Pattern Recognition 42 (2009) 1297 – 1307

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A stochastic graph grammar for compositional object representation and recognition

Liang Lin[a,b,c,*], Tianfu Wu[b,c], Jake Porway[c], Zijian Xu[c]

[a]Beijing Institute of Technology, Beijing 100081, PR China
[b]Lotus Hill Research Institute, Ezhou 436000, PR China
[c]Department of Statistics, University of California, Los Angeles, USA

ARTICLE INFO

Article history:
Received 1 November 2007
Received in revised form 7 September 2008
Accepted 22 October 2008

Keywords:
Object recognition
And–Or graph model
Recursive inference

ABSTRACT

This paper illustrates a hierarchical generative model for representing and recognizing compositional object categories with large intra-category variance. In this model, objects are broken into their constituent parts and the variability of configurations and relationships between these parts are modeled by stochastic attribute graph grammars, which are embedded in an And–Or graph for each compositional object category. It combines the power of a stochastic context free grammar (SCFG) to express the variability of part configurations, and a Markov random field (MRF) to represent the pictorial spatial relationships between these parts. As a generative model, different object instances of a category can be realized as a traversal through the And–Or graph to arrive at a valid configuration (like a valid sentence in language, by analogy). The inference/recognition procedure is intimately tied to the structure of the model and follows a probabilistic formulation consisting of bottom-up detection steps for the parts, which in turn recursively activate the grammar rules for top-down verification and searches for missing parts. We present experiments comparing our results to state of art methods and demonstrate the potential of our proposed framework on compositional objects with cluttered backgrounds using training and testing data from the public Lotus Hill and Caltech datasets.

1. Introduction

As a popular and central research topic in pattern recognition and computer vision, object category recognition contains two key aspects:

(1) How to represent object categories with common parts but a large number of appearances.
(2) How to design effective algorithms for inference.

Although recent research has achieved impressive results for a few specific object categories, such as faces [1,2], humans [3], texture-rich scenes [4,5], and some objects with simple configurations [6–8], recognizing and localizing compositional objects amidst cluttered backgrounds are still a challenging task. This is due to the varied appearances and complex structures of these objects. Compositional object categories refer to objects which can be decomposed hierarchically into constituent components, such as clocks, monitors, bicycles, and many man-made functional categories. Though the number of components and their types in each object are limited, they can create a huge number of combinations, and thus demonstrate radical intra-category structural variance. Fig. 1(a) shows the clock category, which is decomposed into the frame, hands and numbers. Arranging these components allows us to produce various clock instance as shown in Fig. 1(b). A good representation should thus have both *flexible structure* and *rich visual appearance*. In addition, an effective inference algorithm should be compatible with the representation as well. Addressing both of these issues, we focus on a novel stochastic grammar model capable of representing compositional object categories together with a recursive computing strategy integrating bottom-up proposals and top-down verification.

1.1. Related works

In the vision literature, object recognition and representations can be roughly divided into three categories of methods.

Appearance-based approaches achieve simple visual/image representations based on the photometric properties of an individual object or the object category. This field, which gained prominence in the 1990s with holistic appearance models [9], later grew to include local representations using invariant feature points [10], patches [6] and fragments [11]. Because these methods often disregard geometric information about the position of important key points within

[*] Corresponding author at: Beijing Institute of Technology, Beijing 100081, PR China.
E-mail address: linliang@bit.edu.cn (L. Lin).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.10.033

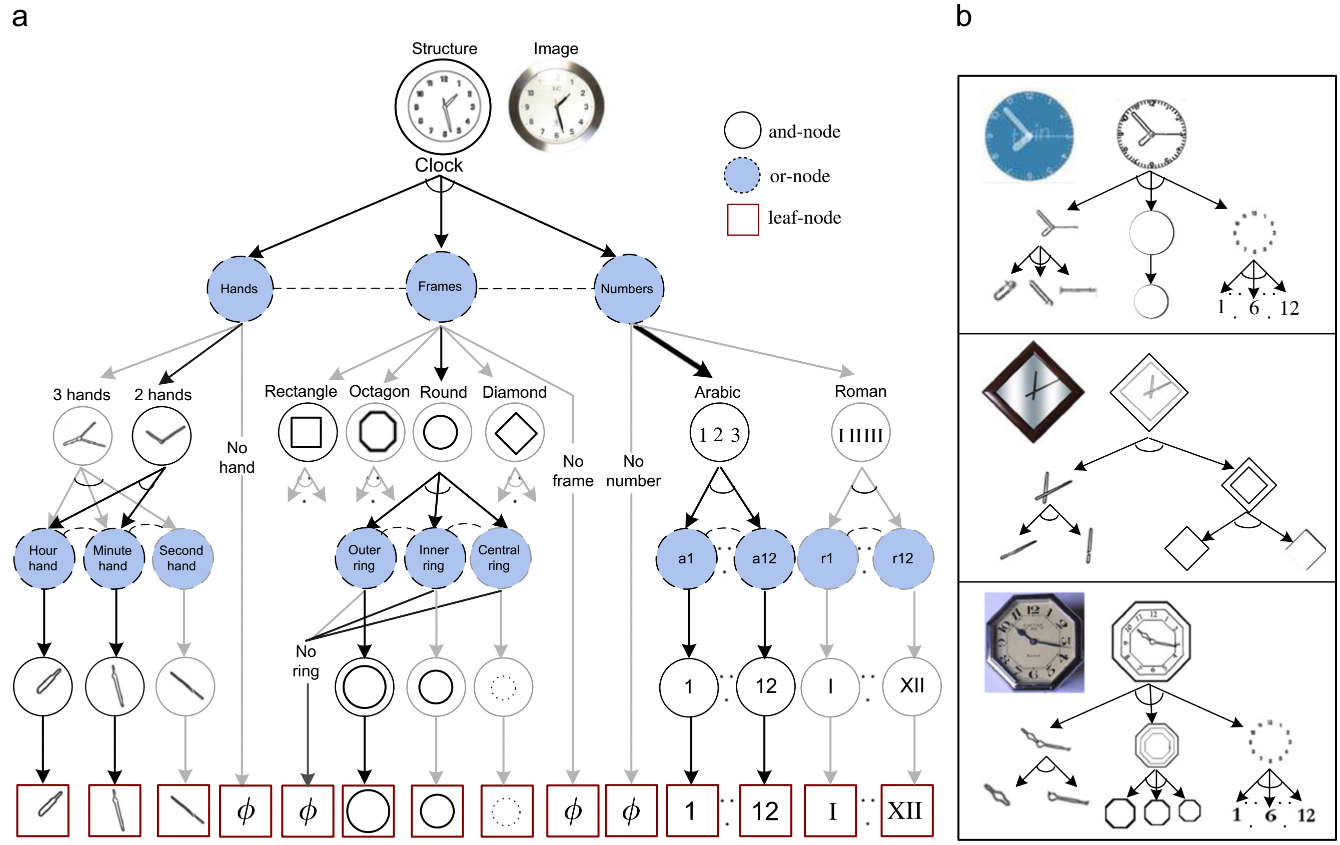

Fig. 1. The attributed grammar for a specific object category can be represented by an And–Or graph. (a) shows one example for the clock category. An Or node (dashed) is a "switching variable" for possible choices of the components. Only one child is assigned for each object instance. An And node (solid) represents a composition of children with certain spatial and appearance relations. The bold arrows form a sub-graph of the And–Or graph (also called a parsing graph) that generates a specific object instance (a clock) in the category. The process of object recognition is thus equivalent to assigning values to these Or nodes to form a "parsing graph". (b) lists three different clock instances and their parsing graphs.

an object, they are not well-suited for recognition in scenarios where pose, occlusion, or part reconfiguration are factors.

Structure-based approaches were introduced in the last decade to account for pictorial deformations [12] and variance in the shapes of patches [6,13]. These methods, such as the constellation model, model relationships between groups of parts using a graph representation (such as a Markov random field, MRF, or implicit shape model), and can thus improve recognition accuracy over purely appearance-based methods. Significant work has been done recently using structure-based approaches to perform general image recognition, including the tasks of scene segmentation and structure inference [5,14].

Though the two methods above have made remarkable progress in the field of object recognition, they have problems overcoming the huge variability in appearances of compositional object categories, which are very common in daily life. The appearance-based models require a huge number of training examples to learn an accurate model due to their lack of compositional and generative structures. They often over-fit a specific training set and can hardly generalize to novel instances or configurations, especially for categories that have large intra-class variations.

Very recently there has been a resurgence in modeling and recognizing object categories through *grammar-based approaches* [15–17]. Early work by [18], Dickinson [19,20], and Ohta [21] introduced these grammars to account for structural variance, but worked primarily on-line drawings and silhouette shaped contours. Han [16] and Chen [15] used attributed graph grammars to describe rectilinear scenes and model clothes, but were hard-coded for one category of images. The further grammar works on object recognition are presented by Lin [17] and Zhu [22]. Outside of the recognition community, Mark et al. designed a constrained grammar for text parsing [23], though its relational constraints were only on neighboring words, thus not incorporating full context.

Following the stream of grammar-based approaches, we present a stochastic grammar model for representing general object categories. This model, combined with the proposed recursive inference algorithm, can model compositional objects well, outperforming the traditional methods mentioned above. The authors presented a related paper using this grammar-based approach [17] to perform stochastic sampling from a grammar model of objects to synthesize new object instances and discussed the benefit of using these samples to improve testing accuracy. That paper serves as additional empirical support to the method proposed in this paper.

1.2. Method overview

The proposed hierarchical generative representation using a stochastic attribute graph grammar is termed an "And–Or graph", borrowing loosely from the knowledge representation terminology coined by Pearl [24]. Fig. 1(a) shows the And–Or graph for the clock category. The Or nodes (dashed) are "switching variables", like nodes in an SCFG, that choose between possible sub-configurations of the object, thus accounting for structural variance. Only one child is assigned to each Or node during instantiation. The And nodes (solid) represent pictorial composition of children with certain spatial relations. The relations include *butting*, *hinged*, *attached*,

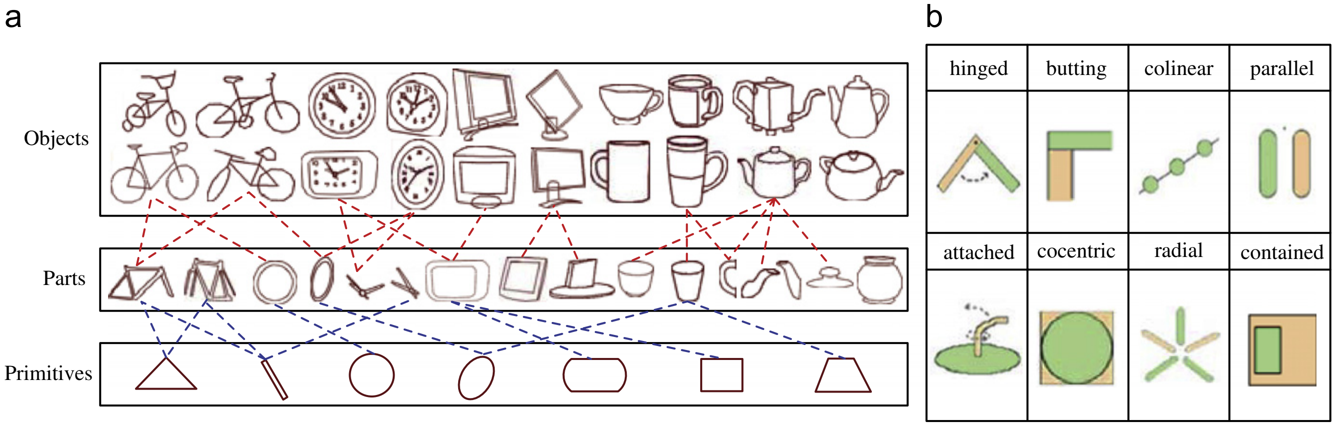

Fig. 2. A hierarchical representation of the generative vocabularies. From top to bottom, the three layers are, respectively, the compositional objects, the part templates and the generalized geometric primitives (circles, ellipses, rectangles, bars, triangles, etc.). The elements in one level are composed of elements in the level below. Different objects can share common parts, while different part templates can share common primitive templates.

contained, *cocentric*, *colinear*, *parallel*, *radial* and other constraints. Porway et al. [25] describe how these relations can be effectively pursued in a minimax entropy framework with a small number of training examples. Assigning values to the Or nodes and estimating the variables on the And nodes produces various object instances as sub-graphs/parsing graphs (Fig. 1(a)). Three different clock instances derived from the clock And–Or graph are shown in Fig. 1(b). The goal of a hierarchical object recognition process is thus equivalent in constructing the parsing graphs on the fly. Fig. 2 shows the hierarchic dictionary of the And–Or graphs. First, objects are decomposed into their constituent parts and then modeled by visual vocabularies similar to [26]. Fig. 2 shows five compositional object categories (bicycles, clocks, cups, monitors and teapots) and the components they are composed of at different levels. Secondly, components at each level are composed to form large structures, and two different objects may share common parts at the lower level. Thirdly, the compositions are specified through a set of attribute grammar rules. Each rule is associated with a number of hard or soft constraints on the attributes of the components, so as to model the pictorial relations. For example in Fig. 1(a), an hour hand, a minute hand and a second hand are grouped together by a “hinged” relation to form the clock hands. Finally, each compositional object category is represented by an And–Or graph which is a graphical representation for the attribute grammar.

The inference procedure follows a stochastic formulation in a two stage process. First the parts/primitives are detected using bottom-up methods. These detections in turn recursively activate the grammar rules for top-down verification and searching for missing parts. This process also allows us to hallucinate occluded/undetected parts in the final configuration based on top-down knowledge from the model.

The bottom-up step detects the parts/primitives of the object parts in two ways:

(a) *Implicit testing*: detecting instances of parts or primitives from the constructed sketch-graph, through a sequence of tests, such as Adaboost [1] or Generalized Hough Transform. For example in Fig. 5, the geometric primitives like ellipses and triangles are detected.
(b) *Explicit binding*: proposing instances of graph nodes by binding their detected children nodes through a sequence of tests on compatibility. For example, two concentric ellipses are grouped into a wheel proposal. Usually the feature detectors we use are trained off-line.

The bottom-up hypotheses in turn activate the grammar rules embedded in the graph model for top-down verification, shown in Fig. 5. This top-down verification via the grammar rules includes three components:

1. *Match* the best composite template from the model to the image given the bottom-up hypotheses.
2. *Search* for weak/miss-detected parts given other verified parts that belong to the same larger structure, subject to certain relationships.
3. *Hallucinate* the occluded parts by sampling from the prior model learned from training data.

The bottom-up and top-down steps are recursively invoked during the whole inference process. As shown in Fig. 5, we not only recognized, but also precisely localized the bicycle and parsed it into its constituent parts. Our recursive algorithm is designed in a similar spirit to *DDMCMC* [27]. Later, we show that deterministic decisions can be made in approximating the globally optimal solution by invoking the bottom-up and top-down steps according to the most reliable bottom-up feature detectors. Similar work was done recently in [15,16], though both these papers focus on specific domains, like rectangles or clothes. We present a general and scalable framework for many general object recognition. Another related work is [28], given that an edge/sketch representation was also used to capture the structures of objects. However in contrast to the outlined-templates used in [28], we use templates that have flexible topologies and are full of intra-contour structures. Experiments showed that our model can be constructed effectively from a relatively small training set (30–50) per category. The efficiency of our inference algorithm is also illustrated on detection and recognition of a set of selected compositional object categories: clocks, bicycles, monitors, cups and teapots in natural images.

This paper is one of a series of papers prepared by the authors and their colleagues in the vision lab at UCLA. This paper was originally written for submission in 2005 and 2006, the technical paper versions of which were then cited in [25]. In addition, a survey of the lab's work was published in [29], which encompassed some of the early results herein. Readers are referred to [25,29] for more details about this work and its context within the greater work being done by the lab.

The remainder of this paper is arranged as follows. We present the And–Or graph representation and grammar in Section 2, followed by the probabilistic formulation in Section 3. The algorithm is discussed in Section 4, and experiments with comparisons are reported in Section 5. The paper is concluded in Section 6 with a discussion of future work.

2. Representing compositional objects

We begin by briefly reviewing the attributed grammar followed by a description of our representation for the parts and constrained grammar rules. We then show their equivalence to the And–Or graph.

2.1. Attribute grammars

An attribute grammar is defined as a 4-tuple as in [25,29]

$$\mathcal{G} = (V_N, V_T, \mathcal{R}, \Sigma)$$

V_N is a set of non-terminal nodes, and V_T is a set of terminal nodes. $\mathcal{R}$ is a set of production rules. Each rule describes how to expand a non-terminal node, for example a clock frame is expanded into a set of ellipses. Σ is a set of configurations that can be produced by repeatedly applying production rules to the root node $S \in V_N$

$$\Sigma(\mathcal{G}) = \{(C, X(C)) : S \xrightarrow{R^*} C\}$$

This is the language of $\mathcal{G}$. For most of the object categories, the total number of valid configurations often largely outnumbers the sum of all the nodes in the And–Or graph

$$|\Sigma| \gg |V_N \cup V_T|$$

Most importantly, Σ includes novel configurations never seen in the training set. This generalization power is critical for modeling compositional object categories. The order in which the grammar rules are applied defines the "parsing graph" for an object instance. In many cases, there may be more than one valid parsing graph for a configuration C, in which case a probability formulation is used to select the most probable graph.

2.2. Primitives for object representation

We use a geometric representation of primitives in this paper where each terminal is defined by a sketch-graph

$$V_T = \{(a, x(a)) : l_a \in \Omega_L, x(a) \in \Omega_a\} \tag{1}$$

$$a = \{\mathcal{V}_a, E_a, I_a, l_a : l_a \in \Omega_L\} \tag{2}$$

$$x(a) = f_i(\mathcal{V}_a, E_a, I_a | l_a), \quad i = 1, 2, \ldots, n(l) \tag{3}$$

Each terminal a consists of a set of vertices $\mathcal{V}$, a set of edges E, the intensity profiles across the edges I, and a label l for its type. The attribute for a terminal node $x(a)$ is defined as a function of its structure, dependent on the type of the primitive. For compositional object categories, the intensity profiles rarely give enough information for recognition, but are instead used for consistency checks, for example ensuring the color around the edge of a frame is similar.

We design a set of sketch-graphs, in which the vertices are arranged in such a way that they well represent generalized geometric primitives (circles, ellipses, rectangles, triangles, etc.). For example, a set of small line segments can be connected sequentially to form a circle, while their centers satisfy the circle equation and their directions point to the tangent. The number of line segments is proportional to the perimeters. We can also define a rectangle with four "L" junctions connected by two pairs of nearly parallel line segments such that the length ratios and corner angles satisfy certain constraints. It, thus, allows us to combine the Generalized Hough Transform and template matching [30,31] techniques for performing flexible and robust bottom-up detection.

2.3. Compositional relations and grammar rules

To account for the constraints between parts, we define a set of relationships, such as "*relative position*", "*relative scale*", "*hinged*", etc. These relationships cover the constraints between every attribute $x(a)$ of every possible node. While the attribute constraints relate the shapes of the node pairs, such as similar interior angles, the spatial constraints enforce relations like cocentricity and collinearity, such as the rings of clock frame shown in Fig. 1(a). Another example is the "hinged" relation among the clock hands. Porway et al. [25] shows how to effectively pursue these relationships using a minimax entropy framework.

We also define three types of production rules. The first rule expands the scene node S into m objects. The second rule expands a node A into $2, \ldots, m$ related nodes. For example, the numbers on the face of a clock would be decomposed into $m = 12$ separate nodes, all related by the "radial" constraint (Fig. 1(a)). The third rule expands a non-terminal node into a terminal node, subject to some attribute constraints. For example, the inner frame of the diamond clock would be constrained to match the outer frame's diamond shape (as shown in Fig. 1(b)).

2.4. The And–Or graph representation

The And–Or graph is a visual representation of the attribute grammar that integrates stochastic context free grammar (SCFG) models and pictorial models. In the And–Or formulation V_t, $\mathcal{R}$, and Σ are the same as in the attribute grammar. The only change is that we classify our non-terminal nodes V_N as *And* and *Or* nodes

$$V_N = V_N^{And} \cup V_N^{Or}$$

An *And* node can be thought of as the result of a production rule. For example, if we follow the production rule $A_1 \to (A_2, A_3)$, we consider the result, (A_2, A_3) an *And* node, as it must consist of both A_2 and A_3. An *Or* node V_N^{Or} is the production rule chosen to expand node A

$$V_N^{Or} = (A, \omega(A))$$

We define a switching variable $\omega(A)$ which indexes the rule chosen. For example, if the production rules $A_i \to A_2$ and $A_i \to (A_3, A_4)$ both exist, $\omega(A_i) = \{1, 2\}$ depending on which rule we use to expand A_i.

The And–Or graph for the clock category is shown in Fig. 1(a), where the root node is an *And* node and is denoted by a solid circle. The dashed circles denote the *Or* nodes, e.g. the *Hands*, which is expanded into 2 *Hands* or 3 *Hands* configurations. The horizontal lines are the spatial relations. Leaf-nodes (primitives) are denoted by rectangles.

3. Probabilistic formulation

As the And–Or graph derives from SCFGs and MRFs, we formulate the probabilistic framework for the And–Or graph as a combination of these two models. This is similar to the constrained stochastic language models in [15,16].

3.1. Stochastic context free grammars

The hierarchical nature of the And–Or graph can be modeled as an SCFG. If we traverse the graph using only *Or* nodes, then we arrive at a parsing tree consisting only of parts pg_t. This parsing graph does not include any spatial relationships as of yet. The prior probability $p(pg_t)$ for tree pg_t is just the product of probabilities of rules, which can be viewed as switch variables on the *Or* nodes

$$p(pg_t)=\prod_{i=1}^{n} p(r_i) \tag{4}$$

This captures the hierarchy of the compositional objects.

3.2. Markov random fields

Given a parsing graph *pg* for an object, like those shown in Fig. 1(b), we embed a MRF on each of the *And* nodes to constrain the node attributes and the relation between pairs of nodes using $\phi(X(A_i))$ and $\psi(X(A_i),X(A_j))$

$$p(A)\propto\prod_{i=1}^{n}\phi(X(A_i))\prod_{i=1}^{n}\prod_{j\in\mathcal{N}_i}\psi(X(A_i),X(A_j)) \tag{5}$$

This defines the probability of a configuration for an *And* node constrained by spatial relations and attributes.

3.3. And–Or graph for compositional objects

The probability model for an And–Or graph is the combination of the probability models for an SCFG and an MRF. By embedding the Markov constraints $\phi(X(A_i))$ and $\psi(X(A_i),X(A_j))$ into the SCFG model, we form a graph that captures object hierarchy from top to bottom as well as spatial and attribute constraints horizontally between nodes at the same level. We can express the prior probability of a graph *pg* as the distribution that minimizes the Kullback–Liebler divergence between some true unknown distribution for the tree structure $f(pg)$ and our prior $p(pg)$, subject to our spatial and relational constraints

$$p^*(pg)=\operatorname{argmin}\sum_{pg} f(pg)\log\frac{f(pg)}{p(pg)} \tag{6}$$

This is equivalent in finding the maximum entropy distribution for the right term above and can be expressed as

$$p^*(pg;\Theta)=\frac{1}{Z}\exp(-E(pg)),\quad \Theta=(\alpha,\phi,\psi) \tag{7}$$

$$E=\sum_{v\in V}\alpha(\omega(v))+\sum_{i=1}^{n}\phi(X(A_i))+\sum_{i=1}^{n}\sum_{j=1}^{n}\psi(X(A_i),X(A_j)) \tag{8}$$

The first term of the energy function is simply the frequencies at each *Or* node, and thus is equivalent to a SCFG. The second and third terms are the singleton and pairwise energies defined by the Markov constraints. Thus the probability of a graph $p(pg)$ is equivalent to the probability of its branching tree, subject to Markov constraints. In [25,29], these constraints are pursued by a minimax entropy framework.

3.4. Likelihood measurement

Combining the SCFG and MRF models, the probability model for the And–Or graph model (as in Eq. (7)) can be learned as a prior term for inference. The likelihood for this model is formed from the probability of matching each terminal graph node (sub-templates) $g_i\in V_T$ with domain Λ_i to the image patch I_{Λ_i} . The likelihood model is

$$P(I_\Lambda|G;\Delta)=\prod_{g_i\in V_T}P(I_{\Lambda i}|g_i;\Delta_i) \tag{9}$$

where $\Delta=\{\Delta_1,\Delta_2,\ldots,\Delta_N\}$ is the dictionary of all object primitives associated with the templates, as shown in Fig. 2(a). We use a graph matching model [30] to measure the goodness of fit $P(I_\Lambda|g_i;\Delta)$ to its image patch Λ_i. This graph matching technique accounts for the photometric, geometrical and topological aspects of the object primitives.

4. Recursive inference

We represent a scene by *n* independent objects and each object is represented by a specific parsing graph $pg_i\in\Sigma(\mathcal{G})$, $i=1,2,\ldots,n$. For a scene that contains three objects, the parsing graph is thus (pg_1,pg_2,pg_3,G_{sk}). G_{sk} is a single layer graph denoting the "free sketches" comprising the background. Object recognition is thus equivalent to optimizing the Bayesian posterior probability

$$\Theta^*=\operatorname{argmax} p(\mathbf{I}|\Theta)p(G_{sk})p(K)\prod_{i=1}^{K}p(pg_i;\Theta) \tag{10}$$

where K is the number of objects in the image ($K=1$ in the mostly cases) and Θ consists of the parameters for the parsing graphs.

As the And–Or graph is defined recursively, we can also define the inference algorithm recursively. This recursive property largely simplifies the algorithm design and makes it easily scalable to an arbitrarily large number of object categories. Similar to [15,16], our algorithm integrates two closely coupled processes, bottom-up detection of parts/primitives from the image and top-down verification using our learned model. These two processes form an iterative loop. This cycle continues back and forth until no further bottom-up steps remain, or until we reach the root node of the parsing graph.

The algorithm keeps two data structures for each graph node as shown in Fig. 3. Thus the bottom-up steps stop when particles weights in the Open List are all lower than some empirical threshold, and top-down verification steps stop when only object nodes are left in the Closed List.

- An *Open List*: stores a number of weighed hypotheses (denoted by particles) that are detected in the bottom-up phase.
- A *Closed List*: stores a set of graph node instances verified in the top-down phase. These instances compose the current parsing graph.

Before entering the iterative cycle, we convert the original image into a "sketch-graph" or "primal sketch" by detecting and grouping edges, corners and junctions [32]. This diminishes the effects of various colors and illuminations while keeping the crucial structural information from the image. The sketch-graph of a partially occluded bicycle image is shown in Fig. 5(a).

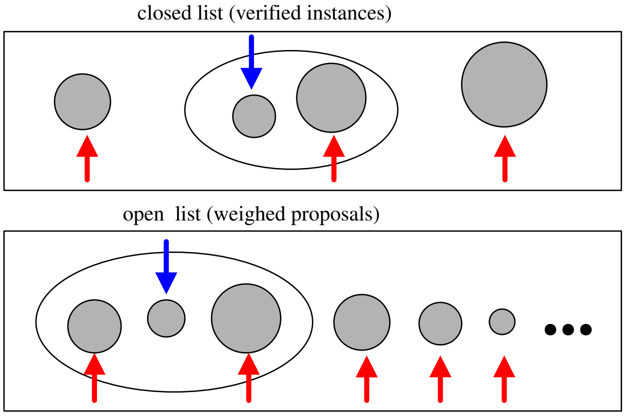
closed list (verified instances)
open list (weighed proposals)

Fig. 3. The open and closed lists used during the bottom-up/top-down inference. The Open List contains particles that are currently under consideration for the current explanation of the scene, while the Closed List contains particles that have already been accepted to explain the scene. The arrows indicate that evidence for the particles can come from bottom-up detections, top-down predictions, or both.

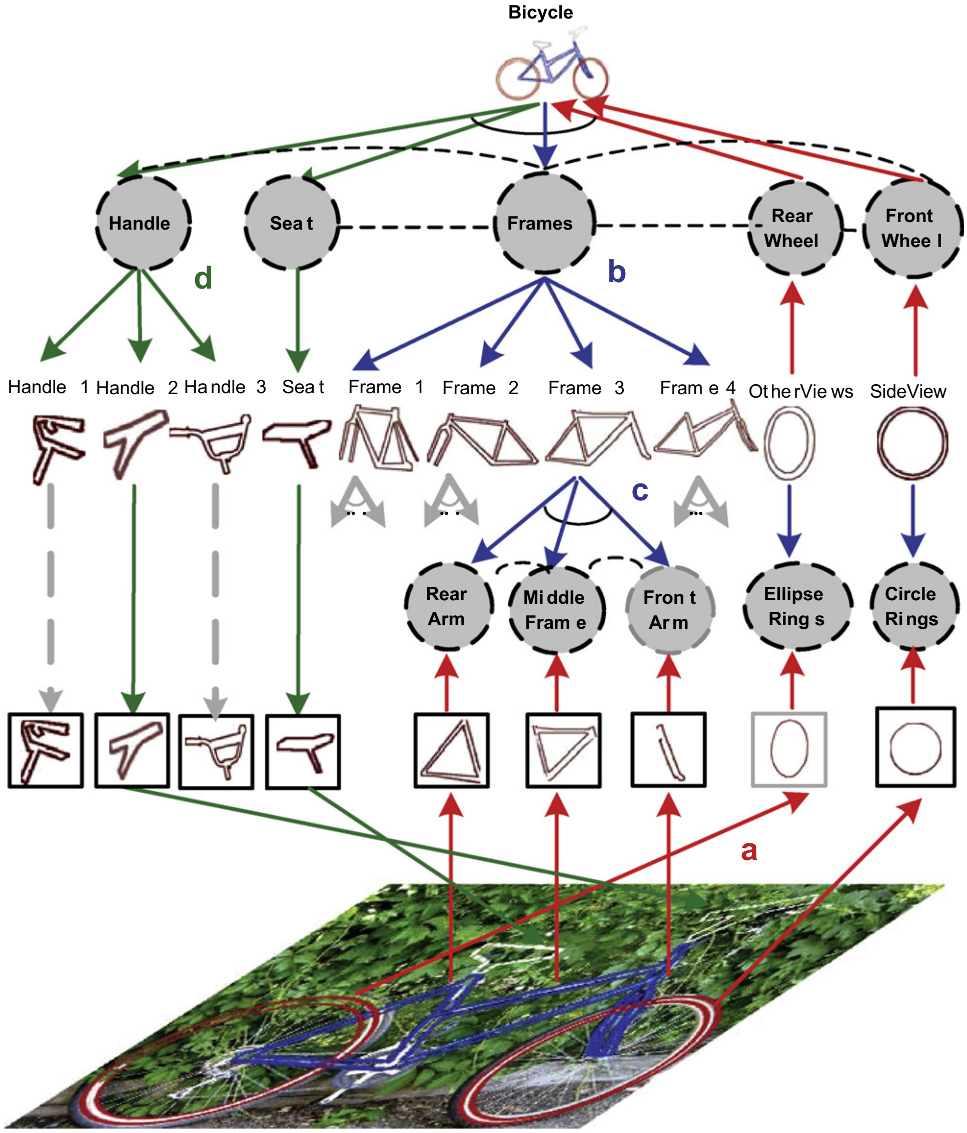

Fig. 4. Bottom-up/top-down inference of the bicycle example. Bottom-up proposals are denoted by red arrows pointing upward, e.g., the circle, ellipse and triangles. Top-down predictions are denoted by blue arrows pointing downward, e.g., the frame. Note that the bottom-up and top-down processes happen recursively. For example, first circles are detected in (a), and a production rule is activated to group two concentric circles into a wheel. The wheel proposal is then accepted and in turn activates the prediction of the frame (b). In (c), a template match component tries to explain the predicted frames using the detected triangles and lines and accepts the best match. In (d), as the seat and handlebars are mostly occluded, they are randomly sampled from the learned prior model keeping the frame and wheels fixed, as shown by green downward arrows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Bottom-up phase

In the bottom-up phase we generate hypotheses of parts or primitives from the sketch-graph (denoted by red upward arrows in Fig. 4) in two ways: *implicit testing* and *explicit binding*. These proposals are weighed according to how frequently they occurred in training examples and how well they match the underlying pixels in the current image. These hypotheses are stored in the *Open List*.

Implicit testing: It entails searching the sketch-graph that we computed beforehand for a set of pre-defined primitives (defined as sub-sketch-graphs), such as circles, ellipse, sticks, triangles and rectangles. We use techniques like AdaBoost [1] and the Generalized Hough Transform to find commonly occurring shapes. Fig. 5(a) shows the proposed ellipses, circles and triangles found using AdaBoost and the Generalized Hough Transform. Because the And–Or graph is defined recursively, many of the parts or primitives in our dictionary are shared across different object categories. This property enables us to produce a large number of configurations from a relatively small set of parts or primitives. It also justifies our use of a fixed pool of feature detectors for proposing these parts and primitives, as we need only to find a small set of common shapes. These detectors are trained off-line.

Explicit binding: We use explicit binding to propose parts that are mostly composed of other easy-to-detect primitives. Given that a couple of nearby primitives are successfully detected, we can bind them into a larger structure through a sequence of tests on their compatibility. For example in Fig. 4, two concentric ellipses are grouped into a wheel proposal. Other relations (Fig. 2(b)) are also checked in the compatibility tests.

Visiting order: The computational robustness and efficiency of detecting a part or primitive may vary a lot depending on its shape.

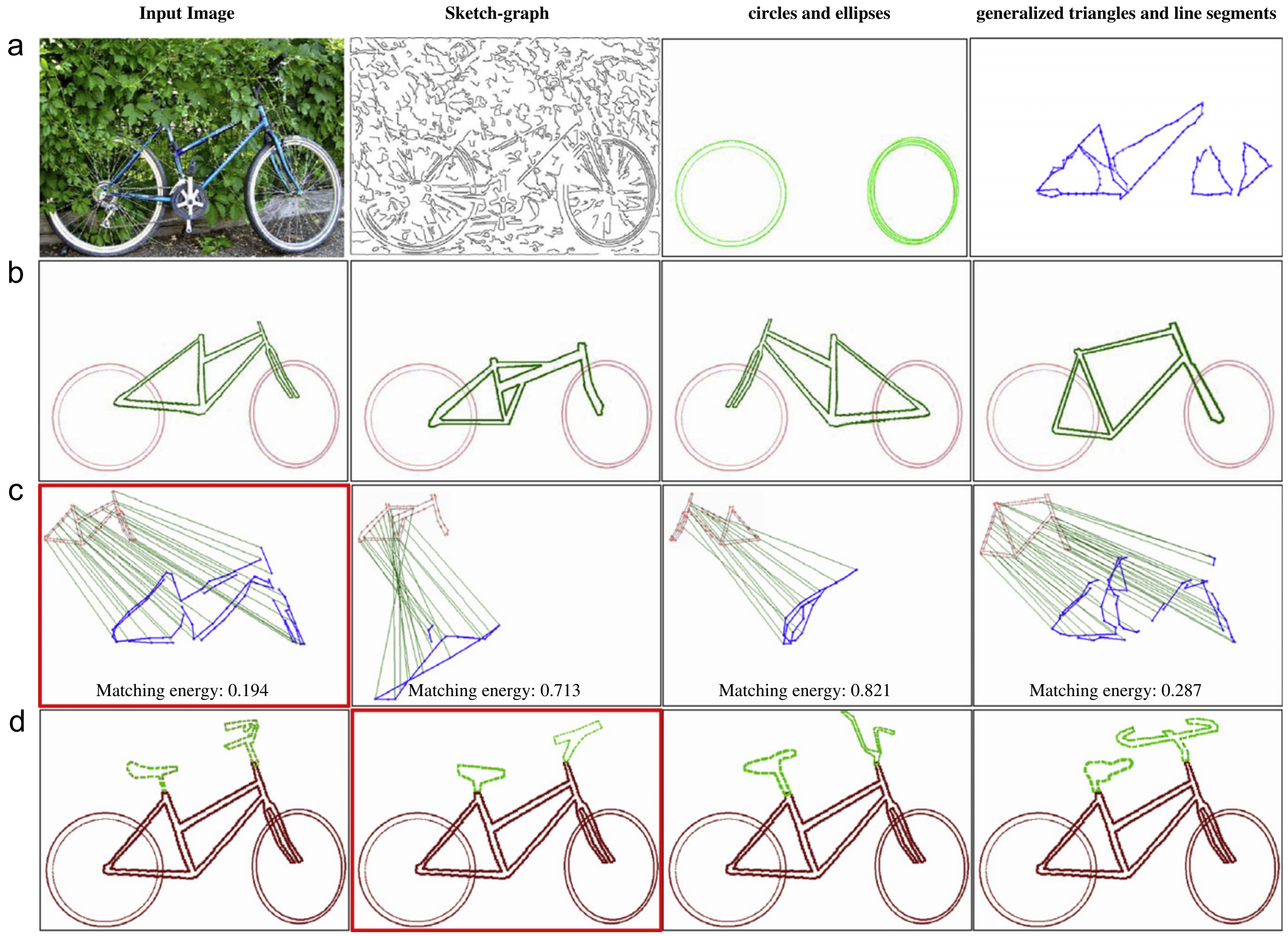

Fig. 5. Example for parsing a partially occluded bicycle. (a) shows the input image. The sketch-graph, a set of small linelets, corners and junctions, is then computed by edge/corner/junction detection techniques. In the initial bottom-up stage, a number of generalized primitives are proposed, such as ellipse and triangles. (b) shows the top-down predictions of a bicycle frame with fixed wheels. The transformed parameters of the frame are sampled from the learned MRF model. As we cannot tell the difference between the front/rear wheels at this moment, the frames are sampled for both directions. (c) shows the template match of the predicted frames, where the one with the minimum matching cost (highlighted in red) is selected. (d) shows the top-down hallucinations for the seat and handlebars. As these two parts are mostly occluded and lack support from the image, they are randomly sampled from the prior model (highlighted in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Input: an input image **I**, and a set of constructed And-Or graphs of compositional object categories.
Output: a parsing graph pg_s of the scene that consists of the parsing graphs of detected objects.

- Repeat the following steps

1 Schedule the next node A to visit from the candidate parts.

2 Call Bottom-up(A) to update A's **open** list.

  i Detect terminal instances of A from the image.
  ii Bind non-terminal instances of A from its children's **open** or **closed** lists

3 Call Top-down(A) to update A's **open** or **closed** lists.

  i Accept hypotheses from A's **open** list to its **closed** list.
  ii Remove (or disassemble) hypotheses from A's **closed** list.
  iii Update the **open** lists for particles that overlap with node A.

- Until the particles in **open** list with weights higher than the empirical threshold are exhausted. Output all parsing graphs whose root nodes are reached.

Fig. 6. Recursive top-down/bottom-up inference algorithm.

For example, the handlebars or seats of the bicycles are usually very hard to accurately detect and blend in with clutter very easily. Also some tests are too costly to perform in the early stages, such as blindly searching for triangles to compose the bicycle frame. We thus traverse the And–Or graph in an orderly manner during inference. In our experiments, the visiting order is decided by comparing

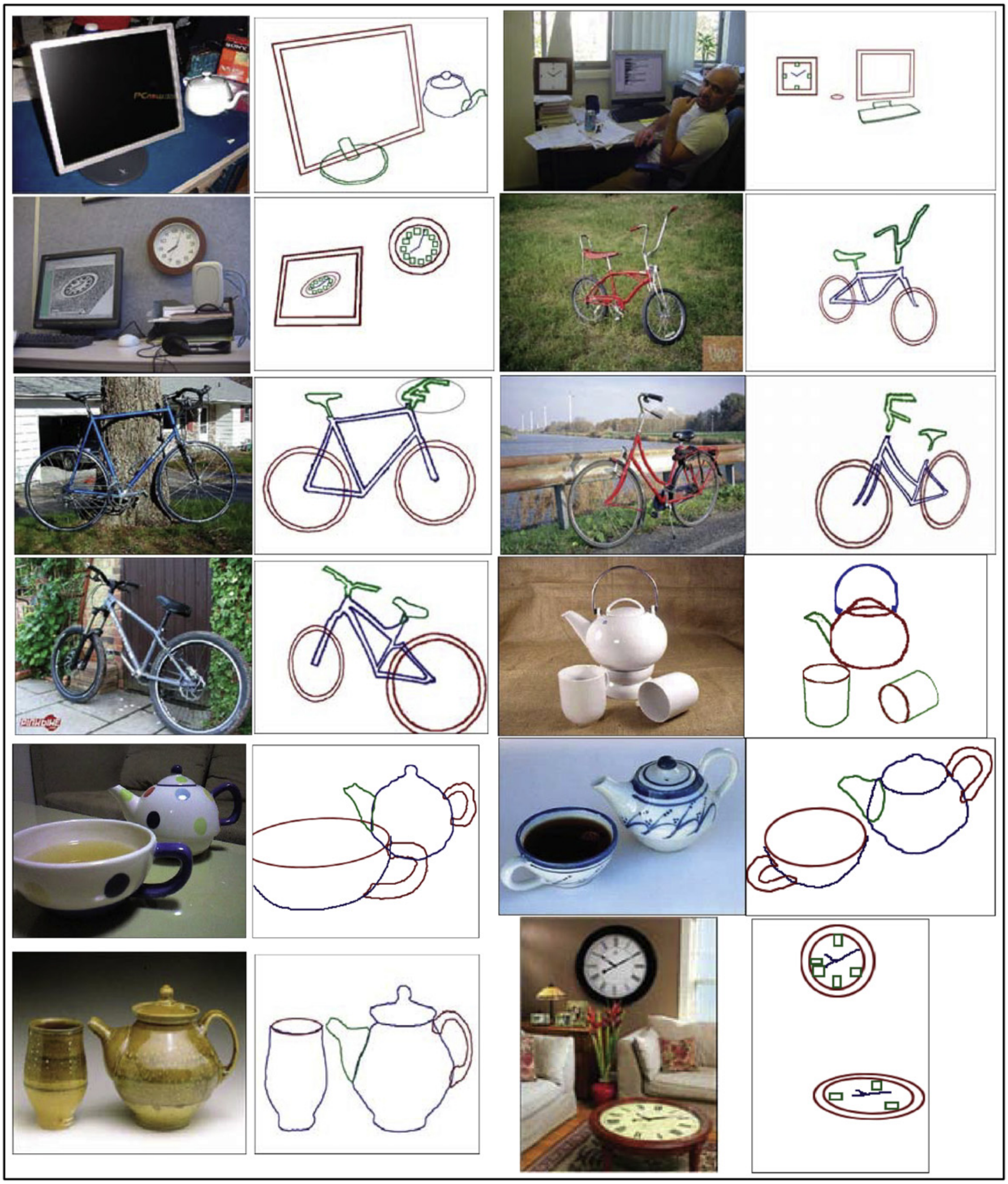

Fig. 7. Parsing/recognition results for compositional object categories, including bicycles, clocks, monitors, cups and teapots. The most informative parts detected in the bottom-up stage are denoted in red, such as the bicycle wheels, monitor frames, clock frames and cup lips. The predicted and matched parts using top-down knowledge are denoted in blue, such as the clock hands, the bicycle frame, cup handles and teapot bases. The parts with weak image evidence (noisy or occluded), which are hallucinated and matched using the prior models, are denoted in green, such as the clock numbers, monitor base, teapot spot and bicycle seat/handlebars. The wrong hallucination results, which do not match human perception, are indicated by circles [25]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the bottom-up detection rate of different parts. As the bottom-up receiver-operating characteristic (ROC) curves in Fig. 8 indicate, the bicycle wheels are detected most reliably among all the parts of the bicycle. When we fix the detection rate to be 100%, the *average false positive number per image* of wheels, frame, seat and handlebars are 27, 322, 296, 268, respectively. Thus, we look for the wheels first as they have the least ambiguity. In doing so we approximate the globally optimal parse by making deterministic decisions about which parts to look for first.

4.2. Top-down phase

Both human intuition and experiments on neuronal response in visual cortical processing [33] indicate that top-down influences are present from the very beginning of image understanding to the end. For example, the first glance at the partially occluded bicycle image in Fig. 5 may only trigger some "circle/ellipse" alarms because they are the most strong stimuli. However our mind, being familiar with bicycles, may immediately pop-out a frame between the two wheels.

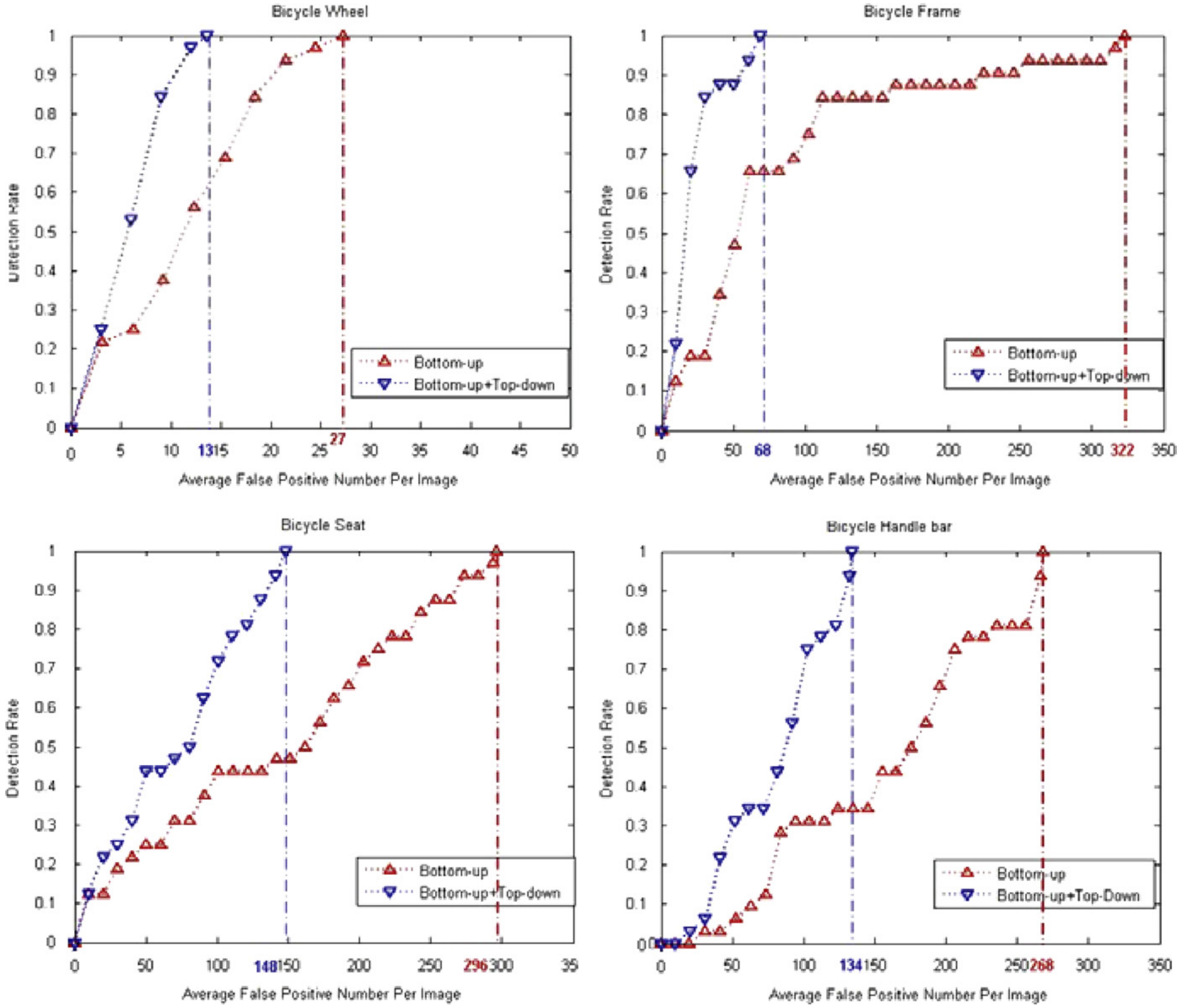

Fig. 8. ROC curves of the detection results for bicycle parts. Each graph shows the ROC curve of the results for a different part of the bicycle using just bottom-up information and bottom-up + top-down information. We can see that the addition of top-down information greatly improves the results. We can also see that the bicycle wheel is the most reliably detected object using only bottom-up cues, so we will look for that part first.

With a quick second glance, even the seat and handlebars may be "seen", though they are actually occluded. Our algorithm simulates the top-down process (indicated by blue/green downward arrows in Fig. 4) in a similar way, using the constructed And–Or graphs.

Verification of hypotheses: Each of the bottom-up proposals activates a production rule that matches the terminal nodes in the graph, and the algorithm predicts its neighboring nodes subject to the learned relationships and node attributes. For example in Fig. 4, a proposed circle will activate the rule that expands a wheel into two rings. The algorithm then searches for another circle of proportional radius, subject to the concentric relation with existing circle. In Fig. 5(b), the wheels are already verified. The candidate frames are then predicted with their ends affixed to the center points of the wheels. Since we cannot tell the front wheels from the rear ones at this moment, frames facing in two different directions are both predicted and put in the *Open List*. In Fig. 5(a), the triangle templates are detected using a Generalized Hough Transform only when the wheels are first verified and frames are predicted. If no neighboring nodes are matched, the algorithm stops pursuing this proposal and removes it from the *Lists*. Otherwise, if all of the neighboring nodes are matched, the production rule is completed. The grouped nodes are then put in the *Closed List* and lined up to be another bottom-up proposal for the higher level. Note that we may have both bottom-up and top-down information being passed about a particular proposal as shown by the gray arrows in Fig. 3. In Fig. 4, the sub-parts of the frame are predicted in the top-down phase from the frame node (blue arrows); at the same time, they are also proposed in the bottom-up phase based on the triangles we detected (red arrows). Proposals with bidirectional supports such as these are more likely to be accepted. After one particle is accepted from the *Open List*, any other overlapping particles should update accordingly.

Template match: The pre-defined part templates, such as the bicycle frames or teapot bodies, are represented by sub-sketch-graphs, which are composed of a set of linked edgelets and junctions. Once a template is proposed and placed at a location with initial attributes, the template matching process is then activated. As shown in

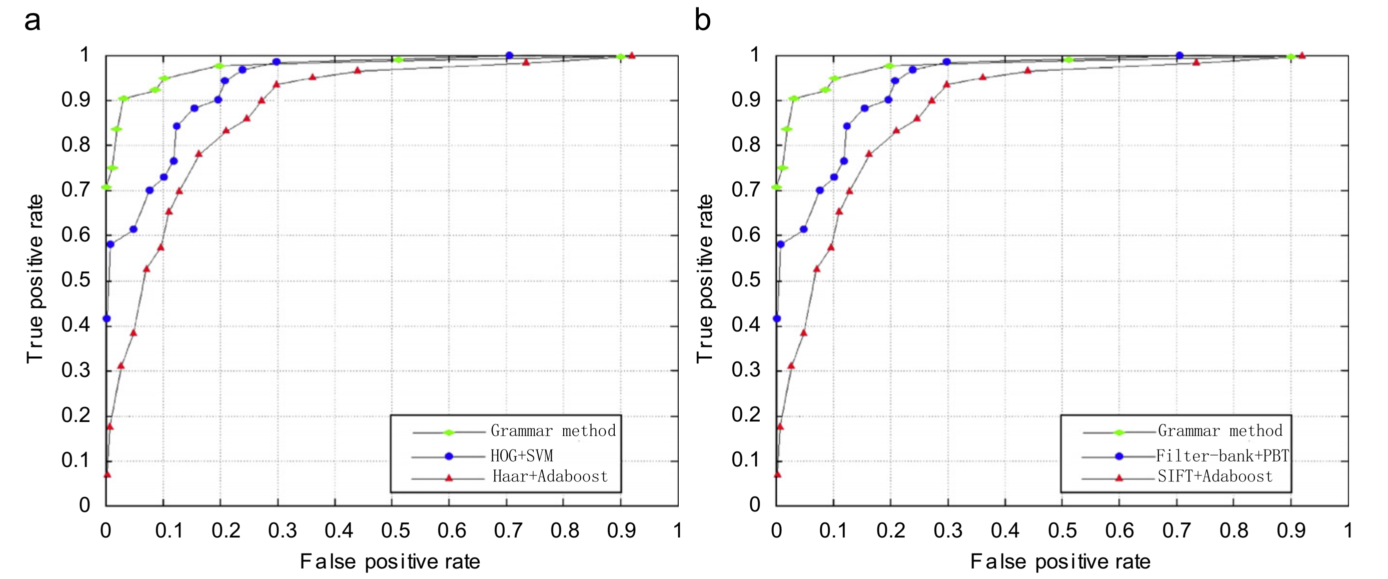

Fig. 9. Recognition results with comparison. (a) shows the ROC curves on bicycle images from LHI dataset. The AUC of the proposed framework is 0.962. (b) shows the ROC curves on rear-car data on Caltech dataset. The AUC of our method is 0.970.

Fig. 5(c), different frames are matched to the sketch-graph of the input image using thin-plate-spline warping with an affine transformation described in [15]. The template with the minimum matching cost is then selected as shown in Fig. 5(c). Some primitives that are detected in the bottom-up stage are used as “seed graphs” to initialize the template matching and greatly speed up the process. Note that two input graphs each consisting of around 50 vertexes (within a 100 × 100 image patch) take about 10 s for matching.

Hallucination: The image noise or occlusion means some object parts are not detectable, even with sufficient context. For example, the bicycle seat and handlebars in Fig. 5 are mostly occluded by the tree-leaves and lack evidence from the image pixels themselves. We thus cannot achieve enough support from the image and have to instead resort to prior knowledge. The missing parts are then randomly sampled from the prior model learned beforehand as in [25]. In Fig. 1(d), the handlebars and seat are hallucinated with reasonable spatial constraints with respect to the verified parts. In the first bicycle example in the second row of Fig. 7, we hallucinate the handlebars that were not detected due to background clutter. These handlebars are circled to indicate that the shape sampled from the prior looks awkward to a human, as they appear backwards. However, we are still able to put the handlebars in the proper position, with the proper scale and orientation.

The complete algorithm proceeds in Fig. 6.

5. Experiments

Since the recognition rates are already very high on relatively clean images (like the Corel dataset), we test six compositional object categories (bicycles, clocks, monitors, teapots, cups, and rear-view car) on challenging LHI [34] and Caltech [4] datsets. With the supervised labeling work supported by Lotus Hill Institute, we are able to learn the And–Or graph model for the compositional objects as in [25].

Fig. 1 shows an And–Or graph of clocks. Figs. 4 and 5 show the parsing graph and running example of a bicycle. A few illustrative recognition results are shown in Fig. 7, which was shown in [25]. The most informative and thus reliably detected parts are denoted in red, such as the monitor frame, clock frame, cup openings and bicycle wheels. The predicted and matched parts using top-down knowledge are denoted in blue, such as the bicycle frame, clock hands, cup handles and teapot vessels. For those parts with weaker image evidence (occluded or noisy), we use green to denote the hallucinated and matched results, such as teapot spout, clock numbers monitor base and bicycle seat/handlebars.

LHI dataset: In order to further quantitatively illustrate the benefit of our grammar model we ran our inference algorithm on 200 bicycle testing images from LHI dataset [34] using a learned And–Or graph model for bicycles. As shown in Fig. 7, the testing images contain cluttered background. For comparison, we train a HOG based SVM [3] classifier and Haar-feature based Adaboost classifier [1] on selected training images (200 positive and 400 negative samples, respectively). Recognition performance is reported as ROC curves, as presented in Fig. 9(a). The area under ROC curve (AUC) of the proposed grammar model with our recursive inference algorithm is 0.962.

Caltech dataset: We performed the same experiment on rear-view car images from the Caltech dataset [4], which contain relatively clean backgrounds. We randomly selected 50 samples from a total of 550 images for testing. We compare our method against the method used in [8], which uses SIFT features in a boosting framework, as well as against the probabilistic boosting tree (PBT) framework with filter bank features [35]. As shown in Fig. 9(b), the AUC of our framework achieves 0.970.

6. Discussion

In this paper, we presented a model for representing compositional object categories as an attribute grammar. This context sensitive approach is novel for the field of object recognition and bridges the gap between appearance models and pictorial models. In addition, the recursive inference algorithm, which alternates between bottom-up and top-down phases is a very powerful method of quickly constraining bottom-up detection and testing top-down constraints.

There are still many problems to overcome for unified grammar-based object category recognition. First, we should flexibly integrate more effective features (including structural and texture features) in the bottom-up module. Mining proper features for a vast number of object categories and specific object parts is a challenging topic.

Secondly, the current computational efficiency of proposed grammar approach is relatively slower than those discriminative approaches (the mentioned Adaboost [1] and SVM + HOG [3]). The average cost time for a testing images is around twice longer. To improve the efficiency of top-down verification, we plan to adopt another generative deformable template matching technique from our group, which achieved high attention in 2007 [31]. In addition, scheduling the implicit testing and explicit binding components for bottom-up detection is also a serious task in recursive inference, i.e. how to adjust the visiting order. Besides comparing the discriminative power as illustrated in this paper, some psychological statistics will also be considered in the future.

Acknowledgments

This work is done when the authors are at the University of California, Los Angeles (UCLA) and Lotus Hill Research Institute (LHI). The project at LHI is supported by Chinese 863 Program (Grant nos. 2007AA01Z340 and 2006AA01Z121) and NSFC (Grant nos. 60673198 and 60672162). The project at UCLA is supported by NSF (Grant no. IIS-0713652). The authors would like to thanks Dr. Song-Chun Zhu for helpful guidance and extensive discussion.

References

[1] P. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 511–518.

[2] Z. Tu, X. Chen, A.L. Yullie, S.C. Zhu, Image parsing: unifying segmentation, detection, and recognition, International Journal of Computer Vision 63 (2) (2005) 113–140.

[3] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Proceedings of European Conference on Computer Vision, vol. 2, 2006, pp. 428–441.

[4] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 524–531.

[5] J. Shotton, J.M. Winn, C. Rother, A. Criminisi, TextonBoost: joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: Proceedings of the ECCV, vol. 1, 2006, pp. 1–15.

[6] M. Weber, M. Welling, P. Perona, Towards automatic discovery of object categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 101–108.

[7] R. Fergus, P. Perona, A. Zisserman, A sparse object category model for efficient learning and exhaustive recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 380–387.

[8] W. Zhang, B. Yu, G.J. Zelinsky, D. Samaras, Object class recognition using multiple layer boosting with multiple features, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 323–330.

[9] S.K. Nayar, H. Murase, S.A. Nene, Parametric appearance representation, in: S.K. Nayar, T. Poggio (Eds.), Early Visual Learning, 1996, pp. 131–160.

[10] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[11] E. Bart, E. Byvatov, S. Ullman, View-invariant recognition using corresponding object fragments, in: Proceedings of European Conference on Computer Vision, vol. 2, 2004, pp. 152–165.

[12] M. Fischler, R. Elschlager, The representation and matching of pictorial structures, IEEE Transactions on Computers 22 (1) (1973) 67–92.

[13] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.

[14] A. Saxena, M. Sun, A.Y. Ng, Learning 3-D scene structure from a single still image, in: Proceedings of International Conference on Computer Vision, Workshop on 3D Representation for Recognition, 2007.

[15] H. Chen, Z. Xu, Z. Liu, S.C. Zhu, Composite templates for cloth modeling and sketching, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 943–950.

[16] F. Han, S.C. Zhu, Bottom-up/top-down image parsing by attribute graph grammar, in: Proceedings of International Conference on Computer Vision, vol. 2, 2005, pp. 1778–1785.

[17] L. Lin, S. Peng, J. Porway, S.C. Zhu, Y. Wang, An empirical study of object category recognition: sequential testing with generalized samples, in: Proceedings of International Conference on Computer Vision, vol. 1, 2007, pp. 353–360.

[18] K.S. Fu, Syntactic Pattern Recognition and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[19] S. Dickinson, A. Pentland, A. Rosenfeld, From volume to views: an approach to 3D object recognition, CVGIP: Image Understanding 55 (2) (1992) 130–154.

[20] Y. Keselman, S. Dickinson, Generic model abstraction from examples, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 1141–1156.

[21] Y. Ohta, Knowledge-based Interpretation of Outdoor Natural Color Scenes, Pitman, London, 1985.

[22] L. Zhu, Y. Chen, A. Yuille, Unsupervised learning of a probabilistic grammar for object detection and parsing, in: Proceedings of Advances in Neural Information Processing Systems, issue 19, 2008, pp. 827–834.

[23] K. Mark, M. Miller, U. Grenander, Constrained stochastic language models, in: S.E. Levinson, L. Shepp (Eds.), Image Models (and Their Speech Model Cousins), IMA Volumes in Mathematics and its Applications, Minneapolis, 1994.

[24] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading, MA, 1984.

[25] J. Porway, B. Yao, S.C. Zhu, Learning compositional models for object categories from small sample sets, in: S. Dickinson, A. Leonardis, B. Schiele, M. Tarr (Eds.), Object Categorization: Computer and Human Vision Perspectives, 2008.

[26] I. Biederman, Recognition-by-components: a theory of human image understanding, Psychological Review 94 (1987) 115–147.

[27] Z. Tu, S.C. Zhu, Image segmentation by data-driven Markov chain Monte Carlo, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 657–673.

[28] V. Ferrari, T. Tuytelaars, L.V. Gool, Object detection by contour segment networks, in: Proceedings of European Conference on Computer Vision, vol. 3, 2006, pp. 14–28.

[29] S.C. Zhu, D. Mumford, Stochastic grammar of images, Foundations and Trends in Computer Graphics and Vision 2 (4) (2006) 259–362.

[30] L. Lin, S.C. Zhu, Y. Wang, Layered graph match with graph editing, in: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, vol. 1, 2007, pp. 885–892.

[31] Y.N. Wu, Z. Si, C. Fleming, S.C. Zhu, Deformable template as active basis, in: Proceedings of International Conference on Computer Vision, 2007.

[32] C.E. Guo, S.C. Zhu, Y.N. Wu, Towards a mathematical theory of primal sketch and sketchability, in: Proceedings of International Conference on Computer Vision, vol. 2, 2003, pp. 1228–1235.

[33] W. Li, V. Piech, C.D. Gilbert, Perceptual learning and top-down influences in primary visual cortex, Nature Neuroscience 7 (6) (2004) 651–657.

[34] B. Yao, X. Yang, S.C. Zhu, Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, in: Lecture Notes in Computer Science, vol. 4697, Springer, Berlin, 2007, pp. 169–183.

[35] Z. Tu, Probabilistic boosting tree: learning discriminative models for classification, recognition, and clustering, in: Proceedings of International Conference on Computer Vision, vol. 2, 2005, pp. 1589–1596.

About the Author—LIANG LIN was born in 1981. He received the B.S. and Ph.D. degrees in Beijing Institute of Technology (BIT) in 1999 and 2008, respectively. He worked in the Center for Image and Vision Science (CIVS) of the University of California, Los Angeles as a visiting scholar in 2006–2007. He is now a post-doc research candidate in UCLA, and a research scientist in Lotus Hill Institute (http://www.lotushill.org). His research interests include but are not limited to computer vision, image understanding, and augmented reality.

About the Author—TIANFU WU received his master degree from HeFei University of Technology in 2005. He worked in the Lotus Hill Institute as a research assistant. Now he is a Ph.D. candidate in the department of Statistics at the University of California, Los Angeles. His major research interests are object recognition and image parsing.

About the Author—JAKE PORWAY received his B.S. in Computer Science from Columbia University in 2004 where he focused on intelligent systems. Now he is a Ph.D. candidate in the department of Statistics at the University of California, Los Angeles, working in the Center for Image and Vision Sciences. His major research interests are object recognition, image understanding, and hierarchical modeling.

About the Author—ZIJIAN XU received the B.E. degree from the Department of Computer Science and Technology, University of Science and Technology of China in 2001 and the doctorate degree from the Statistics Department, University of California, Los Angeles, in 2007, where he studied and researched on computer vision and was supervised by Professor Song-Chun Zhu. He is currently working with Moody's Corp. His research interests include but are not limited to statistical modeling, computer vision, and machine learning.