MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

Implementation of An Object-Oriented & Declarative Language for Image Understanding

E-ren Chuang

Department of Information Management
Kaohsiung Polytechnic Institute
Tahsiung, Kaohsiung County
R.O.C

David Sher

Department of Computer Science
State University of New York at Buffalo
Buffalo, N.Y. 14260
U.S.A

ABSTRACT

In this paper, we discuss a declarative & object-oriented language, VISUAL, for image understanding, in the another words, given relationships among components of an object written in VISUAL, the inference engine of VISUAL will automatically locate the object in the image. The output of the engine is a database of the objects and a $N \times N$ 2-D map which contains the boundary points of each object in the database. Those points are numbered according to the indices of the objects in the database. The inference engine of VISUAL adopts chromatographic search on the map, and choose the nearest neighbors for unification. Therefore, the computation time of unification can be greatly reduced.

1. INTRODUCTION

It is generally believed [3] that model-based system for image understanding include three parts: *feature extraction, object modeling* and *recognition*. There is a recognition engine for the last part. The engine recognizes objects by comparing features extracted from an input image to the object features in the models. There are various mechanisms for the recognition engine, like statistical or syntactical approach, CAD-based vision system, and rule-based or prolog-based approach. Different mechanisms rely on different object modelings.

In the statistical pattern recognition approach [4], an object is described by a vector which elements are global features, like centroid, curvature, moments of inertia, and so on, and matching becomes comparison between the feature vec-curves, which compose the boundary of objects are organized in a structured manner, then syntactic approach can be applied [9, 6]. In the syntactic approach, object models are built by grammars with a set of primitives. Local features in an image are transformed into a string, and then parsing procedure becomes matching between structural models and the string.

CAD-based (Computer-Aided Design) object representation has been applied to model-based vision systems [5, 7, 2]. It is a 3-D representation. This representation can describe an object completely without being constrained by the coordinates of the sensor system. However, objects in 2-D image can not compare directly to the 3-D representation, so sophisticated algorithms are developed to transform the 3D representation into 2-D shapes or trees from all of the possible aspects. Matching process is usually based on the strategy of hypothesis-then-verification, i.e. make a hypothesis of a 2-D shape in the models if a chosen feature is found in the image, then verify whether the rest of the features in the hypothesis can also be found in the image. 3D representation for objects has the properties of completeness, and compactness, however, not every object needs 3D representation, like text, road signs, or traffic lights. Therefore, no matter which representation or matching strategy a system adopts, eventually this system has to match those models in appropriate form against the features extracted from images.

There are two other approaches: rule-based and prolog-based one. In the rule-based approach, knowledge is translated into rules, and each rule will invokes an action which is related for image processing tools, like segmentation, and interpretation. It also provides tools for knowledge acquisition of scene primitive, and spatial constrains. The prolog-based approach [1], provides predicates in image processing and image input/output, to the control structures and inference engine of Prolog to do image understanding tasks. However, Prolog takes a long time to find simple objects, such as line segments, in images because these objects have few constraints or attributes to be described so that there are numerous hypotheses to verify.

This paper is organized as follows: The second section discusses the implementation of the inference engine of VISUAL. The engine includes two components, which are internal representation and chromatographic search, and four procedures, which are *initialization, pre-processing, unification, and plotting procedure*. At last, we will demonstrate the system by examples.

2. IMPLEMENTATION of VISUAL

Since VISUAL is a declarative & object-oriented language. There is an inference engine (INEG) in VISUAL to perform unification. The input of INEG is an object description written in VISUAL. The outputs of INEG are a database and a map. There are two major components and four procedures in INEG. The four procedures are initialization, pre-processing, unification, and plotting. The two major components are internal representation and the chromatographic search. The procedures and the major components are discussed in the following sections.

2.1 INTERNAL REPRESENTATION

The internal representation of an object is object-oriented. The object type is defined in the declaration part of a VISUAL program. Objects of the same type are stored in a database. The boundary point of these objects also are depicted in a map. These points in the map are numbered according to the indices of corresponding object records in the database. For examples, a rectangle is composed by line segments, l_1, l_2, l_3, l_4 which records are stored in the position 1, 2, 3, 4 of the data set, so the edge points of l_1 are all numbered 1; l_2, 2, and so on. Therefore, this numbering becomes the linkage between the object in the map and its corresponding data in the set. In addition, this numbering realizes the chro-

2.2 CHROMATOGRAPHIC SEARCH

The process of CSH is like radar spreading out signal in all directions. CSH directly searches nearest objects on the map. Because the boundaries of objects in the map are numbered according to their record indices in the database, given an object, Therefore, the indices become the numbering becomes the linkage between the object in the map and its corresponding data in the database.

The CSH has the following two merits:

1. Polymorphism. The CSH is polymorphic because it can seearch objects of any types. The polymorphism of CSH is realized by the numbering of boundary points of objects. Although different objects in databases have differnt data types, the type of their corresponding maps is integer array. Therefore, CSH works on 2-D maps of homogenous type rather than on databases of different types.

2. Omnidirection. The process of CSH is like a radar which spreads out signals in all directions. While the signals encounter an object, the number of the object will be reported. Although it takes only $O(\log n)$ time to search the nearest neighbor, it takes $O(m \times n \log n)$ to sort n data in all m directions (if it is possible sort data in the direction like 36 degree). Hence, CSH has advantage to search k nearest objects in all directions.

2.3 FOUR PROCEDURES

There are four procedures of INEG. They are initialization, pre-processing, unification, and plotting. Their functions are described as follows.

1. *Initialization*: The first step of INEG is initialization. The files regarding to imported components and the exported object are opened, then the databases and maps of components are installed.

2. *Pre-processing*: The relationships among components are written in C++ programming, so they will be extracted from the VISUAL program. The control structure, AND, OR, or recursive call in the VISUAL program, will be translated to a corresponding control structure in C programming, and

3. *Unification*: Unification is a process which determines the values for varibles. This procedure takes indexes provided by CSH, and take the data from the database. It will assign at most k values to a variable because of the principle of closeness, therefore, the computation time is reduced.

4. *Plotting procedure*: The output of the INEG include a map, so there is a procedure to draw the boundary of objects. If some components satisfy all of the pre-defined relationships, then one of the described object is found. The boundary points of the object are determined by the boundary points of the satisfactory components.

INEG applies dynamic programming to locate the described object, and choosees the k nearest neighbors for unification as a pruning technique because object are more likely to have close relationship if they are close in space. For instances, two line segments in a local area may form a corner, or parallel lines, so they are more related to each other. By the technique, the computationa time of unification can be greatly reduced.

3. EXPERIMENTS

In this section, we demonstrate the power of VISUAL programs on an image. This image contains a stop sign which is an important landmark for robotic navigation. We write a program in VISUAL to locate candidates of stop sign in a street scene. Because octagons are rare in the natural world, we treat the octagons as strong candidates of stop signs.

First we define a line segment by a edge point attached to another line segment, and an octagon as a polygon composed of eight line segments and eight angles, and the angles are of 135 degree. We encode the definitions by VISUAL programs. The results of locating line segments and octagon are shown in Figure 1(c) and (d), respectively.

4. CONCLUSION

We design a programming language VISUAL for image understand, whcih includes properties of declarative and object-oriented languages. The inference engine of adopt CSH to search nearest objects for unification. Because the output of a VISUAL program may be the input of another one, a hierarchical machine vision system can be built and a described object can be automatically located by VISUAL.

References

[1] Bruce G. Batchelor. *INTELLIGENT IM-*

[2] Bir Bhanu. Cad-based robot vision. *IEEE Computer (Magzine)*, 20, August 1987.

[3] Roland T. Chin and Charles R. Dyer. Model-based recognition in robot vision. *Computing Surveys*, 18, March 1986.

[4] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[5] Patrick Flynn and Anil Jain. Bonsai : 3-d object recognition using constrained search. *PAMI*, 13, October 1991.

[6] King Sun Fu. *Syntactic Pattern Recognition*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1982.

[7] K. Ikeuchi. Generating an interpretation tree from a cad model for 3d object recognition in bin-picking tasks. In *Int. J. Comput. Vision Vol. I, No. 2*, pages 145–165, 1987.

[8] D.M. McKeown, W. Harvey, and L. Wixson. Automating knowledge acquisition for aerial image interpretation. *CVGIP*, 46, 1989.

[9] P. Trahanias and E. Skordalakis. Syntactic pattern recognition of the ecg. *PAMI*, 12, July 1990.

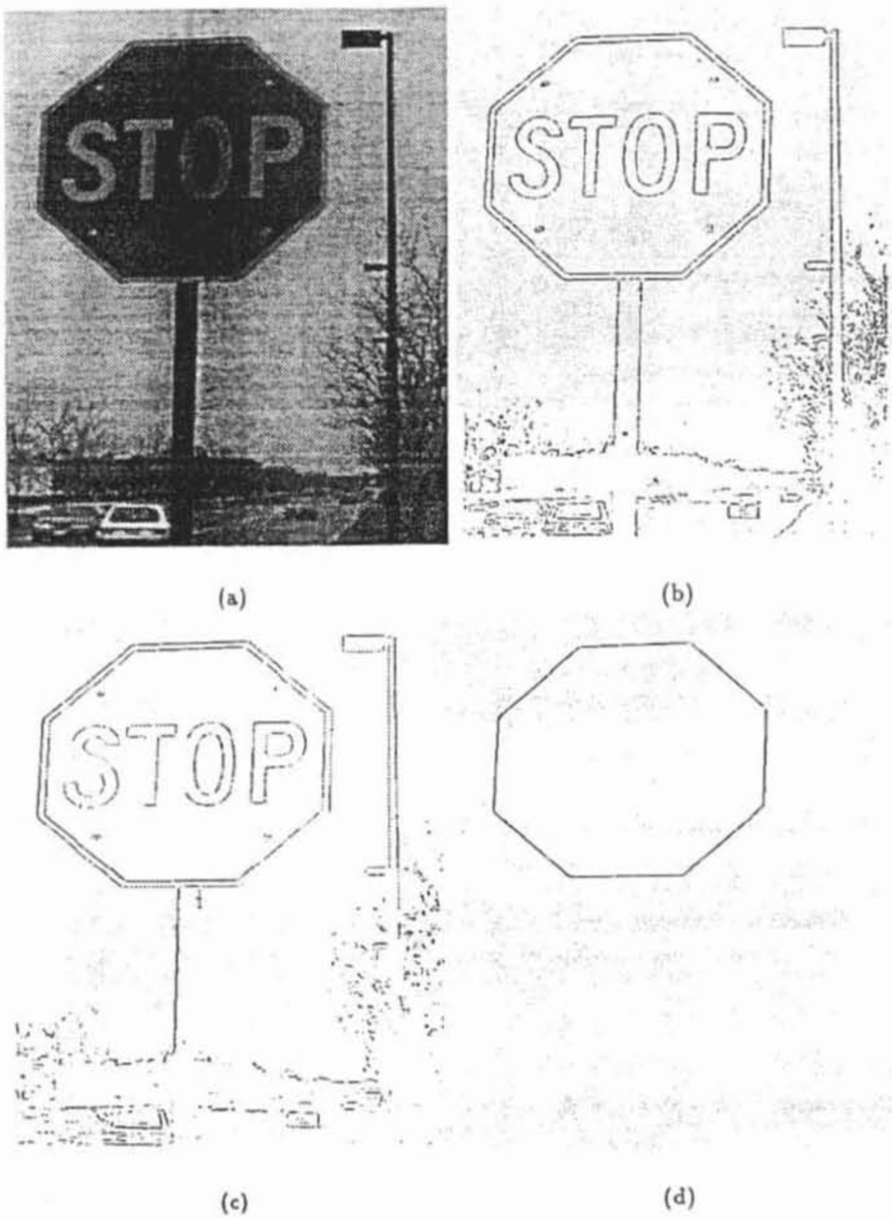

Figure 1: VISUAL programs to locate stop sign in a image. From (a) to (d) are the image, and results of edge detection, locating line segment, and octagon detection.