CHAPTER 12

DECOMPOSITION OF A VISUAL SCENE INTO THREE-DIMENSIONAL BODIES*

Adolfo Guzmán
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

I. INTRODUCTION

We consider visual scenes composed by the optical image of a group of bodies. When such a scene is "seen" by a computer through a film spot scanner, image dissector, or similar device, it can be treated as a two-dimensional array of numbers, or as a function of two variables.

At a higher level, a scene could be meaningfully described as a conglomerate of points, lines and surfaces, with properties (coordinates, slopes, . . .) attached to them.

Still a more sophisticated description could use terms concerning the bodies or objects which compose such a scene, indicating their positions, inter-relations, etcetera.

This paper describes a program which finds bodies in a scene, presumably formed by three-dimensional objects. Some of them may not be completely visible. The picture is presented as a line drawing.

When SEE--the pretentious name of the program--analyzes the scene *TRIAL* (see Fig. 1), the results are:

(*BODY* 1. *IS* :6 :2 :1)
(*BODY* 2. *IS* :11 :12 :10)
(*BODY* 3. *IS* :4 :9 :5 :7 :3 :8 :13)

SEE looks for three-dimensional objects in a two-dimensional scene. The

---

* Reprinted, with permission, from AFIPS Proceedings of the Fall Joint Computer Conference, Vol. 33, pp. 291-304 (Thompson Book, 1968).

Artículo 15

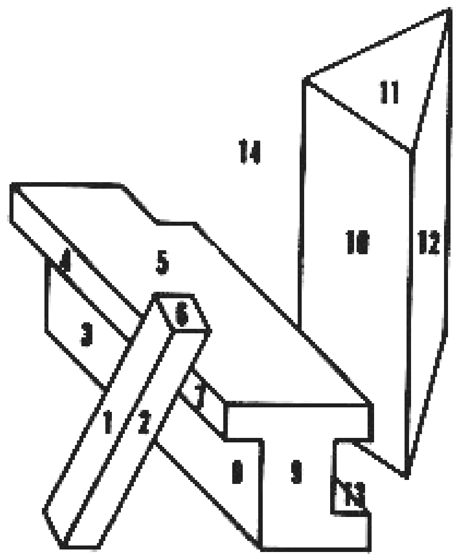

Fig. 1. *'TRIAL'*. The program analyses this scene and finds 3 bodies:
(*BODY* 1 *IS* :6 :2 :1)
(*BODY* 2 *IS* :11 :12 :10)
(*BODY* 3 *IS* :4 :9 :5 :7 :3 :8 :13)

scene itself is not obtained from a visual input device, or from an array of intensities or brightness. Rather, it is assumed that a pre-processing of some sort has taken place, and the scene to be analyzed is available in a symbolic format (to be described in Section III), in terms of points (vertices), lines (edges), and surfaces (regions).

SEE does not have a pre-conceived idea of the form or model of the objects which could appear in a given scene. The only supposition is that the bodies are solid objects formed by plane surfaces; in this way, it can not find "cubes" or "houses" in a scene, since it does not know what a "house" is. Once SEE has partitioned a scene into bodies, some other program will work on them and decide which of those bodies are "houses."

Thus, SEE is intended to serve as a link between a pre-processor [*1, 2*] which transforms intensity pictures into point or line pictures [5], and a recognizer (such as TD [3] or DT [4]), which handles this line picture and finds bodies, objects or zones matching with certain patterns or models. Instead of searching through the whole scene looking for parts to match its models, the work of the recognizer becomes simpler after SEE has partitioned the scene into bodies, because the data to be searched (matched) is smaller and better organized.

The analysis which SEE makes of the different scenes generally agrees with human opinion, although in some ambiguous cases it behaves rather conservatively. Distributed over these pages, the reader will find examples of scenes analyzed by SEE, and the peculiarities and behaviour of the program will become clear.

The program SEE, written in LISP, has been tested in the PDP-6 machine of the Artificial Intelligence Group, Project MAC, at Massachusetts Institute of Technology. A preliminary version, written in CONVERT [6], was used extensively for a quick test of ideas which shaped the program to its actual form. The analysis of a scene takes from 30 to 90 seconds, with the program running interpreted under the interpreter of the LISP programming system.

A more technical description of SEE can be found in an unpublished memorandum [7].

Related work

Rudd H. Canaday [8] in 1962 analyzed scenes composed of two-dimensional overlapping objects, "straight-sided pieces of cardboard." His program breaks the image into its component parts (the pieces of cardboard), describes each one, gives the depth of each part in the image (or scene), and states which parts cover which.

Roberts [9] in 1963 described programs that (1) convert a picture (a scene) into a line drawing and (2) produce a three-dimensional description of the objects shown in the drawing in terms of models and their transformations. The main restriction on the lines is that they should be a perspective projection of the surface boundaries of a set of three-dimensional objects with planar surfaces. He relies in perspective and numerical computations, while SEE uses a heuristic and symbolic (i.e., non-numerical) approach. Also, SEE does not need models to isolate bodies. Roberts' work is probably the most important and closest to ours.

Actually, several research groups (at Massachusetts Institute of Technology [10], at Stanford University [11], at Stanford Research Institute [12]) work actively towards the realization of a mechanical manipulator, i.e., an intelligent automata who could visually perceive and successfully interact with its environment, under the control of a computer. Naturally, the mechanization of visual perception forms part of their research, and important work begins to emerge from them in this area.

Organization of the paper

It is formed by the following sections:

I. Introduction and related previous work.
II. Input Format. The representation of the scene as it is entered to the computer.
III. Format of a Scene. The representation of the scene as SEE expects.
IV. Types of Vertices. Classification of vertices according to their topology
V. The program. Analysis of the algorithm, description of heuristics.
VI. Interesting examples. Discussion. Future work.

II. INPUT FORMAT

For testing purposes, the scenes are entered by hand in a simplified format (called input format), and then some routines convert this to the form required by SEE (this form is described in Section III). Eventually, this data will come from a visual input device, through a preprocessor.

Examples of a Scene

Suppose we want to describe the scene *'CUBE'*. (See Fig. 2.) We begin

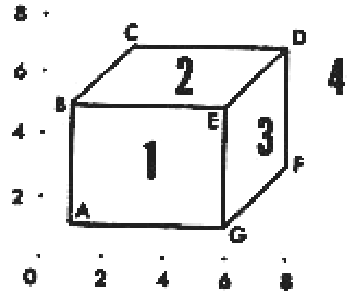

Fig. 2. *'CUBE'*. A scene.

by giving in LISP) a value to '*CUBE*'.

```
(SETQ CUBE (QUOTE (A  1.0  1.0  (:1 B :4 G)
                   B  1.0  5.0  (:1 E :2 C :4 A)
                   C  3.0  7.0  (:2 D :4 B)
                   D  8.0  7.0  (:2 E :3 F :4 C)
                   E  6.0  5.0  (:2 B :1 G :3 D)
                   F  8.0  3.0  (:3 G :4 D)
                   G  6.0  1.0  (:1 A :4 F :3 E)
                   )))
```

Thus we associate with each vertex its coordinates and a list, in counterclockwise order, of regions and vertices radiating from that vertex.

The conversion of the scene, as just given, into the form which SEE expects, is made by the function *LLENA;* thus, *(LLENA CUBE)* will put in the property list of *CUBE* the properties *REGIONS* and *VERTICES;* in the property list of each vertex, the properties *XCOR, YCOR, NREGIONS, NVERTICES* and *KIND* are placed; and in the property list of each region, it places the properties *NEIGHBORS* and *KVERTICES.* It also marks region :4 as part of the background.

In other words, *LLENA* converts a scene from the 'Input Format' to the 'Format of a Scene' described in Section III.

III. FORMAT OF A SCENE

A scene is presented to our program as a scene in a special symbolic format, which we now describe. Essentially, it is an arrangement of relations between vertices and regions.

A scene has a name which identifies it; this name is an atom whose property list contains the properties *'REGIONS'*, *'VERTICES'*, and *'BACKGROUND'*. For example, the scene *ONE* (see Fig. 3) has the name *'ONE'*. In the property list of *'ONE'* we find

REGIONS --	(:1 :2 :3 :4 :5 :6)	Unordered list of regions composing the scene *ONE*.
VERTICES --	*(A B C D E F G H I J K)*	Unordered list of vertices composing the scene *ONE*.
BACKGROUND --	(:6)	Unordered list of regions composing the background of the scene *ONE*.

access to the computational facilities of Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). This paper, having had this support from MAC, may be reproduced in whole or in part for any purpose of the United States Government.

REFERENCES

1 Greenblatt, R. and J.Holloway, Sides 21, Memorandum MAC-M-320 (A.I. Mem.101), Project MAC, M.I.T. (August 1966).

2. Pingle, K. K., A program to find objects in a picture, Memorandum No. 39, Artificial Intelligence Project, Stanford University (January 1966).

3. Guzmán, A., Scene analysis using the concept of model, Technical Report No. AFCRL-67-0133, Computer Corporation of America, Cambridge, Mass., U.S.A. (January 1967). (AD 652017)

4. Guzmán, A., A primitive recognizer of figures in a scene, Memorandum MAC-M-342 (A.I. Memo. 119), Project MAC, M.I.T. (January 1967).

5. Guzmán, A., Some aspects of pattern recognition by computer, M.S. Thesis, Electrical Engineering Dept., M.I.T., Project MAC Technical Report: MAC-TR-37 (February 1967). (AD 656041)

6. Guzmán, A. and H.V. McIntosh, "CONVERT", *Comm. A.C.M.*, 9, pp. 604-615 (1966).

7. Guzmán, A., Decomposition of a visual scene into bodies, Memorandum MAC-M-357 (A.I. Memo. 139), Project MAC, M.I.T. (September 1967).

8. Canaday, R.H., The description of overlapping figures, M.S. Thesis, Electrical Engineering Dept., M.I.T. (January 1962).

9. Roberts. L.G., Machine perception of three-dimensional solids, in *Optical and Electro-optical Information Processing*, J.T. Tippet *et al.*, Eds., pp. 159-197 (M.I.T. Press, 1965).

10. Minsky, M.L. and S.A. Papert, Research on intelligent automata, Status Report II, Project MAC, M.I.T. (September 1967).

11. McCarty, J., Plans for the Stanford Artificial Intelligence Project, Stanford University (April 1965).

12. Rosen, C.A. and N.J. Nilsson, (Eds) Application of intelligent automata to reconnaissance, third Interim Report, Stanford research Institute (December 1967).

13. Guzmán, A., Computer recognition of three-dimensional objects in a visual scene, Ph.D. Thesis, Electrical Engineering Dept, M.I.T., Cambridge, Mass., U.S.A. (December 1968). Also available as MAC-TR-59; Project MAC, MIT.