (1988) 38–64
2. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. PAMI **30** (2008) 1270–81
3. Ferrari, V., Tuytelaars, T., Gool, L.V.: Object detection by contour segment networks. In: ECCV. (2006) 14–28
4. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: ECCV (2). (2008) 774–787
5. Kokkinos, I., Yuille, A.L.: HOP: Hierarchical object parsing. In: CVPR. (2009)
6. Bai, X., Wang, X., Liu, W., Latecki, L.J., Tu, Z.: Active skeleton for non-rigid object detection. In: ICCV. (2009)
7. Russell, B., Freeman, W., Efros, A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR. (2006)
8. Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. IEEE TPAMI **30** (2008) 1–17
9. Kim, G., Faloutsos, C., Hebert, M.: Unsupervised modeling of object categories using link analysis techniques. In: CVPR. (2008)
10. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: CVPR. (2009)
11. Felzenszwalb, P., McAllester, D.: A min-cover approach for finding salient curves. In: CVPR POCV. (2006)
12. Lin, L., Zeng, K., Liu, X., Zhu, S.C.: Layered graph matching by composite cluster sampling with collaborative and competitive interactions. In: CVPR. (2009)
13. Barbu, A., Zhu, S.C.: Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. IEEE TPAMI **27** (2005) 1239–1253
14. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI **24** (2002) 509–522
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
16. Chong, E., S.H., Z.: An introduction to optimization. Wiley-Interscience (2001)
17. Lee, Y.J., Grauman, K.: Foreground focus: Unsupervised learning from partially matching images. In: BMVC. (2008)

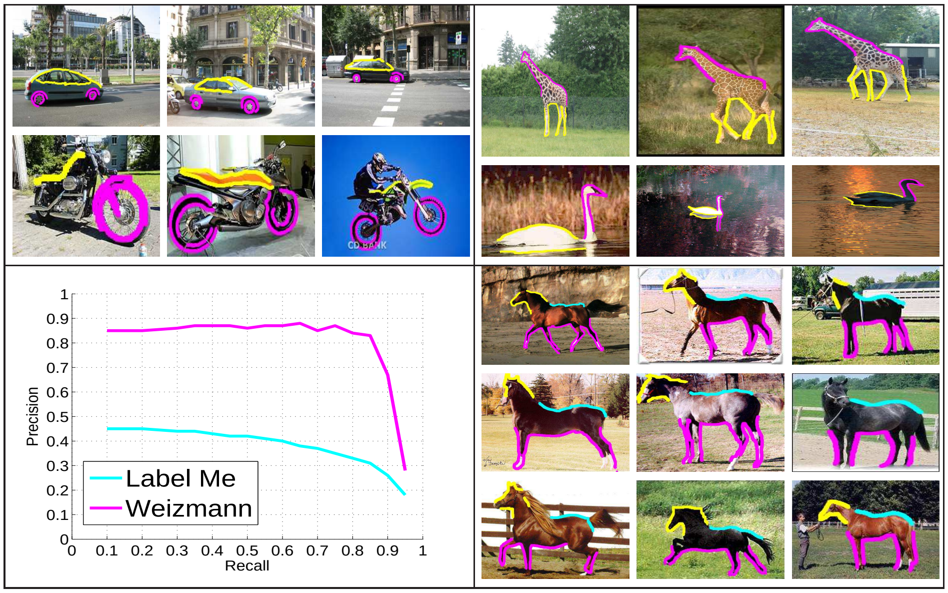


Fig. 7. Unsupervised detection and segmentation of objects in example images from LabelMe (top left), ETHZ (top right), and Weizmann Horses (bottom right). For LabelMe and ETHZ, each row shows images that are grouped within a unique image cluster by CASW in the first stage. Contours that are clustered by CASW in the second stage are highlighted with distinct colors indicating cluster membership. CASW accurately discovers foreground objects, and delineates their characteristic parts. E.g., for LabeMe *Cars sideview* CASW discovers two contour clusters (yellow and magenta), corresponding to the two car parts wheels and roof. (bottom left) ROC curves for LabelMe and Weizmann Horses, obtained by varying the minimum allowed DTW similarity between pairs of matching contours which are input to CASW. (best viewed in color)

18. Fei-Fei L., F.R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR. (2004)
19. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. Technical Report AIM-2005-025, MIT (2005)
20. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: ECCV. Volume 2. (2002) 109–124