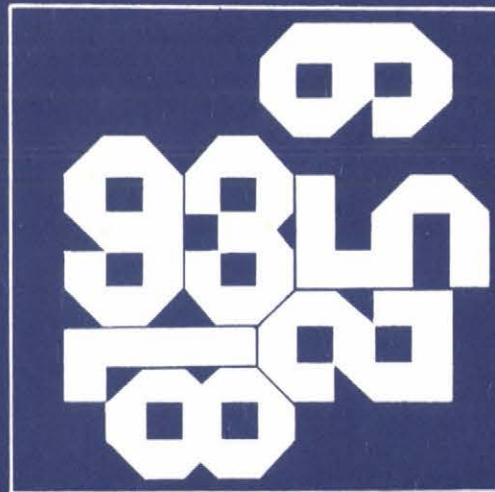


1975 JUL 07



MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADÉMIA  
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET

T. Vámos - Z. Vassy

LIMITED AND PRAGMATIC ROBOT INTELLIGENCE

L. Mérő - Z. Vassy

A SIMPLIFIED AND FASTENED VERSION OF THE HUECKEL OPERATOR FOR  
FINDING OPTIMAL EDGES IN PICTURES

Галло В.

ПРОГРАММА ДЛЯ РАСПОЗНАВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБРАЗОВ, ОСНОВАННАЯ  
НА ЛИНГВИСТИЧЕСКОМ МЕТОДЕ ОПИСАНИЯ И АНАЛИЗА ГЕОМЕТРИЧЕСКИХ  
СТРУКТУР

A kiadásért felelős

Arató Mátyás

Készült az Országos Műszaki Könyvtár és Dokumentációs Központ házi  
sokszorosítójában  
F.v.: Janoch Gyula

C o n t a n t s

1. LIMITED AND PRAGMATIC ROBOT INTELLIGENCE .....	5
2. A SIMPLIFIED AND FASTENED VERSION OF THE HUESKEL OPERATOR FOR FINDING OPTIMAL EDGES IN PICTURES .....	17
3. ПРОГРАММА ДЛЯ РАСПОЗНАВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБРАЗОВ, ОСНО- ВАННАЯ НА ЛИНГВИСТИЧЕСКОМ МЕТОДЕ ОПИСАНИЯ И АНАЛИЗА ГЕОМЕТ- РИЧЕСКИХ СТРУКТУР .....	31



LIMITED AND PRAGMATIC ROBOT INTELLIGENCE

T. Vámos - Z. Vassy



- The intelligent problems for AI should not be consequently artificial -

## 1. INTRODUCTION

In this paper we would like to outline some considerations of our industrial robot experiment reported earlier /1/, /2/. The first basic philosophical issue is the quality of intelligence, which should be contributed to the robot-manipulator. That points to the question how deep and how broad the problem-solver should act. The latter is defined by the tasks:

- pattern recognition from a limited number of objects, which patterns are somehow taught before, but various positions and noise complicate the situation;
- manipulation strategy in a changing environment, i. e. the ability of adaptation to various grasping, transfer and putting or fixing conditions.

The depth of the intelligence is defined by that part of intellectual ability which is required from an unskilled worker who is drilled for each subsequent task. I. e. it /he/ has to fulfil the two problem-solving tasks, mentioned above in all "regular" cases and can communicate with a higher standing capacity /more powerful computer, operator, operations manager with programming capability/, if the problem is above the reach of its /his/ own.

This formulation of AI, viz. the intelligence related to fixed computational potentials and not to the overall power of the usual methods, is a very useful and needed one, the basis of practical decision making for a robot in some practical applications /e. g. in workshop for material handling or assembling what we are aiming at/. The financial and technical limitations of minicomputers should be taken into serious consideration.

The research task - over a certain half practical, half demonstrating and more for problem raising and proving experimental device as a paradigm - is some approximation of this correlation - computing and problem-solving power using all possible and useful mixture of AI methods developed before and having our own vernacular which arises from this certain approach. An intelligent division of labour yields a statistical orientating figure for each kind of problem-solving task, where the reasonable financial and technical limits of the realization with commercial or advanced means of the development period are, what the actual optimum of man-machine cooperation is.



## 2. THE HARDWARE SYSTEM

This is a very common one with some specific features. The TV input is found too noisy for regular environment and industrial tasks. Parallel with some improvements on TV input, which may overcome these difficulties, we concentrate now on a laser-deflector input operating on scanning electron-microscope principle. The acoustic-crystal-deflector is the Institute's development also with a minimum 600 points resolution. The arrangement of the photodetectors solves some 3D problems too.

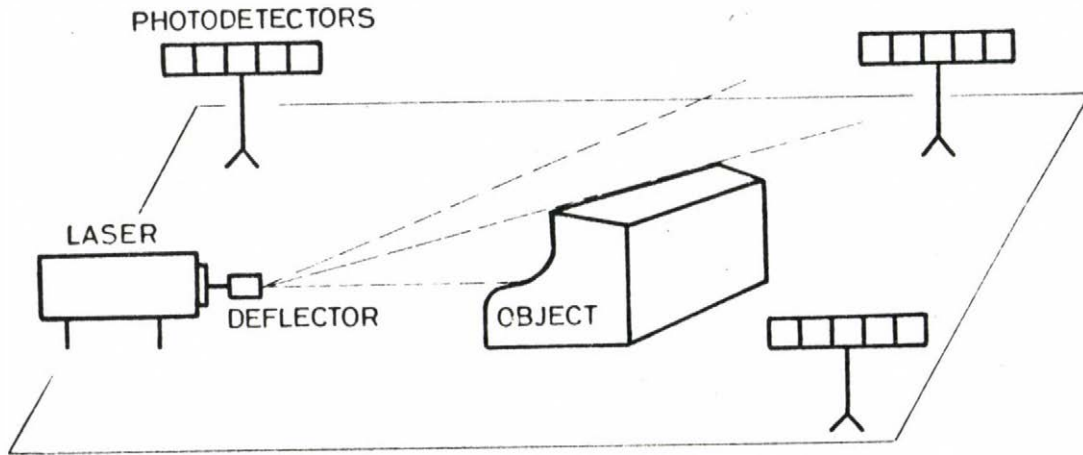


Figure 1

Interface and 16K, 16 bit minicomputer equipped with minidisc /about 50.000 \$ configuration/ and with a display for man-machine communication are the digital facilities. A moderately large computer serves as high capacity background.

We actually use a three degrees of freedom hand hanging from skate, moving in two orthogonal directions, 700 mm above the 1000x630 mm table. One additional freedom is attached to the hand for grasping. We intend to modify this hand in order to perform more complicate manipulations and to make the hand suitable for two-hand operations /see paragraph 5/.

The fingers are equipped with tactile sensors and they can grasp various tools in order to perform assembly tasks.

## 3. THE SOFTWARE SYSTEM

Some basic features of the system are:

- 3.1. The maximum use of the results of the leading groups on these fields /3/, /4/, /5/, /6/, /7/ etc./, as these can specially be mutatis mutandis applied to our task. The whole system is working under a special real-time

monitor disc operating one, this controls the four major components:

- a library with all the pattern recognition, actuating strategy, control routines;
- a dictionary containing the semantic information about the "world" /see /8//, to be operated;
- display-dialogue for man-machine communication;
- control of the devices /eye, hand/.

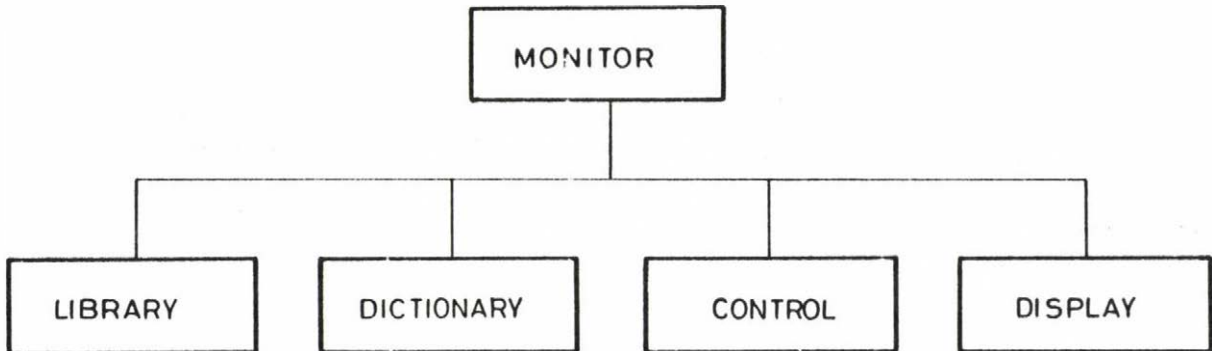


Figure 2

3.2. A multilevel approach is used to economize the computer configuration.

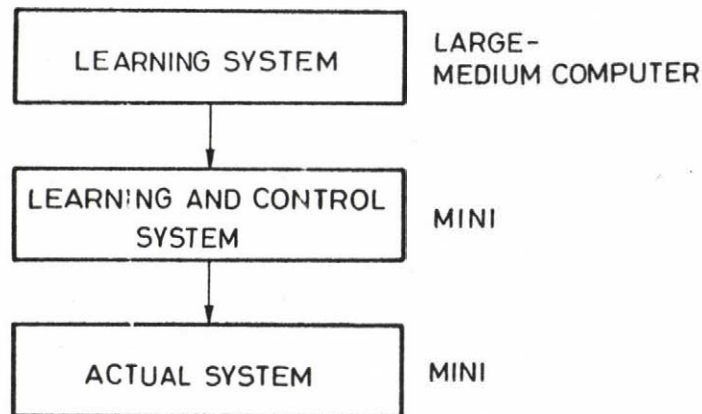


Figure 3

The highest level uses a CDC 3300 type medium large-scale computer, this constructs the grammars of pattern recognition in the learning phase. The highest level can also be used for more complicated problems, although the abandoning of this level is also considered as the minicomputer facilities are growing up to the medium level.

More important are the two lower levels, both realized on the mini. During the learning period all software-tools and "world"-information are needed and this is just the limit of the minicomputer-disc system. /Per defini-

tionem, because if a capacity over this is inevitable, we have to use the first, highest level anyway. As mentioned in the introduction, the study of the limits of this level is just a primary goal of our work./ This is the medium level. For the lowest level a special software-tool, the Generator of Actual System selects from the library, dictionary and control strictly only those routines, lists and data which are really needed for the actual operation. That means e. g. that only two or three objects' information, one pattern recognition or control program should be in the foreground, all other available capacity of the computer can be used either for fast operation on the actual task or to handle other jobs, e. g. the control of a machine-tool. Such an arrangement not only helps in the optimal use of the hardware and makes the project economically more feasible in the future, but a faster operation can also be reached due to the radical decrease of the search processes.

- 3.3. As mentioned above, the whole system is dogmatically pragmatical. That means, all methods of pattern recognition and transfer operation strategy should be possible to be included, the only criterion is, whether the method is more or less practical for the given purpose. This reflects the confidence of the authors, there is no exclusive salvation method in AI, as the human brain similarly uses a pragmatic approach. The pragma is the real life itself, all structures of reasoning distilled and reflected in our brains, are some approximations of the real ones. We are therefore not anxious having too many methods of AI at disposal, as a result of previous work, but too few for the effective solution of the real problem manifold.

The system should therefore be open for various extensions, depending on the task and other circumstances; this extensions should also be written by advanced users. The basic structure of the software has to offer this flexibility and openness.

This philosophy is responsible also for the fact that each progress report of our project contains some parts, which cannot be considered as the logical continuation of the prior ones.

- 3.4. Following the philosophy of the previous point, a new feature extraction routine is added, a very simple template matching with relatively low noise sensitivity. This method is reported in a separate paper /9/. The original heuristic grammatical method /12/ is helped as a preprocessor by a brute force one /2/. Another method called "bleared eye", reported last year /10/, should also join the pattern recognition armory of the system.

All this shall not diminish the fact that the system is essentially based on the grammatical method, especially that of Evans /11/ and the semantics developed by Winograd /8/ and his predecessors. The grammatical method, a mathematical-



computational realization of the use of graphs is the only one, which - as to our knowledge - overcomes in some sense the basic inadequacies of the linear-serial machine representation and can handle a bucket of background information parallel, without the regular foolish processing of long lists. The other superb advantage is the realization of things' structure in a grammatical structure; this offers the possibility of a direct search for a certain part of a known whole. E. g. to grasp a tool, we have to identify the tool itself and after that only those surfaces, which have a role in the grasping operation. All other information can be neglected.

#### 4. SOME MORE DETAILS

In the following a short picture is given about the actual system. Programming is based on the command sentences which yields the backbone of the operation.

4.1. Every command sentence is opened by the word <please>, the sentence has a fixed word order to make the parsing work less complicated.

<sentence>: = <verb><noun><relation><noun> | <verb><noun>

<verb>: = <find> | <move> | <check> | <remember>

find - recognition of the pattern called by <noun>

check - check a prescribed position, situation before action

remember - store a movement or a pattern or a situation

<move>: = <catch> | <transpose> | <position> | <operate>

catch - catch object

transpose - transpose object from its initial <start> place to its <goal> characterized by <rel><noun>

position - a variant of <transpose>, fine adjustment at the goal

operate - execute an operation on the object, e. g. turn a screw in the hole

<noun>: = <object> | <tool> | <hands>

<object>: = <object> | <object part> | <atom> | <object> | <object part>

<object><atom> | <object part><atom> | <object><object part>

<atom>

object and object parts /atoms/, as in /1/, are the hierarchical levels of pattern structure, see Fig. 4; all pattern elements should be understood with their standard attributes and internal relations. This is our device to define joints or other relevant points, lines or surfaces neglecting those, which are irrelevant from the point of view of the actual operation /previous section, point 4./

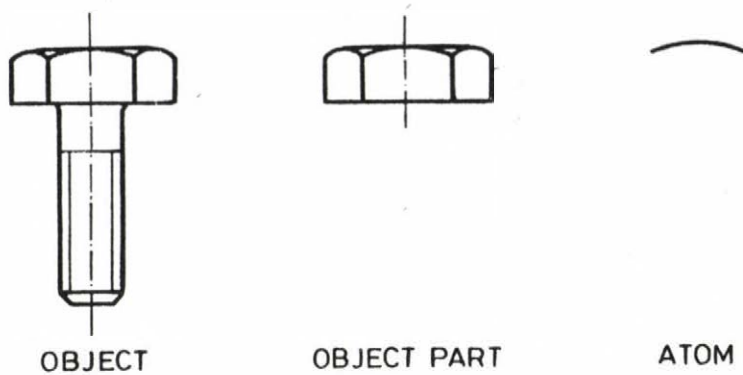


Figure 4

object, tool and hand have a lot of similarities, but to define some differences, is advantageous especially because of the uniformity of the dictionary contents.

<relation>: = <on>|<relative to>|<angle>|<distance><cca>|<logical>

<on> ≡ <in> ≡ <out> - can be understood by the computer on the base of the verb and the object

relative to - is a compound relation, which must be followed by some more definition as:

angle - defines an angle

distance - defines a distance

cca - defines the measure of the tolerance.

Example:

transpose, the biggest gear, relative to cca 0,1 mm dist 5 mm, gearbox second shaft axis endsurface centre,

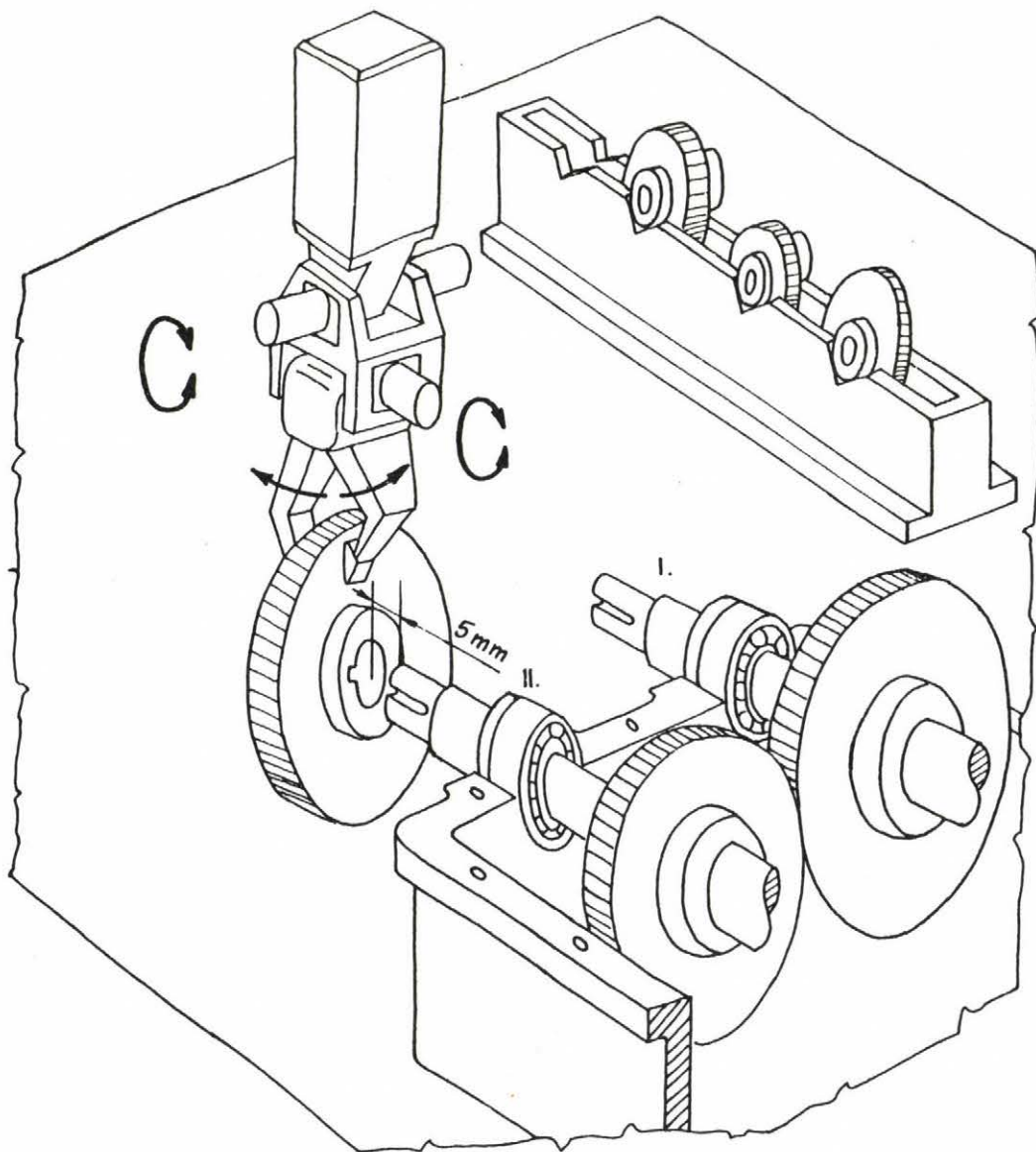


Figure 5

'Transpose the biggest gear relative to  
cca 0,1 mm dist 5 mm gearbox second  
shaft axis endsurface centre'

4.2. Some other instructions:

cond → if...then...else  
repeat repeat /n/ - repeat n-times  
repeat/cond/ - repeat under the condition  
please command sentence start  
begin program start  
wait as usual  
stop  
end

There are two auxiliary instructions, which help the checking of the initial and final place of action, e. g. on the display:

<start>: = <starting object place>  
calculated from the dictionary  
<goal> : = <relation><second object of the sentence>

Instruction words for the dialogue are also parts of the system.

4.3. The dictionary contains the following information for every object /every certain object is a different one for the system as in Winograd /8//:

object\_name  
related\_objects /on the special scene/  
operations  
methods /order of pattern recognition methods recommended/  
descriptions /in the order of the methods one object can have a few descriptions by the same method also because e. g. of the different possible views/  
grasp /indication to the grasping surfaces in the object's grammar/  
history /last place, other previous experiences/  
remarks

4.4. Some examples of the instruction words' further resolution:

find\_object - looks in the dictionary, finds heading of the object, retrieves the recommended method of pattern recognition and the description/s/ belonging to the method. Looks further in the dictionary to find in the history the recommended place of the search. Calls the routine of the feature extraction from the library, executes it and after identification rewrites the actual data in the history /dictionary/.

catch\_object - if a find operation has preceded, looks in the history /dictionary/ to find the place of the object, and also in the dictionary the surface of the grasp. Calls the grasp routine from the library which calculates the absolute grasping coordinates on the basis of the grasp and object place /relative vs. absolute/. Controls the arm /see later/



to execute the grasping operation. Puts a record of the action in the history.

4.5. Library routine examples:

<u>method</u>	brute force
	grammar
	bleareyed
<u>operation</u>	put
	slip
	turn
<u>remark</u>	check condition
	control input /e.g. camera/
	change method
<u>grasp</u>	place /grasping place/
	rel to obj /direction of the grasp/
	force
	transp hand /prescription for the hand position/

5. CONTROL STRATEGIES /FURTHER DEVELOPMENTS/

The previous details are under perfection. This part is only outlined and only the basic philosophical considerations are mentioned, which are always the typical marks of the beginning and the end of a project.

Not one, but a few strategies are provided. That means, that the operator should be able to choose the most simple and apt one for the given case. These are:

- programmed route, with check at or before the goal;
- learned route, otherwise the same;
- learned and/or programmed start /grasp/ and goal, otherwise controlled automatically
  - through the shortest path
  - through single coordinate axis with some preferred sequence, bypass of obstacles, by visual control and following free coordinate axis;
- intelligent start and goal. Calculation of the final grasp situation at the goal, to prepare the operation, if this is not possible at the start, intermediate position. Calculation of obstacles by covering them with more simple forbidden areas /possibly along the coordinate axis, surfaces/. Calculation of momenta joining surfaces, axis.
- Control heuristics. A combination of the above possibilities by generating a control hypothesis, checking it by simulation /e.g. visualization on display/, automatic or manual correction.

We are aware of the fact, that many results preceded our work also in this

field, but a thorough elaboration helps us in problem-understanding and maybe leads us to some new practical and particular ideas.

#### ABSTRACT

Depth of intelligence is discussed which can be contributed to a minicomputer controlled robot-manipulator. Pattern recognition and arm control strategy are the two major elements of the system which has different levels of intelligent action. A learning system with a medium-computer as background, a system of adaptation on the mini and very restricted actual working one complete the hierarchy. A new template matching and other methods lead to feature extraction; recognition is a heuristic grammatical variant, semantics based on Winograd's work complete the system.

#### REFERENCES

- /1/ T.Vámos-Z.Vassy: Industrial Pattern Recognition Experiment - A Syntax Aided Approach. Proc. of the First International Joint Conference on Pattern Recognition, Washington, Oct. 30 - Nov. 1, 1973. IEEE Publ. 73-CHO-821-9C, pp. 445-452
- /2/ T.Vámos-Z.Vassy: The Budapest Robot - Pragmatic Intelligence. Vith IFAC Congress, Boston, Aug. 1975
- /3/ Winston, P.H.: The MIT Robot. Machine Intelligence, 7, pp. 431-463. Meltzler and D. Michie eds. American Elsevier Publ. Comp., 1972
- /4/ Ambler, A.P.: A Versatile Computer Controlled Assembly System. 3rd International Conference on AI, Stanford Univ., Stanford, Aug. 1973
- /5/ Barrow, H.G.-Crawford, G.F.: The Mark 1.5 Edinburgh Robot Facility as ref. /3/, pp. 465-480
- /6/ Fikes, R.E.-Hart, P.E.-Nilsson, N.J.: Some New Directions in Robot Problem Solving. As ref. /3/, pp. 405-429
- /7/ Yoshiaki Tsuboi-Eiichi Tsuda: A Minicomputer Controlled Industrial Robot with Optical Sensor in Gripper. Proc. of the 3rd International Symposium on Industrial Robots, 29-31 May, Zurich, 1973, pp. 343-355
- /8/ T. Winograd: Procedures as Representation for Data in a Computer Program for Understanding Natural Language. Techn. Report AIAI-TR-17, MIT, Cambridge, MA, 1971
- /9/ Z.Vassy-L.Mérő: A Simplified and Fastened Version of the Hueckel operator for finding optimal edges in pictures. 4th International Joint Conference on Artificial Intelligence, Tbilisi, 1975

- /10/ L. Nemes: Pattern Recognition Method for Industrial Robots by  
Extracting the Main Features of Objects. 4th International  
Symposium on Industrial Robots, Tokyo, 19-21. November,  
1974
- /11/ T.G.Evans: A Grammar-Controlled Pattern Analyzer. IFIP Congress, 1968,  
Appl. 3, North Holland, Amsterdam 1968
- /12/ V. Galló: Program for recognition of geometrical patterns, based  
upon description and analysis of geometrical structures by  
grammatical method /in Russian/. 4th International Joint  
conference on Artificial Intelligence, Tbilisi, September  
1975

A SIMPLIFIED AND FASTENED VERSION OF THE HUECKEL OPERATOR FOR  
FINDING OPTIMAL EDGES IN PICTURES

L. Mérő - Z. Vassy



## INTRODUCTION

Hueckel's operator/s/ for edge detection in pictures is one of the best tools which have been invented for this kind of tasks. In very noisy pictures its high effectiveness can be comparable only with the global optimization procedures e. g. by Montanari and Martelli /8/, /9/, but without their drawbacks in processing time and memory requirement. The Hueckel operator processes a picture block containing 52 points on PDP-10 in 10 msec, e. g. if it scanned for example a 256 by 256 point picture so that the blocks overlap by one thirds of each other, the processing time would be about 40 seconds. Of course, by means of sophisticated edge-following methods this time can be reduced significantly, thus the applicability of this operator is not beyond the possibilities to process a real picture. This 10 msec is, however, rather large to look for some modifications based on its basic idea to fasten the operation, even by possible simplifications which keep the excellent properties of the original one.

In this paper we present a method aiming at this by reinterpreting the principle of the Hueckel operator and giving a modification of it based on our interpretation.

## HUECKEL'S METHOD IN ITS ORIGINAL FORM

The original interpretation of the Hueckel operator can be summarized as follows:

- a/ We seek for a function  $S$  on a  $D$  area in the input picture, which function has a constant  $b$  value in either and another constant  $b+d$  value in other side of a certain line, the equation of which is
$$cx + sy = r$$
- b/ This function  $S$  is determined by the least squares method, i. e. we minimize the sum of the squares of differences point by point inside of  $D$  between  $S$  and the input  $P(x,y)$  grey level function;
- c/ For the sake of easy computation we set  $S$  as an infinite Fourier series of some  $H_1$  base functions;
- d/ The infinite set of these  $H_1$  functions is approximated by a finite set of its lowest frequency members  $H_0 \dots H_7$ ;
- e/ We solve the minimization task in this new space. The parameters of  $S$  characterize the parameters of the line segment found in  $D$ .

Hueckel extended this method for finding not only edges but lines as well /6/, but this extended version will not be treated here, as we are interested in edge detection only. The sketches for  $H_0 \dots H_7$  are seen in Figure 1; + and - marks denote the sign of the functions, the lines inside their area  $D$  denote the zero

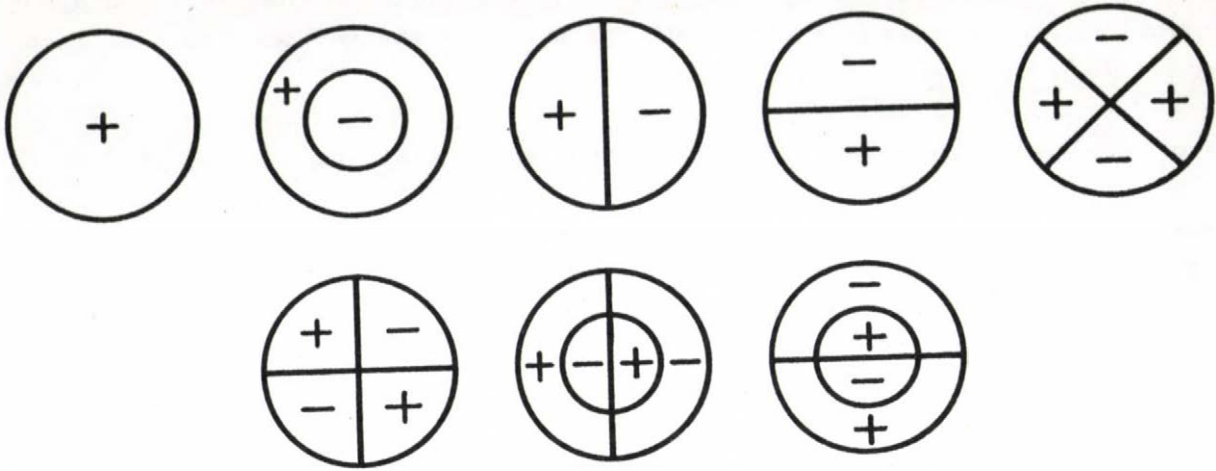


Figure 1

crossings. We also do not present other details, though some of them /e. g. the so-called "solution theorem"/ are also very nice even theoretically.

In the followings we will prove that the Hueckel operator can be viewed as a kind of template matching methods. Before this we are to briefly survey some approaches to template matching.

TEMPLATE MATCHING

Let be given two function,  $f(x,y)$  and  $g(x,y)$  on a  $G$  two-dimensional area, and we seek  $f(x,y)$  in  $g(x,y)$ , i. e. the area in  $G$  where  $g$  identical to  $f$ . In this case we can compute the following convolution:

$$\int_{H G} f(x,y) g(x,y) dx dy = f \bullet g$$

and the sought area  $H'$  is just where this convolution takes the maximum value in  $G$ .  $f(x,y)$  is called template function or simply template /13/.

If  $f(x,y)$  is edge-shaped, it can be used to find edges in pictures. Rosenfeld /14/, Rosenfeld and Thurston /15/, /16/ and others/7/, /10/, /11/, /12/ have worked out several versions of this method and made extensive empirical investigations with them. For example, perhaps the simplest templates for two-dimensional picture are those of Shirai /17/ sketched in Figure 2. The first of

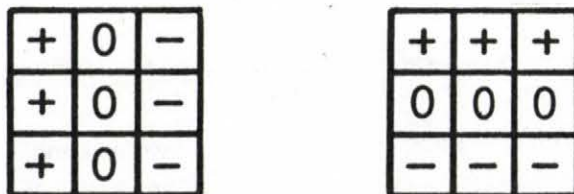


Figure 2



them shows a maximum if it lies exactly on a vertical and the second on a horizontal edge. For making the method less noise-sensitive, one can take into account larger areas e. g. 5x5 or 8x8 etc.

Since Hubel and Wiesel /4/ pointed out that the visual information processing system in mammals operates in a way very close to this method, at least for the local edge and spot extraction on the retina, template matching has a certain theoretical support. Moreover, just this physiological achievement led Deutsch to develop his template called SLEN /Short Line Extractor Neuron /1// used for recognition of numerical characters. For real-time processing of real pictures, however, these methods do not produce so effective results as other ones being though ad hoc without any theoretical suggestions. In the case of edge detection, if the shape and orientation of the template are fixed, the probability of good matching is little because the output value of the convolution ceases very sharply either with the angle between the template edge and the real one or the curvature difference of them /2/. From this two main shortcomings follow. First, the system would not find certain, unlucily oriented edges even from the line-shaped ones. To surmount this difficulty, Hoderman and Kazmierczak /3/ have presented a method with parallel application of several templates to the same picture area around a point. They took care also of noise elimination by averaging, and their output picture after this multiple template operation looked quite good in quality, but even their method could not avoid the following - second - difficulty: in the case of fixed templates the large majority of the required computer operations falls into unsuccessful tryings and, the whole processing time may be too long, especially if the picture is not "angular" enough. The main reason of these shortcomings is obviously the rigidity of the templates applied. In contrast to e. g. Holderman and Kazmierczak's eight templates, we would need only one of them; but that would be the best matching one. The question is, how to find just this "optimal template", i. e. how to solve the opposite problem to that of template matching: fitting the template to a fixed picture area and not the picture area to a/some fixed template/s/. Let D be this area of the picture,  $\tau$  the set of all possible templates, and  $F/x,y/$  the input two-dimensional grey-level function, then we seek  $T/x,y/ \in \tau$  so that

$$\int_D T/x,y/ F/x,y/ dx dy = T \otimes F = \max. \quad /1/$$

In the next section we will point out that Hueckel's operator solves just this task.

#### THE HUECKEL OPERATOR AS A KIND OF TEMPLATE MATCHING METHODS

We will first need to recall some of Hueckel's notations. Let the grey values of the input picture on a circle-shaped area D be  $F/x,y/$ , the sought two-dimen-



sional step function be

$$S/x,y,c,s,r,b,d/ = \begin{cases} b & \text{if } cx + sy > r \\ b+d & \text{if } cx + sy < r \\ 0 & \text{otherwise.} \end{cases}$$

The base functions chosen in the Hilbert space of all the functions on D be  $H_0/x,y/ \dots H_7/x,y/$ .

With these notations the functional equation to be solved, is:

$$\int_D (F/x,y/ - S/x,y,c,s,r,d,b/)^2 dx dy = \min. \quad /2/$$

The Fourier coefficients for F and S are:

$$f_1 = \int_D F/x,y/ H_1/x,y/ dx dy \quad /3/$$

$$s_1/c,s,r,d,b/ = \int_D S/x,y,c,s,r,d,b/ H_1/x,y/ dx dy \quad /4/$$

$$\lambda/c,s,r,b,d/ = \sum_{i=0}^7 (f_i - s_i/c,s,r,b,d/)^2 = \min. \quad /5/$$

Hueckel's operator solves this functional equation<sup>x</sup>. Now we will prove that /5/ can be rewritten to the form of /1/ in the previous section. For this we need a lemma:

Lemma: Let

$$\phi_i /i=1, \dots, N/ \text{ and } \psi_i/p/ /i=1, \dots, N/ \text{ so that}$$

$$\sum_{i=1}^N \psi_i/p/ = \text{const.}$$

where  $p$  a vector of arbitrary dimensions.

If

$$\mu/p/ = \sum_{i=1}^N (\phi_i - \psi_i/p/)^2 \quad \text{and}$$

$$\nu/p/ = \sum_{i=1}^N \phi_i \psi_i/p/$$

then if for some values of  $p$

$$\mu/p/ = \min.$$

then for the same values of  $p$

$$\nu/p/ = \max.$$

The proof is given in Appendix A.

According to this lemma /5/ can be rewritten as

$$\lambda'/c,s,r,b,d/ = \sum_{i=0}^7 s_i/c,s,r,b,d/ f_i = \max.$$

Substituting  $f_1$  from /3/:

$$\lambda'/c,s,r,b,d/ = \sum_{i=0}^7 s_i/c,s,r,b,d/ \cdot \int_D F/x,y/ H_1/x,y/ dx dy = \max.$$

<sup>x</sup>Here we neglect Hueckel's heuristic weighting coefficients as they have no relevance in the problem considered.

With some trivial transcriptions:

$$\lambda' /c,s,r,b,d/ = \int_D F/x,y/ \left[ \sum_{i=0}^7 s_i /c,s,r,b,d/ H_i /x,y/ \right] dx dy = \max.$$

The expression

$$T/x,y,c,s,r,b,d/ = \sum_{i=0}^7 s_i /c,s,r,b,d/ H_i /x,y/$$

is a template by its role in /6/, as the form of /6/ is identical with that of /1/.  $T/x,y,c,s,r,b,d/$  is the optimal template to the  $F/x,y/$  input function constructed as a linear combination from the set of original templates  $H_0 \dots H_7$ .

#### THE POSSIBILITY OF SIMPLIFYING THE HUECKEL OPERATOR

In Hueckel's interpretation the basic aim, i. e. finding an edge, is reached by fitting a two-dimensional function to the input grey-level function in a picture area. Consequently, a relatively large set of both the base functions and edge parameters is required, because it is obvious, that no satisfying degree of fitting could be approached by remarkably simpler base field and parameter vector /containing e. g. two or three functions or values respectively/. If we are, however, considering the method as a template matching one, the fitting problem becomes of no importance and its requirements cannot be taken into account.

First, we may choose a two-dimensional parameter vector  $/c,s/$  instead of  $/c,s,r,b,d/$  as the grey-values of the two sides of the step function are irrelevant, and the exact location of the edge inside the  $D$  area can be computed in a very simple way once we have determined the edge direction. The direction is characterized only by  $c$  and  $s$ , as in the equation of the edge

$$cx + sy = r$$

$c$  and  $s$  are just the direction cosines relative to the  $+x$  axis.

Second, if we need only the edge direction, we may work with only two of Hueckel's original templates  $H_0 \dots H_7$ , because it seems well even intuitively /we will prove it later, of course/ that by the linear combination of  $H_2$  and  $H_3$  the direction of the optimal template can be determined. By the direction of a template we mean the direction of the 0-crossing of a  $H_2$  or  $H_3$  function.

Third, if we do not take care of the mathematical conditions of Fourier function fitting in a Hilbert space - as these conditions determined the choosing of  $H_0 \dots H_7$  in Hueckel's work /Chapter 2/ - the only criterion in choosing our two templates can be the speed of computation which is very crucial in real applications. Thus we can use both constant-valued template functions and a more convenient shape for the  $D$  picture area. The convenience of this shape means also the speed of computing  $c$  and  $s$  or any other quantities characterizing the

direction.

The method we have got to by the considerations above, will be detailed in the following section. It is essentially simpler and faster than Hueckel's one but as it will be seen from the examples presented later, preserves all of its advantages in comparison to other methods. The computation consists of two steps. The first one is the determination of the edge direction by constructing an optimal template from two beginner ones by linear combination. Just like in Hueckel's method, a vast part of the extremization problem is solved analytically and the number of the required arithmetical operations is small. The second step is the determination of the exact location of the edge in D by its two border points. It is solved by some simple logical rules and point reflections.

DESCRIPTION OF OUR EDGE DETECTION OPERATOR

The core of the method is based upon the following elementary theorem:

Theorem: Let D denote the unit square,  $f_1$  be the function being +1 above the subdiagonal of D, and -1 below it,  $f_2$  be the function being +1 above the principal diagonal of D and -1 below it, /see Fig. 3 a, b/ and let f be a function being b+d above a line e in D, and b below this line. /See Fig. 3 c or d./ Then if  $\alpha$  denotes the angle between the subdiagonal of D and e, the following equation will hold:

$$\frac{\int_I f f_2 dx dy}{\int_I f f_1 dx dy} = \text{tg } \alpha$$

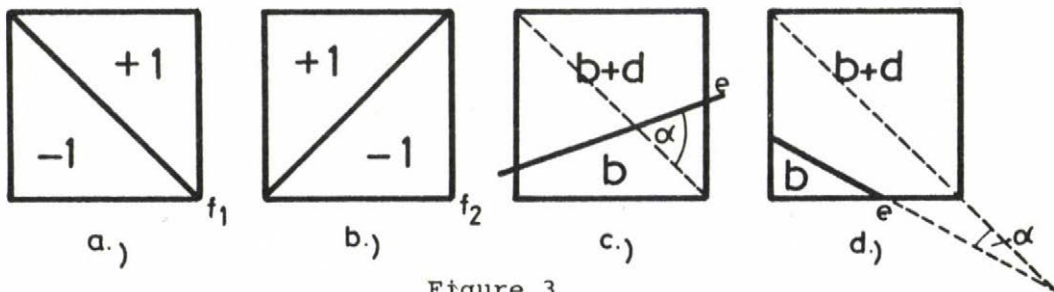


Figure 3

The proof of the theorem will be presented in Appendix B. The odd thing about the theorem is, that the expression at the left side does not depend either on b and d or on the real place of the line but its angle.

By this theorem we are given a very pleasant tool to determine an edge in a little part of the picture. The two integrals can be very easy and fast computed, and because of the integrations this method is of very low sensitivity for noises. /Later we shall show some more considerations about this./



In our work we have chosen  $D$  to be a square consisting of  $8 \times 8$  picture points. Having supposed that there was a straight edge in the picture at  $D$ , using the above mentioned theorem we should determine its place, i. e. its two endpoints in  $D$ .

To begin with we transform the value of the tangent to a better treatable form. We determine from the value of  $\text{tg } \alpha$  that how many picture points are the difference between the right and the left end of a line in  $D$ , the angle of which with the subdiagonal is  $\alpha$ . /It is enough to deal with lines lying in the quarter of the plane shown on Fig. 4, because if the edge should lie in the



Figure 4

other quarter,, then counting with  $|\text{tg}\alpha|$ , after the whole process a reflection of the edge to the principal diagonal of  $D$  - i. e. its endpoints - gives the real edge./

Everything told up to this time, is valid both for pictures with two and for those of more grey levels. Now we shall show the determining process of the real edge at first only for pictures with 2 grey levels and after that we shall show how to generalize that idea for pictures with more grey levels.

In the case with 2 grey levels the sum of the  $l$ 's in  $D$  divided by 8 shows, roughly spoken, that till what a height is  $D$  filled by ones. This means, that e. g. if the edge were horizontal, then the edge should be either the horizontal line at that height, or that reflected to the centre of  $D$  /for  $D$  could be filled by ones from its bottom or from its top/. But however the edge were not horizontal, we should know what the difference is between its left and right endpoints' height, so we could determine the edge. If both the left and right endpoints of this edge were inside of  $D$ , then we should have already only to determine whether this was the real edge or we have still to reflect it to the centre of  $D$ .

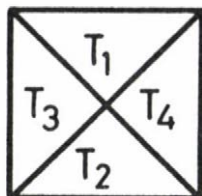
If one of the endpoints of the above received edge were not in  $D$ , it could be sticking out of  $D$  only by 1, 2 or 3 points, because the edge we are working with, now lies already in the plane quarter shown on Fig. 4. But in these cases it is already very easy to determine the real edge.

For pictures with more grey levels we could use the same method if we knew  $b$  and  $d$ , because then the number of the points with greater grey level could be approximated by the following expression:

$$s' = \frac{S - 64b}{d}$$

where  $S$  denotes  $\int_I f dx dy$ , i. e. the sum of the grey levels of all points of  $D$ .

But now using the notations of Fig. 5, let  $T_1, T_2, T_3$  and  $T_4$  denote the sum of



the grey levels of  $f$  /i.e. of the picture/ in these 4 quarters of  $D$ , and  $\frac{\min\{T_i\}}{16}$  resp.  $\frac{\max\{T_i\}}{16}$  approximates well

$b$  resp.  $b+d$ . After this the above described method of finding the edge can be completed in the case with more grey levels, as well.

Figure 5

By this method we have already found the real edge either in  $f$  or reflected to the centre of  $D$ . /See Fig. 6./ To determine which of them is the real edge,

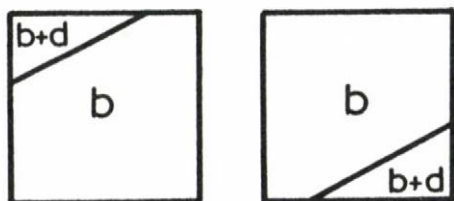
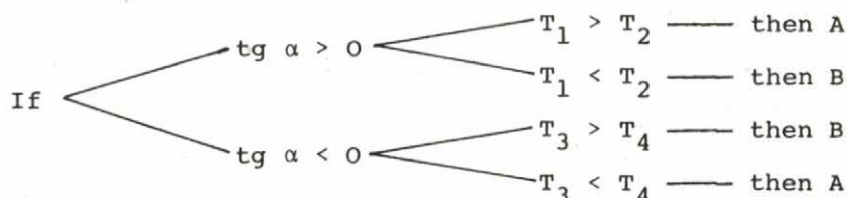


Figure 6

the following table shows a method provided that up to this time we have supposed in the steps that  $D$  was filled by the ones /resp. by  $b+d$ 's/ from its top. The case A means that we have found the real edge, while B means that still we have to reflect our edge to the centre of  $D$ .



This method was implemented by a program on the assembly language of a 16 bit microprogrammed minicomputer of 550 program instructions without any cycles. If there is an edge in  $D$  finding, it needs 250-400 machine operations, i. e. about 500-800  $\mu$ secs. No floating point arithmetics are needed.

About the reliability of the method, rough under-estimations show, that supposing a noise with a binomial distribution with a mean value of 5 error points in a  $8 \times 8$  square, the probability that the difference between the edge determined by this method and the real edge should not be greater than 1 point anywhere, is over 95 %.

Finally we present three examples for the working of the method. /See Fig. 7/

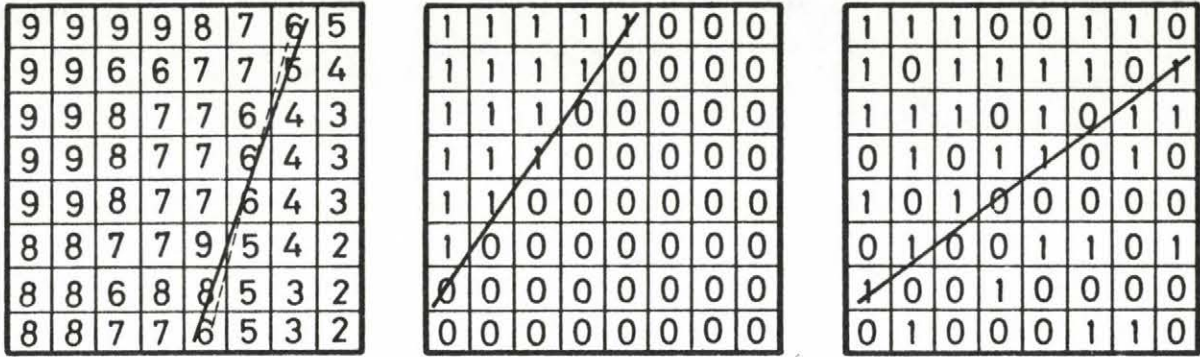


Figure 7

Example 1 is the example showed in Hueckel's article /5/, we have completed it by the corners. The dotted line shows the edge found by Hueckel, the continuous one is the edge found by our method.

APPENDIX A

Proof of the lemma used on page 5.

We will prove that  $\nabla_{\mathbf{p}} \mu / \mathbf{p} / = -\text{const.} \nabla_{\mathbf{p}} \nu / \mathbf{p} /$ .

If it is true, then if

$$\nabla_{\mathbf{p}} \mu / \mathbf{p} / = 0, \text{ then } \nabla_{\mathbf{p}} \nu / \mathbf{p} / = 0, \text{ and } \text{sgn}(\nabla_{\mathbf{p}}^2 \mu / \mathbf{p} /) = -\text{sgn}(\nabla_{\mathbf{p}}^2 \nu / \mathbf{p} /).$$

$$\nabla_{\mathbf{p}} \mu(\mathbf{p}) = \nabla_{\mathbf{p}} \sum_{i=1}^N (\phi_i - \psi_i(\mathbf{p}))^2 = \sum_{i=1}^N -2(\phi_i - \psi_i(\mathbf{p})) \nabla_{\mathbf{p}} \psi_i(\mathbf{p}) = -2 \sum_{i=1}^N \phi_i \nabla_{\mathbf{p}} \psi_i(\mathbf{p}) -$$

$$- \sum_{i=1}^N 2 \psi_i(\mathbf{p}) \nabla_{\mathbf{p}} \psi_i(\mathbf{p}) = -2 \nabla_{\mathbf{p}} \nu(\mathbf{p}) - \nabla_{\mathbf{p}} \left( \sum_{i=1}^N \psi_i^2(\mathbf{p}) \right) = -2 \nabla_{\mathbf{p}} \nu(\mathbf{p})$$

Q.e.d.

APPENDIX B

Let's use the notations of Fig. 8. We shall deal only with the case if  $45^\circ < \alpha < 90^\circ$ , because all other cases can be handled symmetrically. First we show that in the case of the line on Fig. 8, the statement of the theorem is valid. Here the quotient of the two integrals is:

$$\frac{\int_I f_2 f}{\int_I f_1 f} = \frac{T_2 + T_1}{T_2 - T_1}$$

Considering the sides of D to be one, we have

$$y = \text{tg } \beta = \text{tg}(45^\circ - \alpha) = \frac{\text{tg } \alpha - 1}{1 + \text{tg } \alpha}$$

and so

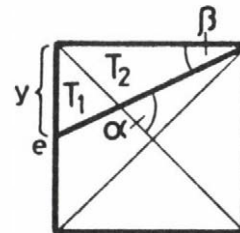


Figure 8



$$\frac{T_2 + T_1}{T_2 - T_1} = \frac{\frac{(1+y)m}{2}}{\frac{(1-y)m}{2}} = \frac{1+y}{1-y} = \operatorname{tg} \alpha$$

Now having this, if  $e'$  lied above the line examined just now, then because of the similarity of all triangles coming in the question /see Fig. 9/, the statement would be true for  $e'$  as well:

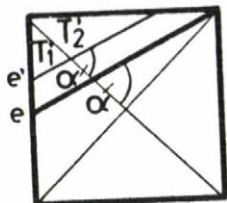


Figure 9

$$T'_1 = kT_1, \quad T'_2 = kT_2,$$

$$\frac{T'_1 + T'_2}{T'_2 - T'_1} = \frac{kT_1 + kT_2}{kT_2 - kT_1} = \frac{T_1 + T_2}{T_2 - T_1} = \operatorname{tg} \alpha.$$

Now if  $e'$  lied between  $e$  and the centre of  $D$ , then using the notations of Fig. 10 because of the similarity of the triangles  $T_1$  and  $T_2UT_3UX$  here and  $T_1$  and  $T_2$  on Fig. 10, we have

$$\frac{T_2 + T + X + T_1}{T_2 + T_3 + X - T_1} = \operatorname{tg} \alpha$$

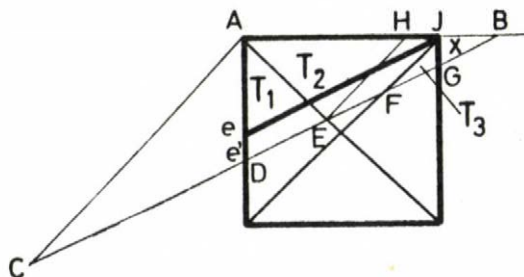


Figure 10

So we have to show only that

$$(\operatorname{tg} \alpha) = \frac{T_2 + T_3 + X + T_1}{T_2 + T_3 + X - T_1} = \frac{T_1 + T_2 - T_3}{T_2 + T_3 - T_1} = \left(\frac{fff_2}{fff_1}\right)$$

Subtracting 1 from both side, taking their reciprocal and adding 1 again, we have

$$\frac{T_2 + T_3 + X}{T_1} = \frac{T_2}{T_1 - T_3} \quad \text{i. e.} \quad \frac{T_1}{T_2 + T_3 + X} = \frac{T_3}{T_3 + X}$$

This means we have to show only

$$\frac{DE}{EB} = \frac{FG}{FB}$$

And this is true for because of the similarities of the triangles  $FJB$  and  $CAB$ ,  $\frac{FG}{FB} = \frac{CD}{CB}$ , so it is enough to show that  $\frac{DE}{EB} = \frac{CD}{CB}$ , i. e. that  $\frac{CD}{DE} = \frac{CB}{EB}$

Now for  $AD$  is the bisector of the angle  $CAE$ , we have:

$$\frac{CO}{DE} = \frac{CA}{AE}, \text{ and for the similarity of the triangles } CAB \text{ and } EHB \quad \frac{CB}{EB} = \frac{CA}{EH}$$



and for the triangle AEH is isocoles  $AE=EH$ , and the theorem is proved in this case, too. If  $e$  were below the centre of  $D$ , the same proof could be presented for negative  $d$ . /We have nowhere made use of being  $d$  positive./ So the whole theorem is proved.

ABSTRACT A simplified version of the Hueckel operator was worked out which preserves all the advantages of the original one, but the running time of which is considerably shorter. Its basic idea is the reinterpretation of the Hueckel operator as a template matching method. This reinterpretation gives the possibility of some theoretical and computational simplifications. As a result, an optimal edge-shaped template is constructed by linear combination of two suitably chosen original ones. The parameters of this optimal template characterize the edge segment found in a picture area.

#### REFERENCES

- /1/ Deutsch, S.: Conjectures on mammalian neuron networks for visual pattern recognition. IEEE Trans. System Science and Cybernetics, SSC-2, pp. 81-85 /1966/
- /2/ Hawkins, J.K.: Photographic techniques for extracting image shapes. Photographic Science and Engineering, 8, pp. 329-335 /1964/
- /3/ Holdermann, F.; Kazmierczak, H.: Preprocessing of grey-scale pictures. Computer Graphics and Image Processing, 1, pp. 66-80 /1972/
- /4/ Hubel, D.H.; Wiesel, T.N.: Receptive fields, binocular interaction, and functional architecture of the cat's visual cortex. J. Physiol. 160, pp. 106-154 /1962/
- /5/ Hueckel, M.H.: An operator which locates edges in digitized pictures. Journal of ACM, 18, pp. 113-125 /1971/
- /6/ Hueckel, M.H.: A local visual operator which recognized edges and lines. Journal of ACM, 20, pp. 634-647 /1973/
- /7/ Levine, M.D.: Feature extraction: a survey. Proc. IEEE, 57, pp. 1391-1407 /1969/
- /8/ Martelli, A.: Edge detection using heuristic search methods. Computer Graphics and Image Processing, 1, pp. 169-182 /1972/
- /9/ Montanari, U.: On the optimal detection of curves in noisy pictures. Comm. ACM, 14, pp. 335-345 /1971/
- /10/ McLeod, I.D.G.: Comments on "Techniques for edge detection". Proc. IEEE, 60, pp. 1344 /1972/
- /11/ Nadler, M.: An analog-digital character recognition system. IEEE Trans. on
- /12/ Nilsson, N.J.: Adaptive pattern recognition: a survey. 1966, Bionics Symp. Dayton, 1966
- /13/ Rosenfeld, A.: Picture processing by computer. Technical report 68-71, Univ. of Maryland, Computer Science Centre, College Park, Maryland /1968/

- /14/ Rosenfeld, A.: A nonlinear edge detection technique. Proc. IEEE, 58, pp. 814-816 /1970/
- /15/ Rosenfeld, A.; Thurston, M.: Edge and curve detection for visual scene analysis. IEEE Trans. on Computers, C-20, pp. 562-569/1971/
- /16/ Rosenfeld, A.; Thurston, M.; Lee, Y.: Edge and curve detection: further experiments. IEEE Trans. on Computers, C-21, pp. 677-715 /1972/
- /17/ Shirai, Y.: A step toward context-sensitive recognition of irregular objects. Computer Graphics and Image Processing, 2, pp. 298-307 /1973/



ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И  
АВТОМАТИЗАЦИИ ВЕНГЕРСКОЙ АКАДЕМИИ НАУК

ПРОГРАММА ДЛЯ РАСПОЗНАВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБРАЗОВ,  
ОСНОВАННАЯ НА ЛИНГВИСТИЧЕСКОМ МЕТОДЕ ОПИСАНИЯ И АНАЛИЗА  
ГЕОМЕТРИЧЕСКИХ СТРУКТУР

ГАЛЛО В.

## ВВЕДЕНИЕ

Программа является частью математического обеспечения системы для распознавания технических чертежей и объектов, о разработке которой сообщалось в докладе Вамоша Т. и Вашли З. [1,7]. Действующий в настоящее время на ЭВМ CDC 3300 вариант программы применяется при распознавании двумерных изображений, подобных изображенным на рис.1. Для изображений характерно, что они состоят из отрезков

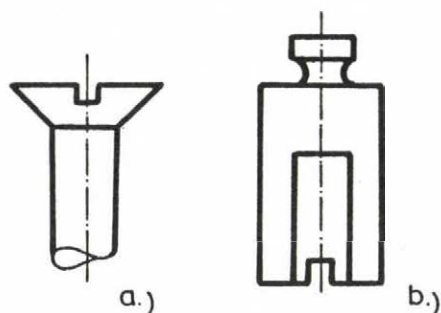


Рис.1

Входом программы является список описаний основных элементов геометрического образа, каковыми являются: отрезок прямой /в дальнейшем: прямая/, дуга и неопределенность малого размера /рис.2/.

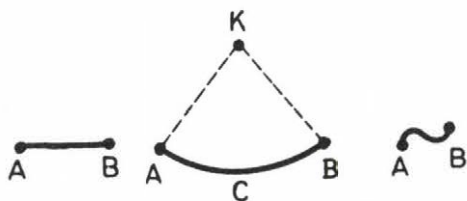


Рис.2

прямых, дуг и - вследствие шума воспринимающего устройства - неопределенностей малого размера. Геометрические соотношения между элементами изображений сложны и разнообразны: совпадение концов отрезков и дуг, параллельность и перпендикулярность линий, равенство и неравенство длин отрезков и кривизны дуг, определенное расположение точек относительно некоторой прямой и т.д.

Описание основного элемента состоит из его идентификатора и количественных характеристик:

координат концов  $A = \begin{matrix} x_A \\ y_A \end{matrix}$  и  $B = \begin{matrix} x_B \\ y_B \end{matrix}$ ,

координат центра  $K = \begin{matrix} x_K \\ y_K \end{matrix}$ ,

координат произвольной точки  $C = \begin{matrix} x_C \\ y_C \end{matrix}$  только для дуги

Входной список или получается в виде двоичной информации от малой машины /СII 10010/, производящей выделение основных элементов изображения реального объекта /о получении и организации этого списка см.[2]/, или же он может являться мануально изготовленным на перфокартах списком описаний основных элементов некоторого идеального геометрического объекта. Происхождение и организация входного списка не принципиальны для работы программы.

Описание объекта может производиться начиная с любого узла и в любом направлении. Каждый основной элемент предполагается описанным только один раз.

Анализ входного списка производится на основе грамматики-структурного описания "знакомых" машине образов. Мы исходили из формализма, предложенного Ивенсом



[3,4], развивая его в смысле возможности четкого описания разнообразных геометрических отношений между элементами образов, экономного конструирования грамматик и программой имплементации.

Терминальные символы грамматики соответствуют основным элементам образов, а нетерминальные символы - более сложным элементам образов. Нетерминальные символы, подобно терминальным, также имеют количественные характеристики. Правила подстановки имеют следующий вид:

Левосторонний нетерминальный символ  $\rightarrow$  | (Конечное множество правосторонних терминальных символов) (Отношения между правосторонними символами или между их количественными характеристиками) (Определение количественных характеристик левостороннего нетерминального символа через количественные характеристики правосторонних символов) (Указание симметричных количественных характеристик<sup>x</sup>).

Допускаемые в системе геометрические отношения и операции получения количественных характеристик нетерминальных символов описываются отдельно и интерпретируются в системе в виде отдельных программных сегментов. Их множество может как угодно расширяться по мере развития системы.

На рис.3 приведено описание характерной детали винта, изображенного на рис.1. Здесь терминальные символы составлены из прописных букв, а нетерминальные - из заглавных. Количественные характеристики левосторонних символов обозначены буквами без индексов, а правосторонних символов - буквами с индексом; индекс указывает порядковый номер соответствующего правостороннего символа в правиле. Правила нумеруются.

Выходом программы при работе с ней, как с самостоятельной программой /например, при испробовании новых грамматик/, является выдача на печать названий распознанных программой на входном списке образов и распечатка описания полного дерева анализа. При использовании программы в системе [1] только окончательные результаты передаются через общее поле данных для других программ.

---

<sup>x</sup>Замечание: Для всех терминальных символов количественные характеристики А и В /координаты концов/ функционально тождественны - условимся говорить, что они "симметричны". Аналогично этому могут быть определены "симметричные" количественные характеристики и для нетерминальных символов. Введение симметричных характеристик позволяет сократить число правил подстановки и допускает произвольность в направлении обхода изображения.

ГРАММАТИКА

Терминальные символы:

arc, его количественные характеристики:  
A, B, C, K (рис.3а)

Нетерминальные символы:

ARC (рис.3а), ARC2 (рис.3б), LOOP (рис.3в), LA (рис.3г)

Правила подстановки:

- 0/ ARC  $\rightarrow$  |(arc)( )( $A=A_1, B=B_1, \varphi=ARCTG(A_1, B_1), K=K_1, C=C_1$ ))(A,B)|  
 1/ ARC2  $\rightarrow$  |(ARC,ARC)( $A_1=A_2, B_1 \neq B_2, K \neq K_2, |\varphi_1 - \varphi_2| > 90^\circ$ )( $A=B_1, B=B_2, K=K_1, L=K_2, C=A_1$ )|  
 2/ LOOP  $\rightarrow$  |(ARC2, ARC)( $A_2=A_2, B_1=B_2, K \neq K_2, L_1 \neq K_2$ )( $A=A_1, B=B_1, C=C_1$ ))(A,B)|  
 3/ LA  $\rightarrow$  |(LOOP, ARC)( $A_1=A_2, B \neq B_2, C_1=B_2$ )|

В системе имеются программные сегменты для проверки геометрических соотношений, обозначенных здесь как:

$$=, \neq, |\varphi_1 - \varphi_2| > 90^\circ$$

и для осуществления операций присваивания, обозначенных здесь как:

$$=, ARCTG$$

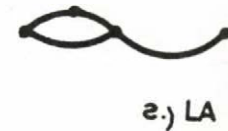
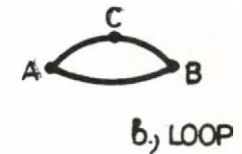
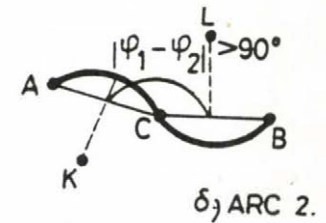
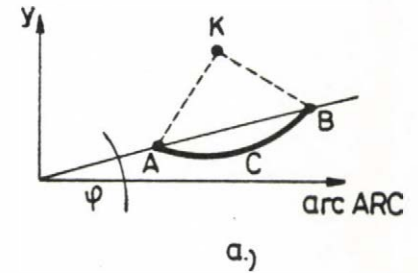


Рис. 3



## ЯЗЫК ПРОГРАММИРОВАНИЯ

Программа написана на языке ФОРТРАН с использованием системы для обработки данных списковой структуры ЛИДИ-72, созданной в Институте Вычислительной техники и Автоматизации ВАН [5,6]. ЛИДИ-72 написана на языке ФОРТРАН.

Система ЛИДИ-72 применима для непосредственного представления в машине любых направленных графов. Вершина может иметь название, при вершине графа может храниться информация в виде линейной последовательности произвольной длины, состоящей из целых чисел, а число дуг может быть произвольным. На рис.4 показано представление вершины графа элементом списковой структуры ЛИДИ-72.

В системе ЛИДИ-72 имеется развитая система программ для обращения со списками: ввод в машину информации, имеющей структуру дерева, описанного в скобочной и некоторых других специальных формах, изменение внутри машины любых графов, условное или безусловное выполнение определяемых пользователем операций на любом заданном подграфе и т.д.

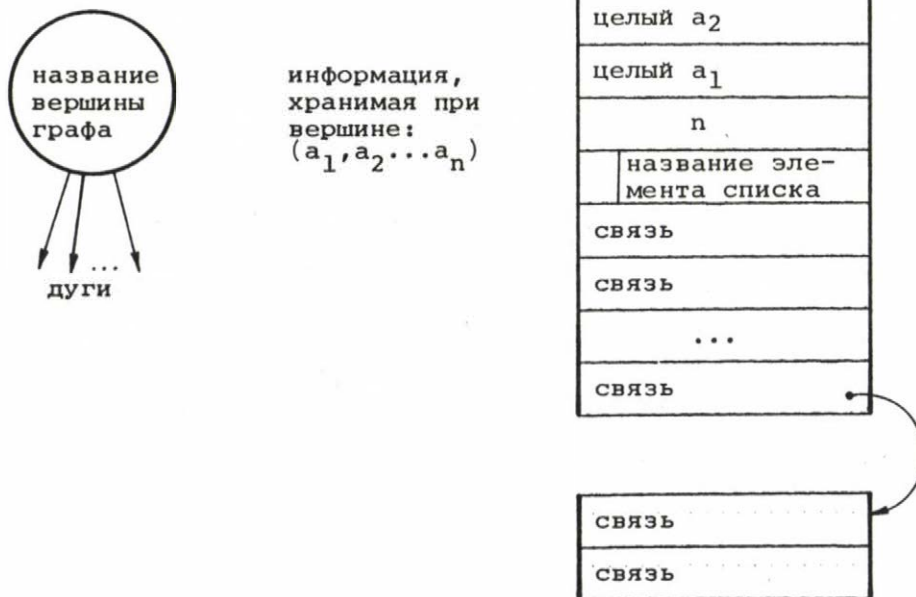


Рис. 4

## ОРГАНИЗАЦИЯ ГРАММАТИЧЕСКОГО АНАЛИЗА

При работе с некоторым определенным, ограниченным классом образов, грамматика, описывающая эти образы, является постоянно используемой и неизменяющейся информацией и организована как часть постоянной структуры данных. На рис.5а показана структура грамматики, и на рис.5б детализирована структура отдельного правила. Более подробное описание построения грамматики дано в Приложении.

Геометрические отношения и операции присвоения значения количественным характеристикам интерпретированы, соответственно, как логические функции и подпрограммы на языке ФОРТРАН. В правилах грамматики указываются только наименования отношений и операций присвоения значения, а также даются описания актуальных параметров. Наименования отношений и операций присваивания собраны в два аналогичных друг другу списка, входящих в постоянную структуру данных. Организация одного из них показана на рис.6. Так называемый переключательный индекс используется для активизации соответственного программного сегмента во время выполнения программы.

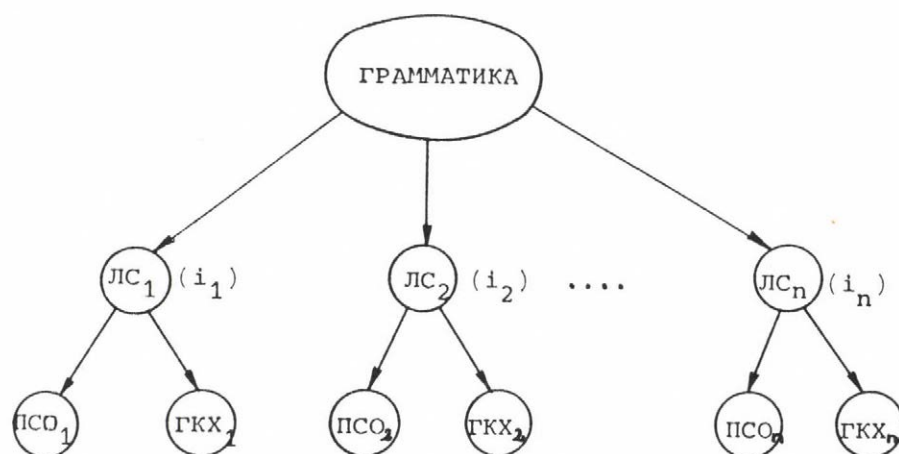


Рис.5а

ЛС - Левосторонний Символ  
*i* - номер правила

ПСО - список Правосторонних Символов  
и Отнoшений  
ГКХ - список для Генерирования Количественных Характеристик Левостороннего символа

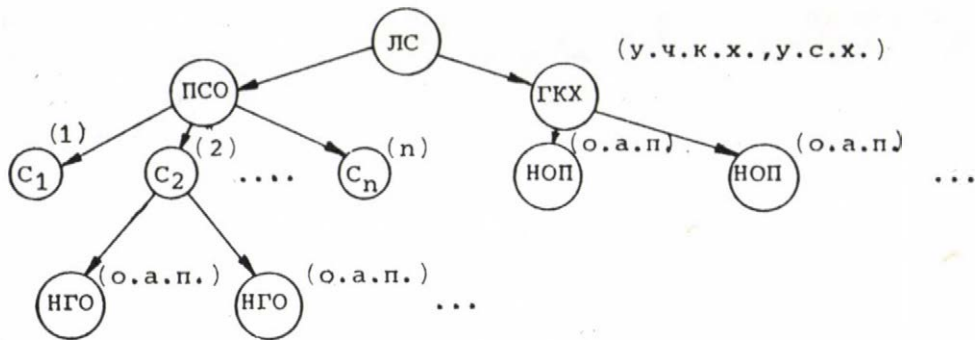


Рис. 56

- |     |  |          |  |
|-----|--|----------|--|
| С   | - Символ                                 | о.а.п.   | - описание актуальных параметров                     |
| НГО | - Наименование Геометрического Отношения | у.ч.к.х. | - указания числа количественных характеристик        |
| НОП | - Наименование Операции Присваивания     | у.с.к.х. | - указание симметричных количественных характеристик |

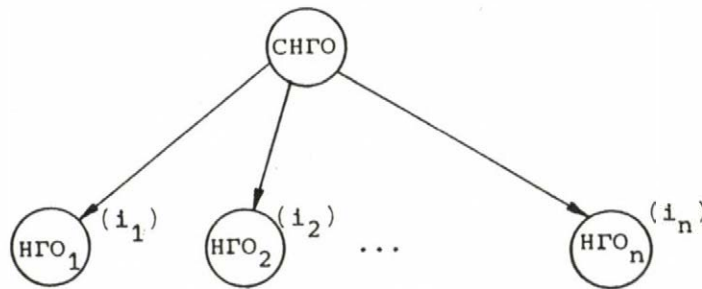


Рис. 6

- СНГО - Список Наименований Геометрических Отношений  
 НГО - Наименование Геометрического Отношения  
 i - переключательный индекс

Анализатор - программа лингвистического анализа - состоит из следующих подпрограмм: подпрограмма анализа согласно заданному правилу, подпрограмма синтеза согласно заданному правилу, подпрограммы для организации работы программных сегментов проверки геометрических отношений и осуществления операций присваивания значений, подпрограмма для организации порядка анализа по правилам грамматики и ряд вспомогательных подпрограмм.

Алгоритм анализа можно охарактеризовать как анализ через синтез. Правила



грамматики пронумерованы по возрастающему порядку иерархии, "снизу-вверх" /по возрастающей сложности описываемых элементов образов/. Алгоритм состоит в обработке информации об актуальном объекте на основе правил грамматики, рассматриваемых в порядке возрастания их номера, в попытках синтезировать левосторонний символ, рассматривая всевозможные комбинации из уже синтезированных символов /синтакстический тип которых совпадает с указанными на правой стороне типами/ и проверяя выполнения геометрических отношений, требуемых данным правилом. При перекрестных рекурсиях между правилами такой "линейный" ход алгоритма нарушается, появляются "петли", однако его основная направленность не меняется. Следует отметить, что большую часть алгоритма составляют подпрограммы анализа по заданному правилу грамматики, на их основе организовать любую стратегию порядка применений правил грамматики.

Количественные характеристики синтезированных при работе алгоритма символов /т.е. распознаваемых программой элементов образа/ записываются в Список Генерированных Количественных Характеристик /СГКХ/, разветвляющийся на подписки по синтакстическим типам. Эти подписки создаются и существуют только во время выполнения программы лингвистического анализа. Выполнение программы начинается введением в оперативную память постоянной структуры данных, затем считыванием входного списка и первичной обработкой его, заключающейся в размещении количественных характеристик основных элементов по трем основным подпискам /прямая, дуга, неопределенность/ в Списке Генерированных Количественных Характеристик /рис.7/, служащим для накопления информации об актуальном образе.

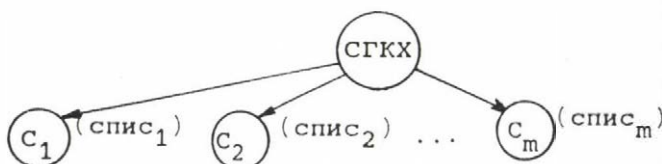


Рис.7

СГКХ - Список Генерированных Количественных Характеристик  
 С - Символ /синтакстический тип элемента изображения/  
 спис - список количественных характеристик элементов синтакстического типа С для актуального входного изображения

Общее представление об организации работы программы можно получить из рис.8. Здесь списковые структуры данных изображены круглыми, а части программы - прямоугольными блоками. Двойными направленными линиями изображена передача информации об актуальном объекте, а одинарными направленными линиями - передача информации, управляющей анализом. При выполнении программы-анализатора над постоянной списковой структурой данных /Грамматика, Список Наименований Геометрических отношений, Список Наименований

Операций Присваивания/ не производится никаких преобразований - поэтому называем ее постоянной -, информация, хранящаяся в ней, используется для управления анализом информации об актуальном объекте; по ходу анализа проводится ряд последовательных структурных преобразований над Списком Генерированных



Количественных Характеристик.

Пользование программой разделяется на два этапа:

1. Создание постоянной структуры данных /Грамматика, Списка Наименований Геометрических Отношений, Списка Наименований Операций Присваивания/ и программирование геометрических отношений и операций присваивания. Эта часть системы может свободно изменяться и расширяться при определении круга задач, решаемых с помощью программы; программа при этом остается неизменной. Постоянная структура данных описывается на входном языке ЛИДИ-72 / в скобочной форме/ на перфокартах. Очень короткая программа организует ввод постоянной структуры данных в память машины с помощью программ ввода ЛИДИ-72. При тестировании структуры данных эта короткая программа по окончании своей работы вызывает программу-анализатор, которая производит анализ какого-нибудь входного списка. После изучения результатов выполнения анализа можно провести исправления в описании структуры данных. Корректная структура данных переносится диск в виде двоичной информации.
2. Использование программы для распознавания образов. При этом постоянная структура данных непосредственно считывается с диска в память машины перед началом работы программы-анализатора.

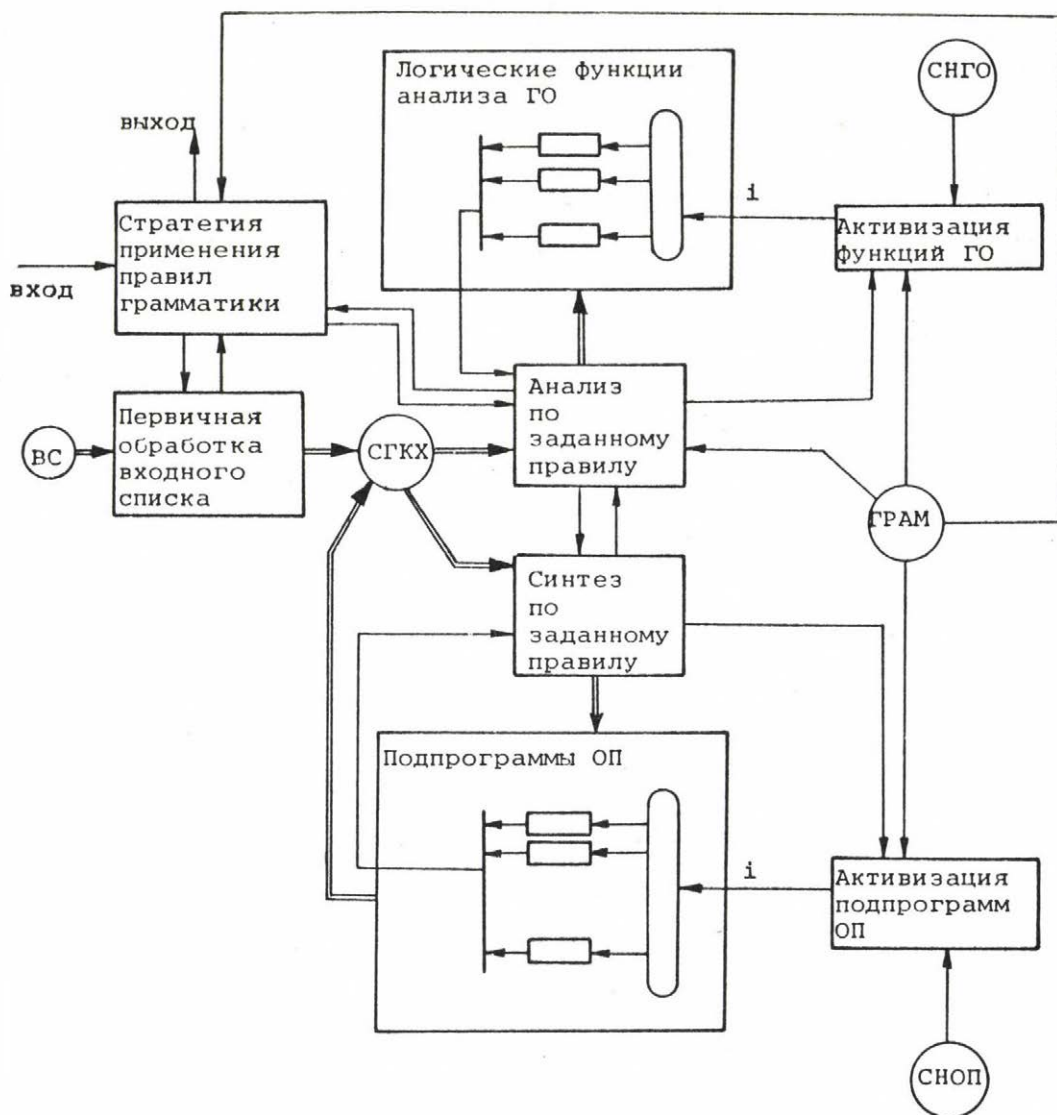


Рис. 8

- ВС - Входной Список  
 ОГКХ - Список Генерированных Количественных Характеристик  
 ГРАМ - ГРАМматика  
 СНГО - Список Наименований Геометрических Отношений  
 СНОП - Список Наименований Операций Присваивания значений  
 количественным характеристикам

ГРАММАТИКА

Правила грамматики разделяем на две группы:

1. Правила, в которых справа стоят только терминальные символы /например, правило 0/ на рис.3/.
2. Правила, в которых справа стоят только нетерминальные символы /например, правила 1/ и 2/ на рис.3/.

Правила первой группы запрограммированы как первичная обработка входного списка, поскольку это общая основа для любых интересных для нас грамматик. Правила второй группы включены в Грамматику, являющейся частью постоянной структуры данных для актуального круга задач. Такое разделение очень упрощает алгоритм, поскольку он имеет дело только с нетерминальными символами.

Порядок нумерации правил грамматики: правила пронумерованы таким образом, что каждый символ встречается сначала как левосторонний. Исключение составляют случаи перекрестной рекурсии между правилами /который также является допустимым в грамматике/. Правила с одинаковыми левосторонними символами получают различные номера и входят в грамматику как отдельные правила.

Структура отдельного правила изображена на рис.56. Каждый символ в Списке Правосторонних Символов получает номер. Нумерация совершенно произвольна, а необходима для однозначного описания геометрических отношений между символами, поскольку в одном и том же правиле может быть несколько символов одного синтаксического типа. При каждом символе указывается список наименований отношений, которым должен удовлетворять символ по отношению к предыдущим правосторонним символам правила, а при каждом наименовании отношения в правиле дается описание актуальных параметров отношения. Для описания параметров принят следующий порядок: каждый параметр описывается парой целых чисел, где первое число - это номер правостороннего символа, который должен удовлетворять этому отношению, а второе число - это номер той количественной характеристики данного символа, значение которой является актуальным параметром отношения.

Количественные характеристики символов идентифицируются их номерами /т.е. целыми числами/; во-первых, потому, что значения количественных характеристик элементов изображения хранятся в линейных списках /при вершинах, названия которых соответствуют их синтаксическому типу (рис.7)/, в установленном порядке:

например, первая - А, вторая - В и т.д.; а во-вторых, - для описания актуальных параметров также используются линейные списки, в которых могут храниться только переменные типа целый.

Подробности построения правила объясним на простом примере. Рассмотрим одно из правил грамматики, приведенной на рис.3. На объясненном во введении метаязыке написания грамматик оно выглядит следующим образом:

$$1/ \text{ARC} \rightarrow |(\text{ARC}, \text{ARC})(A_1=A_2, B_1 \neq B_2, C_1=K_2, |\rho_1 - \rho_2| > 90^\circ)(A=B_1, B=B_2, K=K_2, L=K_2, C=A_1)|$$

На входном языке ЛИДИ-72 названия вершин графа задаются цепочкой букв, а их признаки, т.е. элементы линейного списка - целыми числами. Признаки заключаются между знаками (= =) и разделяются запятыми и замечаниями.

Пусть в системе определены следующие геометрические отношения:

обозначение на метаязыке: =, ≠  
 наименования: EQ, NEQ

и операции присваивания:

обозначение на метаязыке: =  
 наименование: ASSIGN

Все описки правосторонних символов и отношений имеют название LSR /List of Symbols and Relations/, а все списки для генерирования количественных характеристик имеют название LGA /List for Generation of Attributes/.

Структура правила для ARC2 изображена на рис.10. Описание этого правила на входном языке ЛИДИ будет следующим:

C ARC2(=1=)(LSR(ARC(=1=), ARC(=2=)(EQ(=1,1,2,1=),  
 C NEQ(=1,2,2,2=), NEQ(=1,4,2,4=), OBTANG(=1,3,2,3=))),  
 C LGA(=5,0=)(ASSIGN(=1,1,2,2,1,4,2,4,1,1=)))

/С-знак продолжения/.

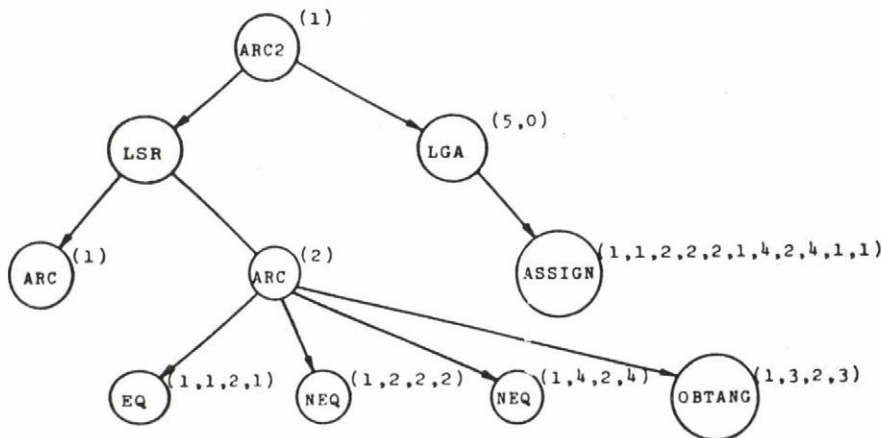


Рис.9



## РЕЗЮМЕ

Дано описание организации программы лингвистического анализа геометрических образов, входящей в систему для распознавания технических чертежей и объектов [1,2,7]

Мы исходили из формализма, предложенного Ивенсом [3,4], развивая в ходе решения конкретных задач способы экономного конструирования грамматик, четкого описания разнообразных геометрических отношений между элементами образов, и свою программную имплементацию. Программа написана на языке ФОРТРАН с использованием системы для обработки данных списковой структуры ЛИДИ-72 [5,6].

Грамматика представлена статической списковой структурой данных, включенные в систему геометрические отношения и структурные преобразования количественной информации имплементированы в виде набора подпрограмм. Алгоритм программы, организующей выполнение анализа, независим от грамматики, а передача управления программным сегментам, обрабатывающим количественную информацию, управляется грамматикой и обратно: выбор информации из грамматики управляется результатами обработки количественной информации об актуальном объекте /а также заданной стратегией применения грамматики/.

## ЛИТЕРАТУРА

- [1] Vámos T., Vassy Z.: Industrial Pattern Recognition Experiment - A Syntax Aided Approach. First International Joint Conference on Pattern Recognition. Washington, Okt.30. - Nov.1. 1973. IEEE Publ. 73-CHO 821-9C pp. 445-452.
- [2] Vámos T., Vassy Z.: Restricted Goal Syntax Aided Pattern Recognition Experiment. Acta Technica Acad.Sci. Hung. Tomus /3-4/
- [3] Evans, T.G.: A grammar-Controlled Pattern Analyzer. IFIP Congress 1968. Appl.3.
- [4] Evans, T.G.: Grammatical Inference Techniques in Pattern Analysis. Software Engineering, 1971.
- [5] Фидрих И.: ЛИДИ-72. MTA SzTAKI Tanulmányok, 21/1974.
- [6] Fidirich I., Uzsoy M.: LIDI-72 Lİstakezelő rendszer a DIGitális osztályon - 1972.évi változat. MTA SzTAKI Tanulmányok, 16/1974.
- [7] Vámos T., Vassy Z.: Limited and Progmatic Robot Intelligence. IV.Международная Объединенная конференция по Искусственному интеллекту в г.Тбилиси в сентябре 1975 г.

A TANULMÁNYOK sorozatban eddig megjelentek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundáns,  $\{\wedge, \vee, \neg\}$  vagy  $\{\text{NOR}\}$  vagy  $\{\text{NAND}\}$  bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására
- 2/1973 Вашкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительной машины
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the Presence of Drift
- 5/1973\* Gyürki J.-Laufer J.-Girnt M.-Somló J.: Optimalizáló adaptív szerszámgépírányítási rendszerek
- 6/1973 Szelke E.-Tótk K.: Felhasználói Kézikönyv /USER MANUAL/ a Folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez
- 7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor
- 8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása
- 9/1973 R.Narasimhan: Picture Processing Using Pax
- 10/1973 Dibuz Á.-Gáspár J.-Várszegi S.: MANU-WRAP hátlaphuzalozó, MSI-TESTER integrált áramköröket mérő, TESTOMAT-C logikai hálózatokat vizsgáló berendezések ismertetése
- 11/1973 Matolcsi Tamás: Az optimum-számítás egy új módszeréről
- 12/1973 Makroprocesszorok, programozási nyelvek. Cikkgyűjtemény az NJSZT és SZTAKI közös kiadásában. Szerkesztette: Legendi Tamás
- 13/1973 Jedlovszky Pál: Új módszer bonyolult rektifikáló oszlopok vegyész-mérnöki számítására
- 14/1973 Bakó András: MTA Kutatóintézeteinek bérszámfejtése számítógéppel
- 15/1973 Ádám György: Kelet-nyugati kapcsolatok a számítógépiparban
- 16/1973 Fidrich I.-Uzsoky M.: LIDI-72 listakezelő rendszer a Digitális Osztályon, 1972.évi változat
- 17/1974 Gyürki József: Adaptív termelésprogramozó rendszer /APS/ termelő-műhelyek irányítására
- 18/1974 Pikler Gyula: MINI-Számítógépes interaktív alkatrész-programíró rendszer NC szerszámgépek automatikus programozásához
- 19/1974 Gertler, J.-Sedlak, J.: Software for process control
- 20/1974 Vámos, T.-Vassy, Z.: Industrial Pattern Recognition Experiment - A Syntax Aided Approach
- 21/1974 A KGST I.-15-1.: "Diszkrét rendszerek automatikus vezérlése" c. témában 1973.februárban rendezett szeminárium előadásai
- 22/1974 Arató, M.-Benczur, A.-Krámli, A.-Pergel, J.: Stochastic Processes, Part I.

---

A \*-gal jelölt kivételével a sorozat kötetei megrendelhetők az Intézet könyvtáránál /Budapest, I., Uri u. 49./.

- 23/1974 Benkó S.-Renner G.: Erősen telített mágneskörök számítógépes tervezési módszere
- 24/1974 Kovács György-Franta Lászlóné: Programcsomag elektronikus berendezések hátlaphuzalozásának tervezésére
- 25/1974 Járdán R.Kálmán: Háromfázisú tirisztoros inverterek állandósult transziens jelenségei és belső impedanciája
- 26/1974 Gergely József: Numerikus módszerek sparse mátrixokra
- 27/1974 Somló János: Analitikus optimalizálás
- 28/1974 Vámos Tibor: Tárgyfelismerési kísérlet nyelvi módszerekkel
- 29/1974 Móricz Péter: Vegyészmérnöki számítási módszerek fázisegyensúlyok és kémiai egyensúlyok vizsgálatára
- 30/1974 Vassy, Z.-Vámos, T.: The Budapest Robot - Pragmatic Intelligence
- 31/1975 Nagy István: Frekvenciaosztásos középfrekvenciás inverterek elmélete
- 32/1975 Singer D., Borossay Gy., Koltai T.: Gázhálózatok optimális irányítása különös tekintettel a Fővárosi Gázművek hálózataira











