

Model-Based Objects Recognition in Industrial Environments for Autonomous Vehicles Control

Joan Martí, Joan Batlle

Alicia Casals

Computer Vision and Robotics Group
University of Girona
17071 Girona - Catalonia - Spain
e-mail: {joanm, jbatlle}@eia.udg.es

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

Abstract

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

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 potential 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 a rule-based cooperative Expert System.

1 Introduction

There are a lot of computer vision applications that require a perceptual system capable of recognizing objects in a scene, describing their spatial relationships, and determining their positions with respect to the system's camera. Typical of such applications is the computer vision system proposed in [1] to assist the navigation of behavior-based navigation autonomous vehicles, or an industrial robot which must move through and interact with its environment as in [2].

Object recognition normally 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 as several works [3, 4, 5] show. 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, it is

advisable to focus on some predefined kind of scenes to keep the system feasible, such as medical images, aerial pictures and so on.

To this effect, we propose a recognition system restricted to operate in industrial environments which are structured at different levels: walls, corridors, machines, etc. 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. The recognition is performed through a rule-based system that deals with information coming from segmentation steps and modeling procedures. This rule-based system has been implemented using CEES, the C++ Embedded Expert System Shell [6] 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.

On the other hand, integration of low-level vision techniques [4, 7] 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 [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.

The proposed recognition system takes advantage of the geometrical properties and the spatial context of structured environments such as industrial ones. This can be expressed in some general assumptions useful for modeling and interpretation purposes:

- 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 a corridor, we can try to find AGVs in it) or to validate an hypothetical object using its context (we try to find the end-effector after identifying a manipulator).

The overall structure of the proposed model-based recognition system is given in Fig. 1.

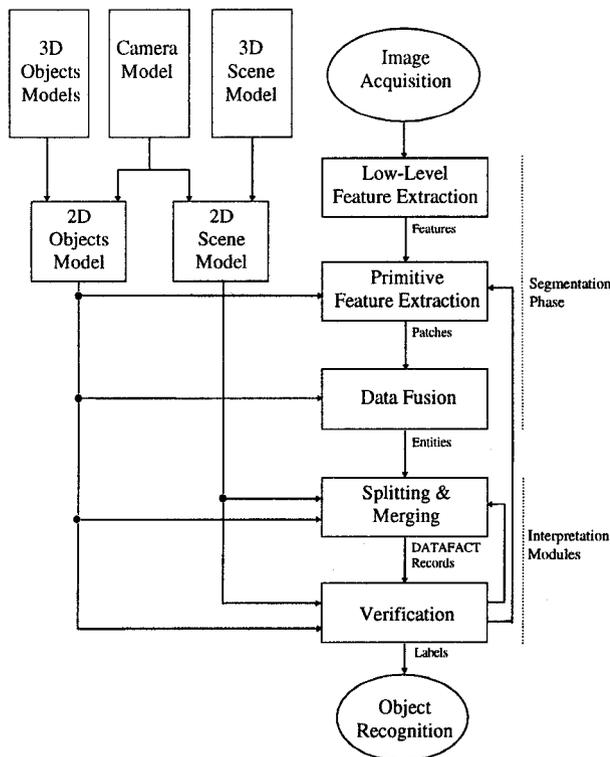


Figure 1: System structure

The paper briefly summarizes the segmentation modules, while it deeply presents the modeling and interpretation modules used, as well as the knowledge representation method chosen in the system. A more complete description of the segmentation modules can be found in [9]. Implementation using the ES is also described. Experimental results on object recognition in industrial environments are finally presented to illustrate the performance of this model-based approach.

2 Image segmentation

The system works with color pictures of indoor scenes belonging to industrial environments which are converted into region descriptions. Each region description is uniform and homogeneous with respect to some segmentation criteria (color, texture, contour and relative position to the vanishing point). It implies a parallel processing of the input raw image where the different evolving procedures gradually group small regions into more meaningful ones attending specific segmentation criterion. Using preliminary 2D information about the objects (basically color and texture) stored in the models database, the segmentation procedures generate *image patches* which are consistent with every segmentation criterion. Data fusion converts the image patches into the initial set of entities to be used by the ES in the interpretation modules, where every entity has an associated feature table containing the incoming patches features, which represents some two-dimensional features of the region.

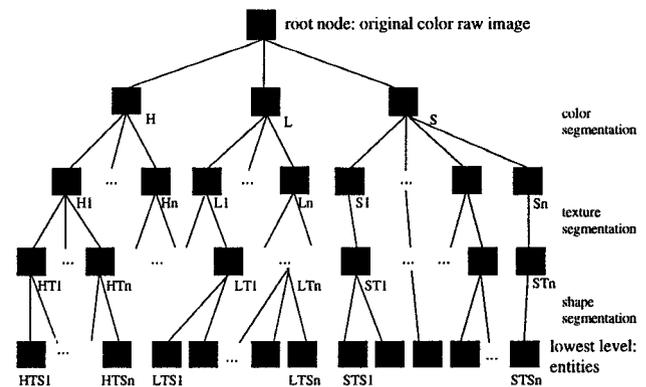


Figure 2: N-node tree representation. Color, texture and shape segmentation techniques are used to generate the entities HTS1, ... STSn and their associated feature table

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 Fig. 2 represents. 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. Additionally, the associated feature table segmentation is used to estab-

lish the correspondences amongst the various nodes of the tree.

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 and straightness.
- a list of adjacent regions and their orientation. For any region, its center of mass is computed and its relative orientation to the located vanishing point is given.

3 Modeling

Industrial environments are usually composed by a wide variety of objects. For the proposed system, rules concerning the models of objects that belong to industrial environments (machines, AGVs, storage shelves, robots, conveyors, handling devices, etc.) have been defined taking into account the following considerations about the used perspective, the objects and the scene:

1. The environment is usually structured orthogonally, as a consequence of its 3D structure approximation. It involves a set of assumptions about the objects and the scene:
 - Linear perspective becomes a central cue used by the system. In particular, lines that are parallel in the scene appear to meet at a distant vanishing point in the image.
 - The objects are roughly described as sticks, plates and blobs. The stick has two end-points, 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.

2. The surface description of the objects includes shape properties such as expected color, size, intensity, boundary shape, surface curvature, and texture.
3. The context information allows to deduce relationships between objects in the scene (e.g. an AGV always lies on the ground plane).

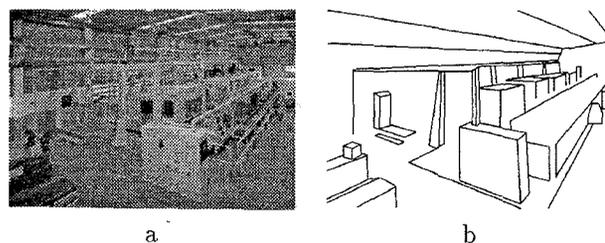


Figure 3: (a) A typical industrial scene, (b) with its modeled regions and objects

As a modeling example, Fig. 3a shows a typical industrial environment where the areas and objects are modeled as sticks, plates or blobs, as shown in Fig. 3b, following the proposed modeling.

4 Interpretation

Once we have obtained the set of features provided by the segmentation module that best initially characterizes the entities, 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.

4.1 Knowledge representation

The integration of evidence from diverse sources of knowledge is used to achieve the object recognition. This is what is generally called knowledge-based methods [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, the C++ Embedded Expert System Shell developed in the Systems Engineering and Automatic Control Laboratory at the University of Girona.

4.2 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 techniques¹ 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, available as **DATAFACT** (whatever knowledge or facts of information), **NUMERIC** (numerical variables), **ACTION** (actions performed on the objects), **INFERENCE_ENGINE** (inference engines expressed as a set of rules) and **MODEL** (a set of stored data that models the objects and the scene). The framework for cooperation is create just as many objects **INFERENCE_ENGINE** as desired to have as many cooperative ESs. 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 a feature table. Inference between visual and model data are performed by **INFERENCE_ENGINE** objects.

Following the CEES description, its most important features are:

- Data abstraction of objects, inherent to the C++ language. So, interaction between them is guaranteed.

¹“Object” as a target recognition for the Computer Vision System must be distinguished from “object” as programming techniques during ES implementation.

- Since there are multiple sources of information and this information could be semantically or in certainty contradictory, some extra processing is done to maintain integrity of information within the Knowledge Base of the receiving ES. This process is called *consensus* and is necessary because of its cooperative features.
- Fuzzy sets are available, and fuzzy reasoning will focus on the logical comparison of variables *equal*, *lower* and *greater*. The certainty of the comparison of a couple of variables will depend on the degree of matching of their values regarding these fuzzy sets.
- Cooperation between ESs implies the ability to add and modify existing models and rules in order to improve the interpretation results. This modularity is easily achieved by incorporating new **MODEL** or **INFERENCE_ENGINE** objects to the system.

4.3 Interpretation modules

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 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 that 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. 4, where a rule concerning the visual appearance of the data appears.

```

Rule 60
  Description "Decide whether an image
              region is a free corridor"
  Certainty 0.85
  Threshold 0.30
  TraceHere Yes
  If (*Region1).FeaturesTable.Adjacency.equal
    ((*Corridor).FeaturesTable.Adjacency)
  And /* A CEES fuzzy condition */
    (*Region1).FeaturesTable.Texture.equal
    ((*Corridor).FeaturesTable.Texture)
  And /* A C++ non fuzzy condition */
    (*Region1).Type == plate
  Then
    deduce(INTERMEDIATE, CORRIDORFOUND)
  EndIf
EndRule

```

Figure 4: Example for a rule in the verification module

This rule attempts to match the visual data contained in **DATAFACT** object record **Region1** with the model **Corridor** using geometrical, textural and adjacency features obtained during the segmentation process. Results (such as new intermediate object records or matching performance) are stored in **DatafactResult** that is later evaluated. The matching performance is evaluated using an homogeneity criterion between the obtained entity and the model to match, based on the following dissimilarity measure ΔS_{rm} :

$$S_p = \sum_{j=1}^f \sum_{i=1}^{n_r} (x_{ij}^r - \bar{x}_j^r)^2 \quad (1)$$

$$\Delta S_{rm} = \frac{n_r n_m}{n_r + n_m} \sum_{j=1}^f (\bar{x}_j^r - \bar{x}_j^m)^2 \quad (2)$$

where n_r and n_m are the number of pixels in the entity r and in the model m , and f is number of features to evaluate in every matching process.

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 subclasses 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 recognition is refused.

5 Experimental results

To evaluate our approach we have used some simple industrial scenes, as shown in Fig. 5a which represents an original gray value image. The set of white lines superimposed on Fig. 5b is the edge map derived from the model in the estimated position of the mobile robot. Fig. 5b shows the entity image labeled as recognized corridor.

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 relevant regions: the corridor, the ceiling, and the walls. The detection of the corridor requires the extraction of its sides, which is greatly simplified by the fact that the world is orthoedrically modeled. Therefore, an essential step in the extraction of the corridor 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.

6 Conclusions and further work

We have presented a recognition system intended to operate on industrial 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-

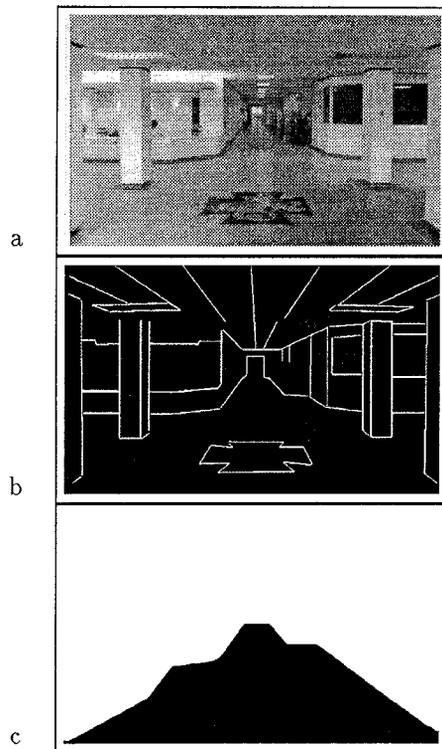


Figure 5: (a) Hallway and corridors in a structured environment (b) Superimposed edge map (c) Recognized corridor

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] 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.
- [2] E. Rivlin and A. Rosenfeld, "Navigational functionalities," *Computer Vision and Image Understanding*, vol. 62, pp. 232–244, September 1995.
- [3] 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.
- [4] 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.
- [5] S. Das, B. Bhanu, and C.-C. Ho, "Generic object recognition using multiple representations," *Image and Vision Computing*, vol. 14, pp. 323–338, 1996.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] J. Martí and A. Casals, "Object recognition using multiple cues," in *Proceedings of the VII National Symposium on Pattern Recognition and Image Analysis*, (Bellaterra, Spain), April 1997. To be published.
- [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] J. Batlle, *Aportation to movement detection independently of camera movement, based on colour image processing*. PhD thesis, Polytechnical University of Catalonia, Barcelona, Spain, September 1993.
- [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.