INVITED PAPER

I2T: Image Parsing to Text Description

The key component of this image parsing system is a database of annotated Internet data generated under the control of an independent organization.

By Benjamin Z. Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu

ABSTRACT | In this paper, we present an image parsing to text description (I2T) framework that generates text descriptions of image and video content based on image understanding. The proposed I2T framework follows three steps: 1) input images (or video frames) are decomposed into their constituent visual patterns by an image parsing engine, in a spirit similar to parsing sentences in natural language; 2) the image parsing results are converted into semantic representation in the form of Web ontology language (OWL), which enables seamless integration with general knowledge bases; and 3) a text generation engine converts the results from previous steps into semantically meaningful, human readable, and query-able text reports. The centerpiece of the I2T framework is an and-or graph (AoG) visual knowledge representation, which provides a graphical representation serving as prior knowledge for representing diverse visual patterns and provides top-down hypotheses during the image parsing. The AoG embodies vocabularies of visual elements including primitives, parts, objects, scenes as well as a stochastic image grammar that specifies syntactic relations (i.e., compositional) and semantic relations (e.g., categorical, spatial, temporal, and functional) between these visual elements. Therefore, the AoG is a unified model of both categorical and symbolic representations of visual knowledge. The proposed I2T framework has two objectives. First, we use semiautomatic method to parse images from the Internet in order to build an AoG for visual knowledge representation. Our goal is to make the parsing process more and more automatic using the learned AoG model. Second, we use automatic methods to parse image/video in specific domains and generate text reports that are useful for real-world applications. In the case studies at the end of this paper, we demonstrate two automatic I2T systems: a maritime and urban scene video surveillance system and a real-time automatic driving scene understanding system.

KEYWORDS | And-or graph (AoG); image parsing; retrieval; text generation

I. INTRODUCTION

A. I2T Overview

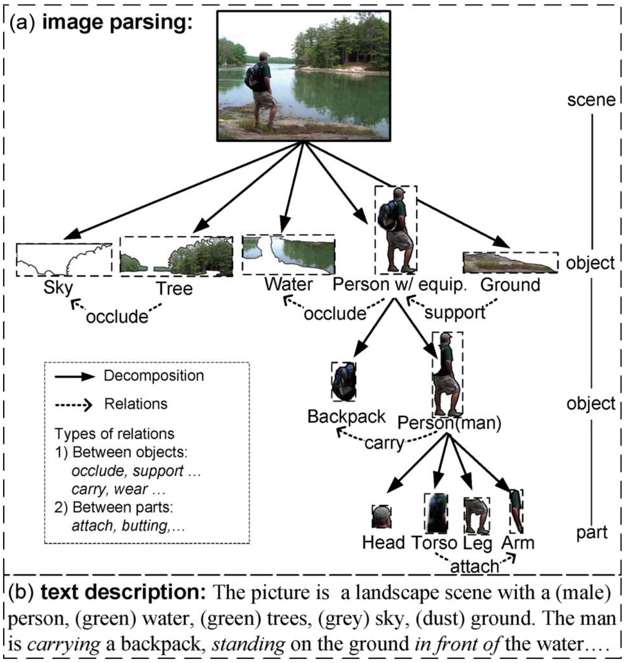
Fast growth of public photo and video sharing websites, such as "Flickr" and "YouTube," provides a huge corpus of unstructured image and video data over the Internet. Searching and retrieving visual information from the Web, however, has been mostly limited to the use of metadata, user-annotated tags, captions, and surrounding text (e.g., the image search engine used by Google [1]). In this paper, we present an image parsing to text description (I2T) framework that generates text descriptions in natural language based on understanding of image and video content. Fig. 1 illustrates two major tasks of this framework, namely, image parsing and text description. By analogy to natural language understanding, image parsing computes a parse graph of the most probable interpretation of an input image. This parse graph includes a tree structured decomposition for the contents of the scene, from scene labels, to objects, to parts and primitives, so

Manuscript received April 27, 2009; revised December 9, 2009; accepted May 5, 2010. Date of publication June 17, 2010; date of current version July 21, 2010. This work was supported by the National Science Foundation (NSF) under Grant IIS-0713652 on image parsing and the Office of Naval Research (ONR) under Grant N00014-07-M-0287. The work at Lotus Hill Institute was supported by two National 863 projects 2006AA01Z121 and 2007AA01Z340.

B. Z. Yao was with the Lotus Hill Institute, Ezhou 436000, China. He is now with the Department of Statistics, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: zyyao@stat.ucla.edu).
X. Yang is with the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: xyang.lhi@gmail.com).
L. Lin was with the Lotus Hill Institute, Ezhou 436000, China. He is now with the Laboratory of Intelligent Information Processing, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: sailalone@gmail.com).
M. W. Lee is with ObjectVideo, Inc., Reston, VA 20191-1410 USA (e-mail: MLee@ObjectVideo.com).
S.-C. Zhu is with the Lotus Hill Institute, Ezhou 436000, China and the Department of Statistics, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: sczhu@stat.ucla.edu).

Digital Object Identifier: 10.1109/JPROC.2010.2050411

0018-9219/$26.00 ©2010 IEEE



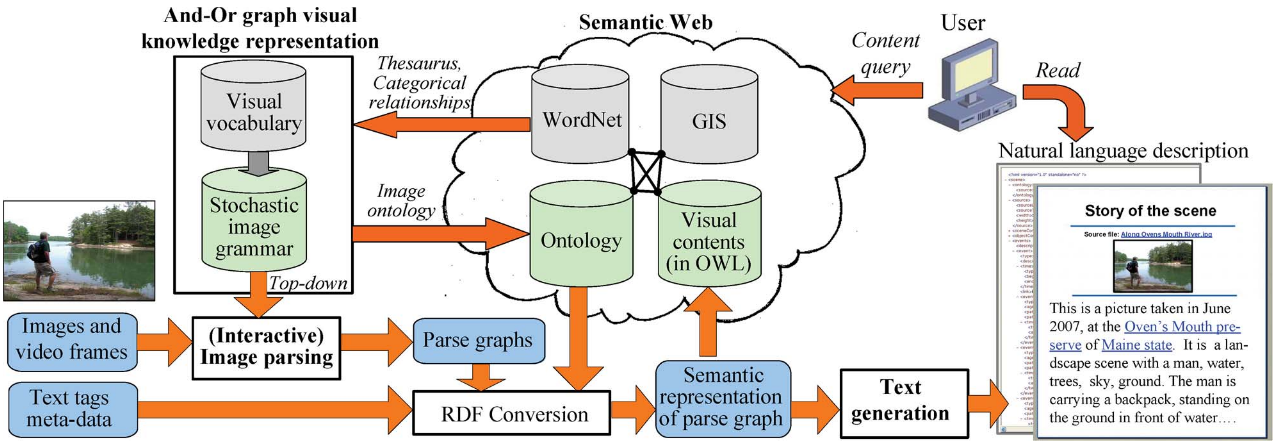
(b) text description: The picture is a landscape scene with a (male) person, (green) water, (green) trees, (grey) sky, (dust) ground. The man is *carrying* a backpack, *standing* on the ground *in front of* the water....

Fig. 1. ***Two major tasks of the I2T framework: (a) image parsing and (b) text description. See text for more details.***

that all pixels are explained. It also has a number of spatial and functional relations between nodes for context at all levels of the hierarchy. The parse graph is similar in spirit to the parsing trees used in speech and natural language understanding [2] except that it can include horizontal connections [see the dashed curves in Fig. 1(a)] for specifying relationships and boundary sharing between different visual patterns. From a given parse graph, the task of text description is to generate semantically meaningful, human readable, and query-able text reports as illustrated in Fig. 1(b).

To achieve the goal illustrated in Fig. 1, we propose an I2T framework, which has four major components as shown in Fig. 2.

1) **An image parsing engine** that parses input images into parse graphs. For specific domains such as the two case study systems presented in Section VII, the image/video frame parse is automatic. For parsing general images from the Internet for the purpose of building a large scale image data set, an interactive image parser (IIP) is used as discussed in Section III-B.
2) **An and–or graph (AoG) visual knowledge representation** that embodies vocabularies of visual elements including primitives, parts, objects, and scenes as well as a stochastic image grammar that specifies syntactic (compositional) relations and semantic relations (e.g., categorical, spatial, temporal, and functional relations) between these visual elements. The categorical relationships are inherited from WordNet, a lexical semantic network of English [3]. The AoG not only guides the image parsing engine with top–down hypotheses but also serves as an ontology for mapping parse graphs into semantic representation (formal and unambiguous knowledge representation [4]).
3) **A semantic web** [5] that interconnects different domain specific ontologies with semantic representation of parse graphs. This step helps to enrich parse graphs derived purely from visual cues with other sources of semantic information. For example, the input picture in Fig. 2 has a text



Story of the scene

Source file: Along Ovens Mouth River.jpg

This is a picture taken in June 2007, at the Oven's Mouth preserve of Maine state. It is a landscape scene with a man, water, trees, sky, ground. The man is carrying a backpack, standing on the ground in front of water....

Fig. 2. ***Diagram of the I2T framework. Four key components (as highlighted by bold fonts) are: 1) an image parsing engine that converts input images or video frames into parse graphs; 2) an AoG visual knowledge representation that provides top–down hypotheses during image parsing and serves as an ontology when converting parse graphs into semantic representations in RDF format; 3) a general knowledge base embedded in the semantic web that enriches the semantic representations by interconnecting several domain specific ontologies; and 4) a text generation engine that converts semantic representations into human readable and query-able natural language descriptions.***

tag "Oven's mouth river." With the help of a geographic information system database embedded in the semantic web, we are able to relate this picture to a geolocation: "Oven's mouth preserve of Maine state." Another benefit of using semantic web technology is that end users not only can access the semantic information of an image by reading the natural language text report but can also query the semantic web using standard semantic querying languages.

4) **A text generation engine** that converts semantic representations into human readable and queryable natural language descriptions. We will come back to discuss these components in more detail in Sections I-C–E.

As simple as the I2T task in Fig. 1 may seem to be for a human, it is by no means an easy task for any computer vision system today—especially when input images are of great diversity in contents (i.e., number and category of objects) and structures (i.e., spatial layout of objects), which is certainly the case for images from the Internet. But given certain controlled domains, for example, the two case study systems presented in Section VII, automatic image parsing is practical. For this reason, our objective in this paper is twofold.

1) We use a semiautomatic method (interactive) to parse general images from the Internet in order to build a large scale ground truth image data set. Then, we learn the AoG from this data set for visual knowledge representation. Our goal is to make the parsing process more and more automatic using the learned AoG models.
2) We use automatic methods to parse images/videos in specific domains. For example, in the surveillance system presented in Section VII-A, the camera is static, so we only need to parse the background (interactively) once at the beginning, and all other components are done automatically. In the automatic driving scene parsing system discussed in Section VII-B, the camera is forward looking at roads and streets. Although the image parsing algorithm may produce some errors, it is fully automatic.

B. Previous Work

Over the past two decades, many researchers from both computer vision and content-based image retrieval (CBIR) domain have been actively investigating possible ways of retrieving images and video clips based on low-level visual features such as color, texture, and object shape. A number of domain-specific CBIR systems have achieved success (see surveys [6]–[10]), but these CBIR systems cannot compete with human visual system in terms of robustness, flexibility, and shear number of object categories recognizable. The major challenge is a so-called semantic gap [6], which is defined as the discrepancy between human interpretations of image information and those currently derivable by a computer.

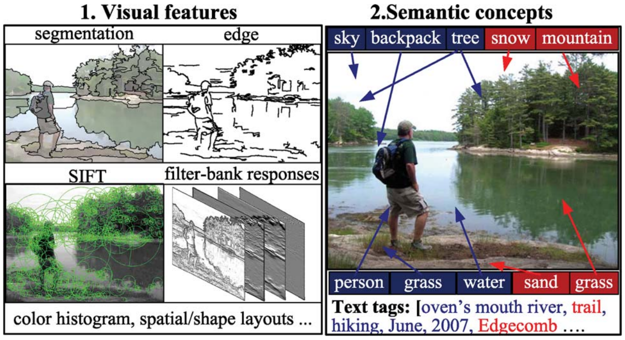


Fig. 3. ***Categorical representation of visual contents. Task1. Extract low level visual features from raw images. Task2. Map visual features into semantic concepts. Concepts in blue color are correct ones that are related to the image content. Concepts in red are irrelevant, and are generated by either classification error or unrelated text tags.***

From an artificial intelligence (AI) point of view, bridging the semantic gap is equivalent to solving a visual symbol grounding problem [11]. Therefore, as suggested by Harnad [12], we may further decompose the symbol grounding problem in the visual domain into two levels.

- *Categorical representations*, which are learned and innate feature detectors that pick out the invariant features of object and event categories from their sensory projections (images). Each category corresponds to an elementary symbol (visual element).
- *Symbolic representations*, which consist of symbol strings describing semantic relations between elementary symbols, such as category membership relations (e.g., a *zebra* is a *horse* that has *stripes*), functional relations (e.g., in Fig. 1, the *man* is the *owner* of the *backpack*), and so on. With these semantic relationships, basic elementary symbols grounded in categorical representations (e.g., *horse* and *stripe*) can be used to compose new grounded symbols using rules (e.g., $zebra = horse + stripes$).

Previous work can be roughly classified into two groups accordingly.

Group 1 studies the categorical representation for objects and events from visual signals (e.g., object recognition from images). This has been the mainstream for computer vision research for the past decades. Fig. 3 illustrates a rough but intuitive idea about two major tasks of the categorical representation:

- *Task1*: Extract low-level visual features such as coherent regions (segmentation) [13]–[15], salient edges [16], [17], filter-bank responses (textons) [18], [19], SIFT descriptors [20], color histograms

[1] A complete reference will essentially include all computer vision literature. We only list a few representative methods as examples.

[21], shape descriptors [22], and scene layout [23] from raw images.[1]

- *Task2*: Map these low-level visual features into high-level semantic concepts. For example, detecting and recognizing the pedestrian from Fig. 3 requires a combination of edge features and shape information [24].

Extensive previous work in image annotation (e.g., [25]–[27]) and video annotation (e.g., reported under the TREC Video Retrieval Evaluation program (TRECVID) [28]) has been mainly focused on addressing categorical representation, but none of them performs full segmentation and recognition over an entire image simultaneously. Also, as illustrated by Fig. 3, local concepts are prone to error due to a lack of global context. For example, the tree crown and white sky in Fig. 3 are easily confused with a mountain covered in snow. This kind of inconsistency can be mitigated by modeling scene context and relations between objects, which is a relatively new topic in computer vision. A few recent approaches have proposed interesting models for image understanding using contextual information [29]–[32]. The image parsing algorithm in the I2T framework is similar to these algorithms in spirit.

Group 2 pays attention to the symbolic representation, i.e., semantic relations between visual symbols. Marszalek and Schmid used semantic hierarchies (categorical relations) of WordNet to integrate prior knowledge about interclass relationships into the visual appearance learning [33]. Following this line, J. Deng *et al.* launched an ImageNet project aiming to populate a majority of the synsets in WordNet with an average of 500–1000 images selected manually by humans [34]. While the semantic hierarchy provided by WordNet is very useful to shed light on interclass relationships between visual elements, we believe, however, it is not sufficient for modeling symbolic representation of a visual system. On the one hand, it does not account for some important aspects of semantic relationships such as co-occurrence relations studied in [35], spatial relationships [30], [32], and functional relationships. On the other hand, many semantic relationships in the WordNet hierarchy are purely conceptual with absolutely no correspondence in visual domain. For example, in WordNet, the "combustion engine" is connected with the "car" by the is-part-of relation. For most people (except car mechanics), however, this relationship has no visual evidence and thus cannot provide any useful information to recognize motorized vehicle from images. WordNet is, after all, a lexical dictionary designed for language, not vision. In this paper, we propose a different approach from the ImageNet database. As discussed in more detail in Section II-A, the semantic relations of the AoG are learned from our own large scale ground truth image database. Therefore, the semantic relationships in the AoG are grounded to image or video instances.

To properly evaluate the contribution of the I2T framework, we would also like to review literature related to image parsing. Early work on image parsing can be dated back to Fu's syntactic image grammar in the 1970s [37]. Due to difficulties posed by the semantic gap and limitation in computation power, image grammar work at that time was limited to artificial images such as line drawings. With great developments in both appearance-based modeling and computation strategies, the recent vision literature has observed a trend for returning to the grammatical and compositional methods. In a series of work [14], [38]–[40], Zhu and his collaborators posed image parsing as a unified framework for image segmentation, object detection, and recognition. The image parsing problem is then solved optimally under the Bayesian framework by a data-driven Markov chain Monte Carlo (DDMCMC) algorithm. A dedicated survey paper [36] reviewed histories and recent progresses of image parsing and summarized them under a unified stochastic image grammar framework.

C. AoG Visual Knowledge Representation

Image parsing has been previously demonstrated in a small number of images and object categories [14], but the difficulty and complexity of this task grows significantly as the number of object and scene categories increases. A key problem here is how to represent and organize a huge number of diverse visual patterns systematically and effectively. Objects with similar semantic meaning may differ greatly in image appearance. For example, an ontology engineer will probably classify "round clock" and "square clock" into the same category "clock," but as shown in Fig. 4, these two types of clocks have dramatically different visual look. This is a similar problem to the "semantic ambiguity" in natural language processing (NLP), which can be solved by defining multiple senses for a single word. For example, in WordNet, there are five alternative senses for the noun "car": 1) automobile; 2) rail car; 3) cable car; 4) gondola of a balloon; and 5) elevator car. In the same spirit, we introduce an AoG representation, first proposed in [41],[2] that is able to describe all possible structural variabilities of visual object categories.

To illustrate the basic idea of the AoG, let us first look at the two parse graphs of clocks shown in the left panel of Fig. 4. We may notice that both clocks share the component "frame" and "hands," and a couple of relations: the hinge joint to connect clock hands and a concentric relation to align the frames. Therefore, by summarizing all shared components and relations from a number of parse graphs of clocks, we may arrive at a combination graph as shown in the right panel of Fig. 4, which is called an AoG because it has or-nodes pointing to alternative subconfigurations and and-nodes decomposing into a number of components. It is a graph (as opposed to a tree) because there are links between or-nodes representing shared

[2] The term "and–or graph" has been previously used by Pearl in AI search and is not related to the AoG discussed here.

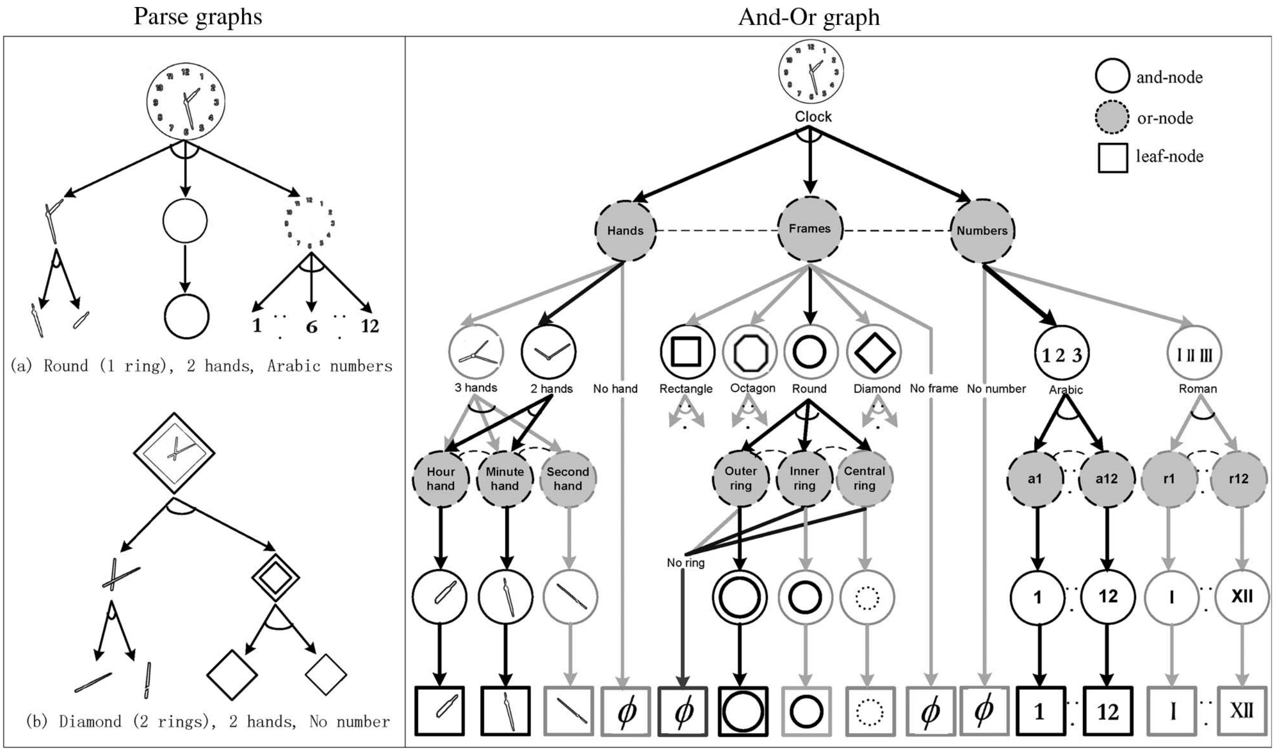


Fig. 4. *Categorical representation with AoG. The left panel illustrates two parse graphs of clocks. The right panel shows an AoG of a clock generated from merging multiple parse graphs together. The dark arrows in the AoG illustrate one parse graph (a round clock). Some leaf nodes are omitted from the graph for clarity. Edited from [36].*

relations. The relations shown in this example are mainly spatial/geometric relationships between object parts. For example, the "frame" and the "numbers" share a *concentric* relation, and the "hands" are connected by a *hinged* relation. These relations can be learned from parse graphs as they are commonly shared by most of the examples. The "clock" example provides a way for using an AoG to represent images of an object category.

In addition, to achieve the visual symbol grounding problem described in the previous section, there are still two challenges to overcome. First, we need to collect a set of image examples for each visual element (symbol) in order to learn an AoG model that grounds the symbol into visual signals. Second, we need to discover semantic relationships between these visual elements in order to build the symbolic representation of visual knowledge. The solution to the first challenge is to build a large scale ground truth image data set, which is discussed in the next section. The second challenge is handled by the design of the AoG model. As we will discuss in Section II, the relationships of the AoG are not limited to the spatial/geometric relationships shown in the "clock" example above; they also include object-level semantic relationships such as categorical membership relationships, functional relationships, etc. These relationships can also be learned from the large scale ground truth image data set.

D. Image Database and Interactive Image Parsing

Building an image data set with manually annotated parse graphs provides training examples needed for learning the categorical image representation in the AoG model. Properly annotated data sets also provide training examples needed for learning semantic relationships. This data set must be large scale in order to provide enough instances to cover possible variations of objects. To meet this need, S.-C. Zhu founded an independent nonprofit research institute called Lotus Hill Institute (LHI, Ezhou, China), which started to operate in summer 2005. It has a full time annotation team for parsing image structures and a development team for annotation tools and database construction. Each image or object is parsed interactively into a parse graph where objects are associated with WordNet synsets to inherit categorical relationships. Functional relationships such as "carry" and "eat" are also specified manually. Fig. 10 lists an inventory of the current ground truth data set parsed at LHI. It now has over 1 million images (and video frames) parsed, covering about 300 object categories. To cope with the need of labeling tens of thousands of images, an interactive image parsing software, named IIP, was developed to facilitate the manual annotation task (see more details in Section III-B). As stated in a report [42], this data set provides ground truth annotation for a range of vision tasks from high-level scene classification and object segmentation to low-level edge detection and edge attributes annotation. Comparing with other public data sets collected in various groups, such as MIT LabelMe [43], ImageNet [34], the MSRC data set [44], Caltech 101 and 256 [45], [46], and Berkeley segmentation [47], [48], the LHI data set not only provides finer segmentation but also provides extra information such as compositional hierarchies and functional relationships.

E. Image Ontology and Text Generation

In a knowledge sharing context, the term *ontology* stands for a "formal, explicit specification of a shared conceptualization" [4]. As discussed by Town [11], the most important issue for designing an image ontology is how to provide a systematic way to ground the terms of the ontology to the actual visual data. As the AoG provides a unified model for both categorical and symbolic representation, it is reasonable to believe that the AoG can serve as a domain-specific image ontology.

Using the AoG, the image content inferred by image parsing can be expressed by a standard semantic representation language called the resource description framework (RDF). Another semantic representation language called the video event markup language (VEML) [49] is also used for annotating video events. Both of these languages can be merged into the Web ontology language (OWL) [5], [50]. Using OWL representation, an ontology engineer can declare how knowledge concepts defined in different ontologies are related. With this ontological mapping, multiple OWL documents can be interconnected. This promotes reuse of existing ontologies and encourages the development of domain-specific ontologies to address the diverse needs of different applications. With this framework, image and video content can be published on the semantic web and allow various semantic mining and inference tools to retrieve, process, and analyze video content. Some of the semantic concepts such as object classes can be mapped to well-established concepts defined in general knowledge bases such as open geospatial consortium (OGC) and WordNet. This improves the accessibility and portability of the inferred video content. The whole integrated system is illustrated in Fig. 2.

While OWL provides an unambiguous representation for image and video content, it is not easy for humans to read. Natural language text remains the best way for describing visual content to humans and can be used for image captions, scene descriptions, and event alerts. In the later sections, we will further present text generation techniques that convert the semantic representation to text description using natural language generation (NLG) techniques.

F. Outline of the Paper

The remainder of the paper is organized as follows. In Section II, we discuss the AoG in detail, including representation, inference, and statistics. In Section III, we introduce several improvements in automatic image parsing algorithms guided by the AoG model as well as the IIP software that integrates semiautomatic algorithms such as interactive segmentation, shape matching, etc., with human annotations. In Section IV, we introduce the data structure and design of the LHI data set. In Section V, we discuss in detail the semantic representation of visual content, including how to derive it from parse graphs and how to integrate it with the general knowledge base. In Section VI, an NLG algorithm is introduced. In Section VII, we introduce two case studies of the proposed I2T framework: a real-time automatic driving scene understanding system and a maritime and urban scene video surveillance system.

II. AoG REPRESENTATION

The AoG is a compact yet expressive data structure for representing diverse visual patterns of objects (such as a clock). In this section, we will formally define the AoG as a general representation of visual knowledge, which entails 1) a stochastic attribute image grammar specifying compositional, spatio–temporal and functional relations between visual elements; and 2) vocabularies of visual elements of scenes, objects, parts, and image primitives.

A. Stochastic Image Grammar

The AoG representation embodies a stochastic attributed image grammar. Grammars, studied mostly in language [52], are known for their expressive power to generate a very large set of configurations or instances (i.e., their language) by composing a relatively small set of words (i.e., shared and reusable elements) using production rules. Therefore, the image grammar is a parsimonious yet expressive way to account for structural variability of visual patterns. We use Fig. 5 as an example to review the representational concepts of the image grammar in the following.

1) **An and–or tree** is an AoG without horizontal connections (relations). As shown in Fig. 5(a), an and–or tree includes three types of nodes: and-nodes (solid circles), or-nodes (dashed circles), and terminal nodes (squares). An and-node represents a decomposition of an entity into its parts. There are two types of decompositions: a) object → parts, and b) scene → objects. The object → parts decomposition has a fixed number of child nodes, which correspond to the grammar rules, for example:

$$A \longrightarrow BCD, \quad H \longrightarrow NO.$$

This is equivalent to other part-based models such as the constellation model [53] and the pictorial structures [54].

The scene → objects decomposition has a variable number of child nodes, which correspond to the grammar rules, for example:

$$A \longrightarrow B(n_1)C(n_2)D(n_3)$$

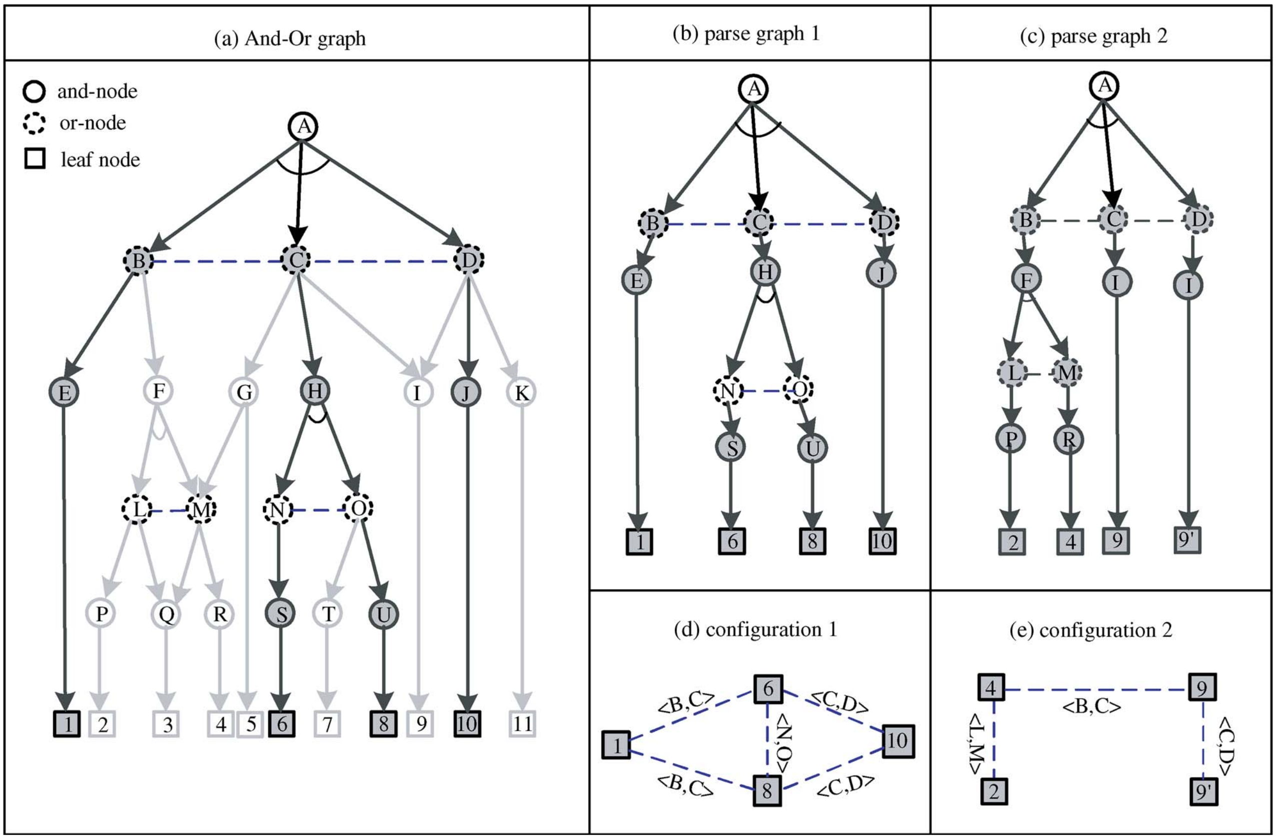


Fig. 5. *Illustrating the AoG representation: (a) an AoG embodies the grammar productions rules and contexts; it contains many parse graphs, one of which is shown in bold arrows; (b) and (c) are two distinct parse graphs obtained by selecting the switches at related or-nodes; (d) and (e) are two graphical configurations produced by the two parse graphs, respectively. The links of these configurations are inherited from the AoG relations. From [36].*

where n_i denotes the number of occurrences of a type of object in a scene. One example of the aerial image scene parsing is illustrated in Fig. 6. We can use this model to create a vast amount of unique scene configurations by sampling as shown in Fig. 7. This model is similar (and can be converted) to the model presented by Sudderth *et al.* [55], [56]. For more details, refer to [51].

The or-nodes act as "switches" for alternative substructures, and stand for labels of classification at various levels, such as scene category, object classes, parts, etc. They correspond to production rules like

$$B \longrightarrow E|F, \quad C \longrightarrow G|H|I.$$

Due to this recursive definition, one may merge the AoGs for many objects or scene categories into a larger graph. In theory, all scene and object categories can be represented by one huge AoG, as is the case for natural language. The nodes in an AoG may share common parts. For example, both cars and trucks have rubber wheels as parts, and both clock and pictures have frames.

2) **Relations** represent the horizontal links for contextual information between the children of an and-node in the hierarchy at all levels, as shown by the dashed lines in Fig. 5(a)–(c). Each link may represent one or several relations. There are three types of relations of increasing abstraction for the horizontal links and context. The first type is the bond type that connects image primitives into bigger and bigger graphs. The second type includes various joints and grouping rules for organizing the parts and objects in a planar layout. The third type is the functional and semantic relation between objects in a scene.

- *Relations Type 1: Bonds and Connections (Relation Between Image Primitives)*. Leaf nodes of the and–or tree are image primitives. Fig. 8(i-a) illustrates a dictionary of image primitives called "texton" (blobs, bars, terminators, and junctions) as proposed in [57]. The type 1 relations are bonds and connections between image primitives. Each image primitive has a number of open bonds, shown by the half-disks, to connect with others to form bigger image patterns. For instance, Fig. 8(i-b)

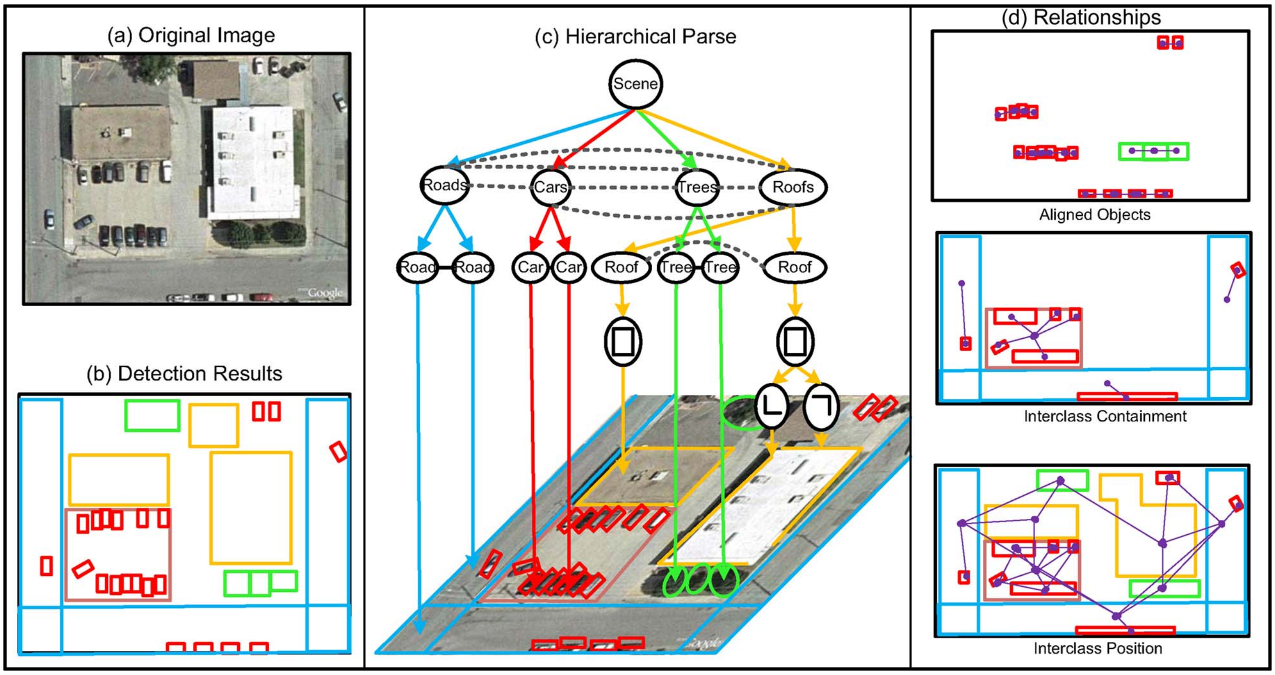


Fig. 6. *An example of a hierarchically parsed aerial image. (a) The original image. (b) A flat configuration of objects in the scene. (c) A hierarchical parse graph of the scene. (d) Three typical contextual relationships and related objects. From [51].*

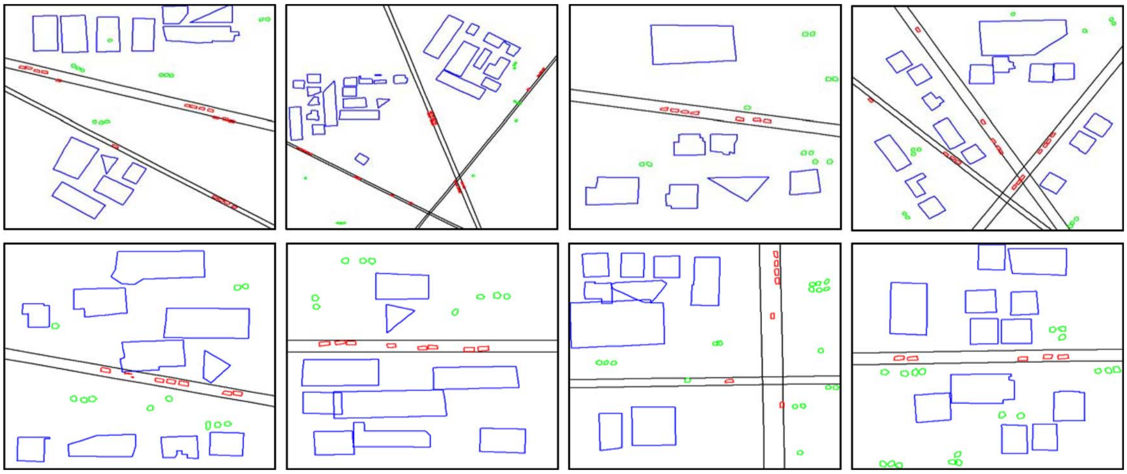


Fig. 7. *Samples from our learned model (blue = roofs, red = cars, black = roads, green = trees). These are not images directly sampled from the training data, but collections of objects obeying the statistics of our learned model. We can create a vast amount of unique object configurations even though we have never observed them directly. Adapted from [51].*

shows an example composing a big "T"-shape image using seven primitives. Two bonds are said to be connected if they are aligned in position and orientation. The bonds and connections can be considered as random variables pointing to neighbors of an image primitive forming a mixed random field (see details in [58]).

- *Relations Type 2: Joints and Junctions (Relation Between Object Parts)*. When image primitives are connected into larger parts, some spatial and functional relations must be found. Beside its open bonds to connect with others, usually its immediate neighbors, a part may be bound with other parts in various ways. Parts can be linked over large distances by collinear, parallel, or symmetric relations. This is a phenomenon sometimes called gestalt grouping. Fig. 8 displays some typical relations of this type between object parts.

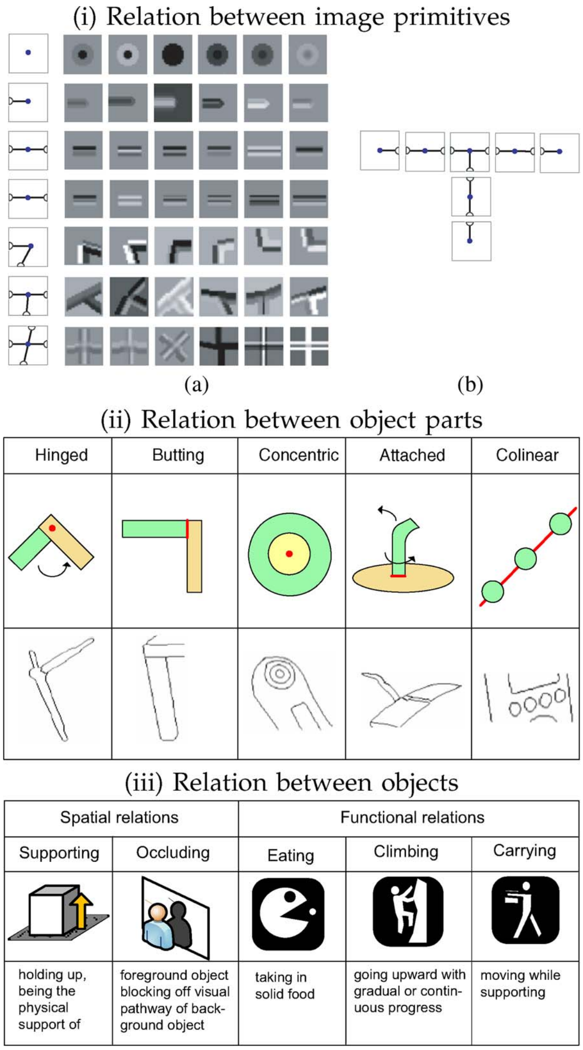

(i) Relation between image primitives
(a)
(b)
(ii) Relation between object parts
Hinged
Butting
Concentric
Attached
Colinear
(iii) Relation between objects
Spatial relations
Functional relations
Supporting
Occluding
Eating
Climbing
Carrying
holding up, being the physical support of
foreground object blocking off visual pathway of back-ground object
taking in solid food
going upward with gradual or contin-uous progress
moving while supporting

Fig. 8. ***Examples of three types of relations in the AoG. Modified from [36].***

- *Relations Type 3: Interactions and Semantics (Relation Between Objects)*. When letters are grouped into words, semantic meanings emerge. When parts are grouped into objects, semantic relations are created for their interactions. Very often these relations are directed. For example, the occluding relation is viewpoint dependent binary relation between objects or surfaces, and it is important for figure-ground segregation. A supporting relation is a view point independent relation. There are other functional relations among objects in a scene. For example, a person is carrying a backpack, and a person is eating an apple. These directed relations usually are partially ordered. Examples of object interaction relations are shown in the parse graph of the outdoor scene shown at the beginning of this paper (Fig. 1).

3) **A parse graph**, as shown in Fig. 1, is a hierarchical generative interpretation of a specific image. A parse graph is augmented from a parse tree, mostly used in natural or programming language by adding a number of relations, shown as side links, among the nodes. A parse graph is derived from the AoG by selecting the switches or classification labels at related or-nodes. Fig. 5(b) and (c) shows two instances of the parse graph from the AoG in Fig. 5(a). The part shared by two nodes may have different instances, for example, node I is a child of both nodes C and D. Thus, we have two instances for node 9.

4) **A configuration** is a planar attribute graph formed by linking the open bonds of the primitives in the image plane. Fig. 5(d) and (e) shows two configurations produced by the parse graphs in Fig. 5(b) and (c), respectively. Intuitively, when the parse graph collapses, it produces a planar configuration. A configuration inherits the relations from its ancestor nodes, and can be viewed as a Markov network (or deformable template [59]) with a reconfigurable neighborhood. We introduce a mixed random field model [58] to represent the configurations. The mixed random field extends conventional Markov random field models by allowing address variables, which allows it to handle nonlocal connections caused by occlusions. In this generative model, a configuration corresponds to a primal sketch graph [17].

5) **The language** of a grammar is the set of all possible valid configurations produced by the grammar. In stochastic grammar, each configuration is associated with a probability. As the AoG is directed and recursive, the subgraph underneath any node A can be considered a subgrammar for the concept represented by node A. Thus, a sublanguage for the node is the set of all valid configurations produced by the AoG rooted at A. For example, if A is an object category, say a car, then this sublanguage defines all the valid configurations of a car. In an exiting case, the sublanguage of a terminal node contains only the atomic configurations and thus is called a dictionary.

B. Visual Vocabularies

Another important component of the AoG representation is the visual vocabulary. Language vocabulary is a collection of terms or codes composing its atomic elements (e.g., words). Similarly, a visual vocabulary consists of the terminal nodes of the AoG, which comprise atomic entities of visual patterns. Due to the scaling property, the terminal nodes could appear at all levels of the AoG, which is different from language, where the terminal nodes only appear at the lowest level. Each terminal node takes instances from a certain set. The set is called a dictionary and contains image patches of various complexities. The elements in the set may be indexed by variables such as type, geometric transformations, deformations, appearance changes, etc. Each patch is augmented with anchor points and open bonds to connect with other patches.

An example of visual vocabularies at each level of the AoG is shown in Fig. 9. For each dictionary, there is a set of or-nodes (solid ellipses) representing abstract visual

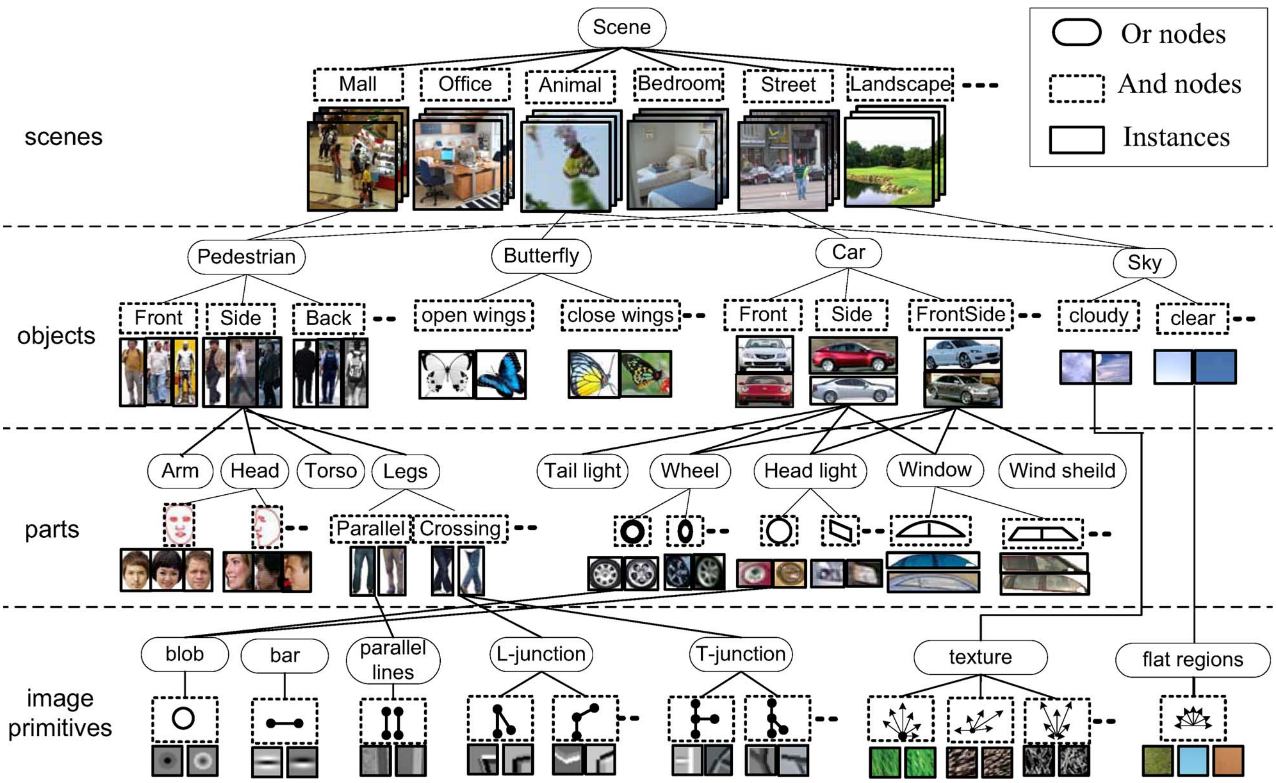


Fig. 9. *Visual vocabularies of scenes, objects, parts, and image primitives compiled into an AoG representation. Each or-node (sold ellipses) represents for a visual concept and is pointed to alternative subconfigurations. Each and-node (dashed rectangles) is associated with a set of instances (solid rectangles) from an annotated image database and can be decomposed into a number of components in a lower level. See text for detailed discussion of each dictionary. For clarity purposes, only part of the vertical links is shown.*

concepts. Each or-node is connected to a set of and-nodes (dashed rectangles) representing alternative subconfigurations of that concept. Each and-node is associated with a set of instances from real-world images. And-nodes can be further decomposed into a number of components of lower level. There are four types of visual dictionaries illustrated in Fig. 9. From top to bottom, they are scenes, objects, parts, and image primitives, respectively. We will now briefly discuss them.

- *Scene*. Human vision is known to be able to perform scene classification in a very short time $\leq$ 400 ms [60]. There are several previous works (such as [61]) proving that scene classification can be solved using simple global features. Therefore, the highest visual dictionary in the AoG is scene (see top level of Fig. 9 for example).
- *Objects* are the most important elements for understanding images. There has been much research on visual dictionaries of commonly seen object categories, such as face, car, pedestrians, etc. An example of an object dictionary is illustrated in the second level of Fig. 9.
- *Parts* have long been believed to be important visual elements for object detection, especially deformable objects. Examples of object parts are illustrated in the third level of Fig. 9. Another visual dictionary of human body parts is illustrated in Fig. 19.
- *Image primitives* are the leaf nodes of the AoG. They represent a set of image patches. The lowest level of Fig. 9 illustrates three commonly recognized image primitives. 1) Textons. As defined by Julesz in [62], textons are shape structure frequently observed in natural images, such as "blob," "bar," "parallel lines," and "junctions." Each texton is denoted with a symbolic sign. 2) Textures. Different types of texture are denoted with different filter response histograms (small arrows with different lengths). The length of the arrow represents the strength of the filter response at its orientation. 3) Flat regions. They are denoted with filter response histograms with even bars.

C. AoG Statistics

Aiming at providing ground truth annotation for a wide range of computer vision research, the LHI data set contains ground truth annotations for high-level tasks such as scene classification, object detection, aerial image understanding, text detection, and recognition, as well as low-level tasks such as edge detection and edge attribute annotation. Fig. 10 shows a recently updated inventory of

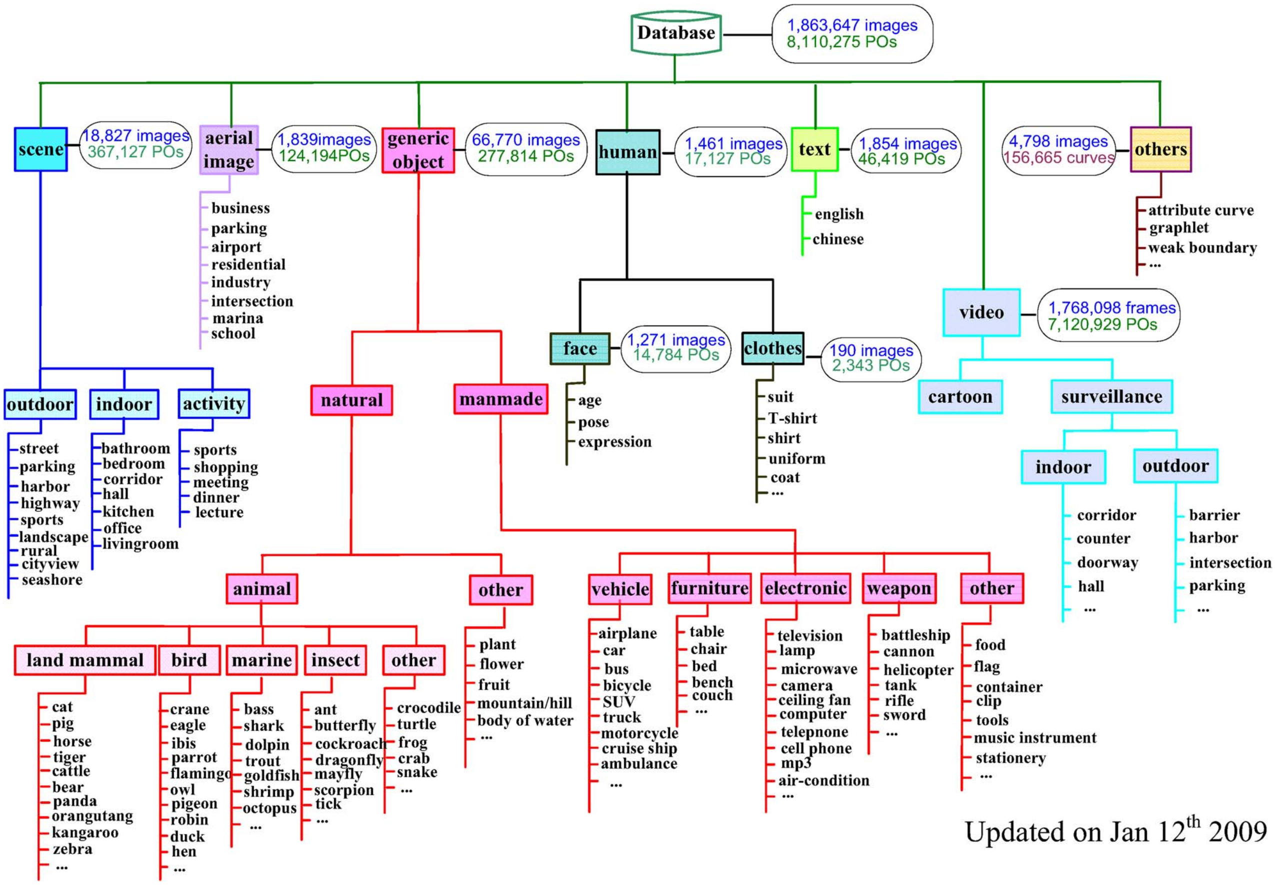
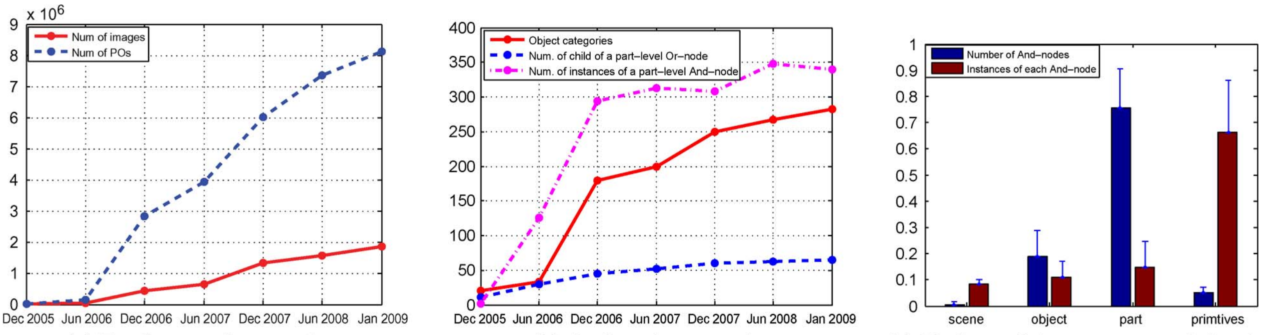


Fig. 10. ***Inventory of the LHI ground truth image database. Annotation examples of each category in this figure as well as a small publicly available data set can be found on at www.imageparsing.com.***

the LHI data set, where a physical object (PO) represents a meaningful entity in an image, such as an object or an object part. We further use Fig. 11 to study more statistics of the LHI data set. Fig. 11(a) displays the growth in image and PO number of the LHI data set over time. Fig. 11(b) shows the growth of the AoG over time. The solid line stands for the number of object categories. The dashed line is the average number of children of a part-level or-node, while the dash–dot line is the average number of instances of a part-level and-node. We may notice that the increase



(a) Database scale over time

(b) AoG nodes over time

(c) Nodes and instances at each level.

Fig. 11. ***Statistics of the LHI database. (a) Number of annotated images and POs over time. (b) Number of object categories over time, average number of children of a part-level or-node and average number of instances of a part-level and-node over time. (c) Average number of and-nodes and instances at "scene," "object," "part," and "primitive" levels, respectively (estimated from 20 object categories and normalized to 1).***

in or-nodes gradually slows down, which means that when there are a lot of instances, a new instance is more likely to merge into already existing subconfigurations, rather than create a new subconfiguration of its own. Fig. 11(c) compares the number of and-nodes and instances at "scene," "object," "part," and "primitive" levels. The numbers are estimated from 20 object categories (with a confidence margin marked on top of each bar). Numbers are normalized so that they sum to 1. This figure implies two facts that match common sense. 1) Different levels of visual elements are observed with different frequencies. Image primitives are observed most frequently. Scene level nodes have the smallest population. 2) Middle level nodes, i.e., objects and parts, are more "uncertain" and contain more varying patterns. Therefore, there are many more and-nodes of parts and objects than of scenes and primitives. It is also worth mentioning that primitives are not explicitly annotated in the data set; they are clustered from the annotated data using the method proposed in [63].

III. IMAGE PARSING GUIDED BY THE AoG

As discussed in the previous section, parse graphs can be derived from the AoG by selecting the switches or classification labels at related or-nodes. Image parsing can be interpreted as generating a parse graph from the AoG that best matches the input image. Automatic image parsing algorithms guided by the AoG have been discussed in detail in a survey paper [36]. Here we extend the discussion with several improvements, namely, a bottom–up/top–down inference algorithm from [64] and a cluster sampling algorithm from [51]. We also introduce an IIP, which is developed for building a large scale ground truth image data set effectively and accurately.

A. Automatic Image Parsing

1) Bottom–Up/Top–Down Inferences With AoG: We extend the previous algorithm in [40] to work on an arbitrary node A in an AoG, as illustrated in Fig. 12. We define three inference processes for each node A in an AoG.

1) *The α process.* The α process handles situations in which node A is at middle resolution without occlusion. Node A can be detected directly (based on its compact image data) and alone (without taking advantage of surrounding context) while its children or parts are not recognizable alone in cropped patches. An example of $\alpha(\text{face})$ process is shown in Fig. 13. Most of the sliding window detection methods in computer vision literature belong to this process. It can be either bottom–up or top–down in terms of whether discriminative models such as the Adaboost method [65] or generative models such as the active basis model [59] are used.

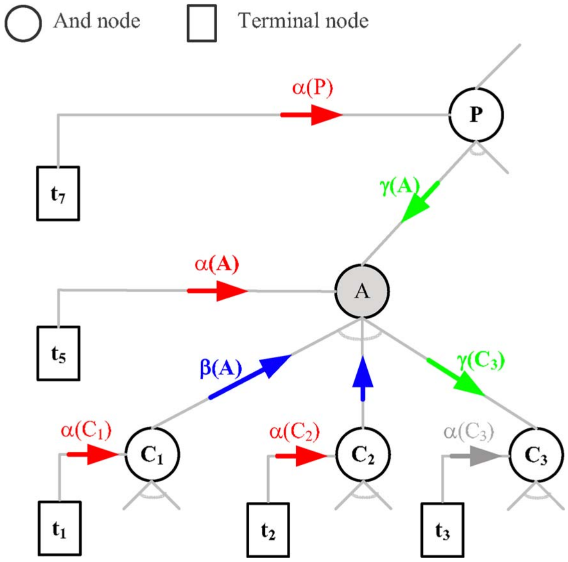


Fig. 12. ***Illustration of identifying the $\alpha(A)$, $\beta(A)$, and $\gamma(A)$ inference process for node A in an AoG (see text for detail definitions). The $\alpha(A)$ process is directly based on the compact image data of node A (either bottom–up or top–down), the $\beta(A)$ process generates hypotheses of node A by bottom–up binding the α processes of some child node(s) [e.g., $(\alpha(C_1),\alpha(C_2)) \to \beta(A)$], and the $\gamma(A)$ process predicates hypotheses of node A from the α processes of some parent node(s) [e.g., $\alpha(P) \to \gamma(A)$ in a top–down fashion]. In computing, each process has two states: "on" or "off." For example, the α process of node C_3 is off and we show it in gray. Because AoG is often recursively defined, each node has its own α, β, and γ processes. Modified from [64].***

2) *The β process.* When node A is at high resolution, it is more likely to be occluded in a scene. Node A itself is not detectable in terms of the α process due to occlusion. A subset of node A's children nodes can be detected in cropped patches (say, their α processes are activated). Then, the $\beta(A)$ process computes node A by binding the detected child nodes bottom–up under some compatibility constraints. An example of $\beta(\text{face})$ process is illustrated in Fig. 13. Most of component [66], [67], fragment [68], or part-based methods, the constellation models [69], and the pictorial models [70] belong to this process.

3) *The γ process.* The γ process handles situations in which node A is at very low resolution. Node A cannot be detected alone based on $\alpha(A)$, and neither can its parts. Then, the $\beta(A)$ process also fails. Information outside the local window must be incorporated; an example of $\gamma(\text{face})$ process is illustrated in Fig. 13. The $\gamma(A)$ process top–down predicts node A top–down from a parent node whose α process is activated. In this paper, we let the parent node pass contextual information, such as information from some sibling nodes or other spatial contexts. Most of the context-based methods [31], [32] belong to this process.

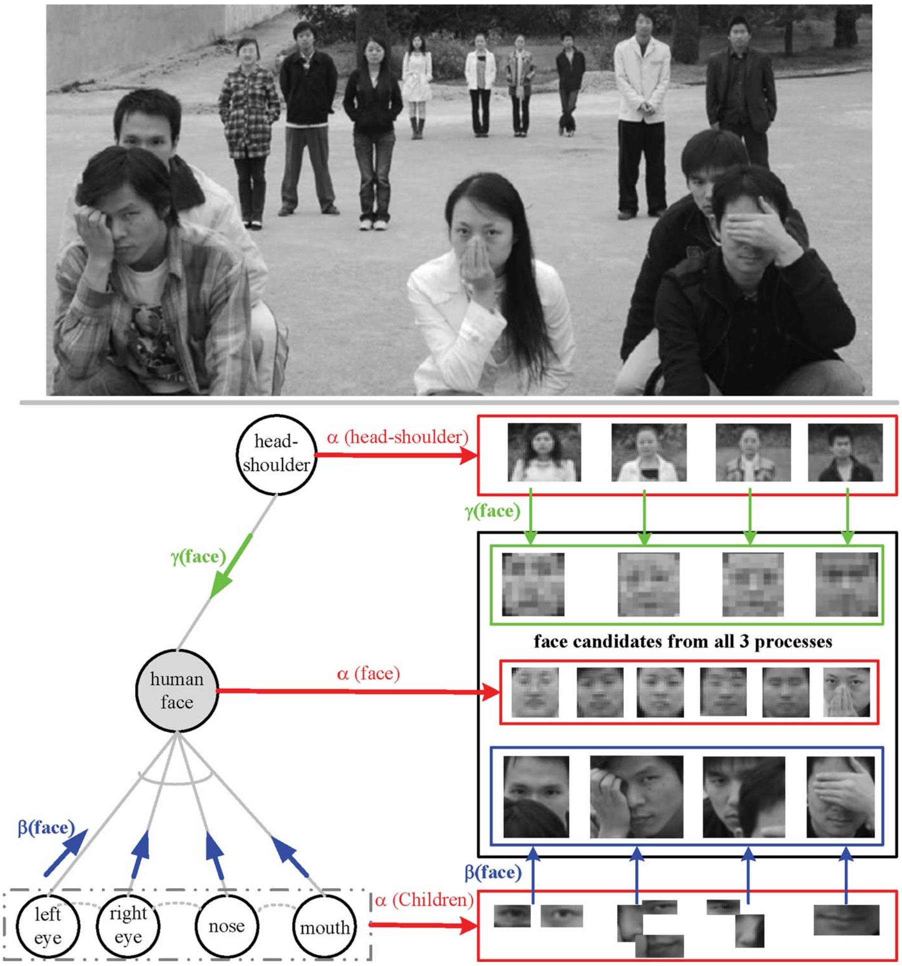


Fig. 13. ***Illustration of integrating the α(face), β(face), and γ(face) in the human face AoG for face detection. The three inference processes are effective in complementary ways depending on the scale and occlusion conditions. The typical situations shown here are common to other object categories. Modified from [64].***

For node A, all three inference processes $\alpha(A)$, $\beta(A)$, and $\gamma(A)$ contribute to computing it from images in complementary ways. The effectiveness of each process depends on the scale and occlusion conditions. As shown in Fig. 13, the three cases of human faces can be handled by the α(face), β(face), and γ(face), respectively. Intuitively, for robust inference, we should integrate them. As an AoG is a recursive structure, the three inference processes are also defined recursively, and each and-node has its own α, β, and γ inference processes (except that the γ process of the root node and the β processes of leaf nodes are always disabled). We would like to refer interested readers to [64] for detailed theoretical proofs and experimental results.

2) *Cluster Sampling:* Aside from bottom–up/top–down detection of a single object, another important issue pertinent to *automatic image parsing* is how to coordinate detection of multiple objects in one image. For example, in Fig. 3, there are nine candidate objects (i.e., person, grass, etc.) overlapping and competing with each other. It is important to have an algorithm that can optimally pick the most coherent set of objects. Previous methods such as [40] and [64] commonly used a greedy algorithm, which first assigned a weight to each of the currently unselected candidate objects based on how well it maximized the posterior. The object with the highest weight was selected to be added to the running parse of the scene. The objects were then reweighted according to how much the remaining objects would improve the overall explanation of the scene and this process iterated until no objects above a certain weight remained. The problem with this approach is that it is greedy and cannot backtrack from a poor decision. For example, by selecting the snow covered mountain in Fig. 3, the algorithm will now give an exceedingly low weight to the trees as we virtually never see trees on a snow summit. Had the snow covered mountain been selected first, we would not have arrived at

the correct interpretation. We would like our new algorithm to be able to backtrack from these mistakes.

We use an algorithm called clustering via cooperative and competitive constraints (C4) [51] to deal with these problems. It differs from classic cluster-sampling algorithms such as Swendsen–Wang clustering [71], [72] in two major ways.

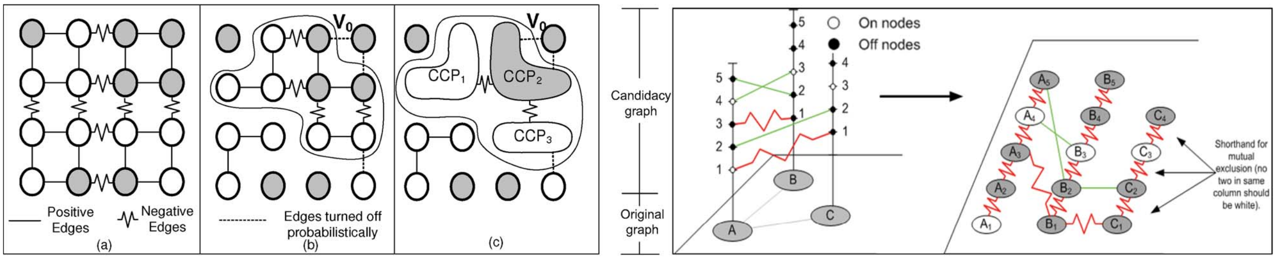
1) *Negative edges*: In addition to the "positive" edges in Swendsen–Wang clustering, in which nodes were encouraged to have the same label, C4 incorporates negative edges, dictating that neighboring sites should not be labeled similarly. We use them here to indicate that two explanations of the scene cannot both exist at once. For example, we could have negative edges between two overlapping cars to indicate that they cannot both be in the same explanation at the same time.
2) *Composite Flips*: Traditional Swendsen–Wang cut (SWC) updates the label of a single cluster in one step. In our model, the new labels for one cluster may cause it to violate constraints with neighboring clusters, so we may need to update the labels of many clusters simultaneously. We thus form composite components consisting of conflicting clusters that all need their labels reassigned at once to remain consistent. Fig. 14(b) shows C4 on a toy model. In this example, we have introduced a backbone of negative edges down the middle of the lattice, requiring that nodes on one side have the same color, but each side has a different color. Traditional Gibbs sampling attempts to update one site at a time, which creates a low probability state. SWC updates an entire side at one time, but only updates one cluster and ignores negative edges, thus creating another low-probability state. C4 clusters the entire system and relabels the individual clusters subject to both positive and negative constraints, creating a high-probability state.

We extend the example from the Ising model in Fig. 14(b) to handling general problems in candidacy graphs in Fig. 14(a). Objects that have a high prior probability of being on together are grouped together with "positive" edges, while objects that have low prior probability of being on together are grouped by "negative" edges. Here we have added positive and negative edges based on pairwise energies from the exponent in the AoG.

B. Semiautomatic Image Parsing

The IIP is a software for image annotation developed at the LHI to improve the efficiency of manual image annotation in order to cope with the need of annotating tens of thousands of images. The IIP has the following components.

1) **Segmentation**. Manual segmentation of objects from an image, especially fine segmentation as illustrated in Figs. 16 and 17, is the most time consuming part of any image annotation task. We cannot rely on automatic tools to do this job because results from even the state-of-the-art image segmentation algorithms are far from satisfactory compared with human results. One way out of this dilemma is a hybrid of manual and automatic segmentation—an interactive segmentation tool, where the human can provide guides (e.g., initialization) to the segmentation algorithm and is able to edit small defects in the segmentation results. Luckily, there has already been some powerful interactive object segmentation tools available, such as the GrabCut [73] and GraphCut [15] currently integrated into the IIP. Fig. 15(a) illustrates the user interface of the IIP when performing the interactive object segmentation. To get the boundary of the target object in an image, a human labeler will first draw a rectangular box surrounding the target and activate the GrabCut algorithm, which will automatically generate a rough boundary of the object. A human labeler will then use the GraphCut algorithm to modify small



(a) C4: cluster-sampling

(b) Candidacy graph

Fig. 14. ***(a) A visualization of the C4 algorithm. Here negative edges down the center encourage the four corners to be opposite colors, creating a checkerboard pattern. The right panel shows a connected component V_0 formed after edges are turned on and off. V_0 can be broken down in the final panel into subcomponents of like-colored nodes connected by negative edges. (b) Conversion from a candidacy graph to a binary graph with positive/negative edges. From [51].***

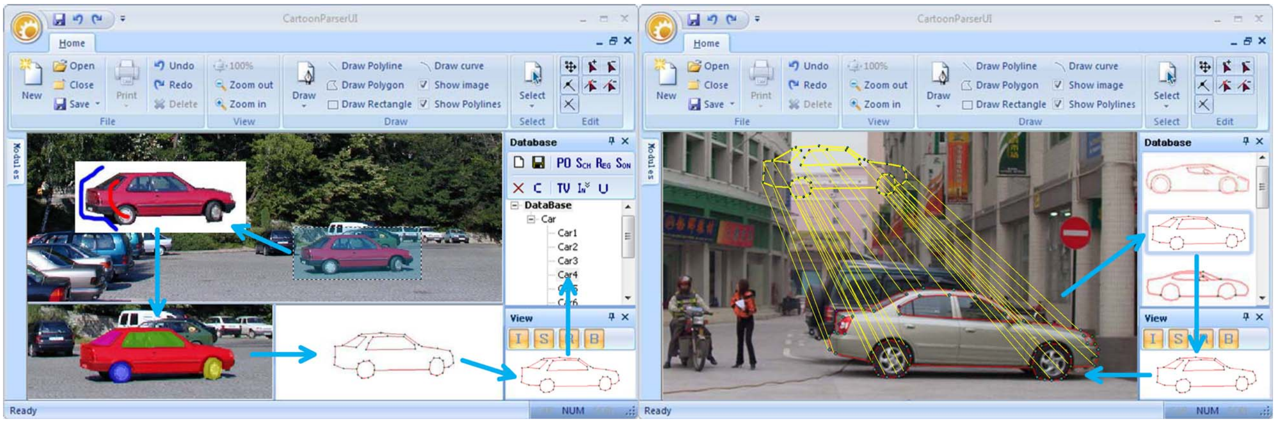


Fig. 15. ***Interactive parsing: Left panel shows an example of interactive object segmentation, where the boundaries of the car and its parts are derived by GrabCut and GraphCut algorithms. A sketch graph of the car is further derived by composing all the boundaries. The right panel shows an example of shape retrieval from the AoG database. The sketch graph of a new instance of car is matched with all stored sketch graphs in the car category; the instance is then associated with the and-node with the best match score.***

defects on the boundary, by specifying pixels belonging to the object (the red line) and to the background (the blue line), respectively.

2) **AoG database**. Since the task of annotating tens of thousands of images cannot be done by a single person on a computer (in fact, there is a team of about ten labelers at the LHI), it is very important to have a consistent structure for different human labelers. The AoG, as discussed before, is a systematic way to summarize visual knowledge. The IIP has a centralized database storing the AoG and a number of terminal interfaces that support multiple users working concurrently.

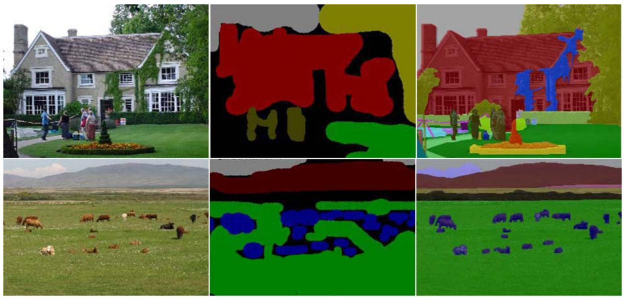
3) **Shape retrieval and matching**. Given the database, the task of annotation is to associate the parse graph of a new image instance with the existing AoG. This is essentially a retrieval problem: given a new instance of object (with sketch graph generated by interactive segmentation), find the most similar node from the AoG database. This is by no means a trivial task given the potential number of all and-nodes in the AoG [as shown in Fig. 11(c)]. We develop a retrieval algorithm based on shape matching and semantic pruning. First, good candidates are automatically selected from the AoG. Then, a shape matching algorithm such as *shape context* [22] and *graph match* [74] is used to fit the template onto an object. The shape matching results are further pruned by semantic meaning. For example, a car wheel may be matched to the frame of a round clock, but if it is known to be a "car" part, the IIP will prune out the semantically incompatible matches. Thus, the labeling procedure is sped up dramatically. If there is a new instance that does not resemble any previously observed and-nodes in the AoG, then the human labeler will add a new and-node as a new subconfiguration under the same or-node concept. A similar methodology has been used in a work by Hays *et al.* [75].

IV. THE LHI DATA SET

In this section, we use examples from the LHI data set to illustrate more details about the data structures and design issue of the data set.

A. Object Segmentation

The task of object segmentation is to create a label map where different objects presented in the original image are annotated with different colors. One important feature of the LHI data set is that it provides fine segmentation. For example, Fig. 16 compares segmentation label maps from both the LHI data set and the MSRC data set [44]. It is



(a) Original Image (b) MSRC annotation (c) LHI annotation

Fig. 16. ***(a) Two example images from the MSRC data set [44]. (b) Segmentation label maps from the MSRC data set. (c) Segmentation label-maps from the LHI data set (label maps are set to be transparent for better illustration).***

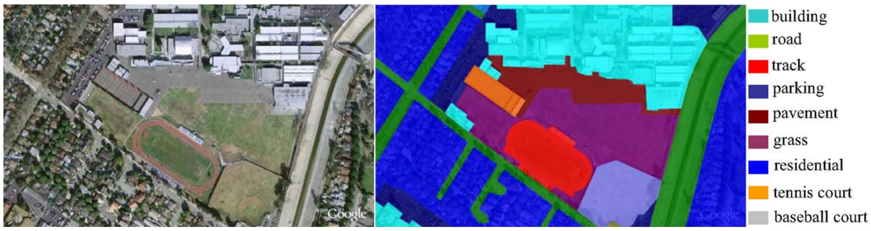


Fig. 17. ***Example segmentation label map of an aerial image (a school scene).***

clear that the segmentation label maps of the LHI data set are much more accurate. Fine segmentation not only provides better accuracy for evaluation, but also makes it possible to annotate small objects in complicated scenes (e.g., Fig. 17).

B. Sketch Graph

A segmentation label map only provides silhouettes of objects, which cannot represent internal structures. We adopt a new data structure called *sketch graph* in addition to the segmentation label map. As shown in Figs. 18 and 19, a sketch graph is composed of a set of control points representing landmark positions, which are further grouped into a set of *curves* to form "sketches." This is similar in spirit to the active shape models (ASMs) [76], and is closely related to the primal sketch representation [17]. Fig. 18 shows that the sketch graphs can capture internal structural information of cloth, such as the folds, sewing lines, and albedo/lighting changes. A dictionary of human body components is illustrated in Fig. 19, where each element is a small sketch graph and has open bonds (anchor points) for connecting with other parts.

Sketch graphs can be further augmented to include low-middle level vision elements by adding attributes to each curve (*attributed curves*). As illustrated in Fig. 20, the curves on the Winnie the Pooh are differently colored (different attributes) and represent *occlusion*, *surface normal change*, and *lighting/albedo change*, respectively. The attributed curve is also useful for representing illusory (occluded) contours in an image (as illustrated in Fig. 21). This is closely related to the "2.1D sketch," which is proposed in [77] to model low-level depth reconstruction exhibited in early human vision.

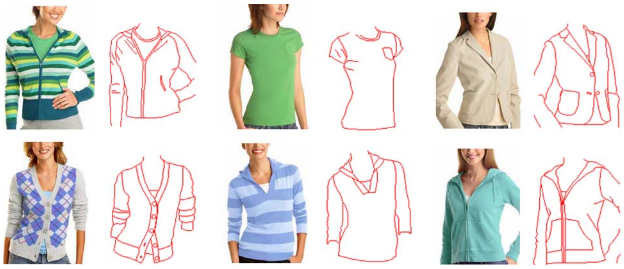


Fig. 18. ***Sketch graph representation of cloth. The sketch graph can effectively capture internal structural information of cloth, such as folds, sewing lines, and albedo/lighting changes.***

C. Hierarchical Decomposition

As shown in Fig. 1, each image in the LHI data set is decomposed hierarchically from scene to parts to generate a parse graph. Horizontal links are also added between nodes to represent relationships between objects and parts. Both the hierarchical decomposition and horizontal relationships will become a part of an AoG.

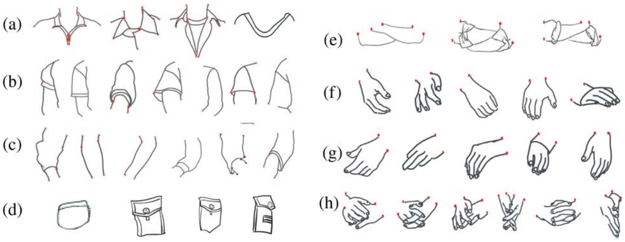


Fig. 19. ***Dictionary of cloth and body components. Each element is a small sketch graph and has open bonds (red points) for connecting with other parts. Modified from [41].***

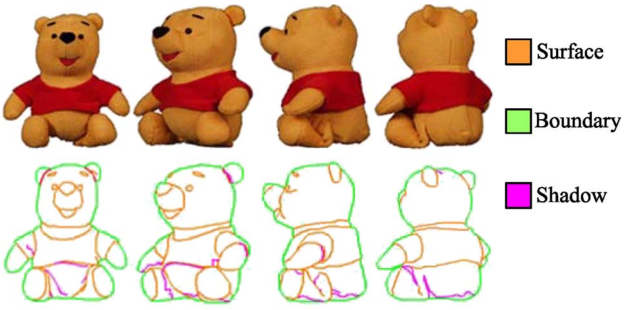


Fig. 20. ***Attributed curves. "Winnie the Pooh" is labeled with three types of curve attributes: surface: curves generated by surface norm change; boundary: curves on object boundary; shadow: curves generated by shadow.***

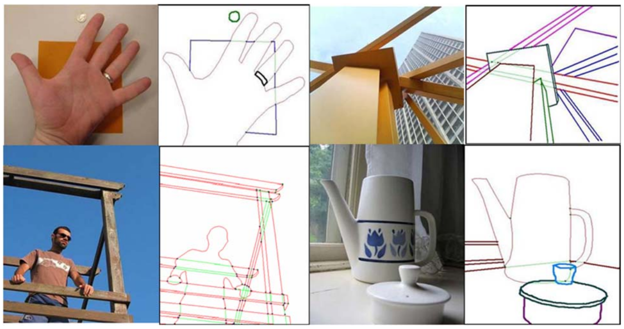


Fig. 21. ***Sketch graph with 2.1D layered representation. Green lines stand for illusory (occluded) contours.***

D. Semantic Annotation Using WordNet Vocabulary

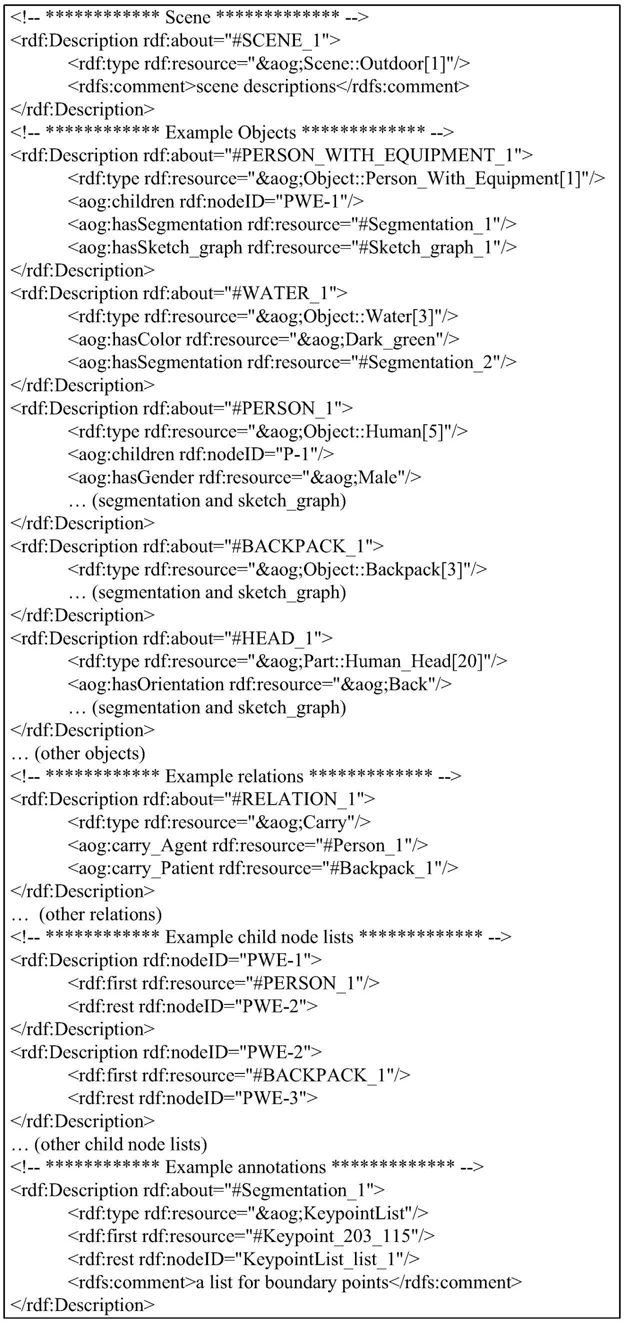
WordNet [3] contains a large vocabulary, which has been systematically annotated with word sense information and relationships such as synonyms, hyper- and hyponyms, moronyms, etc. We annotate the LHI data set according to the vocabulary from WordNet. Therefore, the LHI data set inherits the semantic relationships between concepts from WordNet. The following steps are adopted when a name is chosen for an object: 1) words must be selected from the WordNet; 2) the sense in WordNet should be mentioned; 3) descriptive words can be added to provide further information (e.g., *[male]human*, *[race]car*). This process has to be done manually by a human.

V. SEMANTIC REPRESENTATION AND QUERY

A. Semantic Representation

Semantic representation is a formal way of expressing inferred image and video content and provides a bridge to content knowledge management. This is needed for a variety of information exploitation tasks, including content-based image and video indexing and retrieval, forensic analysis, data fusion, and data mining. The representation should be unambiguous, well formed, flexible, and extensible for representing different object classes, their properties, and relations.

With an image ontology based on the AoG, we can convert the parse graph representation of an image into semantic representation using RDF format. For example, the XML report in Fig. 22 shows a section of the semantic representation converted from the parse graph illustrated in Fig. 1. In this report, each object is associated with an and-node in the AoG. For example, the object node "*#WATER_1*" has resource "*&aog;Object::Water[3]*," where "*&aog;*" points to a web resource defining the image ontology based on the AoG, "*Object::Water*" means an or-node named "Water" in the "Object" level, and "*[3]*" stands for the third subconfiguration under the or-node, which pointed to an and-node. The hierarchical structure of the parse graph is represented in a node-list format. For example, the object "*#PERSON_WITH_EQUIPMENT_1*" has a child with node ID "PWE-1," which is associated with the object "*#PERSON_1*" and also points to its sibling "PWE-2." Relations are also defined based on the AoG. For example, the relation "*#RELATION_1*" is associated with an AoG concept "*&aog;Carry*" and has an agent "*#PERSON_1*" and a patient "*#BACKPACK_1*." Similarly, with the AoG image ontology, we can translate any parse graph into a semantic representation.



```
<!-- ************ Scene ************* -->
<rdf:Description rdf:about="#SCENE_1">
        <rdf:type rdf:resource="&aog;Scene::Outdoor[1]"/>
        <rdfs:comment>scene descriptions</rdfs:comment>
</rdf:Description>
<!-- ************ Example Objects ************* -->
<rdf:Description rdf:about="#PERSON_WITH_EQUIPMENT_1">
        <rdf:type rdf:resource="&aog;Object::Person_With_Equipment[1]"/>
        <aog:children rdf:nodeID="PWE-1"/>
        <aog:hasSegmentation rdf:resource="#Segmentation_1"/>
        <aog:hasSketch_graph rdf:resource="#Sketch_graph_1"/>
</rdf:Description>
<rdf:Description rdf:about="#WATER_1">
        <rdf:type rdf:resource="&aog;Object::Water[3]"/>
        <aog:hasColor rdf:resource="&aog;Dark_green"/>
        <aog:hasSegmentation rdf:resource="#Segmentation_2"/>
</rdf:Description>
<rdf:Description rdf:about="#PERSON_1">
        <rdf:type rdf:resource="&aog;Object::Human[5]"/>
        <aog:children rdf:nodeID="P-1"/>
        <aog:hasGender rdf:resource="&aog;Male"/>
        … (segmentation and sketch_graph)
</rdf:Description>
<rdf:Description rdf:about="#BACKPACK_1">
        <rdf:type rdf:resource="&aog;Object::Backpack[3]"/>
        … (segmentation and sketch_graph)
</rdf:Description>
<rdf:Description rdf:about="#HEAD_1">
        <rdf:type rdf:resource="&aog;Part::Human_Head[20]"/>
        <aog:hasOrientation rdf:resource="&aog;Back"/>
        … (segmentation and sketch_graph)
</rdf:Description>
… (other objects)
<!-- ************ Example relations ************* -->
<rdf:Description rdf:about="#RELATION_1">
        <rdf:type rdf:resource="&aog;Carry"/>
        <aog:carry_Agent rdf:resource="#Person_1"/>
        <aog:carry_Patient rdf:resource="#Backpack_1"/>
</rdf:Description>
…  (other relations)
<!-- ************ Example child node lists ************* -->
<rdf:Description rdf:nodeID="PWE-1">
        <rdf:first rdf:resource="#PERSON_1"/>
        <rdf:rest rdf:nodeID="PWE-2">
</rdf:Description>
<rdf:Description rdf:nodeID="PWE-2">
        <rdf:first rdf:resource="#BACKPACK_1"/>
        <rdf:rest rdf:nodeID="PWE-3">
</rdf:Description>
… (other child node lists)
<!-- ************ Example annotations ************* -->
<rdf:Description rdf:about="#Segmentation_1">
        <rdf:type rdf:resource="&aog;KeypointList"/>
        <rdf:first rdf:resource="#Keypoint_203_115"/>
        <rdf:rest rdf:nodeID="KeypointList_list_1"/>
        <rdfs:comment>a list for boundary points</rdfs:comment>
</rdf:Description>
```

Fig. 22. ***Semantic representation of the parse graph in Fig. 1 based on an image ontology embodied in the AoG representation.***

Recent emergence of semantic web technology has encouraged the growth of distributed yet interconnected ontologies published on the Internet using the OWL. Using the OWL representation, an ontology engineer can declare how knowledge concepts defined in different ontologies are related. With this ontological mapping, multiple OWL documents can be interconnected. This promotes reuse of existing ontologies and encourages the development of domain-specific ontologies to address the diverse needs of different applications. The semantic description of a parse graph in RDF format can be translated into the OWL format. The collective ontology can express more complex image content and video events. With this framework, visual content can be published on the semantic web, allowing various semantic mining and inference tools to retrieve, process, and analyze the content. Some of the semantic concepts such as object classes can be mapped to well-established concepts defined in standard knowledge bases such as the Open Geospatial Consortium (OGC) and WordNet. This improves the accessibility and portability of the inferred video content.

B. User Queries

With visual content published in OWL, a user can now perform content-based searches using SPARQL, the query language for semantic web [78], released by the World Wide Web Consortium (W3C). With SPARQL, users or autonomous data mining engines can perform searches by expressing queries based on semantics. This improves usability as the details of database models are hidden from the user. The versatile nature of SPARQL allows the user to query multiple OWL documents collectively and this enhances data integration from multiple knowledge sources. For example, suppose that a car in an image is annotated as a "sedan" while the user performs a search using the term "automobile"; SPARQL is still able to retrieve the result because WordNet identifies that the two words are synonyms.

VI. TEXT GENERATION

While OWL provides an unambiguous representation for image and video content, it is not easy for humans to read. Natural language text remains the best way for describing the image and video content to humans and can be used for image captions, scene descriptions, and event alerts. NLG is an important subfield of natural language processing. NLG technology is already widely used in Internet applications such as weather reporting and for giving driving directions. A commonly used NLG approach is template filling, but it is inflexible and inadequate for describing images. An image NLG system should be able to consume OWL data, select relevant content, and generate text to describe objects in images, their properties, events, and relationships between other objects.

The text generation process is usually designed as a pipeline of two distinct tasks: text planning and text realization. The text planner selects the content to be expressed, and decides how to organize the content into sections, paragraphs, and sentences. Based on this formation, the text realizer generates each sentence using the correct grammatical structure.

A. Text Planner

The *text planner* module translates the semantic representation to a sentence-level representation that can readily be used by the text realizer to generate text. This intermediate step is useful because it converts a representation that is semantic and ontology based, to a representation that is based more on functional structure. The output of the text planner is based on a *functional description* (FD) which has a feature-value pair structure, commonly used in text generation input schemes [79]. For each sentence, the functional description language specifies the details of the text that is to be generated, such as the process (or event), actor, agent, time, location, and other predicates or functional properties. An example of functional description is shown in Fig. 23. The text planner module also organizes the layout of the text report document. The planning of the document structure is strongly dependent on the intended application. For example, a video surveillance report may contain separate sections describing the scene context, a summary of objects that appeared in the scene, and a detailed list of detected events. Other applications, such as an e-mail report or instant alert would warrant different document structures, but the underlying sentence representation using functional description remains the same.

B. Text Realizer

From the functional description, the text realizer generates each sentence independently using a simplified head-driven phrase structure grammar (HPSG) [80]. HPSG consists of a set of production rules that transform the functional description to a structured representation of grammatical categories. Grammatical categories include the following part-of-speech tags: S (sentence), VP (verb phrase), NP (noun phrase), DET (determiner), and PP (prepositional phrase) among others. Examples of production rules include: S → NP VP, NP → DET (A) N, VP → V NP, VP → V PP, VP → V ADV, and PP → P NP. Rules with features are used to capture lexical or semantic properties and attributes. For example, to achieve *person–number agreement*, the production rules include variables so that information is shared across phrases in a sentence: S → NP(per,num) VP(per,num). A unification process [81] matches the input features with the grammar recursively, and the derived lexical tree is then linearized to form the sentence output. An example of text realization is shown in Fig. 23.

While general-purpose text realization is still an active research area, current NLG technology is sufficiently capable of expressing video content. The lexicon of visual

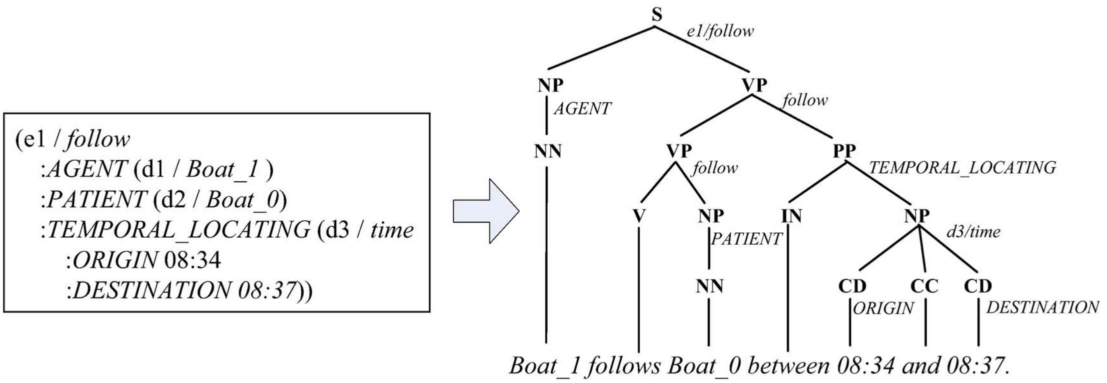


Fig. 23. *Example of a functional description (FD) of a video event being converted to natural language text by the text realizer. The FD is first transformed to a part-of-speech (POS) tree where additional syntactic terms ("between" and "and") are inserted. The POS tree is then linearized to form a sentence. Notice that the FD and the POS tree share similar hierarchical structure, but there are notable differences. In the FD, children nodes are unordered. In the POS tree, the ordering of children nodes is important and additional syntactic nodes are inserted.*

objects and relationships between objects is relatively small. In addition, textual descriptions of visual events are mostly *indicative* or *declarative* sentences and this simplifies the grammar structure of the resulting text significantly.

VII. CASES STUDIES

A. Video Surveillance System

In this section, we demonstrate an end-to-end system from surveillance videos to text reports using the I2T framework. The overall architecture of the system is shown in Fig. 24(a), which resembles the diagram for static images shown in Fig. 2 except two extra steps for analyzing video content, namely object tracking and event inference. First, an IIP generates a parse graph of scene context from the first frame of the input video. The parse graph is further translated into a semantic representation using techniques described previously. Since the camera in this system is static, the parsing result of the first frame can be used throughout the entire video. Second, the system tracks moving objects in the video (e.g., vehicles and pedestrians) and generates their trajectories automatically. Third, from the scene context and object trajectories, an event inference engine extracts descriptive information about video events, including semantic and contextual information, as well as relationships between activities performed by different agents. The video event markup language (VEML) [49] is adopted for semantic representation of the events. Finally, a text generation engine is used to convert the semantic representation of scene context and video events into a text description. In the following sections, we describe the system in detail.

1) Event Inference: For event inference, we leverage the existing state-of-the-art in knowledge representation and focus on extracting descriptive information about visual

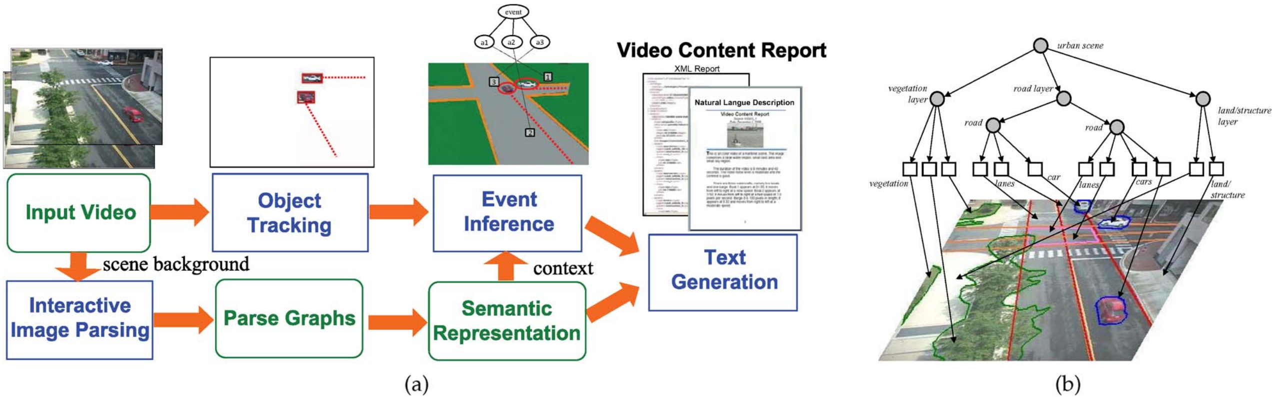


Fig. 24. *(a) Diagram of the video-to-text system. (b) Parse graph of the scene context.*

events, including semantic and contextual information as well as relationships between activities performed by different agents. A grammar-based approach is used for event analysis and detection. In the following, we discuss different aspects of event inference.

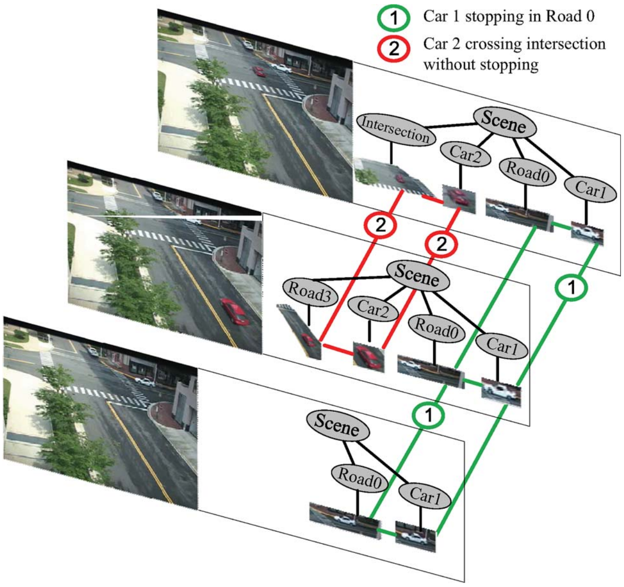
a) Scene region analysis: Scene region analysis enhances the scene understanding by analyzing the functional and contextual property of scene regions. Pixel-level scene element classification can be further analyzed to derive higher level scene content. For instance, in road analysis, the aim is to extract road structure, junctions, and intersections. To analyze road structure, we expand the taxonomy and object class properties, and derive object class relations. Based on these relations, roads are detected using the data-driven approach with data from the observed trajectories of vehicles. Combining road information with superpixel-based scene element classification, the boundaries of roads can be extracted fairly accurately. Junctions are then detected as intersections of roads. Similar inference can be made on other types of scene regions, such as waterway (used by watercraft) and sidewalk (used by pedestrians).

A key benefit of scene region extraction is the automatic demarcation of region-of-interest (ROI) zones for higher level analysis. A zone is a generic term to describe an image region that has semantic, contextual, or functional significance. Examples of zones include road junctions, port docking areas, and entrances to buildings. A zone serves as a spatial landmark; the position and motion of other objects can be described with respect to this landmark. This allows us to detect semantic actions and events, and it facilitates the textual description of the events thereafter.

b) Spatio–temporal analysis: For spatio–temporal analysis, we assume that an object is moving on a ground plane and the detected trajectory is a series of tracked "footprint" positions of the object. The trajectory is then approximated by a series of image-centric segments of straight motions or turns, such as "move up," "turn left," etc. The trajectory can be described concisely in terms of these motion segments. A trajectory is also described in relation to the zones that are demarcated in the scene, such as entering and exiting a zone. The system analyzes the motion properties of objects traveling in each zone, such as minimum, maximum, and average speeds. From a collected set of trajectories, histogram-based statistics of these properties are learned. By comparing new trajectories to historical information, abnormal speeding events inside the zone can be detected.

Speed information is generally expressed in image-centric measure (pixel per second). The image size of an object is then used to coarsely estimate the ground sample resolution (meter per pixel) to compute true speed (e.g., mile per hour). More accurate estimation can be obtained by calibrating the camera, either manually or automatically [82].

Complex events are composed of subevents and can be represented by a spatio–temporal parse graphs (see Fig. 25). Contextual information is important for video event understanding and is specified by the spatial, temporal, and functional relations between moving objects and background objects. The context includes approaching, entering, and exiting a location, providing semantic-based inference and descriptive event annotation.



1 Car 1 stopping in Road 0
2 Car 2 crossing intersection without stopping
Scene
Intersection
Car2
Road0
Car1
Road3
1
2

Fig. 25. ***Example of a video event detection involving a failure-to-yield incident. Each image frame is represented by a parse graph and objects are tracked over frames. In the event grammar, complex events are composed of elementary events observed as the motion of objects (cars) with respect to scene context (road lanes and intersection). There are two events in this example: (1) car-1 stopping in road-0; (2) car-2 crossing intersection without stopping.***

2) Results: Our evaluation focused on urban traffic and maritime scenes, and it consisted of two parts. We evaluated event detection and metadata/text generation with sequences of different scenes. We processed ten sequences of urban and maritime scenes, with a total duration of about 120 min, that contain more than 400 moving objects. Visual events were extracted and text descriptions were generated. Detected events included: entering and exiting the scene, moving, turning, stopping, moving at abnormal speed, approaching traffic intersection, entering and leaving traffic intersection, failure-to-yield violation, watercraft approaching a maritime marker or a land area, and an object following another object.

When annotating these events in both metadata and text description, the system extracts and provides information about the object class, scene context, position, direction, speed, and time. Examples of text descriptions and corresponding video snapshots are shown in Fig. 26. With the text description, user can search for video events using keywords. Full text search engines are commercially available to provide word-based indexing and searching functionalities.

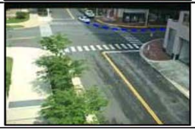
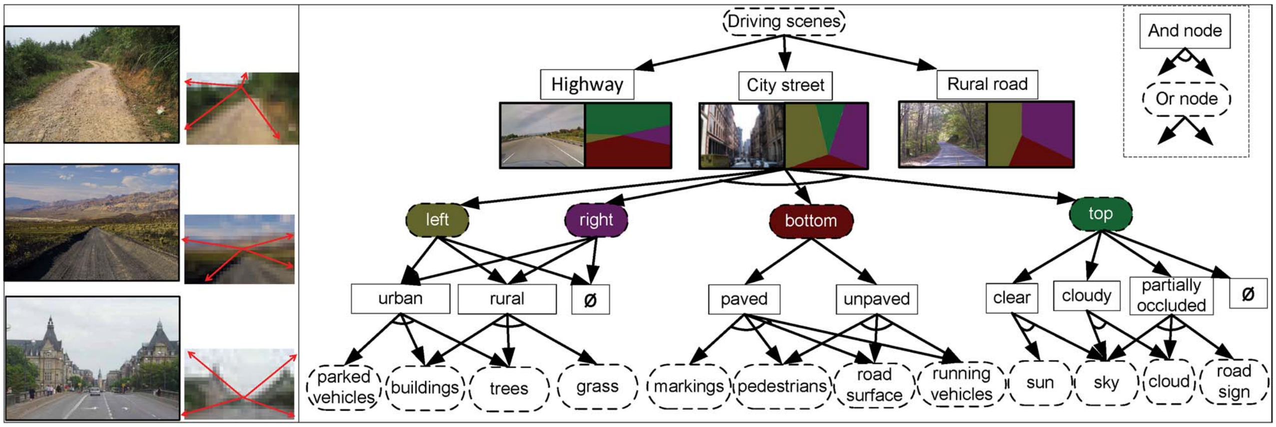
	Boat_2 enters the scene on water region at 19.50. Boat_2 approaches maritime marker at 20.09.
	Boat_4 follows Boat_3 between 35:36 and 37:23
	Boat_7 turns right at 55:00.
	Land_vehicle_359 approaches intersection_0 along road_0 at 57:27. It stops at 57.29. Land_vehicle_360 approaches intersection_0 along road_3 at 57:31.
	Land_vehicle_360 moves at an above-than-normal average speed of 26.5 mph in zone_4 (approach of road_3 to intersection_0) at 57:32. It enters intersection_0 at 57:32. It leaves intersection_0 at 57:34. There is a possible failure-to-yield violation between 57:27 to 57:36 by Land_vehicle_360.
	Land_vehicle_359 enters intersection_0 at 57:35. It turns right at 57:39. It leaves intersection_0 at 57:36. It exits the scene at the top-left of the image at 57:18.

Fig. 26. *Samples of generated text and corresponding video snapshots.*

B. Automatic Driving Scene Parsing

The second application of the proposed I2T system is an ongoing project on automatic driving scene parsing. As illustrated in Fig. 27, we build a novel AoG for driving scenes using an X-shaped-rays model at low resolution to obtain efficient scene configuration estimation. Then, we further detect interesting objects in the "foreground" such as cars and pedestrians and classify regions at high resolution under the scene context.

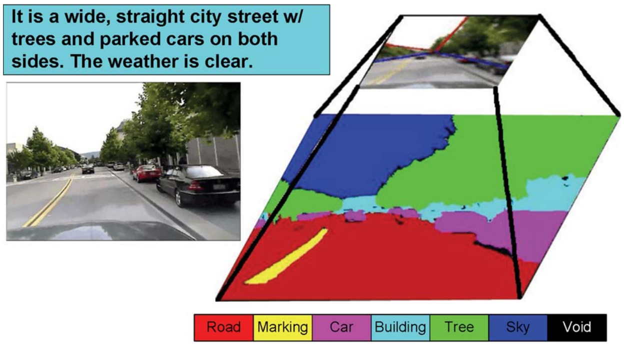
We exploit several useful features from the X-shaped-rays model to classify different scenes, such as the four intersection angles, the area ratio of sky to the whole image, the area ratio of building to the whole image, etc. For detecting cars and pedestrians, we adopted the active



(a) X-shape-rays model

(b) An And-Or graph for driving scenes

Fig. 27. *(a) Under a low resolution (e.g., 32 × 32 pixles), a driving scene can be approximated by an X-shape-rays model with four components (left, right, bottom, and top). (b) The AoG used for parsing driving scenes. means a component is missing.*



It is a wide, straight city street w/ trees and parked cars on both sides. The weather is clear.

Fig. 28. ***Example of parsing and text generation results.***

basis model [59]. One example of a scene parsing and text generation results are illustrated in Fig. 28.

VIII. CONCLUSION AND DISCUSSION

This paper proposes a framework that provides an end-to-end solution for parsing image and video content, extracting video event, and providing semantic and text annotation. One major contribution is the AoG visual knowledge representation. The AoG is a graphical representation for learning categorical image representations and symbolic representations simultaneously from a large scale image. It not only provides top–down guides during the image parsing process but also connects low-level image features with high-level semantically meaningful concepts so that the parsed image can be seamlessly transformed to a semantic metadata format and finally to a textual description. The I2T framework is different from, but complementary to, existing technology in keyword-based image and video shots categorization/annotation, because it provides richer and semantically oriented annotation of visual contents. With image and video contents expressed in both OWL and text format, this technology can be easily integrated with a full text search engine, as well as SPARQL queries, to provide accurate content-based retrieval. Users can retrieve images and video clips via keyword searching and semantic-based querying.

For future work, we are going to explore connotative messages from an image. At the moment, the I2T system, like most other CBIR systems, can only process denotative messages from an image. To explain the concept of connotative and denotative messages, we would like to use a simple example from [83]: Consider a drawing of a cute rabbit. Based on this image, most viewers would perceive a rabbit. Viewer from the Western or Christian cultural background might also be reminded of the Easter Bunny and associate the image with Easter themes. In this example, the identifiable object, a rabbit, functions as the denotative message. Additional messages such as Easter themes are connotative. Viewers conceive connotative messages from an image based on visual perception (denotative messages) as well as their own cultural background. It is widely recognized that images convey both denotative and connotative messages, whereas the connotative message is oftentimes more important for representing a searcher's intention. It is obvious that deriving connotative messages from images requires integrating knowledge from several related domains such as art history, social and religious culture, among others. Since the semantic web technology is a perfect tool for integrating diverse domain knowledge, the I2T framework provides a possible solution for indexing and retrieving connotative message from images. ■

Acknowledgment

The authors would like to thank Dr. H. Gong and Mr. T. Wu for their help in developing the interactive image parser and Mr. S. Li and other members of the Lotus Hill Institute (LHI, Ezhou, China) annotation team for their great efforts in constructing the LHI dataset. The authors would also like to thank the anonymous reviewers for their suggestions that improved the presentation of this paper.

REFERENCES

[1] A. Zipern, *News Watch; A Quick Way to Search for Images on the Web*. [Online]. Available: http://www.nytimes.com/2001/07/12/technology/news-watch-a-quick-way-to-search-for-images-on-the-web.html

[2] C. Manning, H. Schütze, and M. Press, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[3] G. A. Miller, "Wordnet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[4] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *Int. J. Human Comput. Studies*, vol. 43, no. 5, pp. 907–928, 1995.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, May 2001.

[6] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[7] Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *J. Vis. Commun. Image Represent.*, vol. 10, pp. 39–62, 1999.

[8] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, Feb. 2006.

[9] C. Snoek and M. Worring, "Multimodal video indexing: A review of the state-of-the-art," *Multimedia Tools Appl.*, vol. 25, no. 1, pp. 5–35, 2005.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, Apr. 2008, article 5.

[11] C. Town, "Ontological inference for image and video analysis," *Mach. Vis. Appl.*, vol. 17, no. 2, pp. 94–115, 2006.

[12] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, no. 1–3, pp. 335–346, 1990.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[14] Z. W. Tu, X. R. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 113–140, Jul. 2005.

[15] R. Zabih, O. Veksler, and Y. Y. Boykov, "Fast approximate energy minimization via graph cuts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 1, pp. 377–384.

[16] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision: Issues, Problems, Principles and Paradigms*. San Francisco, CA: Morgan Kaufmann, 1987, pp. 184–203.

[17] C. E. Guo, S.-C. Zhu, and Y. N. Wu, "Primal sketch: Integrating structure and texture," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 5–19, Apr. 2007.

[18] S.-C. Zhu, Y. N. Wu, and D. Mumford, "Filters, random-fields and maximum-entropy (frame): Towards a unified theory for texture modeling," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 107–126, Mar. 1998.

[19] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 29–44, Jun. 2001.

[20] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157.

[21] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 729–736, Jul. 1995.

[22] S. J. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[23] C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2169–2178.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. I:886–I:893.

[25] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 985–1002, Jun. 2008.

[26] D. Xu and S. F. Chang, "Video event recognition using kernel methods with multilevel temporal alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1985–1997, Nov. 2008.

[27] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, 2003.

[28] NIST, *Trec Video Retrieval Evaluation*. [Online]. Available: http://www.nlpir.nist.gov/projects/trecvid/

[29] L. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2036–2043.

[30] X. He, R. Zemel, and M. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. II-695–II-702.

[31] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 3–15, Oct. 2008.

[32] A. Torralba, K. Murphy, and W. Freeman, "Contextual models for object detection using boosted random fields," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2004.

[33] M. Marszalek, C. Schmid, and M. Inria, "Semantic hierarchies for visual object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, DOI: 10.1109/CVPR.2007.383272

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[35] D. Parikh and T. Chen, "Hierarchical semantics of objects (hSOs)," in *Proc. IEEE Int. Conf. Comput. Vis.*, DOI: 10.1109/ICCV.2007.4408960

[36] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, 2006.

[37] K. S. Fu, *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[38] Z. Tu and S.-C. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 657–673, May 2002.

[39] Z. Tu and S.-C. Zhu, "Parsing images into regions, curves, and curve groups," *Int. J. Comput. Vis.*, vol. 69, no. 2, pp. 223–249, 2006.

[40] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing by attribute graph grammar," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1778–1785.

[41] H. Chen, Z. J. Xu, Z. Q. Liu, and S.-C. Zhu, "Composite templates for cloth modeling and sketching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 943–950.

[42] B. Z. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2007, ser. Lecture Notes in Computer Science, pp. 169–183.

[43] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 157–173, May 2008.

[44] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 1–15.

[45] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[46] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, Tech. Rep., 2007.

[47] D. R. Martin, C. C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, pp. II:416–II:423.

[48] K. Barnard, Q. F. Fan, R. Swaminathan, A. Hoogs, R. Collins, P. Rondot, and J. Kaufhold, "Evaluation of localized semantics: Data, methodology, and experiments," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 199–217, 2008.

[49] R. Nevatia, J. Hobbs, and B. Bolles, "An ontology for video event representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, p. 119.

[50] *Web ontology language*. [Online]. Available: http://www.w3.org/TR/owl-features/

[51] J. Porway, Q. Wang, and S.-C. Zhu, "A hierarchical and contextual model for aerial image parsing," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 1–30, Jun. 2010, DOI: 10.1007/s11263-009-0306-1.

[52] S. P. Abney, "Stochastic attribute-value grammars," *Comput. Linguistics*, vol. 23, no. 4, pp. 597–618, 1997.

[53] M. Weber, M. Welling, and P. Perona, "Towards automatic discovery of object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, vol. 2, pp. 101–108.

[54] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.

[55] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1331–1338.

[56] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Describing visual scenes using transformed Dirichlet processes," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 18. Cambridge, MA: MIT Press, 2006, pp. 1299–1306.

[57] C. E. Guo, S.-C. Zhu, and Y. N. Wu, "Primal sketch: Integrating structure and texture," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 5–19, Apr. 2007.

[58] R. X. Gao, T. F. Wu, S.-C. Zhu, and N. Sang, "Bayesian inference for layer representation with mixed Markov random field," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2007, ser. Lecture Notes in Computer Science, pp. 213–224.

[59] Y. N. Wu, Z. Z. Si, C. Fleming, and S.-C. Zhu, "Deformable template as active basis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, DOI: 10.1109/ICCV.2007.4408980.

[60] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, 1996.

[61] K. Murphy, A. Torralba, and W. T. Freeman, "Graphical model for recognizing scenes and objects," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2003.

[62] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[63] K. Shi and S.-C. Zhu, "Mapping natural image patches by explicit and implicit manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, DOI: 10.1109/CVPR.2007.382980.

[64] T. Wu and S.-C. Zhu, "A numerical study of the bottom-up and top-down inference processes in and-or graphs," *Int. J. Comput. Vis.*, May 2010, DOI: 10.1007/s11263-010-0346-6.

[65] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple

features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. I-511–I-518.

[66] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychol. Rev.*, vol. 94, no. 2, pp. 115–147, 1987.

[67] B. Heisele, T. Serre, and T. Poggio, "A component-based framework for face detection and identification," *Int. J. Comput. Vis.*, vol. 74, no. 2, pp. 167–181, 2007.

[68] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature Neurosci.*, vol. 5, no. 7, pp. 682–687, 2002.

[69] R. Fergus, P. Perona, and A. Zisserman, "Weakly supervised scale-invariant learning of models for visual recognition," *Int. J. Comput. Vis.*, vol. 71, no. 3, pp. 273–303, 2007.

[70] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.

[71] R. Swendsen and J. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Phys. Rev. Lett.*, vol. 58, no. 2, pp. 86–88, 1987.

[72] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1239–1253, Aug. 2005.

[73] C. Rother, V. Kolmogorov, and A. Blake, "'Grabcut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[74] L. Lin, S.-C. Zhu, and Y. T. Wang, "Layered graph match with graph editing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, DOI: 10.1109/CVPR.2007.383190.

[75] J. Hays and A. Efros, "Scene completion using millions of photographs," *Commun. ACM*, vol. 51, no. 10, pp. 87–94, 2008.

[76] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[77] M. Nitzburg and D. Mumford, "The 2.1-D sketch," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1990, pp. 138–144.

[78] *Sparql Query Language for rdf*. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[79] I. Langkilde-Geary and K. Knight, *Halogen Input Representation*. [Online]. Available: http://www.isi.edu/publications/licensed-sw/halogen/interlingua.html

[80] C. Pollard and I. Sag, *Head-Driven Phrase Structure Grammar*. Chicago, IL: Chicago Univ. Press, 1994.

[81] K. Knight, "Unification: A multidisciplinary survey," *ACM Comput. Surv.*, vol. 21, no. 1, pp. 93–124, 1989.

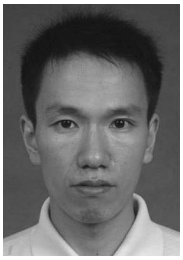
[82] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1513–1518, Sep. 2006.

[83] R. Barthes and S. Heath, *Image, Music, Text*. New York: Hill & Wang, 1977.

ABOUT THE AUTHORS

Benjamin Z. Yao received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2003 and the M.S. degree in electrical engineering from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2006. Currently, he is working towards the Ph.D. degree at the Department of Statistics, University of California, Los Angeles (UCLA).

During 2006-2007, he worked as a Research Assistant at Lotus Hill Institute, Ezhou, China. His research interest includes human action detection and recognition, human annotated image database, and video surveillance.



Xiong Yang received the M.S. degree from the Department of Computer Science, Huazhong Normal University, Wuhan, China, in 2005. Currently, he is working towards the Ph.D. degree at the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China.

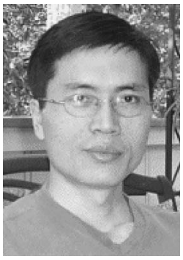
During 2006-2009, he worked as a Research Assistant at Lotus Hill Institute (LHI), Ezhou, China. He is now with the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology. His research interests are focused on pattern recognition and computer vision.

Liang Lin received the B.S. and Ph.D. degrees from Beijing Institute of Technology (BIT), Beijing, China, in 1999 and 2008, respectively. He studied in the Department of Statistics, University of California, Los Angeles (UCLA), as a visiting Ph.D. student during 2006-2007.

He was a Postdoctoral Research Fellow at the Center for Image and Vision Science (CIVS), UCLA, and a Research Scientist at Lotus Hill Institute, Ezhou, China, during 2007-2009. Currently, he is an Associate Professor at Sun Yat-Sen University (SYSU), Guangzhou, China, and the Deputy Director of the Laboratory of Intelligent Information Processing at SYSU. His research interests include but are not limited to object recognition, graph and shape matching, image parsing, and visual tracking.

Mun Wai Lee received the B.Eng. and M.Eng. degrees from the Department of Electrical Engineering, National University of Singapore, Singapore, in 1998 and 1999, respectively, and the Ph.D. degree from the Department of Computer Science, University of Southern California, Los Angeles, in 2006.

He has been a Research Scientist at Object-Video Inc., Reston, VA, since 2006. His research interests include computer vision, machine learning, pattern recognition, and artificial intelligence.



Song-Chun Zhu received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1991 and the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1994 and 1996, respectively.

Currently, he is a Professor at the Department of Statistics and the Department of Computer Science, University of California, Los Angeles (UCLA). Before joining UCLA in 2002, he worked in the Division of Applied Math at Brown University, the Department of Computer Science at Stanford University, and the Department of Computer and Information Science at Ohio State University. His research interests include computer vision and learning, statistical modeling, and stochastic computing, and visual arts. He has published more than 100 papers in computer vision. In 2005, he founded, with friends, the Lotus Hill Institute for Computer Vision and Information Science, Ezhou, China, as a nonprofit research organization.

Dr. Zhu received a number of honors, including the 2008 J. K. Aggarwal prize from the International Association of Pattern Recognition, the David Marr Prize at the 2003 IEEE International Conference on Computer Vision, the Marr Prize honorary nominations, in 1999 and 2007, respectively, the 2001 Sloan Fellowship in Computer Science, the 2001 U.S. National Science Foundation (NSF) Career Award, and the 2001 U.S. Office of Naval Research (ONR) Young Investigator Award.