Model-Based Objects Recognition in Man-Made Environments

Joan Martí

Computer Vision and Robotics Group
University of Girona
17071 Girona - Catalonia - Spain
e-mail: joanm@ei.udg.es

Alícia Casals

Dept. of Automatic Control and Computer Eng.
Polytechnical University of Catalonia
08028 Barcelona - Catalonia - Spain
e-mail: casals@esaii.upc.es

Abstract

In this paper we describe a model-based objects recognition system which is part of an image interpretation system intended to assist autonomous vehicles navigation. The system is intended to operate in man-made environments. Behavior-based navigation of autonomous vehicles involves the recognition of navigable areas and the potential obstacles.

The recognition system integrates color, shape and texture information together with the location of the vanishing point. The recognition process starts from some prior scene knowledge, that is, a generic model of the expected scene and the potencial objects. The recognition system constitutes an approach where different low-level vision techniques extract a multitude of image descriptors which are then analyzed using a rule-based reasoning system to interpret the image content. This system has been implemented using CEES, the C++ Embedded Expert System Shell developed in the Systems Engineering and Automatic Control Laboratory (University of Girona) as a specific rule-based problem solving tool. It has been especially conceived for supporting cooperative Expert Systems, and uses the object oriented programming paradigm.

1 Introduction

Automatic interpretation of complex scenes is not possible just using general image processing routines without introducing semantics and other background knowledge, which is usually the province of human experts. With this thought Artificial Intelligence (AI) and Expert Systems (ES) joined Computer Vision to deal with the problem of Image Understanding (IU) which means the transformation of two-dimensional spatial (and, if appropriate to the problem domain, time-varying) data into a description of the three-dimensional spatiotemporal world. This procedure involves the design and experimentation of computer systems that integrate explicit models of a visual problem domain with one or more algorithms for feature extraction from images and one or more methods for matching features with models [1, 2]. A general purpose vision system must contain a very large number of models that represent prototypical objects, events, and scenes, but it is computationally prohibitive to match image features with all of them. Therefore, Image Understanding Systems (IUS) usually focus on some predefined kind of scenes to keep the system feasible, such as medical images, aerial pictures and so.

To this effect, we propose a recognition system restricted to operate on man-made environments which spans a range of complexity from outdoor scenes such as urban streets and highways to highly structured indoor scenes like corridors or hallways. Given an image, the overall goal is to analyze it with the purpose of recognizing some requested object types that can be used to assist the navigation of a behavior-based navigation autonomous vehicle [3].

Attending the geometrical properties and the spatial context for man-made environments [4, 5], the following general assumptions can be taken on:

- The 3D structure of the viewed scene can be described, in a first approximation, as a simple orthohedral world. This preliminary 3D description assumes that the scene is composed by simple blocks with sides converging to the vanishing points.

- There are constraints for the spatial relations between objects. They are expressed as rules and can be used to infer a request to search an object (e.g. once we have found the street, we can try to find cars in it) or to validate an hypothesis object using its context (we try to find car wheels after locating the car on the street).

On the other hand, integration of low-level vision techniques [1, 6] is a key step for the useful extraction of features from individual objects and for creating object records that are later processed with actions specified by the rule-based system [7, 8]. We have chosen color processing, texture analysis, vanishing point location and shape information as vision techniques for image segmentation. These techniques can operate over an image or on a part of it based on the interpretation requirements. The output of this segmentation module is a set of image regions called *entities* by the ES for interpretation purposes. The interpretation modules are primarily responsible for labeling the entities as recognizable objects. They accept entity parameters and, based on models, attempt to label image regions [1, 2, 9]. Fig. 1 shows a block diagram for the structure of our proposed model-based recognition system.

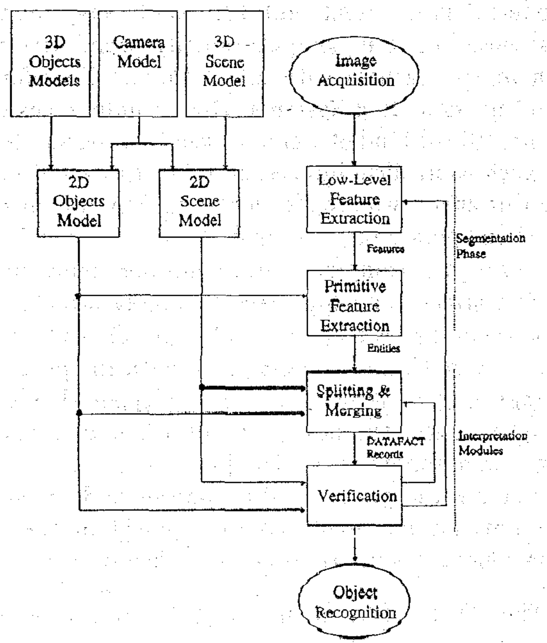

Figure 1: System structure

In the following sections we briefly summarize the segmentation and interpretation modules, as well as the knowledge representation method chosen in the system. Implementation using the ES is also described. Experimental results on object recognition in urban scenes are finally presented to illustrate the performance of this model-based approach.

2 Image segmentation

In order to interpret a 2D acquired image, the image is first partitioned into regions, where each region is uniform and homogeneous with respect to some segmentation criteria. The segmented regions will form the initial set of image entities used by the ES in the interpretation modules. Currently, four low-level vision techniques can be invoked with actions specified by the rule-based system:

- a vanishing point location algorithm
- a color-based segmentation processor
- a texture-based segmentation processor
- an edge/contour-based segmentation module

Such segmentation processes are evolving procedures which usually start from the original acquired image and gradually group small regions into more meaningful ones. During such evolution, some grouping or decision making may go wrong due to a variety of reasons. Therefore it should be possible to return to a more primitive status and make a new decision according to the knowledge bases. Bearing these requirements in mind, the whole segmentation process can be viewed as a "N-node tree", as shown in Fig. 2.

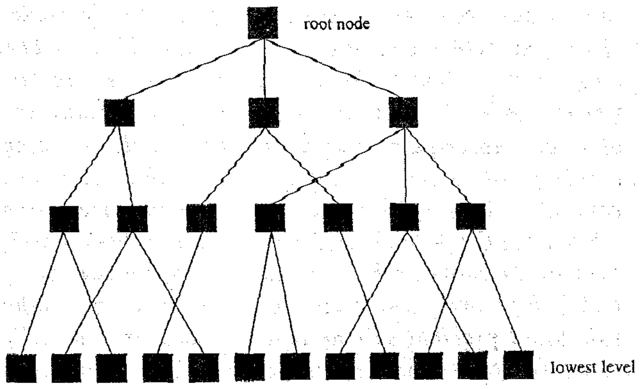

Figure 2: N-node tree representation

The "N-node tree" can be seen as an extension of a quad tree, and with such extension the number of children under a node is changeable. The segmentation tree consists of a number of levels, with each level representing segmentation results at different stages. Each node in one level describes a complete region which has no overlapping with other regions in the same level. For a node on one level, one can find out its associated original image pixels by tracing down the tree through its children until the lowest level is reached, where each node represents the image pixel indexed from left to right and from top to bottom on the image. Each node of the segmentation tree has a feature table associated, which represents some two-dimensional features of the region, and is used to establish the correspondences amongst the various nodes of the tree. Feature tables should consist of many independent features which are not affected by the image scale. The set of entity features used in the trials include:

- geometrical data [10], consisting of the centroid and its maximum and minimum coordinates, and area.
- color data [11], consisting of hue, saturation and luminance measures.
- profile data [12], consisting of a full description of the profile of the perimeter; this could be used directly as evidence, and also indirectly in the calculation of shape and partial-shape properties.
- textural data [13], consisting of an evaluation for the kind of texturing based on the following textural parameters: blurriness, granularity, discontinuity, abruptness, straightness and curviness.
- a list of adjacent regions and their orientation.

In order to avoid the effect of image scale, the area and perimeter are normalized according to the parent region, while position features are described as positions relative to rectangles circumscribed around the parent regions and normalized to 1 x 1.

3 Knowledge representation

Once we have obtained the set of features provided by the segmentation module that best initially characterizes the entities, we must integrate evidence from diverse sources of knowledge to arrive at an object recognition. This is what is generally called knowledge-based methods [7, 8], in which the recognition process is controlled according to the structure of the available background or world knowledge. This is used to make hypotheses about the image and embodies certain semantic constraints that are to be satisfied amongst entities in the image. Integration of knowledge takes into account:

- Knowledge of the image segmentation process, which includes the vision techniques used to extract the image primitives, and the appearance of these primitives in the image, including relevant object models.
- Knowledge of spatial relationships (such as "above", "between", "left of") and constraints between the scene domain primitives.
- Knowledge of models composition (such as "part of") that considers the aggregation of concepts into more abstract ones or the decomposition of concepts into more primitives ones.
- Knowledge of methods for combining model compositions into complete scene interpretations.

The knowledge representation structure we use is adapted from the rule-based reasoning system CEES [14], the C++ Embedded Expert System Shell developed in the Systems Engineering and Automatic Control Laboratory at the University of Girona.

3.1 Description of CEES

Essentially, CEES is a specific rule-based problem solving tool that has been specially conceived for supporting cooperative Expert Systems, and uses object oriented programming techniques[1] as the solution to obtain completely independent agents, ESs.

CEES implements on objects the different ESs and interaction (communication) between them will be supported via C++ methods. This implementation is designed to support communication amongst different knowledge bases, inference engines and simulators. Therefore, communication is based on methods (messages) amongst objects.

CEES has defined all its information structured in objects. Whatever knowledge or facts of information (**DATAFACT**), numerical variables (**NUMERIC**), actions (**ACTION**), inference engines (**INFERENCE_ENGINE**) and models (**MODEL**) are the basic objects that are available in CEES. This hierarchy of objects is represented in Fig. 3 and the framework for cooperation is now quite easy: just create as many objects INFERENCE_ENGINE as desired to have as many cooperative ESs.

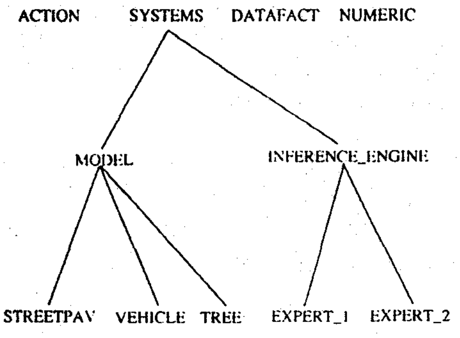

Figure 3: CEES classes hierarchy

[1] Although we tried to choose another expression, we failed finding an equivalence to the word *object*. So, it must be distinguished "object" as a target recognition for the Computer Vision System from "object" as programming techniques during ES implementation.

These cooperative ESs can exchange information by means of DATAFACT objects. They can also ask for information to MODEL objects that will provide NUMERIC objects that contain structured numerical variables. DATAFACT objects contain the entity descriptions provided by the segmentation module while MODEL objects contain the stored models expressed as rules. Inference between visual and model data are performed by INFERENCE_ENGINE objects.

3.2 Modeling

Man-made environments are usually composed by a wide variety of objects. Although there are not any formulas underlying their spatial arrangement, there must exist a set of rules, no matter how many, but a finite number of them. For the proposed system, rules concerning the models of objects that belong to urban environments (road network, buildings, trees, grass lands, cars, street signs, traffic lights, streetlamps, etc.) have been defined taking into account the following considerations:

1. The scene has only 3 vanishing points corresponding to the 3D orthogonal directions, as a consequence of its 3D structure approximation (the streets are straight and with constant width, many buildings are approximately solid blocks with parallel or orthogonal edges —being these edges orthogonal or parallel to the street ones—).

2. The objects are roughly described as sticks, plates and blobs. The stick has two endpoints, a set of interior points, and a center of mass that can be specified as connection points. The plate has a set of edge points, a set of surface points, and a center of mass. The blob has a set of surface points and a center of mass. In general, sticks will project to long, thin regions of the image; plates will project to compact regions; and blobs will project to one or more connected regions.

3. "A coarse-to-fine" description of the objects allows to incorporate additional attributes than those related to their shape. In this way, other color and texture information are assigned in a fuzzy way to the objects, as well as the scale factor between objects size. This complementary information can become essential in some interpretation tasks.

4. Context information is added to deduce relationships between objects in the scene (e.g. a car always lies on the ground plane).

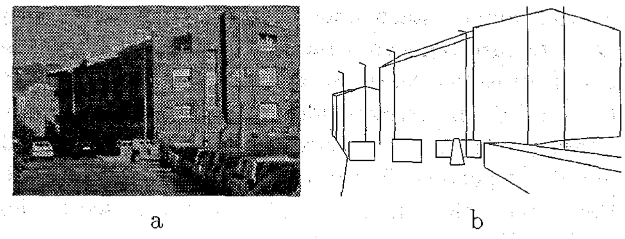

Figure 4: (a) A typical urban scene, (b) with its modeled regions and objects

As a modeling example, Fig. 4a shows a typical urban environment where the buildings are modeled as solid blocks while the remaining regions and objects follow the stick, plate and blob models (e.g. the streetlamps, the pavement and the cars, respectively), as shown in Fig. 4b.

3.3 Interpretation modules

The task of the rule-based system is to generate a plausible recognition methodology for the objects in the image by building successively more specific interpretations based on the modeling of the generic objects to be recognized. Following the CEES structure, the interpretation is performed upon DATAFACT objects recorded in the ES as:

```
DATAFACT
   Name
   Type
   FeaturesTable
   DatafactResult
```

where each DATAFACT identified by the label Name is a dynamically changing record according to the rules expressed in the different INFERENCE_ENGINE objects. The item Type belongs to one of the three categories defined in the modeling section (sticks, plates and blobs), while FeaturesTable is the updated table of features that corresponds to the geometrical, color, profile, textural and adjacency data contained in the DATAFACT.

Initially, every entity provided by the segmentation module generates a single DATAFACT object in the knowledge base of EXPERT_N that can evolve during the interpretation process. The interpretation is performed by two independent modules, and results are stored in DatafactResult:

- a pre-processing module for merging and splitting DATAFACT objects, and
- a verification module that tries to label these DATAFACT objects.

The rule-based pre-processing module can interact with the segmentation process refining some entity features. It can modify **DATAFACT** objects merging or splitting their data so the rules of this module may call any of the low-level vision techniques to compute the necessary additional feature values. The verification module tries to perform the object recognition using a matching between visual data (expressed as **INFERENCE_ENGINE** objects) and models (stored as **MODEL** objects) by means of production rules. The rules have the general structure *Situation — Action*, as is shown in Fig. 5, where a rule concerning the visual appearance of the data appears.

```
Rule 100
  Description "Decide whether an image
         region is the street pavement"
  Certainty 0.75
  Threshold 0.40
  TraceHere Yes
  If (*Region1).FeaturesTable.Geometric.equal
     ((*StreetPav).FeaturesTable.Geometric)
    And /* A CEES fuzzy condition */
     (*Region1).FeaturesTable.Color.equal
     ((*StreetPav).FeaturesTable.Color)
    And
     (*Region1).FeaturesTable.Adjacency.equal
     ((*StreetPav).FeaturesTable.Adjacency)
    And /* A C++ non fuzzy condition */
     (*Region1).Type == plate
    Then
      deduce(INTERMEDIATE, STREETPAVFOUND)
  EndIf
EndRule
```

Figure 5: Example for a rule in the verification module

This rule attemps to match the visual data contained in **DATAFACT** object record **Region1** with the model **StreetPav** using geometrical, color and adjacency features obtained in the segmentation process. Results (as new intermediate object records or matching score) are stored in **DatafactResult** that is later evaluated. There are also rules concerning the spatial relationships between entities and rules that describe the model composition for objects and scenes. *Situation* is a logical AND of predicates, declared in sublasses of **INFERENCE_ENGINE**. Predicates are logical evaluations of feature comparisons stored as **NUMERIC** in the **FeaturesTable** structure in **DATAFACT** and represent the conditions of rules (each condition describes a possible situation of visual data).

An *Action* occurs when all conditions of *Situation* are globally satisfied with a confidence level expressed by **Threshold**. The system attempts to verify the rules using the features extracted from the image. If this attempt succeeds, the actions are executed with a **Certainty** score. Typical actions are generation of new **DATAFACT** objects as intermediate results that are later processed as new records and matchings between visual and model data.

If alternative matches are obtained, further rules can select the proper ones. When a match fails, the recognizing is refused.

4 Experimental results

To evaluate our approach we have used some simple urban scenes, as is shown in Fig. 6a which represents an original gray value image. Fig. 6b shows the entity image labeled as recognized street pavement while Fig. 6c shows the entities recognized as vehicles.

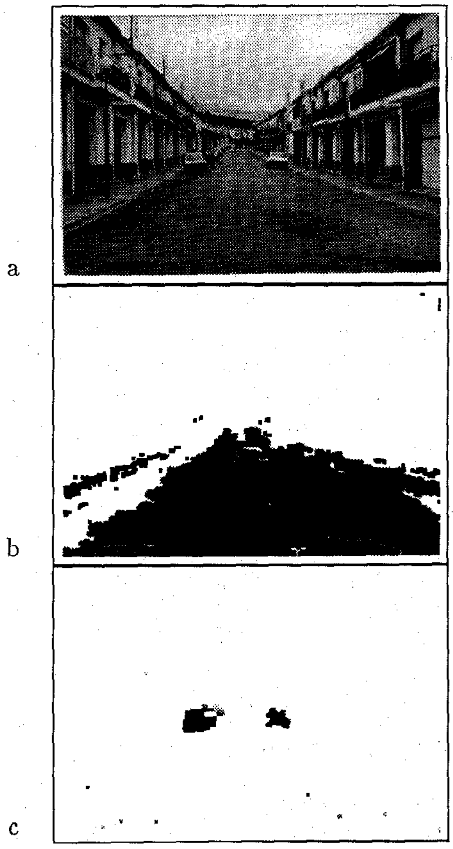

Figure 6: (a) An image to be analyzed (b) Recognized street (c) Recognized vehicles

Images of distant objects are typically small and of low contrast. Only few features may be extracted, so object recognizing in such a kind of images makes use of the recovery of 3D structure of the viewed scene by segmenting the image into four relevant regions: the road, the sky and the left and right building areas. The detection of the road requires the extraction of its sides, which is greatly simplified by the fact that the distance between the sides is usually constant. As a consequence it is possible to assume that road sides form parallel lines. Therefore, an essential step in the extraction of the road sides is the detection of vanishing points. Modeling images taken under (almost) central perspective will give high scoring to the spatial relationships between entities and the detected vanishing points on the Expert System.

5 Conclusions and further work

We have presented a recognizing system intended to operate on man-made environments. Its main capabilities are the integration of multiple segmentation data and the use of spatial knowledge by means of a rule-based system which allows to model the objects to be recognized and the environment. The use of fuzzy reasoning has demonstrated to be useful in the CEES implementation.

Applications of the system are path-planning for autonomous vehicles navigation that use behavior-based navigation instead of coordinate-based. Therefore, we plan to extend the capabilities of our system by integrating temporal knowledge.

Further research is needed to determine what kind of matching criterion is most suitable to this approach, and to reduce the computation time required for the process. Moreover, inclusion of this recognition system in a complete image understanding system is intended.

References

[1] T. O. Binford and T. S. Levitt, "Model-based recognition of objects in complex scenes," in *Proceedings: Image Understanding Workshop*, (Monterey, CA, USA), ARPA, 1994.

[2] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *Computing Surveys*, vol. 18, pp. 67–108, March 1986.

[3] U. Regensburger and V. Graefe, "Visual recognition of obstacles on roads," in *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 980–987, September 1994.

[4] P. Garnesson and G. Giraudon, "Spatial context in an image analysis system," in *Proceedings of the 1st European Conference on Computer Vision*, (Antibes, France), pp. 579–582, April 1990.

[5] H. Ishiguro, T. Maeda, T. Miyashita, and S. Tsuji, "Building environmental models of man-made environments by panoramic sensing," *Advanced Robotics*, vol. 9, no. 4, pp. 399–416, 1995.

[6] M.-P. Dubuisson and A. K. Jain, "Fusing color and edge information for object matching," in *Proceedings of the IEEE International Conference on Image Processing. Vol III*, (Austin, Texas), pp. 982–986, November 1994.

[7] T. Matsuyama, "Expert systems for image processing: Knowledge-based composition of image analysis processes," *Computer Vision, Graphics and Image Processing*, no. 48, pp. 22–49, 1989.

[8] L. B. Gamage, R. G. Gosine, and C. W. de Silva, "Extraction of rules from natural objects for automated mechanical processing," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 26, pp. 105–120, January 1996.

[9] D. Koller, K. Daniilidis, T. Thórhallson, and H.-H. Nagel, "Model-based object tracking in traffic scenes," in *Proceedings of the 2nd European Conference on Computer Vision*, (Santa Margherita, Italy), pp. 437–452, May 1992.

[10] T. A. Cass, "Robust affine structure matching for 3d object recognition," in *Proceedings of the 4th European Conference on Computer Vision. Vol I*, (Cambridge, UK), pp. 492–503, April 1996.

[11] M. Celenk, "Color scene analysis," in *Proceedings of the Conf. on Human Vision, Visual Processing and Digital Display*, pp. 407–417, SPIE, Volume 2179, 1994.

[12] P. Parodi and G. Piccioli, "3D shape reconstruction by using vanishing points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 211–217, February 1996.

[13] A. Casals, J. Amat, and A. Grau, "Texture parametrization method for image segmentation," in *Proceedings of the 2nd European Conference on Computer Vision*, (Santa Margherita, Italy), pp. 160–164, May 1992.

[14] J. L. De la Rosa, J. Aguilar, and I. Serra, *Heuristics for Cooperation of Expert Systems. Application to Process Control.* Girona, Spain: PIAR, University of Girona, 1994.