Spatio-temporal Reasoning for Traffic Scene Understanding

Raluca Brehar∗, Carolina Fortuna†, Silviu Bota∗, Dunja Mladenic‡, Sergiu Nedevschi∗
∗Computer Science Department
Technical University of Cluj-Napoca
Email: Raluca.Brehar@cs.utcluj.ro, Silviu.Bota@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro
†Department of Communication Systems
Josef Stefan Institute
Email: Carolina.Fortuna@isj.si
‡Department of Knowledge Technologies
Josef Stefan Institute
Email: Dunja.Mladenic@isj.si

Abstract—In this paper we introduce a system for semantic understanding of traffic scenes. The system detects objects in video images captured in real vehicular traffic situations, classifies them, maps them to the OpenCyc ontology and finally generates descriptions of the traffic scene in CycL or cvasi-natural language. We employ meta-classification methods based on AdaBoost and Random forest algorithms for identifying interest objects like: cars, pedestrians, poles in traffic and we derive a set of annotations for each traffic scene. These annotations are mapped to OpenCyc concepts and predicates, spatio-temporal rules for object classification and scene understanding are then asserted in the knowledge base. Finally, we show that the system performs well in understanding traffic scene situations and summarizing them. The novelty of the approach resides in the combination of stereo-based object detection and recognition methods with logic based spatio-temporal reasoning.

I. INTRODUCTION

The field of traffic scene understanding is closely related to image and video understanding. It has numerous applications like active intelligent surveillance, summarization and indexing of video data, unobtrusive home care for the elderly, and hands-free human-computer interaction.

Our method tries to improve the understanding of human actions and interactions in a traffic environment. The novelty of the approach resides in the combination of stereo-based object detection and recognition methods with logic based spatio-temporal reasoning.

The objects appearing in a traffic scene are complex and usually traffic scenes are represented by highly cluttered environments. Hence the separation between the objects and the background is extremely difficult. In our approach the segmentation of the foreground is done using stereo-vision object detection techniques. Even if the segmentation provides good results, a more difficult task is the recognition of the objects. We apply classification algorithms for finding three major classes of objects in a traffic scene: pedestrians, cars and poles. All other objects that may appear are grouped in the class labeled ‘other objects’ (also referred as ‘unknown’).

Due to the high inter and intra class variance of pedestrians these are the most difficult to recognize. Nevertheless, cars have also complex particularities due to different illumination conditions, orientation angle with respect to camera and also due to their different aspects (sedan, coupe, etc). Poles may also have different shapes and dimensions, but we may consider they are the easiest to recognize. Once recognized, relationships between objects can be complex, therefore we use semantic technologies for modeling the objects and infer relationships between them.

The method we propose detects objects in video images captured in real vehicular traffic situations, classifies them, maps them to the OpenCyc ontology and finally generates descriptions of the traffic scene in CycL or cvasi-natural language. We employ meta-classification methods based on AdaBoost and Random forest algorithms for identifying interest objects like: cars, pedestrians, poles in traffic and we derive a set of annotation for each traffic scene. These annotations are mapped to OpenCyc concepts and predicates, spatio-temporal rules for object classification and scene understanding are then asserted in the knowledge base.

The paper is structured as follows. In Section II we summarize related work. Section III described the architecture of the system. Section IV presents the experimental results while Section V concludes the paper.

II. RELATED WORK

Semantic image understanding is an active field of research. Most state of the art methods deal with semantic image understanding and less have as research topic traffic scene understanding. There are several directions in the field of semantic image understanding that are somewhat related to traffic scene understanding. A first direction focuses on understanding video events [1] - those high-level semantic concepts that humans perceive when observing a video sequence. A comprehensive survey is done by [1] that offers a “Bird’s
Eye View of the Video Event Understanding Domain”. They consider the video event understanding process takes an image sequence as input and abstracts it into meaningful units. These units may refer to pixel based features, object based attributes or logic based features. The result of the abstraction is used by an event model to determine if an event of interest has occurred. The event model may comprise state models (Bayesian networks, Hidden Markov Models), pattern recognition methods (Support Vector Machines, Nearest Neighbors, Neural Network), semantic models (Petri Nets, Grammars, Logic). Output of a video event understanding process may be a decision on whether a particular event has occurred or a summary of events in the input sequence.

Most video event understanding methods are related to human activities and their interactions with the environment. For example, the authors in [2] propose a method for understanding continued and recursive human activities like fighting, greeting approach, depart, point, shake-hands, hug, punch, kick, and push. The understanding is done using a context free grammar representation scheme. A more elaborate method is employed by [3] that propose a framework for recognizing human actions and interactions in video by using three levels of abstraction: the poses of individual body parts recognized by Bayesian networks, the overall body pose, the actions of a single person modeled using a dynamic Bayesian network and time juxtaposition for identifying interactions.

Object hypothesis generation (object detection) is usually performed by grouping together some low level features. In [4] clusters of 6D vectors with common motion are used to generate object hypotheses. In [5] a grid based method for object localization is presented. Approaches to object recognition in two-dimensional images are based on invariant features like Scale-invariant feature transform (SIFT) [6], Speeded Up Robust Features (SURF)[7], Histogram of Oriented features [8], wavelets fed into different classification schemes like neural networks, ensemble methods or support vector machines. Other approaches comprise contour matching or bag of words representations of objects [9], [10].

The method we propose has three layers of abstraction: (1) object detection, (2) object recognition and (3) spatio-temporal reasoning. It is different from the body of related work in the approach and components, especially by using a powerful reasoning engine and comprehensive ontology for scene understanding and English language transliteration.

III. SYSTEM ARCHITECTURE

The general context in which the proposed method applies is the analysis of traffic scenes. A stereo based driving assistance system mounted on a car captures information about the traffic scene. Figure 1 shows the general context. A vehicle equipped with two cameras captures information about the outside environment. Using complex processing methods for 3D stereo reconstruction it generates hypotheses of objects. The object hypotheses are further processed using methods for:

- extracting objects’ parameters like: dimension, speed, depth and position in scene
- object recognition that generates class hypotheses. These can be: car, pedestrian, pole ore unknown.

Based on these methods annotations are generated for each object in the scene. Next the annotations are fed into a spatio-temporal reasoning module that improves the object detection results based on semantic reasoning and generates a better model of the traffic scene.

A. OBJECT DETECTION AND OBJECT ATTRIBUTES GENERATION

Our system uses stereo-vision in order to construct object hypotheses. Two cameras mounted on a vehicle in a stereo configuration are used for acquiring images. Figure 2 shows the position of the cameras, the orientation of the coordinate system and the cuboidal object model.

After acquiring the images, as a preprocessing stage we apply undistortion, scaling and rectification in order to obtain a canonical rectified image pair. Then, dense stereo reconstruction is performed using the TYZX [11] hardware. Optical flow is then computed using the pyramidal Lucas-Kanade corner based optical flow algorithm described in [12].

The object detection stage generates cuboidal objects by grouping together individual 3D points. First, the ground’s profile is detected using an elevation map [13]. Then, the points are classified according to their height above the ground.
Fig. 2. Coordinate system and object model

Points located above the ground are then projected on the xOz plane and grouped together. A oriented cuboid is the generated around the grouped points. A complete description of this algorithm can be found in [14]. An alternative algorithm based on density maps [15] is also used for detecting small objects (such as pedestrians). Speeds are then computed, for detected objects, by averaging the optical flow vectors contained in each object.

After detection, objects are tracked using Kalman filters. Tracking improves detection by integrating information across multiple frames. It also allows to maintain the identity of the object and it improves the speed computed using optical flow.

Objects are modeled as cuboids, right prisms with rectangular bases. The bases of the prism are always parallel to the xOz plane. Therefore, each object can be fully described by following parameters:

- The (x, z) coordinates of its base corners (the near left, near right, far left and far right corners). We call these parameters NearLeftX_m, NearLeftZ_m, NearRightX_m, NearRightZ_m, FarLeftX_m, FarLeftZ_m, FarRightX_m, FarRightZ_m. They are expressed in meters.
- The y coordinates for the top and bottom bases.
- Speed vector components, along the x and z axes, expressed in km/h, SpeedX_km and SpeedZ_km. As objects move only on the ground surface there is no need to consider speed along the y axis.

From these parameters we also derive the following values:

- The object’s mass center, given by the CenterX_m, CenterY_m and CenterZ_m coordinates.
- The object’s height, Height_m obtained by the difference of the top and bottom y coordinates.
- Object’s width and length, Width_m and Length_m.
- Object’s z distance from the ego vehicle, Depth_meters.

B. Object Recognition

An important feature that can be added to semantic annotations is the class to which the objects belong. In order to recognize the objects that appear in a traffic scene we have employed several machine learning methods for identifying relevant classes of objects in traffic:

- pedestrian
- pole
- car
- unknown - class that contains objects other than pedestrians, poles, cars.

Our methodology comprises the following:
1) Construct a database of models for each class
2) Extract relevant features for the models in each class
3) Train level 1 classifiers. These learners are trained on histogram of oriented gradients features for the Pole and Pedestrian class. Their recognition score will be used as input feature (among other features) for a meta-classifier.
4) Train meta-classifier on all features and on the responses of level 1 classifiers.

1) The database of models: The database of models was obtained by manually labeling several traffic video sequences. When choosing the sequences we have tried to have a large variety in illumination conditions, background, weather conditions and intersection configurations. Each object in the three classes was perfectly framed. Directed by the detection results, we have introduced in the class - 'Unknown' - 2D bounding boxes in which two objects were grouped together, or bounding boxes that contain parts of an object. Samples from the database are shown in Figure 3:

Fig. 3. Models used for object recognition

2) Relevant feature extraction: For each sample in the database we have computed the following features:
   a) Object dimension - width and height - are computed based on the cuboids generated by the segmentation of 3D points. The cuboids capture the position and the spatial extent of each obstacle.
   b) Motion features are represented by: (i)Speed (on Ox and Oz axis): obtained from optical flow filtered by tracking and (ii) Motion signature – the horizontal variance of the 2D motion field.
   c) The HoG Pedestrian score and the HoG Pole Score are given by previously trained boosted classifiers on Histogram of Oriented Gradient features for the class pole and pedestrian.
   d) The pattern matching score is given by the degree of match between the contour of an obstacle and templates from a previously generated hierarchy of pedestrian contours.
   e) The vertical texture dissimilarity measures the maximum vertical dissimilarity that can be found in the object’s area.

3) Train level 1 classifiers: For the images in the class pedestrian and pole we have extracted Histogram of Oriented Gradient (HoG) features [8] and on each class we have trained a cascaded boosted classifier.
The HoG features are obtained by dividing the 2D image corresponding to the 3D cuboid surrounding the object, into non-overlapping cells of equal dimension. For each the gradient magnitude and orientation are computed. Based on them, a weighted histogram of orientations is build within each cell. Next, cells are grouped in overlapping blocks and the values of the histograms contained by a block are normalized. Two different cascades of boosted classifiers are trained using the AdaBoost algorithm, one for poles and one for pedestrians, Figure 4.

4) Meta-classification for object recognition in traffic scenes: A generic meta-classifier combines the output from the level 1 classifiers with the other features (motion, pattern matching, object dimension, texture) as shown in Figure 5.

We have experimented with two types of classification schemes: adaboost and random forest. Section “Experimental Results” describes the performance of the recognition task.

C. Ontological Modeling of the World

For the ontological modeling of the world, by world referring to the acquired images, objects and information about the objects using the previously described modules, we used OpenCyc. The OpenCyc knowledge base is composed of ontology (i.e. an information model and relations between its elements), instances of concepts, relations and rules. The world model is mapped and modeled via OpenCyc specific abstractions. For each scene (image), the Object recognition module provides a list of objects described by a set of attributes. These attributes are mapped to existing OpenCyc concepts or expressed using corresponding predicates. For attributes with no equivalence in the OpenCyc ontology, new corresponding concepts and predicates have been defined and linked. The basic mapping of the descriptors provided by the Object recognition model is presented in Table I. The detected object type is mapped to one of four concepts (Automobile, UtilityPole, Person or Obstacle for the unclassified objects).

The four points determining the projection on the XoZ plane of the object are defined as points in a list of vertices. For some unclassified objects, the four points determining the projections are not available, so for identifying the objects’ position in space, the 3D coordinates of the center are used. As a result, we use the positionAlongAxisInCoordinateSystem predicates on the object itself as opposed to on the point delimiting the footprint in the XoZ plane. The dimensions of objects is specified using the height/length/widthOfObject predicates. For inserting knowledge about 2D speed of the objects, two predicates were created: speedAlongXAxisInCoordinateSystem and speedAlongZAxisInCoordinateSystem. These are similar to the positioning predicates just that they denote speed.

A set of objects depicted by an image represents from ontological point of view, a PhysicalSeries - a series of spatially ordered objects. Each image depicts a corresponding PhysicalSeries and objects which belong to this particular case. For instance, the knowledge base description of a video clip containing a sequence of images is presented in...
Figure 6 and a description of an image is presented in Figure 7.

Individual : VideoClip0
isa : Series
seriesLength : 15
seriesMembers :
ImageA000288 ImageA000287 ImageA000286
ImageA000285 ImageA000284 ImageA000283
ImageA000282 ImageA000281 ImageA000280
ImageA000279 ImageA000278 ImageA000277
ImageA000276 ImageA000275 ImageA000274
seriesOrderedBy : after

Fig. 6. Representation of a time ordered sequence of images

Individual : ImageA000282
isa : Image TemporalThing
imageDepicts :
Unclassified10A0000282 Unclassified9A0000282
Unclassified8A0000282 Unclassified7A0000282
Unclassified6A0000282 Unclassified5A0000282
Pole4A0000282 Unclassified3A0000282
Pedestrian2A0000282 Unclassified1A0000282
Unclassified0A0000282 PhySeriesA0000282
seriesLength : 15
seriesMembers VideoClip0 VideoClip1 VideoClip2 VideoClip3
seriesOrderedBy : after

Fig. 7. Representation of an image in a sequence

Starting from the descriptors listed in the first column of Table I, additional information can be extracted and inserted in the knowledge base. Based on the speed on X and Z axis, a movement direction can be determined as well as a speed in that direction. For instance, an object with SpeedZ > 0 and SpeedX = 0, moves in direction North with speed equals to SpeedZ. The appropriate OpenCyc constants for representing this ale listed in the first two rows of Table II.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction of motion</td>
<td>movesInDirection-Average, North- Generally, Northeast-Generally West-Generally, Northwest-Generally</td>
</tr>
<tr>
<td>Speed of object</td>
<td>speedOfObject-Translation, velocityOfObject</td>
</tr>
<tr>
<td>Distance between objects</td>
<td>distanceBetween</td>
</tr>
<tr>
<td>Relative position between objects</td>
<td>near, farFrom, northOf, northeastOf, eastOf, southeastOf, spatiallyIntersects, overlappingSpaceRegions</td>
</tr>
<tr>
<td>Action execution</td>
<td>performedBy, doneBy</td>
</tr>
<tr>
<td>Events</td>
<td>Walking, Jogging, Running</td>
</tr>
</tbody>
</table>

TABLE II
OTHER KNOWLEDGE

Furthermore, the distance between two objects can be determined based on the distance between their footprints (note that objects are delimited by a rectangular cuboid with a rectangular footprint). Based on the distance between objects and the width of the scene, subjective estimations are relative distances between objects can be made using predicates such as farFrom and near. Also, relative positioning predicates such as northOf and spatiallyIntersects can be inserted.

Once the relevant concepts have been modeled and introduced in the knowledge base, generic or application specific rules can be defined to achieve the desired functionality. Our scope in this paper is to define rules to improve the classification and to achieve better scene understanding.

1) Scene Understanding: Scene understanding can be achieved just by inserting appropriate instances and predicates in the knowledge base. With the additional knowledge already in the knowledge base (OpenCyc knowledge base constrains over 47,000 concepts and 306,000 facts) rich inferences can be performed and more abstract queries can be answered to. Additional rules can be added to the knowledge base to allow more application specific reasoning if this is not already covered by the existing (common sense) knowledge. For instance, rules for inferring the directionBetweenObjects based on relative positioning, rules for classification and rules for action recognition.

2) Action Recognition: Action recognition in the context of urban traffic scenes typically deal with detecting people cars, motorbikes, trams, etc. and recognizing actions they perform. For instance a person which moves on two legs is performing one of the following actions: walking, jogging, running. When the person is on the bike, then the action is riding a bike. Cars are driven be persons who can drive straight, take a turn, drive with high speed or according to the rules, etc. Depending on their positioning in the scene and their movements, actors may approach each other. Such events as approaching should be detected for safety reasons especially when a person is approaching a car or vice versa. We introduce rules for spatial reasoning over relative positions and motion (speed, direction) of actors to detect approaching events.

D. English transliteration

The semantic representation of each traffic scene is provided in the CycL language and stored in the OpenCyc knowledge base using specific representation. For some type of relations, including the ‘isa’ predicate, transliterations can be obtained through the API. For more complex queries, answers in CycL need to be parsed by the custom transliteration modules and output in a specific manner. The complete transliteration of a traffic scene can be seen as a textual summary of the scene. In other words, our system is able to take images, detect objects, classify objects, infer relationships between objects, understand the complex situation in a scene and output this in a quasi-natural language manner using the transliteration.

IV. EXPERIMENTAL RESULTS

A. Object Detection and Recognition

The experimental results of object detection are better understood if presented via the correlation with object recognition.

In performing the recognition task our database of models contained: 27000 cars, 26000 pedestrians, 34000 pole models, 100000 samples in the unknown class. For all these models
we have extracted the features mentioned in section III-B2 and we have applied the meta-classification scheme.

For meta-classification we have experimented with two types of multi-class learners: random forest and adaptive boosting. The random forest represents a meta-learner composed of many individual decision trees. It is designed to operate quickly over large datasets and to be diverse by using random samples to build each tree in the forest. It constructs random forests by bagging ensembles of random trees. We have used 20 random trees in the implementation. The adaptive boosted classifier is also a meta-learner and in our implementation it is formed by the linear combination of J48 weak learners.

The validation on such a classification system is very difficult. A manual validation is extremely time consuming and it requires plenty of effort from the user. We have validated the system on the database of build models.

The results of the meta-classifier trained using random forest and AdaBoost are presented in what follows. The evaluation is done using stratified cross-validation on the database of models.

<table>
<thead>
<tr>
<th>Classifier: Random Forest evaluated using stratified cross-validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly Classified Instances</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
</tr>
<tr>
<td>Kappa statistic</td>
</tr>
<tr>
<td>Mean absolute error</td>
</tr>
<tr>
<td>Root mean squared error</td>
</tr>
<tr>
<td>Relative absolute error</td>
</tr>
<tr>
<td>Root relative squared error</td>
</tr>
<tr>
<td>Total Number of Instances</td>
</tr>
</tbody>
</table>

**TABLE III**  
**RANDOM FOREST META-CLASSIFIER EVALUATION**

<table>
<thead>
<tr>
<th>Classifier: AdaBoost with J48 decision stumps evaluated using stratified cross-validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly Classified Instances</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
</tr>
<tr>
<td>Kappa statistic</td>
</tr>
<tr>
<td>Mean absolute error</td>
</tr>
<tr>
<td>Root mean squared error</td>
</tr>
<tr>
<td>Relative absolute error</td>
</tr>
<tr>
<td>Root relative squared error</td>
</tr>
<tr>
<td>Total Number of Instances</td>
</tr>
</tbody>
</table>

**TABLE IV**  
**ADABoost META-CLASSIFIER EVALUATION**

These results are promising but not perfect. The spatio-temporal reasoning that we have presented shows how these results can be improved.

**B. Ontological Modeling of the World**

**1) Scene Understanding:** To improve the classification provided by the Object recognition module we defined several rules, one of them is presented in Figure 8. These rules use spatio-temporal reasoning for improving object classification. For instance, the rule in Figure 8 asserts that, if there are two consequent images in a video, in both images an object was detected in a location, in one of the images, the object is classified by the Object recognition module as unknown (Obstacle in the knowledge base) and in the other image, it is classified as a Pedestrian (Person in the knowledge base), and the two objects have similar dimensions and relative positions to the origin of the coordinate system, then probably the unknown object is a pedestrian. In the case of image A282, the classification algorithms in the Object recognition module detected 11 objects and classified two (a utility pole and a pedestrian) as can be seen at the top of Figure 9. The pole is surrounded by a green rectangle and the person by a yellow one. Some of the unclassified objects we as humans can distinguish: object 3 is a car, object 8 is a pole. Furthermore, we can tell that there is another car in the image which was not detected, there is a publicity panel supported by legs which look like poles, there is a fence with metal bars, the publicity panel is behind the fence. Telling which exactly are the support pillar of the publicity panel is hard even for us. It can be seen, that the Object recognition algorithm performs poorly here (and it’s rather inconsistent with respect to the detected objects across several similar images). The spatio-temporal reasoning improves the classification of unknown objects by recognizing the car, the utility pole on the right side of the image (check Table V). However, it misclassifies many of the unknown objects (i.e. metal fence, publicity panel) as poles. In this particular case, the misclassification error is not expensive since all these objects are static, and
the shapes of a pole or a support pillar of the fence are similar. However, the misclassification of a person as a pole would be more expensive. These rules we designed for spatio-
temporal reasoning perform well on the small number of images we manually verified (25 images). However we believe that they can be further optimized especially by involving more complex spatial reasoning including surrounding objects, their classification and their relative position. Complex rules for classification based on spatio-temporal reasoning applied to images where the object detection algorithm produces noisier results may though underperform. The listing below the image in Figure 9 represents part of the description of the scene as modelled in the knowledge base. The complete description of a scene is rather lengthy as the description of one object consists of over 100 assertions. Using appropriate queries, a list of things depicted by the image can be obtained, the conceptualization of things (i.e. uncategorized3a000282 is a car.), inferred conceptualization based on the knowledge already existing the the knowledge base (i.e. uncategorized3a000282 is a mechanicaldevice, uncategorized3a000282 is a powereddevice.) as well as relative positioning of objects in space (i.e. pedestrian2a000282 farFrom uncategorized9a000282).

2) Action Recognition: We introduce rules for spatial reasoning over relative positions and motion (speed, direction) of actors to detect approaching events. Figure 10 shows how the action recognition task is modeled in Cyc.

C. English transliteration

The English language transliteration is generated based on the knowledge present in the knowledge base. This consists of transliterating the reasoning (i.e. the proof) for a query. For instance, when a query is fired asking for all SpatialThings in

### Table V

<table>
<thead>
<tr>
<th>Image name</th>
<th>Classification algorithm [TP/FP/detected]</th>
<th>Spatio-temporal reasoning [TP/FP/detected]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A274</td>
<td>0/0/6</td>
<td>0/0/6</td>
</tr>
<tr>
<td>A275</td>
<td>1/0/6</td>
<td>0/0/6</td>
</tr>
<tr>
<td>A276</td>
<td>0/0/6</td>
<td>3/2/6</td>
</tr>
<tr>
<td>A277</td>
<td>1/0/9</td>
<td>4/4/9</td>
</tr>
<tr>
<td>A278</td>
<td>3/0/12</td>
<td>4/5/12</td>
</tr>
<tr>
<td>A279</td>
<td>2/0/1</td>
<td>5/2/11</td>
</tr>
<tr>
<td>A280</td>
<td>2/0/8</td>
<td>4/2/8</td>
</tr>
<tr>
<td>A281</td>
<td>2/0/9</td>
<td>4/2/9</td>
</tr>
<tr>
<td>A282</td>
<td>2/0/11</td>
<td>4/11</td>
</tr>
<tr>
<td>A283</td>
<td>4/0/11</td>
<td>5/3/11</td>
</tr>
<tr>
<td>A284</td>
<td>2/0/9</td>
<td>5/2/9</td>
</tr>
<tr>
<td>A285</td>
<td>2/0/8</td>
<td>3/1/8</td>
</tr>
<tr>
<td>A286</td>
<td>3/0/7</td>
<td>4/1/7</td>
</tr>
<tr>
<td>A287</td>
<td>3/0/8</td>
<td>3/0/8</td>
</tr>
<tr>
<td>A288</td>
<td>3/0/8</td>
<td>3/0/8</td>
</tr>
</tbody>
</table>

Table V Improving classification by means of spatio-temporal reasoning

Fig. 9. Scene with its description

Motion modeling by translation and direction:

- pedestrian2a000282 moves in direction northwest–generally, speed of object kilometersperhour 17.205.
- uncategorized3a000282 moves in direction northwest–generally, speed of object kilometersperhour 17.205.

Action recognition:

- actionpedestrian2a000282 performed by pedestrian2a000282, actionpedestrian2a000282 isa running.
- actionpedestrian2a000282 performed by pedestrian2a000282, actionpedestrian2a000282 isa ambulation.
- actionpedestrian2a000282 performed by pedestrian2a000282, actionpedestrian2a000282 isa movingalllegs.
- movement02a000282 done by pedestrian2a000282, movement02a000282 isa approaching.
- movement02a000282 done by pedestrian2a000282, movement02a000282 isa translation-locationchange.
- movement02a000282 done by pedestrian2a000282, movement02a000282 isa movement-translationevent.

Fig. 10. Description of a scene
an image, the reasoning will go up the hierarchy via the isa predicate and will determine whether the object is an instance of the abstract SpatialThing concept. In Figure 11 can be seen that the object instance denoted by the name unclassified0a000282 is a utility pole (as it was classified by the spatio-temporal reasoning), and in the OpenCyc ontology it is represented that utility pole is a specialization of spatial thing, therefore the transliterated answer to the above question can be read in the figure. Such transliterations, or transliterations of the scene can be used to generate a textual summary of the image.

![Image](ImageA000282.png)  
**Image:** ImageA000282 depicts SpatialThing?  
unclassified0a000282 is a utility pole,  
every utility pole is an open-air,  
every open-air is a localized spatial thing,  
every localized spatial thing is a spatial thing.  
unclassified0a000282 is an obstacle,  
every obstacle is a tangible thing,  
every tangible thing is a three dimensional thing,  
al three dimensional thing is a thing with two or more dimensions,  
every thing with two or more dimensions is a spatial thing with one or more dimensions,  
every spatial thing with one or more dimensions is a spatial thing.

![Image](ImageA000282.png)  
**Image:** ImageA000282 depicts ObjectWithUse?  
unclassified0a000282 is a utility pole,  
every utility pole is a post,  
every post is a shaft,  
every shaft is a rod,  
every rod is an implement,  
every implement is a device,  
every device is an object with uses.  
unclassified3a000282 is a car,  
every car is a device that is not a weapon,  
every device that is not a weapon is a device,  
every device is an object with uses.

![Image](ImageA000282.png)  
**Image:** ImageA000282 depicts SpatialThing?

![Image](ImageA000282.png)  
**Image:** ImageA000282 depicts ObjectWithUse?

**Fig. 11.** Transliteration of semantic representations

## V. CONCLUSIONS

This paper has presented a novel method of spatio-temporal reasoning for understanding traffic scenes. Our method combines stereo-vision object detection approaches with meta-classification object recognition algorithms in order to generate a first level semantic annotation of the traffic scene. This semantic annotation is further mapped into OpenCyc concepts and predicates, spatio-temporal rules for object classification and scene understanding are then asserted in the knowledge base. The method performs well in understanding traffic scene situations and summarizing them.

## VI. ACKNOWLEDGMENTS

This work was partially funded by the PlanetData, Intersafe2 and the Romanian-Slovenian bilateral cooperation project.

### REFERENCES


