

SAVE: A Framework for Semantic Annotation of Visual Events

Mun Wai Lee, Asaad Hakeem, Niels Haering
ObjectVideo
{mlee, ahakeem, nhaering}@objectvideo.com

Song-Chun Zhu
Dept. of Statistic and Computer Science
University of California, Los Angeles
sczhu@stat.ucla.edu

Abstract

In this paper we propose a framework that performs automatic semantic annotation of visual events (SAVE). This is an enabling technology for content-based video annotation, query and retrieval with applications in Internet video search and video data mining. The method involves identifying objects in the scene, describing their inter-relations, detecting events of interest, and representing them semantically in a human readable and query-able format. The SAVE framework is composed of three main components. The first component is an image parsing engine that performs scene content extraction using bottom-up image analysis and a stochastic attribute image grammar, where we define a visual vocabulary from pixels, primitives, parts, objects and scenes, and specify their spatio-temporal or compositional relations; and a bottom-up top-down strategy is used for inference. The second component is an event inference engine, where the Video Event Markup Language (VEML) is adopted for semantic representation, and a grammar-based approach is used for event analysis and detection. The third component is the text generation engine that generates text report using head-driven phrase structure grammar (HPSG). The main contribution of this paper is a framework for an end-to-end system that infers visual events and annotates a large collection of videos. Experiments with maritime and urban scenes indicate the feasibility of the proposed approach.

1. Introduction

The proliferation of video cameras and networked video storage systems is generating enormous amounts of video data. Efficient automatic video analysis is required to enable retrieval via human readable queries, either by searching the meta-data or text description. Existing video search tools rely mainly on user-annotated tags, captions, and surrounding text to retrieve video based on broad categories. The goal of content-based visual event retrieval is to allow queries based on specific events and event attributes in the video. It requires a more detailed understanding of objects, scene elements and their inter-relations. It also involves inference of complex events including multi-agent activities [6]. Effective annotation

should provide rich information surrounding the visual events, such as “*a red car enters the traffic intersection at a speed of 40 mph at 3:05p.m.*”. Attributes such as *object class* (e.g. car), *scene context* (e.g. traffic intersection), *speed*, and *time*, provide important semantic and contextual information for accurate retrieval and data mining.

We propose a framework *SAVE: Semantic Annotation of Visual Events*; and the architecture is summarized in Figure 1. We adopted the modeling and conceptualization methodology of *stochastic attribute image grammar* [20] to extract semantics and contextual content, where a visual vocabulary is defined from pixels, primitives, parts, objects and scenes. The grammar provides a principled mechanism to list visual elements and objects present in the scene and describe how they are related, where the relations can be spatial, temporal or compositional. Guided by bottom-up analysis and object detection, a bottom-up top-down strategy is used for inference to provide a description of the scene and its constituent elements. With the image parsing result, an *event inference engine* then extracts information about activities and produces a semantic representation. A *text generation engine* then converts the semantic representation to text descriptions. This paper describes the SAVE framework. To date, we have focused on urban and maritime environments to achieve rich annotation of visual events. We plan to later extend the framework to videos in other domains.

1.1. Related work

In literature, extensive work in video annotation has been reported under the TREC Video Retrieval Evaluation program (TRECVID) [15] to categorize video shots using a list of media content concepts. Our proposed method is significantly different as it provides descriptive annotation of each activity in the video. Our work is more closely related to the work by Kojima et al. [9] which generated text description of single-human activities in a laboratory using *case frames*. Compared to [9], we use grammar-based approaches for inferring and annotating a broader range of scenes and events, including multi-agent complex events.

A variety of approaches have been proposed for detecting events in video. Most of these approaches can be arranged into two categories based on the semantic

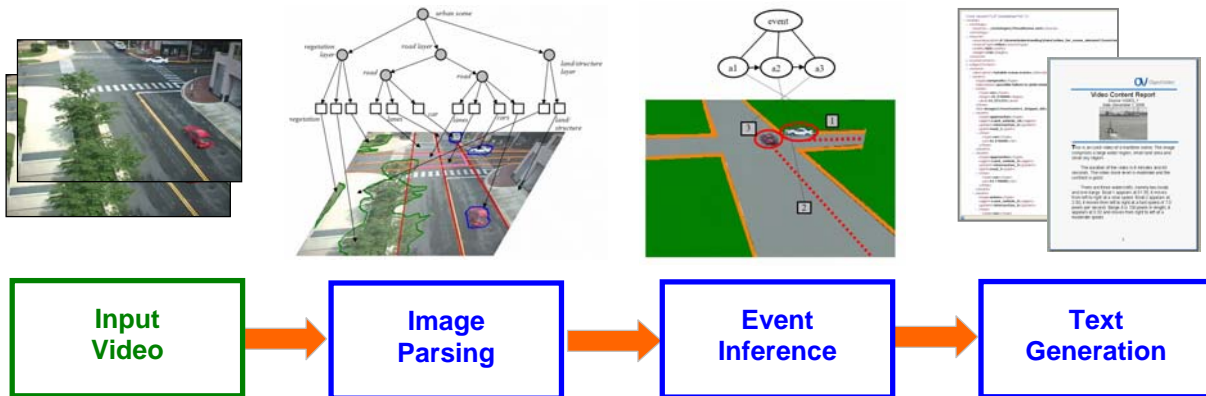


Figure 1: The SAVE framework for semantic annotation of visual events. The three main components are: *image parsing*, *event inference*, and *text generation*.

significance of their representations. Approaches where representations do not take on semantic meaning include Stochastic Context Free Grammars (SCFG) [13], learning methods such as Bayesian Networks (BN) [7], and Hidden Markov Models (HMM) [2]. On the other hand, semantically significant approaches like the state machines [10], and PNF Networks [16] provide varying degrees of representation to the actions and agents involved in the events. Also, the Video Event Representation Language (VERL) was proposed in [14] where complex events are semantically represented with a hierarchical structure.

Detecting and annotating complex visual events in a broader range of scenes would first require the understanding of the scene to provide contextual information surrounding events. Image parsing with scene element labeling [20] is an important stage towards this goal. This motivates the design of the SAVE framework, where the goal is to provide an end-to-end system that infers visual events and annotates a large collection of videos, in a human readable and query-able format.

2. SAVE: Semantic Annotation of Visual Events framework

The overall architecture for SAVE is shown in Figure 1, which consists of three main components. The first component is an *image parsing engine* that consists of the *stochastic attribute image grammar* [20]. Grammars, which are mostly studied in natural language processing, are known for their expressive power, i.e. the capability of generating a very large set of configurations from a small vocabulary using production rules. Transferring the idea of a grammar from natural language processing to computer vision, a visual vocabulary is defined from pixels, primitives, parts, objects and scenes, and specify their spatio-temporal or compositional relations and a bottom-up top-down strategy is used for inference. The grammar also provides a principled mechanism to list visual elements and objects present in the scene and describe how they are related. Also, the bottom-up image analysis techniques include edge detection, segmentation,

and appearance-based object detection.

The output of the image parsing engine is further processed by the second component: the *event inference engine*. In this component, descriptive information about visual events is extracted, including semantic and contextual information, as well as, relationships between activities performed by different agents. The *Video Event Markup Language* (VEML) [14] is adopted for semantic representation and a grammar-based approach is used for event analysis and detection. Finally, in the *text generation engine*, the semantic representation is converted to text description using *head-driven phrase structure grammar* (HPSG) [17]. The following sections describe each of these components in detail.

2.1. Image parsing

The first component in the SAVE framework is the image parsing engine to classify the imagery into scene elements and objects. It consists of a *stochastic attribute image grammar* [20] that serves as a unified methodology for analysis, extraction, and representation of the visual elements and structure of the scene, such as the roads, sky, vehicles, and humans. These image elements form the basis of a visual vocabulary of scenes. At the lowest level of the grammar graph are the image *primitives* such as image patches, lines or color blobs. Serving as basic cues, these primitives are combined to form larger objects and scene structure. The *production rules* realize the composition of the image elements with attributes. As illustrated in Figure 2, graphs are used to represent the grammars where the nodes represent the visual elements and the edges depict the rules defining the relations between the elements.

Under the stochastic attribute image grammar methodology, the image content extraction is formulated as a graph parsing process to find a specific configuration produced by the grammar that best describes the image. The inference algorithm finds the best configuration by integrating bottom-up detections and top-down hypotheses. As illustrated in Figure 2, with a maritime scene as an example, bottom-up detection includes

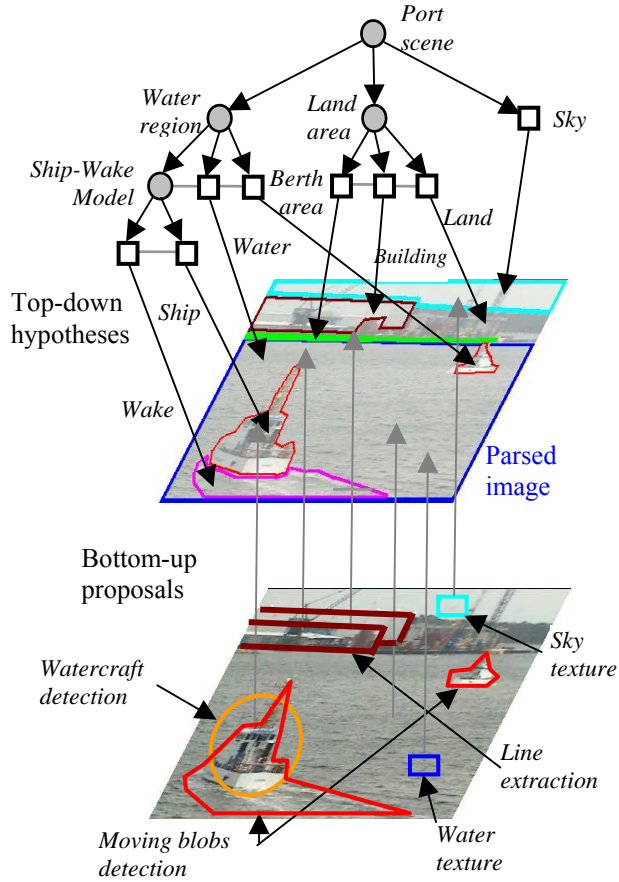


Figure 2: Image content inference with bottom-up proposals and top-down hypotheses. The parsed image is represented by a semantic graph consisting of the image elements and their relationships.

classification of image patches as regions of sky, water, and watercraft, which generate data-driven candidates for the scene content. Top-down hypotheses, on the other hand, are driven by scene models and contextual relations represented by the attribute grammar. The fusion of bottom-up and top-down information yields a more robust image content extraction method.

A parsed image is an instance or configuration of the attribute graph grammar that aims to find a parse-graph that maximizes the posterior probability, under the Bayesian framework. The objective is to find the graph configuration that best represents the input image. The process is initiated with the bottom-up approach that generates candidate proposals by changing the labeling of scene elements based on local features. These proposals are used, in a stochastic manner, to activate the instantiation of *production rules* in the *attribute grammar*, which in turn generate *top-down hypotheses*. The top-down inference guides the search based on domain knowledge and contextual relations between objects and scene elements, as illustrated in Figure 2. These rules specify how attributes of elements are estimated and passed along the parse-graph through the use of constraint equations. *Data-Driven Markov Chain Monte Carlo* (DDMCMC) [19] is used to maximize the posterior probability for scene content inference.

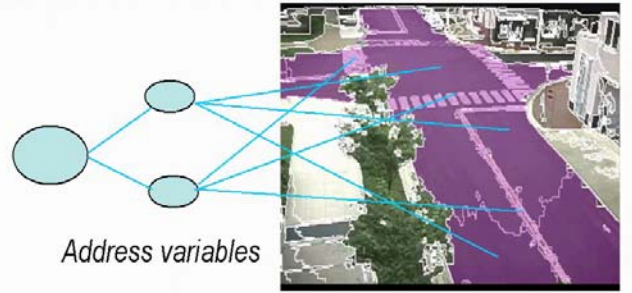


Figure 3: Regions of the same type are connected using address variables in the layered image representation framework. A mixture of Gaussians is used for the generative model for the appearance of each region type.

The *attribute grammar* method is formulated as a graph partition problem [1] where the graph is denoted by $G = \langle V, E \rangle$ which is an adjacency graph where V is a set of vertices representing image elements and E is a set of edges or links representing adjacency relations between elements. In image parsing, the objective is to partition the vertices into subsets, $\pi = \{V_1, V_2, \dots, V_n\}$, where vertices in each subset V_i belong to the same element class, and n is the number of partitions. Given the observable data, the objective is to find the partition that maximizes the posterior probability, $p(\pi) = p(\pi | \text{Data})$ over the set of all possible partitions Ω . Each edge in the partition graph is denoted by $e = \langle s, t \rangle \in E$, where s, t , are vertex indices. When computing the posterior probability, a discrimination model is used to measure the coherency between adjacency nodes. This local probability is denoted by, $q_e = q(e | F(s), F(t))$, where $F(s), F(t)$ are local feature vectors (such as color, texture, and location) used as inputs to the discrimination model. The image statistics is learned from a set of annotated training data.

The bottom-up analysis involves detecting image features and other low level processing for the classification of the objects and scene elements. For moving objects, we used the Adaboost classification method which utilizes a set of features to detect humans, vehicles, and watercrafts. The set of features include Histogram of Oriented Gradient (HOG) [3] and C2 features [18]. For scene elements, we initially perform over-segmentation to divide the image into super-pixels using the mean-shift color segmentation method. Since adjacent pixels are highly correlated, analyzing scene elements at the super-pixel level reduces the computational complexity. For each super-pixel, a set of local features are extracted including color, hue, saturation, intensity, spatial texture, difference of oriented Gaussian filter, image location, size, shape and normalized 'x' and 'y' means. These local features are used for *image parsing* within the *attribute grammar* framework. The detected scene elements include water, road, sky, vegetation, and land.

Top-down scene model is represented using a *mixture Markov random field* [5]. Compared to traditional MRF, the mixture MRF can handle nodes and edges of different

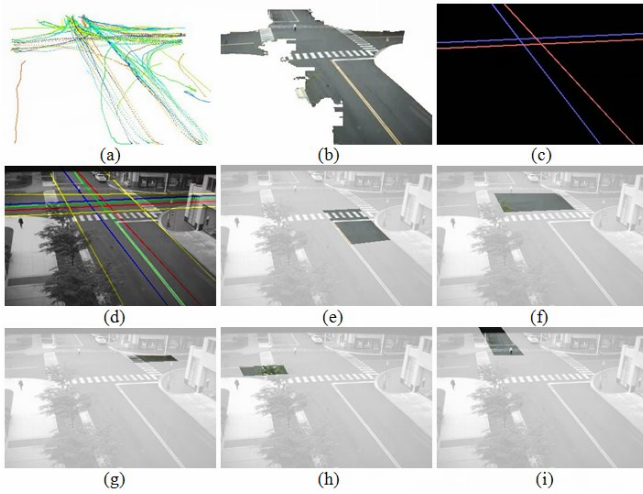


Figure 4: Road extraction and zone demarcation. (a) Object trajectories, (b) Road scene element detection, (c) Extracted roads (color-coded with traffic direction), (d) Extracted roads boundaries and intersection, (e)-(i) Extracted ROI zones on roads. The extracted semantics information and ROI zones are used for event detection and description.

types. For scene modeling, mixture MRF can be used to represent relations between non-adjacent scene elements which, in the graphical model, are connected by edges known as *address variables* (see Figure 3). Thus the method is more flexible and can handle more complex relations. In a typical image, regions of the same element type (e.g. roads, water) share similar characteristics even though they might not be connected. This is modeled with a *layered image representation* [5], where the color distribution of each image layer is represented by a *mixture Gaussian model*. For each scene element type, the *prior* on the number of Gaussian components is modeled using *Poisson distribution* and is estimated from training data.

To classify scene elements in the image, the *data-driven Markov chain Monte Carlo* (DD-MCMC) [19] is used for Bayesian inference. This is an iterative process modeled as a stochastic Markov chain. Suppose A and B are two different states (i.e. different graph configurations), at each iteration, the acceptance probability of a Markov transition from A to B is given by

$$\alpha(A \rightarrow B) = \arg \min \left\{ 1, \frac{q(B \rightarrow A) p(B)}{q(A \rightarrow B) p(A)} \right\},$$

where $q(B \rightarrow A)$ is the proposal probability formulated from bottom-up observation. The output of DD-MCMC algorithm is a parsed image with labeled scene elements.

2.2. Event inference

The second component in the SAVE framework is the event inference engine which leverages the existing state-of-the-art in knowledge representation and natural language processing, and focuses on extracting descriptive information about visual events, including semantic and contextual information as well as relationship between

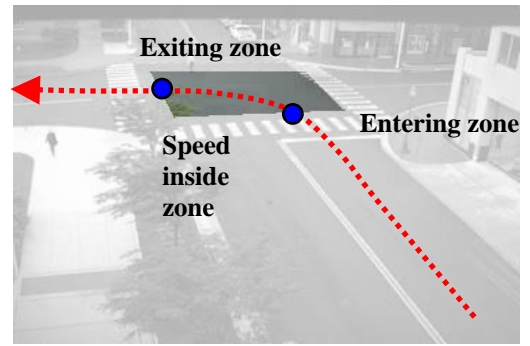


Figure 5: Events with respect to zone include entering and exiting. The speed of the object inside the zone is measured and checked for abnormality.

activities performed by different agents. The *Video Event Markup Language* (VEML) [14] is adopted for semantic representation, while a grammar-based approach is used for event analysis and detection. The module is composed of four sub-components described next.

2.2.1 Scene Region Analysis

The first sub-component of the event inference engine deals with the scene region analysis. This sub-component 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 using information from existing transportation ontology [12]. 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 (the object class that travels on road), which are clustered to extract roads. The vehicles are detected and tracked using background subtraction and Kalman filtering. 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. The algorithmic steps of road extraction process are illustrated in Figure 4 (a)-(d). Based on ontology, similar inference can be made on other types of scene regions, such as waterway (used by watercraft), and sidewalk (used by pedestrians).

<p style="text-align: center;">Scene Understanding <i>Report generated on <date and time></i></p> <p>Source Information <i>Information on video source ...</i></p> <p>Scene Context <i>Scene context description ...</i></p> <p>Object Summary <i>Brief description of objects in the scene ...</i></p> <p>Events of Significance <i>Significant events detected ...</i></p> <p>Detailed Object Information <i>Detailed events description ...</i></p>	<pre> <document> <title>Scene Understanding</title> <section> <title>Source Information</title> <paragraph> <sentence> ... </sentence> </paragraph> </section> <section> <title>Scene Context </title> ... </section> </document> </pre>
(a) Text report layout	(b) Corresponding text-planner structure (XML)

Figure 6: An example of document structure for text report.

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. Examples of zone demarcation result are shown in Figure 4 (e)-(i).

2.2.2 Spatio-Temporal Analysis

The second sub-component deals with the analysis of the spatio-temporal trajectories of moving objects. Assuming that an object is moving on a ground plane, 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. Comparing new trajectories to historical information, abnormal speeding events inside the zone can be detected (see example in Figure 5).

Speed information is generally expressed in image-centric measure (pixel-per-second). Objects’ image sizes in then used to coarsely estimate the ground sample resolution (meter-per-pixel) to compute metric-based speed measure (e.g. mile-per-hour). More accurate estimation can be obtained by automatic calibration method but this is outside the scope of this paper.

2.2.3 Event Detection

The third sub-component utilizes the information obtained from the previous sub-components to perform

event detection. We use the *stochastic context-free grammar* (SCFG) [13] to detect events. The grammar is used to represent activities where production rules describe how activities can be broken down into sub-activities or actions. In the stochastic approach, a probability is attached to each production rule for the sequence of actions, depicting the likelihood of a particular production rule to be utilized in the generation of the activity.

Given an input sequence of detected actions, the *Earley-Stolcke Parsing algorithm* [4] is used to parse the sequence based on the SCFG. The parsing algorithm is an iterative process for each input sequence of actions. Each iteration consists of three steps: *prediction*, *scanning*, and *completion*, in a manner similar to the top-down bottom-up approach of the image attribute grammar. The parsed result defines the event that occurred. To overcome the presence of unrelated concurrent actions, we use semantics and contextual information to identify *associations* between visual elements and actions based on factors such as target type, time, location, behavior and functional roles. This process removes unrelated data for event detection, thereby reducing computational complexity.

For instance, this method is used to analyze multi-agent events around traffic intersections, where vehicle’s actions include approaching and stopping before an intersection, entering and exiting the intersection. By observing sequences of these actions, the system learns the stochastic grammar for these actions and later uses the grammar to detect unusual events. In addition to using learning-based method, the system also detects user-defined events expressed in VERL [14], where complex events are represented as hierarchies of sub-events or atomic actions.

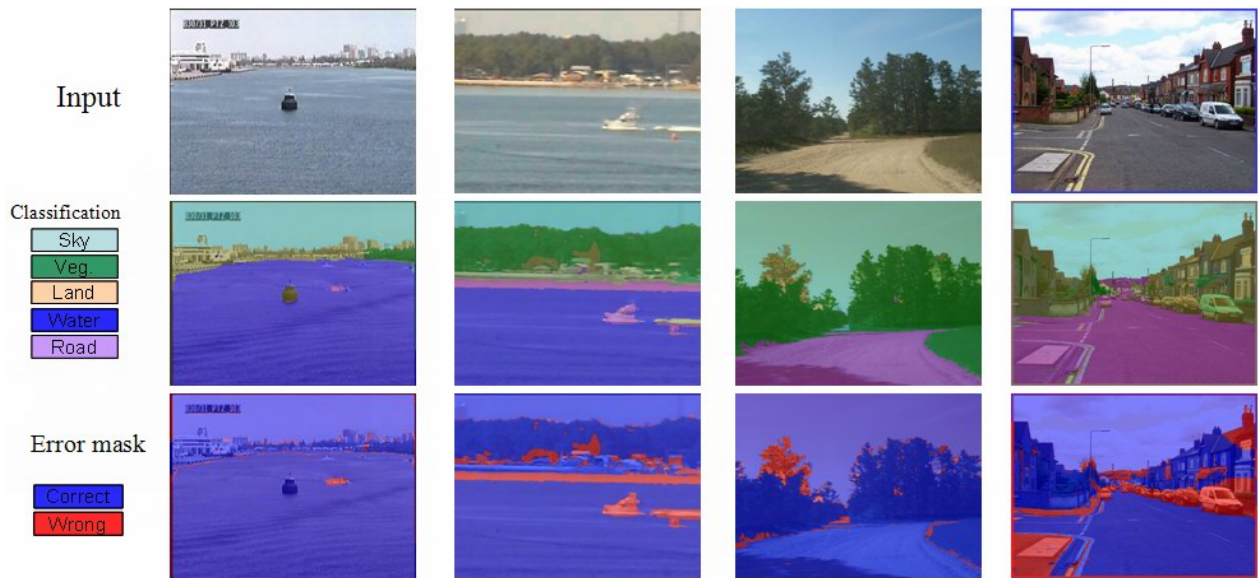


Figure7: Examples of scene element classification result.

Actual\Detected	Road	Vegetation	Water	Land	Sky
Road	0.781	0.086	0.026	0.098	0.009
Vegetation	0.069	0.677	0.001	0.240	0.013
Water	0.126	0.008	0.824	0.018	0.023
Land	0.223	0.171	0.025	0.527	0.053
Sky	0.025	0.056	0.021	0.132	0.766

Table 1. Confusion matrix at pixel level.

	Precision	Recall
Sky	0.90	0.75
Water	0.93	0.74
Road	0.69	0.83
Vegetation	0.79	0.67
Land	0.41	0.51

Table 2. Classification result for each scene element type.

2.2.4 Semantic Representation

The final sub-component expresses the extracted information to a semantic representation that encodes the detected scene elements, moving objects, events, and the spatial and temporal relation between them in a semantic representation. We adopted the *Video Event Markup Language* (VEML) [14] for semantic representation which is based on XML and therefore can be easily generated and parsed using standard software tools. The format of the output XML file consists of the following main sections: *Ontology* that lists the location of the ontology file and describes the visual entities, subtype relations, other relations, and their properties; *Streams* that identifies the input video source; *Context* that describes scene contextual information related to the video which include static scene elements such as roads, intersections, etc; *Objects* that describe objects that are present in the scene; and *Events* that describes events detected in the scene. The *Events* section is further divided into *Significant Events* (e.g. traffic violation, abnormal events) and *Detailed Events* (a complete list of all detected events).

2.3. Text generation

The third component in the SAVE framework is the text generation engine which generates text reports based on the output of the event inference engine. The text generation process is a pipeline of two distinct tasks: text

planning and text realization. The *text planner* selects the content to be expressed, specifies hard sentence boundaries, and organizes content information according to these boundaries. Based on this formation, the *text realizer* generates the sentences by determining grammatical form and performing word substitution. We now describe these tasks in detail in the following sections.

2.3.1 Text Planner

The *text planner* module translates the semantics representation to a 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 lexical structure. The output of *text planner* is based on text generator input representation known as a *functional description* (FD) which has a *feature-value pair* structure, commonly used in text generation input schemes. Our system uses the *HALogen* representation [11]. 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*, *predicates*, and other functional properties. The text planner module also organizes the text report document. The document structure is dependent on applications. For this paper, a simple text report document structure is designed, as shown in Figure 6.

2.3.2 Text Realizer

A simplified *head-driven phrase structure grammar* (HPSG) [17] is used to generate text sentences during the text realization task. HPSG consists of two main components: a highly structured representation of grammatical categories; and a set of descriptive constraints for phrasal construction. The generation grammar represents the structure of features with production rules. Textual descriptions of visual events are mostly *indicative* or *declarative* sentences and this simplifies the grammar structure significantly. The grammatical categories in HPSG consist of: *S*, sentences; *NP*, noun phrase; *VP*, verb phrase; *PP*, preposition phrase; *N*, noun; *V*, verb; *A*, adjective; *DET*, determiner; *Pron*, pronoun; *P*, Preposition; and *ADV*, Adverb. The set of production rules or descriptive constraints include: $S \rightarrow NP VP$, $NP \rightarrow DET (A) N$, $VP \rightarrow V NP$, $VP \rightarrow V PP$, $VP \rightarrow V ADV$, and $PP \rightarrow P NP$. 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: $S \rightarrow NP(per,num) VP(per,num)$. A unification process [8] matches the input features with the grammar in a recursively manner, and the derived lexical tree is then linearized to form sentence output.

As an example, the sentence “*Boat_1 follows Boat_0 between 08:34 to 08:37*” is generated from the following functional description

```
(e1 / follow
:AGENT (d1 / Boat_1 )
:PATIENT (d2 / Boat_0)
:TEMPORAL_LOCATING (d3 / time
:ORIGIN 08:34
:DESTINATION 08:37)).
```

3. Results

We focused our evaluation on urban traffic and maritime scenes and it consists of two parts. First, we evaluated our method for scene labeling with static images. Second, we evaluated event detection and meta-data/text generation with sequences of different scenes. The evaluation of object tracking is outside the scope of this paper. The results and discussion follow.

3.1. Image parsing

For evaluation, a dataset of 90 different scenes is collected, of which 45 is used for training and the remaining for testing. These include maritime and urban scenes. Figure 7 shows some result of scene element classification and error masks. The overall classification accuracy is 73.6%. For comparison, the SVM method was used to classify each superpixel region independently and the overall accuracy by SVM is 60.4%.

Table 1 shows the confusion matrix at the pixel level

and Table 2 shows the breakdown of *recall* and *precision* for each scene element type. The performance is reasonable for *sky*, *water* and *road*, while there is some confusion between *land* and *road*, as well as *land* and *vegetation*, which have similar appearance. In future work, we plan to investigate the use of global feature models to improve the overall classification.

3.2. Event detection and text generation

We processed 10 sequences of urban and maritime scenes, with a total duration of about 120 minutes and contain more than 400 moving objects. Visual events were extracted and text descriptions are generated. Detected events include: *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* maritime marker or land area, and an object *following* another object.

When annotating these events in both meta-data and text description, the system extracts and provides information about the object class, scene context, position, direction, speed, and time. Examples of text description and corresponding video snapshots are shown in Figure 8.

The detected events are compared with manual annotation for selected events, and the recall and precision measures are shown in the table below. The results are promising and illustrate the system’s ability to detect events using context information (e.g. events at road intersection). The recall for “turning” events is relatively low because of poor estimation in motion direction from low perspective views in some scenes; whereas human can use other appearance cues to infer turning motion.

Events	Recall	Precision
Enter/leave scene	0.93	0.96
Turning	0.66	0.75
Moving at abnormal speed	0.75	0.90
Crossing traffic intersection	0.88	0.94
Failure-at-yield at intersection	0.84	0.91
Watercraft approaching marker/land	0.67	0.80

4. Discussion

This paper proposes the SAVE framework that provides an end-to-end automatic system for parsing video, extracting visual event content, and providing semantic and text annotation. A key feature of this framework is the use of various grammar-based approaches to represent and infer visual content so that it can be seamlessly transformed from parsed image to semantic meta-data format and finally to textual description. We have applied this approach on selected scenarios in urban traffic and maritime scenes and demonstrated capabilities in visual event inference and text description generation. The framework can be extended to other domains although the fundamental object detection and classification technology


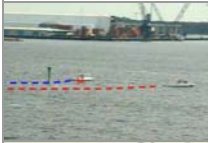





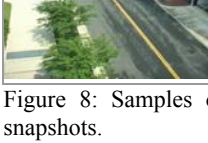
	Boat_2 enters the scene on water region at 19:50. Boat_2 approaches maritime marker at 20:09.
	Boat_4 follows Boat_3 between 35:36 and 37:23
	Boat_7 turns right at 55:00.
	Land_vehicle_359 approaches intersection_0 along road_0 at 57:27. It stops at 57:29.
	Land_vehicle_360 approaches intersection_0 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.

Figure 8: Samples of generated text and corresponding video snapshots.

needs to be improved.

This work is different from but complementary to existing technology in video shot categorization and caption-based video annotation, by providing richer and semantic-oriented annotation of visual events in video. With video content expressed in both XML and text format, this technology can be easily integrated with full text search engine as well as XML-query or relational database search engine to provide accurate content-based video retrieval. As a future work, we are developing a web-based video service from which users can retrieve video via keyword searches and semantic-based queries using standard web interface.

Acknowledgement

This research was supported by the Office of Naval Research under Contract # N00014-07-M-0287.

References

- [1] A. Barbu and S.C. Zhu. "Graph partition by Swendsen-Wang cut." *ICCV*, Nice, France, October, 2003.
- [2] A. Bobick and A.D. Wilson, "A state based approach to the representation and recognition of gesture", *PAMI* 19(12):1325-1337, December 1997.
- [3] N. Dalai and B. Triggs, "Histograms of oriented gradients for human detection" *CVPR* 2005, vol. 1, pp 886-893.
- [4] J.C. Earley, *An Efficient Context-Free Parsing Algorithm*. PhD thesis, Carnegie-Mellon University, 1968.
- [5] R.X. Gao, T.F. Wu, N. Sang, and S.C. Zhu "Bayesian Inference for Layer Representation with Mixed Markov Random Field" *EMMCVPR, Springer LNCS 4679*, Ezhou, China, Aug 2007.
- [6] A. Hakeem and M. Shah, "Learning, Detection and Representation of Multiple Agent Events in Videos" *Artificial Intelligence Journal*, 2007.
- [7] S. Hongeng, F. Bremond and R. Nevatia, "Bayesian Framework for Video Surveillance Application", *ICPR*, vol I, pp. 164-170, September 2000.
- [8] K. Knight, "Unification: A Multidisciplinary Survey." *ACM Computing Surveys*. 21 (1) (1989).
- [9] A. Kojima, T. Tamura, and K. Kukunaga, "Natural Language Description of Human Activities of Video Images Based on Concept Hierarchy of Actions," *International Journal of Computer Vision*, vol. 50, pp. 171-184, 2002.
- [10] D. Koller, N. Heinze, and H. Nagel, "Algorithmic Characterization of Vehicle Trajectories from Image Sequences by Motion Verbs". In *Proc. of Computer Vision and Pattern Recognition*, pp.90-95, 1991.
- [11] I. Langkilde-Geary and K. Knight, "HALogen Input Representation", <http://www.isi.edu/publications/licensed-sw/halogen/interlingua.html>.
- [12] B. Lorenz, H. J. Ohlbach, L. Yang "Ontology of Transportation Networks", *Rewerse*, 2005.
- [13] D. Moore and I. Essa, "Recognizing Multitasked Activities using Stochastic Context-Free Grammar" *CVPR* 2001
- [14] R. Nevatia, J. Hobbs and B. Bolles, "An Ontology for Video Event Representation", *IEEE Workshop on Event Detection and Recognition*, June 2004.
- [15] NIST, *TREC Video Retrieval Evaluation*, <http://www-nlpir.nist.gov/projects/trecvid/>.
- [16] C. Pinhanez and A. Bobick, "Human Action Detection Using PNF Propagation of Temporal Constraints." *CVPR*, pp.898-904, 1998.
- [17] C. Pollard and I.A. Sag, (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.
- [18] T. Serre, L. Wolf and T. Poggio, "Object Recognition with Features Inspired by Visual Cortex", *CVPR* 2005.
- [19] Z.W. Tu and S.C. Zhu, "Parsing images into regions, curves and curve groups", *IJCV*, 69(2), 223-249, August, 2006.
- [20] S.C. Zhu and D.B. Mumford, "Quest for a stochastic grammar of images", *Foundations and Trends of Computer Graphics and Vision*, 2006.