ANNALS of the ORADEA UNIVERSITY.

Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008

ALGORITHM FOR RECOGNIZING 3D OBJECTS UTILIZING NOTIONS OF THE FORMAL LANGUAGES THEORY

Marius NICA[1], Lucian, Macedon GANEA[2], Gheorghe DONCA[1],

[1]. eng., drd., University of Oradea, [2]. prof., PhD., eng., University of Oradea

e-mail: mariusnica@yahoo.com

Keywords : formal language, object categorization, characteristic views, topological recognition, 3-D object recognition

ABSTRACT: Vision systems work in real time in order to rapidly solve problems in flexible manufacturing systems. In this paper we present a theoretical investigation on the use of formal description for topological recognition from multiple views of polyhedral objects. This method is based on the characterization of 3D simple objects by a set of 3 characteristic views.

1. INTRODUCTION

The original algorithm presented below solves the problem of the recognition of tri-dimensional objects based on bi-dimensional images of these, utilizing a symbolic formal description of the topology of the objects. The algorithm is presented for two simple objects, parallelepiped and pyramid, and the expansion of the algorithm is default.

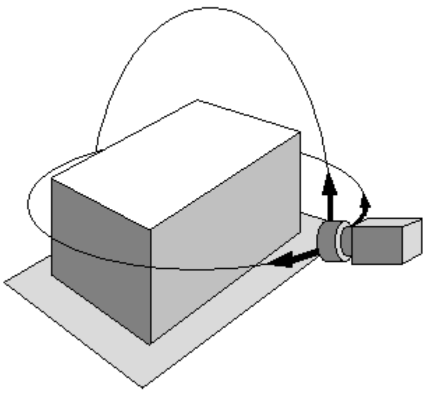

Fig. 1. Camera positioning for image aquisition

Any tri-dimensional body has an infinite number of views. The views that do not differ from a topological point of view are topologically equivalent (TE). For each class of TE views, a characteristic view is assigned. Any tri-dimensional body possesses an infinite number of characteristic views. For example, the parallelepiped from Fig.1 possesses three characteristic views, presented in Fig.2.

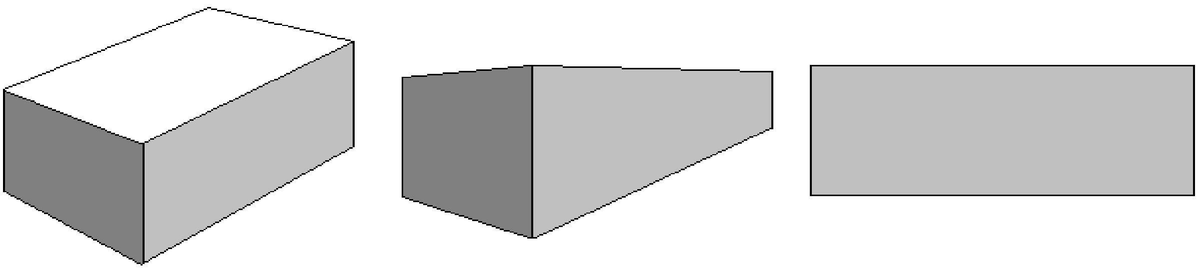

Fig. 2. Different views of the same object.

2. THE RECOGNITION ALGORITHM

The logical scheme of the algorithm is presented in Fig.3.

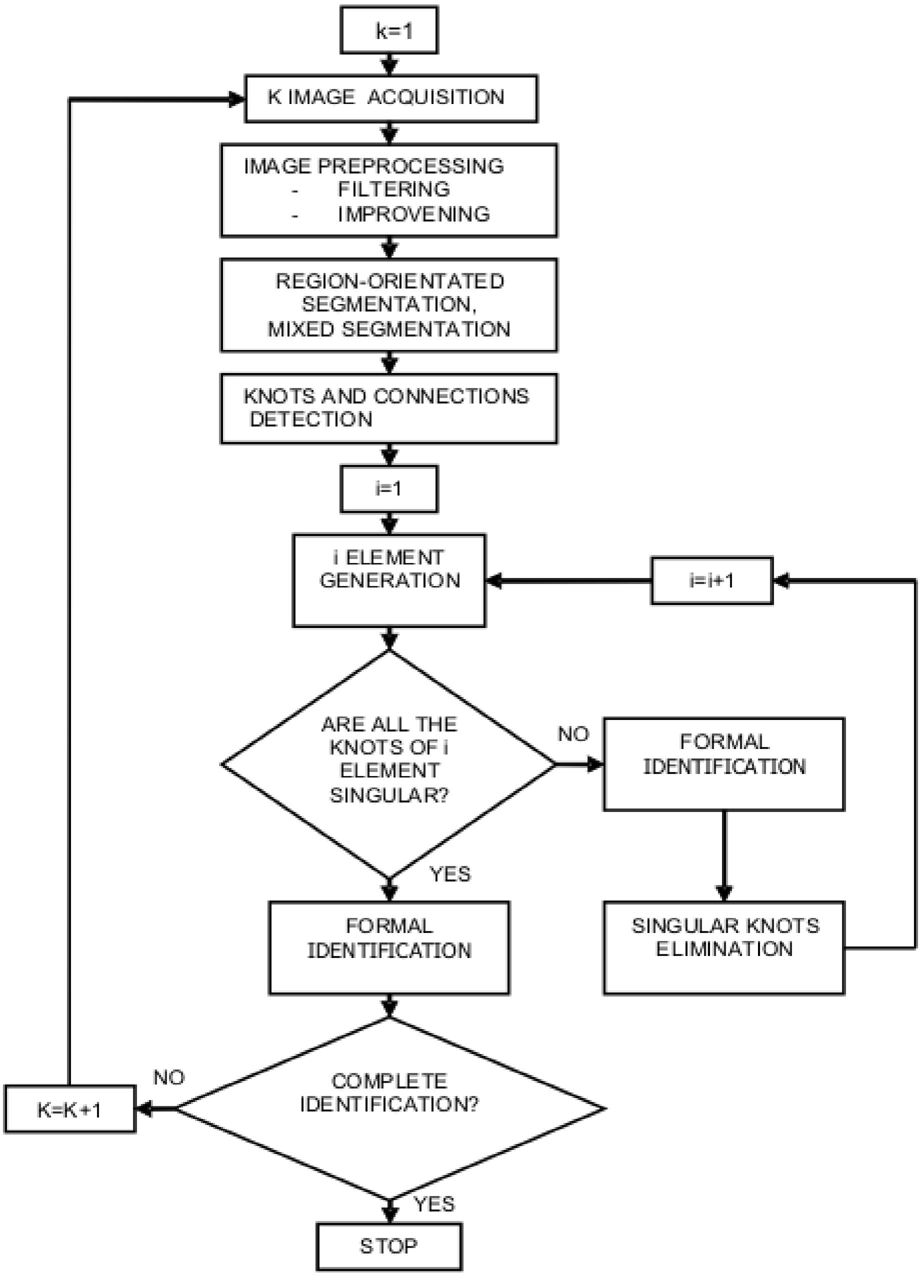

Fig. 3. Recognition algorithm flowchart

After filtration and segmentation, images of the object to those presented in Fig.4 are obtained. On these images, the knots (a, b, c, ….) and the connections between them are identified. The knots are noted arbitrarily with alphabetic symbols, and the words and phrases that describe every characteristic view are generated. The relations between the words of each phrase describe the topology of the characteristic views.

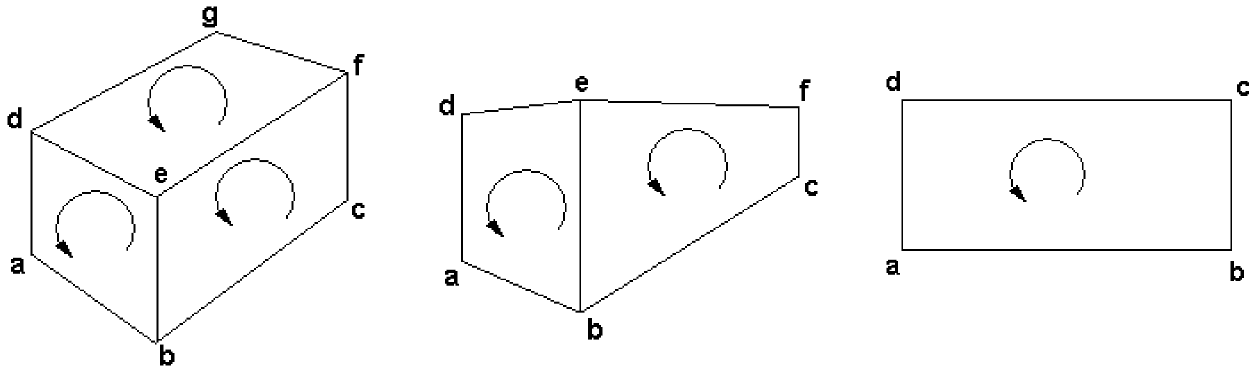

Fig. 4. The arrows indicate the routing direction of the elements in the process of phrase generation.

For the characteristic views from Fig.4 the following phrases are generated:

abed bcfe efgd.abed bcfe.abcd

Here, the symbol “ “ is utilized as a divider between words and the symbol “.” as a divider between phrases.
The number of phrases is added: Nf.
The number of words in each phrase is added: Nc(i); i = 1... Nf.
The number of symbols in each word of every phrase is added:
Ns(i, j); i = 1... Nf; J = 1... Nc

The information defined above is saved in a database as being the description of parallelepiped kind objects.
As a part of utilizing the algorithm of recognition, at a given time, an unknown object is presented (for illustration, yet another parallelepiped, presented in Fig.5).

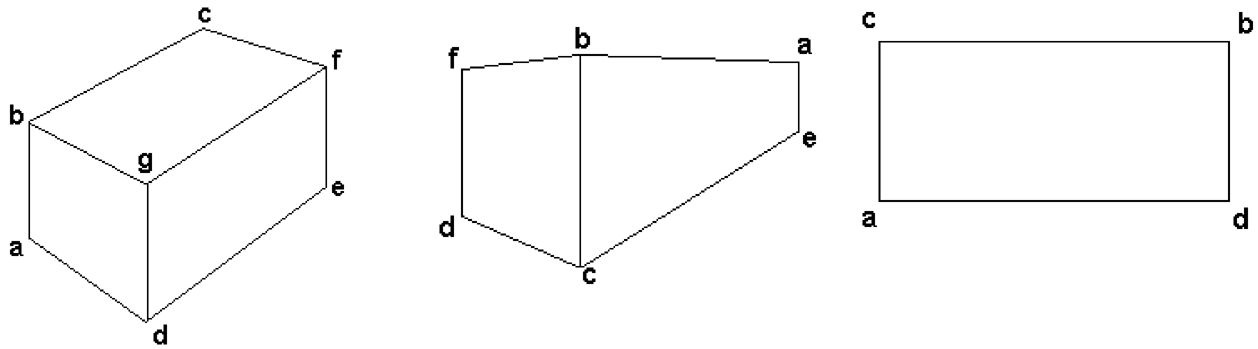

Fig. 5.A new object with automatic knot numbering.

For the unknown object the algorithm generates the characteristic phrases:

adgb/defg/gfcb; dcbf/ceab; adbc

From here begins a process of identification, of the phrases that describe the unknown object, with the characteristic phrases saved in the database.
The algorithm is tested on different objects, such as the one from Fig.6.

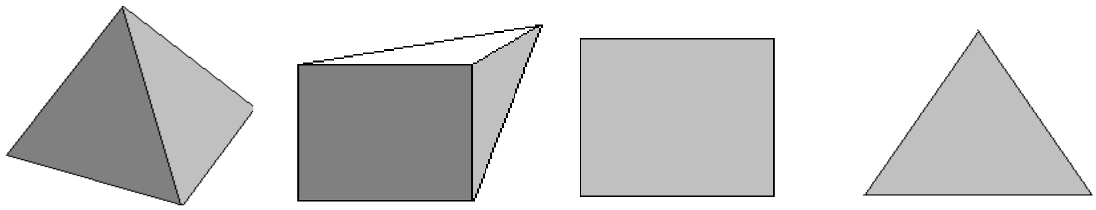

Fig. 6. Different views for a prism type object

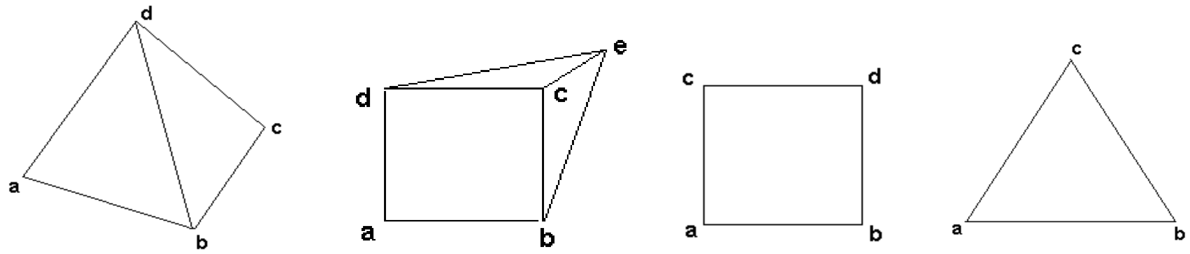

Fig. 7. Knots numbering for prisms views in database.

abd/bcd; abcd/bec/ced; abdc; abc

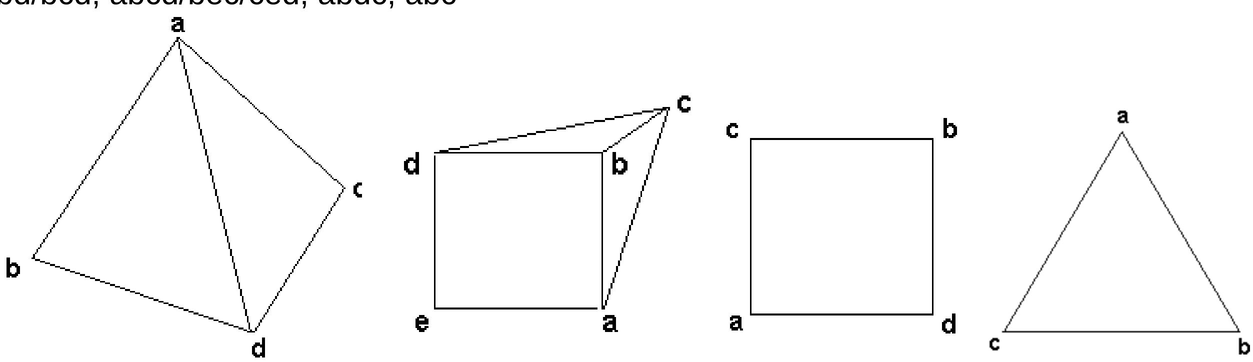

Fig. 8. . Knots numbering for prisms views for a new object.

bda/dca; eabd/acb/bcd; adbc; cba

3. CONCLUSIONS

The formal description of the objects represents an abstract representation of these objects [1],[2],[3],[4]. From an informatics methods viewpoint, the algorithm utilizes chain functions and does not require complicated analytical geometry computing, which leads to a relatively short time of data processing.

REFERENCES

[1] A. G. Hamilton, *Logic for Mathematicians*, Cambridge University Press, 1978, ISBN 0 521 21838 1.
[2] Michael A. Harrison: *Introduction to Formal Language Theory*, Addison-Wesley, 1978
[3] Grzegorz Rozenberg, Arto Salomaa, *Handbook of Formal Languages: Volume I-III*, Springer, 1997, ISBN 3 540 61486 9
[4] Seymour Ginsburg, *Algebraic and automata theoretic properties of formal languages*, North-Holland, 1975, ISBN 0 7204 2506 9