FIRST-ORDER LOGICAL-PROBABILISTIC INTERPRETATION OF TRAFFIC LANES

Angelo Gurzoni Jr*, Paulo E. Santos†, Carlos R. C. Souza‡, Fabio G. Cozman§

*Escola Politécnica da Universidade de São Paulo
São Paulo, Brazil

†Dept. of Electrical Engineering,
Centro Universitário da FEI
São Bernardo do Campo, SP – Brazil

‡Mercedes-Benz do Brasil
S. Bernardo do Campo, SP, Brazil

§Escola Politécnica da Universidade de São Paulo
São Paulo, Brazil

Emails: jgurzoni@ifc.org, psantos@fei.edu.br, carlos.roberto@daimler.com, fgcozman@usp.br

Abstract— This paper presents a probabilistic logical model of traffic rules with the goal to provide high-level interpretation of the traffic rules regarding lane signalisation in video scenes, as observed from a vehicle's viewpoint. The images are provided by a monocular camera attached to a vehicle driving in normal traffic situations. A low-level computer vision algorithm classifies the type of lane dividers and their relative positions with respect to the vehicle, then sends the information to be used as evidences by a probabilistic inference engine that reasons about the vehicle's location and actions. The inference is accomplished using a first-order knowledge base formalising right-handed traffic rules. Uncertainties inherent to the sensors are treated within a probabilistic framework, the Markov Logic Networks, and results are compared to a well-known baseline classifier, the Naïve Bayes. We evaluate the use of these techniques under real-world traffic situations.

Keywords— Spatial reasoning, Image Understanding, Markov Logic Networks

Resumo— Este artigo apresenta um modelo lógico-probabilístico de regras de trânsito com o objetivo de fornecer uma interpretação de alto nível sobre faixas de rodagem a partir de cenas de vídeo, tal como observado do ponto de vista de um veículo. As imagens são fornecidas por uma câmara monocular ligada à um veículo em condições normais de trânsito. Um algoritmo de visão computacional de baixo nível classifica o tipo de separadores de pista e as suas posições relativas em relação ao veículo. Em seguida, esta informação é utilizada como evidência por um mecanismo de inferência probabilística cujo objetivo é determinar as ações do veículo. Esta inferência é realizada utilizando uma base de conhecimento de primeira ordem representando um subconjunto de regras de trânsito. Incertezas inerentes aos sensores são tratadas dentro de uma estrutura probabilística, as Redes Lógica de Markov, e os resultados são comparados com um classificador *Naïve Bayes*. O uso destas técnicas foi avaliado em situações de tráfego urbano real.

Palavras-chave— Qualitative Spatial Reasoning, Interpretação de cenas, Redes Lógicas de Markov.

1 Introduction

In this paper, we investigate the use of a probabilistic relational model to bring contextual information into consideration for the task of interpreting the function of lane dividers from videos obtained on a car in normal traffic situations. The lane dividers' position and type (e.g., *continuous*, *dashed*) are extracted by a computer vision system and sent to a probabilistic inference engine, where they serve as evidence to infer the location and actions of the vehicle. To model the traffic rules, a probabilistic framework with capabilities to deal with the uncertainties inherent to the sensors and environment is used, the Markov Logic Network (MLN) (Domingos and Lowd, 2009). In our previous work (Souza and Santos, 2011), we present preliminary results suggesting that Markov Logic Networks could be used to efficiently encode relational rules representing contextual knowledge (traffic rules) about the traffic domain, while also providing the appropriate means to infer the functionality of traffic lanes from visual observation. However, in (Souza and Santos, 2011) no comparison was presented between the results obtained from the MLNs with more traditional methods such as Naïve Bayes. This comparison is the main contribution of the present paper. Besides this, the experiments presented in (Souza and Santos, 2011) were conducted on a preliminary dataset within which not all of the events were represented. In the present work, however, the experiments were conducted on a more complete dataset containing containing examples of every event formalised in this work.

2 Related Work

The extent of which MLNs can be applied to the task of interpreting traffic scenes from video is an issue that we address in the present work. In this context, the seminal work of Nagel (Nagel, 2004) was the first to attempt the automatic description of image sequences from traffic situation using concepts. On a broader sense, these ideas were at the starting point of the field called *semantic scene understanding*, which aims at abstracting an image sequence into meaningful units that are further used to check if an event of interest has occurred, a survey of this field is given in (Lavee et al., 2009). In particular, the work presented in (Brehar et al., 2011) considers the visual classification of objects in real traffic scenes and the representation of the observed situations by means of the OpenCyc ontology[1]. Although this work considers high-level representation and reasoning via the Cyc technology, the inferences are deterministic, since no probabilities are assigned to the domain predicates and rules.

In terms of probabilistic-logic theories for moving vehicles we can cite the work described in (Parker et al., 2007) that uses optimisation methods in order to define the probability about where a moving objects is located at a given time. This theory, however, assumes that the observer is static and has a global viewpoint of the objects, as the aim of that work is to track objects as seen from radars.

To the best of our knowledge, the investigation reported in (Tran and Davis, 2008) was the first to use Markov logic networks to the task of event modelling from video sequences. Similar to the work described in this paper, the authors in (Tran and Davis, 2008) use probabilistic inference for querying the scenes from a domain represented by first-order rules. In these works, scenes from a static viewpoint observing human actions in structured scenarios are analysed (e.g. two agents playing basketball); whereas, in the present paper we consider normal traffic scenes as observed by a moving vehicle.

Markov logic networks were also used in (Hensel et al., 2010; Souza and Santos, 2011) to infer object relations in traffic scenes. In contrast to the present paper, the work reported in (Hensel et al., 2010) assumes a bird's eye view of the scenes, leaving for a future work the scene interpretation and reasoning from the viewpoint of a moving vehicle.

3 Methods

This section presents the two stages of the proposed LAS system: the low-level vision system that detects the lane dividers condition and the probabilistic-logic inference engine that reasons about this information.

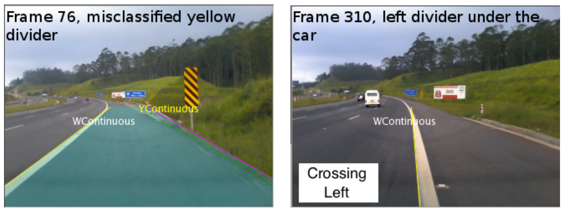
[1] http://www.opencyc.org/opencyc

3.1 Low-Level Vision system

The low-level vision part of the method is based on an off-the-shelf Hough transform algorithm for lane-boundary detection (Yu and Jain, 1997)., which outputs the classification of lane dividers on the left and right of the vehicle at each camera frame. These lane dividers are classified with the following *types*: *WContinuous*, representing a *white continuous* divider; *WDashed*: representing a *white dashed* divider; *YContinuous*: standing for a *yellow continuous* divider; *YDashed*: that represents a *yellow dashed* divider; and, *Merge*: which represents the access to enter the road and to depart from it.

For every frame, the vision module outputs a pair $\langle divider(Left, type, status, time), divider(Right, type, status, time)\rangle$, representing the types of the lane dividers located on the left and right of the vehicle (where *status* states whether the divider is at the sides or under the vehicle, and *time* is the frame where the information was acquired). It is also output by the vision system whether the vehicle is crossing towards the left or right (*crossingLeft* or *crossingRight*).

Figure 1 presents some frames processed with this algorithm.



(a) (b)

Figure 1: Snapshots of traffic scenes.

3.2 Markov Logic Networks

A Markov logic network L is a set of formulae F_i in first-order logic with a weight w_i (real number) attached to each formula. This can be viewed as a template for constructing Markov networks $M_{L,C}$, where C is a set of constants. The probability distribution of the domain variables, over possible worlds x (interpretations of F_i), is given by:

$$P(X=x) = \frac{1}{Z} exp\left(\sum_i w_i f_i(x)\right) = \frac{1}{Z}\prod_i \phi_i(x_{\{i\}})^{n_i(x)}, \quad (1)$$

where $n_i(x)$ is the number of true groundings of F_i in x, $x_{\{i\}}$ is the state of the i-th clique which has a corresponding feature $f_i(x) \in \{0,1\}$ and an associated weight $w_i = log\phi_i(x_{\{i\}})$. Z is a normalisation factor, also known as the partition function, $Z = \sum_{x\in\mathcal{X}}\prod_i \phi_i(x_{\{i\}})$.

The basic idea of MLN is to extend the innate determinism of the first-order knowledge bases (KB), allowing these KBs to deal with probabilistic uncertainty. The weight associated to each formula reflects its strength as a constraint. Formulae with infinite weights rule out worlds in which they are not satisfied, whereas lower weight formulae make them less probable.

Inference in MLN can be probabilistic or logical, with complexity #P-Complete and NP-Complete, respectively (Domingos and Lowd, 2009). Efficient algorithms for both exact and approximate inference methods, such as the *Markov Chain Monte Carlo* (MCMC) or Belief Propagation, are possible in MLNs.

Learning in MLN can be discriminative or generative, and the most common algorithm for both learning and inference is the MC-SAT. The MC-SAT is a slice sampling MCMC which uses a combination of satisfiability testing and simulated annealing to sample from the slice of a distribution (Domingos and Lowd, 2009). MC-SAT treats deterministic or near-deterministic dependencies by efficiently finding isolated modes in the distribution, resulting in a fast mixing of Markov chain.

The next section describes the formalisation of the contextual information about the use of traffic lane dividers and its application as MLN formulae.

4 Contextual Information From Traffic Scenery

Contextual information is injected into the inference engine through the addition of predicates encoding traffic rules related to the use of lane dividers in traffic situations. One important characteristic of a lane assistance system is the fact that the agent of perception is inside the context, e.g., the point of view of the agent is the same of the driver. Therefore, the formalisation described in this section follows this concept: the predicates defined below will be always relative to the driver's viewpoint.

This work assumes a typed first-order language, whose sorts are *type*, *way*, *lanepos*, *status* and *time*. The domains of these sorts are defined as follows:

- *type* $\in$ {*YContinuous, WContinuous, YDashed, WDashed, Merge*}, representing the divider's type (as mentioned in Section 3);
- *way* $\in$ {*One, Two*} that represents the road's allowed traffic directions, i.e. a road may be either a *One* or a *Two*-way road;
- *lanepos* $\in$ {*Left, Centre, Right*}, representing vehicle's or lane's position by the constants *Left*, *Centre* or *Right*. The vehicle position is given relative to the road (e.g., "the vehicle is on the centre lane"), whereby *Centre* is considered every position that is not in the extreme left or right lanes. The divider position is relative to the vehicle (e.g., "the divider is on the left side of the vehicle");
- *status* $\in$ {*Sides, Under*}, that indicates if a divider is *Under* the vehicle (e.g., during a crossing) or at the vehicle *Sides*;
- *time*, represented as t, is defined on the set of frames obtained by the vision apparatus.

Using these terms, a number of predicates were defined (described below).

- *divider(lanepos,type,status,t)* : represents the divider identification with lane position *Left, Centre, Right*, its type (*Dashed, Continuous*), its status (*Sides, Under*), at a time t. Ground facts of *divider*/4 are output by the vision algorithm at every frame;
- *carRelPos(lanepos,t)*: represents the car's relative position with respect to the road (*lanepos*);
- *crossingLeft(t)*: represents that the vehicle is crossing to the left lane;
- *crossingRight(t)*: represents that the vehicle is crossing to the right lane;
- *emergencyLane(t)*: represents that the vehicle is on the emergency stop lane;
- *wrongWay(t)*: represents that the vehicle is on the wrong way of the road;
- *roadWay(way,t)*: represents the road's allowed traffic directions.

With these predicates, MLN formulae were constructed to encode traffic rules about right-handed traffic and the knowledge about the use of lane dividers, as shown in the formulae below (first presented in (Souza and Santos, 2011)). It is worth pointing out that, initially, the formulae below are only first-order logical sentences without weight. As described in the next section, the weights were further learnt from the training data using MC-SAT on the related Markov network.

In the formulae below we assume that constants are written in uppercase letters, variables in lowercase and that, unless explicitly written, every free variable is universally quantified.

1. If the vehicle perceives a yellow continuous divider under it or there is one at its right side, it is doing a prohibited manoeuvre:
$divider(Left, YContinuous, Under, t) \vee$
$divider(Right, YContinuous, sides, t) \Rightarrow$

$wrongWay(t) \wedge$
$prohibitedManoeuvre(t) \wedge roadWay(Two, t) \wedge$
$carRelPos(Left, t) \wedge \neg carRelPos(Centre, t) \wedge$
$\neg carRelPos(Right, t) \wedge \neg emergencyLane(t).$

2. If there is no evidence of a two way road, consider it to be a one way road, where p is a variable for position and s is a variable for *status*:
$\neg divider(p, YContinuous, s, t) \wedge$
$\neg divider(p, YDashed, s, t) \Rightarrow$
$roadWay(One, t) \wedge \neg roadWay(Two, t).$

3. If there is a yellow continuous or a yellow dashed divider at any position, the road has two ways:
$divider(Left, YContinuous, s, t) \vee$
$divider(Left, YDashed, s, t) \vee$
$divider(Right, YContinuous, s, t) \vee$
$divider(Right, YDashed, s, t) \Rightarrow$
$roadWay(Two, t) \wedge \neg roadWay(One, t) \wedge$
$carRelPos(Left, t) \wedge \neg carRelPos(Centre, t) \wedge$
$\neg carRelPos(Right, t) \wedge \neg emergencyLane(t).$

4. If the left divider is white continuous and the right divider is white dashed, the road has one way and the vehicle is on the left lane:
$divider(Left, WContinuous, sides, t) \wedge$
$divider(Right, WDashed, sides, t) \Rightarrow$
$roadWay(One, t) \wedge \neg roadWay(Two, t) \wedge$
$carRelPos(Left, t) \wedge$
$\neg carRelPos(Centre, t) \wedge \neg carRelPos(Right, t) \wedge$
$\neg prohibitedManoeuvre(t) \wedge \neg wrongWay(t).$

5. If the divider is dashed on both sides of the view, the vehicle is on the centre of the road:
$divider(Left, WDashed, sides, t) \wedge$
$divider(Right, WDashed, sides, t) \Rightarrow$
$carRelPos(Centre, t) \wedge \neg carRelPos(Left, t)$
$\wedge \neg carRelPos(Right, t) \wedge \neg wrongWay(t) \wedge$
$\neg crossingLeft(t) \wedge \neg crossingRight(t) \wedge$
$\neg prohibitedManoeuvre(t) \wedge$
$\neg emergencyLane(t).$

6. If the left divider is not white dashed and the right is white dashed, the vehicle is on the left lane:
$\neg divider(Left, WDashed, sides, t) \wedge$
$divider(Right, WDashed, sides, t) \Rightarrow$
$carRelPos(Left, t) \wedge \neg carRelPos(Centre, t) \wedge$
$\neg carRelPos(Right, t) \wedge \neg crossingLeft(t) \wedge$
$\neg crossingRight(t) \wedge$
$\neg prohibitedManoeuvre(t) \wedge$
$\neg emergencyLane(t).$

   In the following x and y are variables for lane *type*.

7. If the divider is yellow dashed and it is at the right side of the vehicle, the car is on the wrong way. It is not a prohibited manoeuvre because this condition is permitted during an overtake manoeuvre:
$divider(Right, YDashed, sides, t) \wedge$
$divider(Left, x, sides, t) \Rightarrow wrongWay(t) \wedge$
$carRelPos(Left, t) \wedge \neg carRelPos(Centre, t) \wedge$
$\neg carRelPos(Right, t) \wedge \neg crossingLeft(t) \wedge$
$\neg crossingRight(t)$
$\wedge \neg prohibitedManoeuvre(t) \wedge$
$\neg emergencyLane(t).$

8. If the left divider is white dashed and the right divider is white continuous or merge, the vehicle is on the right lane:
$(divider(Right, WContinuous, sides, t) \vee$
$divider(Right, Merge, sides, t)) \wedge$
$divider(Left, WDashed, sides, t) \Rightarrow$
$carRelPos(Right, t) \wedge$
$\neg carRelPos(Centre, t) \wedge \neg carRelPos(Left, t) \wedge$
$\neg wrongWay(t) \wedge \neg crossingLeft(t) \wedge$
$\neg crossingRight(t) \wedge \neg emergencyLane(t).$

9. If the vehicle is over a right divider or over a left divider, with a right divider at its *sides*, the car is crossing to the right (and analogously for crossing to the left). This prevents one predicate overlapping with another when, as given by the vision system, the divider changes from left to right (or vice-versa) whenever it crosses the middle of the vehicle's field of view:
$divider(Right, x, Under, t) \vee$
$(divider(Left, y, Under, t) \wedge$
$divider(Right, y, sides, t)) \Rightarrow$
$crossingRight(t) \wedge \neg crossingLeft(t).$

$divider(Left, x, Under, t) \vee$
$(divider(Right, y, Under, t) \wedge$
$divider(Left, y, sides, t)) \Rightarrow$
$crossingLeft(t) \wedge \neg crossingRight(t).$

The negation of some predicates on the formulae above denotes unchanged characteristics of the domain given the evidences.

In the next section we show results of the application of the model on real traffic situations.

5 Results

The approach described in this paper was evaluated using 3600 frames, half of which composed the training set and the other half the test set. All the processing (image processing, weight learning and probabilistic inference) was accomplished on a laptop with a processor AMD Turion X2, 2200MHz and 4Gb of RAM.

The weight learning for each formula was executed with MC-SAT algorithm (Domingos and Lowd, 2009) on the training set. MC-SAT was also applied in the inference, whereby the output from the vision system was used as evidences with which the following predicates were queried within the domain's MLN: $carRelPos(Centre, t)$, $carRelPos(Left, t)$, $carRelPos(Right, t)$, *crossingLeft*(t), *crossingRight*(t), *emergencyLane*(t), *wrongWay*(t), *roadWay*(One, t) and *roadWay*(Two, t), for every frame t in the test set.

The ground truth was manually labelled, considering that an event *lane crossing* occurs every time the inferior extremity of a divider appears at the bottom line of the frame (as also inbuilt in the vision system during data acquisition). Crossing a divider was kept in true state until the divider changes the side with respect to the vehicle (i.e., when it crosses the frame centre). We have used individual decision thresholds for each predicate queried that were found empirically.

In order to evaluate the model described in the previous section, we executed the inference engine against the test set and measured the accuracy (Acc), sensitivity (Sens), precision (Prec) and specificity (Spec) of the responses. The results are shown in Table 1 .

The same training and test sets were used to train and execute a naïve Bayes classifier with four features: the Left and Right divider types (*WContinuous*, *WDashed*, *YContinuous* and *YDashed*) and their status (*Ok* and *Under*). These features were all modelled as multivariate multinomial distributions. Results of the naïve Bayes classifier execution are shown in Table 2.

Table 1: Results obtained from probabilistic inference on the domain's MLN

Predicate	**Acc (%)**	**Sens (%)**	**Prec (%)**	**Spec (%)**
carRelPos(Centre,t)	91.0	98.9	75.2	88.1
carRelPos(Left,t)	93.9	93.3	89.5	93.3
carRelPos(Right,t)	90.8	64.4	87.2	97.6
crossingLeft(t)	87.3	74.6	65.3	90.4
crossingRight(t)	76.4	89.2	41.3	73.8
emergencyLane(t)	80.9	78.8	45.9	81.4
wrongWay(t)	98.0	79.3	89.6	99.3
roadWay(One,t)	96.7	99.9	95.4	89.6
roadWay(Two,t)	98.4	99.8	95.3	97.7
Mean value	*91.3*	*86.5*	*76.1*	*91.1*
Standard deviation σ	*7.2*	*12.1*	*19.6*	*7.9*

Table 2: Results obtained from the Naïve Bayes classifier

Predicate	**Acc (%)**	**Sens (%)**	**Prec (%)**	**Spec (%)**
carRelPos(Centre,t)	89.2	98.8	83.8	78.3
carRelPos(Left,t)	96.8	93.6	80.9	97.2
carRelPos(Right,t)	87.7	66.6	97.9	99.2
crossingLeft(t)	89.2	77.7	51.5	90.7
crossingRight(t)	89.4	93.2	61.9	88.7
emergencyLane(t)	97.3	87.2	90.5	98.7
Mean value	*91.6*	*86.2*	*77.8*	*92.1*
Standard deviation σ	*4.3*	*12.0*	*17.7*	*8.1*

6 Discussion

The results of the queries to the domain's MLN, shown in Table 1, indicates that the system was successful on reasoning about the relative position of the car, about the vehicle's actions (whether it is crossing to the emergency lane, or crossing to the left or right of a lane) and the functionality of the road it is driving on (whether a two-way or a one-way road).

The average values obtained with the probabilistic inference were around 91% for accuracy, 86% for sensitivity, 76% for precision, and 91% specificity. attesting for the success of using a probabilistic first-order representation for the task of image interpretation from traffic situations.

The results from the naïve Bayes classifier, though, indicate that, despite its lack of capabilities to model dependencies among predicates, its performance was very close to the one of the proposed MLN. This is not surprising when one analyses the defined MLN's structure (defined by the formulae shown in Section 4). Mostly, the logic rules show dependencies only among the divider's type and status, which is exactly the structure of our naïve Bayes classifier. Therefore, the MLN used in this work is implementing a naïve Bayes model. This fact leads to the question of why should one go into the burden of defining a first-order model, when a simple naïve Bayes would do the job. The answer relies on the representational power of a first-order language, such as the Markov Logic Network, which allowed us to represent simple, common-sense, facts about our knowledge on the use of traffic lanes and these facts become automatically the classifier's structure. The naïve Bayes model was obtained from the MLN structure due to the simplified application domain: the interpretation of traffic-lane function is just a subset of the far more complex domain of traffic scene interpretation. Once we extend this work towards the interpretation of the behaviour of the other agents, the first-order model should naturally lead to more complex Markov network structures.

It is also worth mentioning that the advantage of an MLN model, in contrast to a non-probabilistic logical model, is that the set of rules defining the knowledge base does not need to be complete, or even entirely correct. Such imprecisions in the formulae will be considered within the probabilities assigned to them. Besides, as these probabilities are learnt from real data, they will also account for user-specific interpretations of lane functions.

Besides the MLN model, we attempted to model the traffic lane domain using PRISM (Sato and Kameya, 2008), a well-known probabilistic logic programming language. However, unlike the MLN, PRISM's learning algorithm did not gracefully accept the noisy examples present in the training data. Its explanation-search parameter learning algorithm tries to find explanations to all examples in the database, thus requiring either the explicit modelling of all possible sensory failure conditions encountered in the data or the definition of a *failure* predicate that would represent these situations. To the best of our knowledge, the difficulties in defining a failure predicate in PRISM prevent the direct use of this tool to our domain.

The vision system used in this work did not have an homogeneous performance with respect to the type of lane dividers it could detect, what has produced a higher than normal rate of false positives. A more robust vision module was not pursued as a mean to to verify to what extent querying the developed Markov logic network would give meaningful results under this kind of sensor uncertainty. For most of the experiments, the MLN dealt with the vision classification problems, except in the case of the yellow dashed divider. This was because the misclassification rate at certain points in the video became indeed too high, with correct detection under 50%.

Although the vision system alone is also capable of interpreting the vehicle actions $crossingLeft$ and *crossingRight*, this information was not used during the inference procedure (which involved only the formulae described in Section 4 and the lane classification from the vision system). However, when comparing the interpretation of these actions as obtained by the probabilistic inference and the vision alone, we noticed that the probabilistic inference had considerably better sensitivity (around 74% for *crossingLeft* and 89% for *crossingRight*) than the vision system alone (of around 54% for *crossingLeft* and 51% for *crossingRight*). The overall results from the inference procedure also presented high values for accuracy, precision and specificity for the remainder predicates.

7 Conclusion

This paper proposed a probabilistic logic formalisation of a number of rules related to the use of lane dividers in right-handed traffic. These rules provided the structure of a Markov logic network (MLN) that was used for the inference of the relative vehicle's position in the road, the vehicle's actions and the functionality of the road lanes. Evidences used in the inference procedure were provided by a simple off-the-shelf computer vision system that provided the classification of the road lanes 15 times per second. The results obtained from MLN inference were compared with the results of a simple Naïve Bayes classifier. From this comparison we conclude that both inferences have similar behaviour wrt accuracy, sensibility, precision and specification. The advantage of a relational model, in this case, is its representational power, which allows the representation of simple, common-sense, facts about our knowledge on the use of traffic lanes whereas these facts become automatically the classifier's structure.

References

Brehar, R., Fortuna, C., Bota, S., Mladenic, D. and Nedevschi, S. (2011). Spatio-temporal reasoning for traffic scene understanding, *IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP)*, pp. 377 –384.

Domingos, P. and Lowd, D. (2009). *Markov Logic: an interface layer for artificial intelligence*, Morgan & Claypool.

Hensel, I., Bachmann, A., Hummel, B. and Tran, Q. (2010). Understanding object relations in traffic scenes., *VISAPP*, pp. 389–395.

Lavee, G., Rivlin, E. and Rudzsky, M. (2009). Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **39**(5): 489 –504.

Nagel, H.-H. (2004). Steps toward a cognitive vision system, *AI magazine* **25**(2): 31–50.

Parker, A., Yaman, F., Nau, D. S. and Subrahmanian, V. S. (2007). Probabilistic go theories., *in* M. M. Veloso (ed.), *IJCAI*, pp. 501–506.

Sato, T. and Kameya, Y. (2008). New advances in logic-based probabilistic modeling by PRISM, *in* L. Raedt, P. Frasconi, K. Kersting and S. Muggleton (eds), *Probabilistic Inductive Logic Programming*, Vol. 4911, Springer Berlin Heidelberg, pp. 118–155.

Souza, C. and Santos, P. (2011). Probabilistic logic reasoning about traffic scenes, *Towards Autonomous Robotic Systems*, Vol. 6856 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 219–230.

Tran, S. D. and Davis, L. S. (2008). Event modeling and recognition using Markov Logic Networks, *Proc. of the ECCV*, Springer-Verlag, pp. 610–623.

Yu, B. and Jain, A. (1997). Lane boundary detection using a multiresolution Hough transform, *Proc. of Int. Conf. on Image Processing*, Vol. 2, pp. 748 –751.