Reprinted from Proceedings of the
Second Hawaii International
Conference on System Sciences.
Honolulu, Hawaii, January 1969.
pages 479-482. Western Periodicals
Co.

OBJECT RECOGNITION:
DISCOVERING THE PARALLELEPIPEDS IN A VISUAL SCENE

by

Adolfo Guzmán
Artificial Intelligence Group of Project MAC
Massachusetts Institute of Technology; Cambridge, Mass.

Synopsis

An interesting pattern recognition problem in automatic photo-interpretation consists of the analysis of a picture or other suitable two-dimensional representation of a three-dimensional scene, in order to isolate or identify in it desired objects or classes of objects, where in general these bodies are three-dimensional in nature, but the input picture contains only two-dimensional information about them.

Techniques are here presented, in the form of a computer program, which allow identification of parallelepipeds (three-dimensional solids limited by six planes, parallel two-by-two) contained in a picture or visual scene. The scene to be analyzed is represented as a two-dimensional parallel projection consisting of points (vertices) and lines (edges). The scene is supposed to contain arbitrary objects formed by straight edges, from which the program will select the parallelepipeds. In order to deal successfully with partially occluded parallelepipeds, the program uses heuristics that link local evidence about the kinds of vertices with somewhat more global evidence about their colinear relations.

Introduction

POLYBRICK is a program that solves the following problem:
A scene contains noise-free parallelepipeds without perspective

Artículo 18

effects, but partially occluding one another. Extraneous rectilinear objects other than parallelepipeds may be present.

__Problem__: What parallelepipeds (hereafter called "cubes") are there and where are they (in the 2-dim scene)?

__Input to the Program__ Eventually the program will read its data directly from the real world; right now, the picture is transformed (by hand) to a list of corners and points of intersection (real or virtual) of lines, and their two-dimensional coordinates in the picture, together with their nearest adjacent points.

__Format of the Answer__ An example is given. We use the CONVERT processor and apply the function __cube__ (in a given input-file), our current implementation of POLYBRICK, to the picture GORDO. Answers are

```
(THERE ARE AT LEAST 3 OR 3 CUBES)
(CUBE 1 IS (N (W O M) W (N J L) L (M K W)) )
(CUBE 2 IS (I (J H X) G (F X H) X (E G I) E (X F D )))
(CUBE 3 IS (P (A O Q) R (S Q T) Q (B R P) B (Q C A )))
```

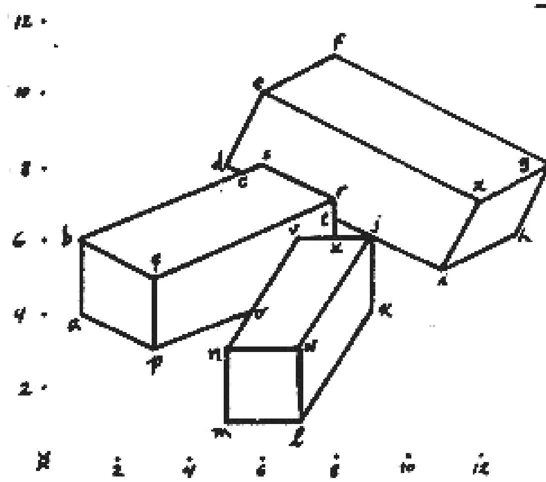
__Algorithms and Heuristics employed__ The vertices of the scene are classified according to whether they form CORNERS (two lines), Y's (three non-colinear lines meeting), T's (three lines meeting, two of them colinear) or ANY's (anything else).

Each Y is characterized by a set of 3 numbers: the slopes of its edges. Thus, the Y's of GORDO are partitioned into classes:

slopes: 2., 0.5, -0.5	slopes: 0.4, ∞ , -0.5	slopes 0,1.5,∞

At a first approximation, each of these classes could be considered as a parallelepiped by POLYBRICK. In any event, it is impossible for a cube to be in two classes simultaneously; therefore, in what follows, the program works with one class at a time, ignoring all the vertices in the other classes.

__Senses or Signs of the Direction of the Lines of a Y__ Two cubes

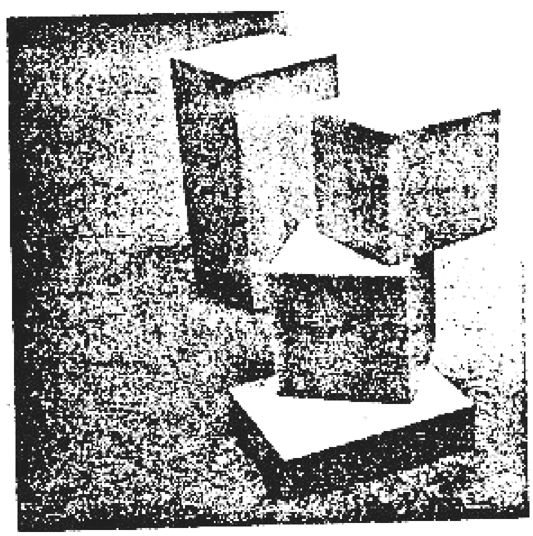


Vertex	Coordinate X	Y	neighbors
A	1	4	B P
B	1	6	A Q C
C	5 4/9	7 7/9	B D S
D	5	8	E C
E	6	10	D X F
F	8	11	E G
G	14	8	X H F
H	13	6	I G
I	11	5	X H J
J	9	6	I T U WK
K	9	4	J L
L	7	1	K W M
M	5	1	N L
N	5	3	O W M
O	5.7213	4.0909	P V
P	3	3	A Q O
Q	3	5	B P R
R	8	7	Q S T
S	6	8	C R
T	8	6.5	R U J
U	8	6	T V J
V	7	6	U D
W	7	3	N J L
X	12	7	I G E

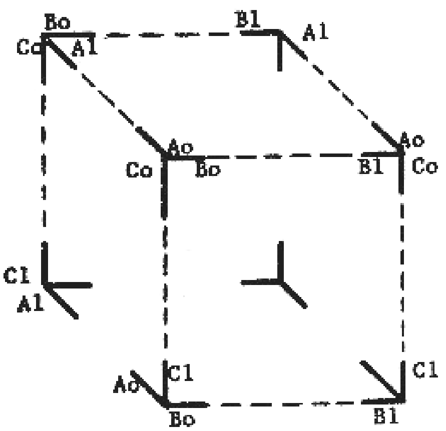
Fig. GORDO

To its right is its description list, the input to the program.

may fall in the same class, if their edges are approximately parallel (see figure at right). To discriminate among them, the eight possible Y's of an ideal cube are uniquely characterized by associating with each of them the sense or sign of the direction of the lines forming it (see figure "SENSES"); for a given direction, sense 1 means the opposite of sense 0, choice of sense 0 being arbitrary.

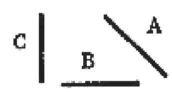


Vertices that fall on the same edge of a cube disagree in exactly one sense; for instance, Ao Bo Co and Ao Bo C_1. During the analysis of a given Y, these signs allow POLYBRICK to know what kind of vertex to expect to be connected to such a Y in each of its three directions; for instance, from A_1BoCo we expect to find AoBoCo in the direction A_1, A_1B_1Co in the direction Bo and A_1BoC_1 in the direction Co.



Types of vertices

A	B	C	
0	0	0	"Y"
0	0	1	"arrow"
0	1	0	
1	0	0	
1	1	0	"L"
1	0	1	
0	1	1	
1	1	1	"invisible Y"



DIRECTIONS

Figure 'S E N S E S'
The three possible directions are A, B, and C. Each one has two senses: 0 and 1. When POLYBRICK analyzes a 'Y', these senses provide information as to the type of vertex expected to be connected to the Y in question.

Further Information and References Polybrick is written in the CONVERT language:

1. Guzmán, A., and McIntosh, H. V. "CONVERT" Communications of the ACM 9, 8 (August 66), pp. 605-615.

More heuristics and rules are used; these and the program are in:

2. Guzmán, A. Polybrick: Adventures in the Domain of Parallelepipeds. Project MAC Memo MAC M 308 (AI Memo 96). Massachusetts Institute of Technology. June 1966.

3. Guzmán, A. Some Aspects of Pattern Recognition by Computer. M. S. Thesis, Electr. Eng, Dept. MIT. Febr 67. Also available as Project MAC (MIT) report MAC TR 37

A program that finds any kind of rectilinear objects in a scene, even in the presence of inaccurate (noisy) data is found in

4. Guzmán, A. Computer Recognition of Three-Dimensional Objects in a Visual Scene. Ph. D. Thesis, Electr. Eng. Dept., M. I. T. December 1968. To appear also as a Project MAC, MIT Technical report: MAC-TR-59.