A logical approach to image recognition with spatial constraints*

R.K. FEDOROV, A.O. SHIGAROV

Institute of System Dynamics and Control Theory Siberian Branch, Russian Academy of Sciences

Lermontov St. 134, Irkutsk, 664033

RUSSIA

fedorov@icc.ru, shigarov@icc.ru

Abstract**—**In this paper an approach to recognizing objects on images is proposed. The approach is based on a logical inference in CLP Prolog using structural descriptions of objects. Searching edges of objects on image is performed as a unification of built-in predicate line satisfying a set of constraints defined by the description. Structural description is presented as rules of CLP Prolog.

Keywords**—** Syntactic Pattern Recognition, Constraint Logic Programming, Prolog, Spatial Constraints.

I. Introduction

The object recognition on raster image is rather complicated problem. Automatic object extraction has been an active research topic in the field of digital photogrammetry and computer vision for many years. There are many methods for object recognition on image. But they can't completely recognize objects on some classes of images. For example automatic building detection on space images is unsolved task. Possible applications for automatic building detection are the creation and verification of maps and GIS data, automatic land use analysis, etc. Human can recognize better than them. Therefore we have a need to improve and develop new methods.

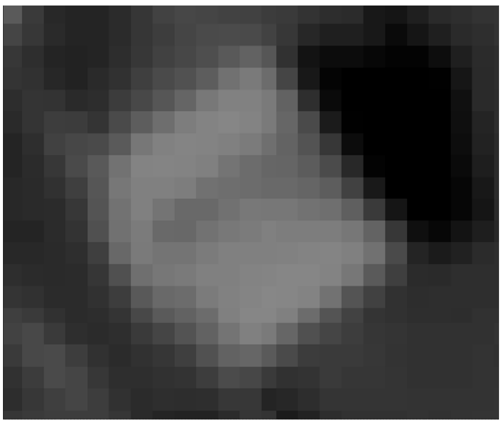


Fig. 1. A building on a space image

* This work was supported by RFBR (projects 09-07-12017-ofi_m and 08-07-00163-a), the grant of the RF President (MK-3592.2010.8), and the Lavrent'ev grants for youth projects of SB RAS.

II. Problem formulation

In the paper we consider processing grayscale images in visual spectrum. The image usually contains some noise and blur, and a lot of objects to be recognized and unrecognized. Moreover these objects can overlap each other and can have complex textures (e.g. tile roofing). Adjacent objects can have same texture. It results in partial loss of boundaries between them. Moreover different object parts (e.g. roof slopes) can be displayed differently because of illuminations. As a result the objects can be represented by inhomogeneous regions and some object regions can be merged. According to the above mentioned image features we can state that methods of image analysis should be able to use partial information about object contours represented by intensity gradients.

There are many different methods for edge detection. Let's consider two main categories of methods into which most of them are [1,2,3,4]. Many edge-detection methods are based upon the 1st derivative of the intensity - this gives us the intensity gradient of the original data. Using this information we can search boundaries on an image for peaks in the intensity gradient (Sobel, Roberts, Cross, Prewitt, Canny). Some other edge-detection methods are based upon the 2nd derivative of the intensity. This is essentially the rate of change in intensity gradient. In the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient [1,4]. For example edge-detection methods are applied for muscle biopsy image analysis [5].

The image of Fig. 1 is presented in 3D view on Fig. 2 where a value of z coordinates is an intensity of pixels.

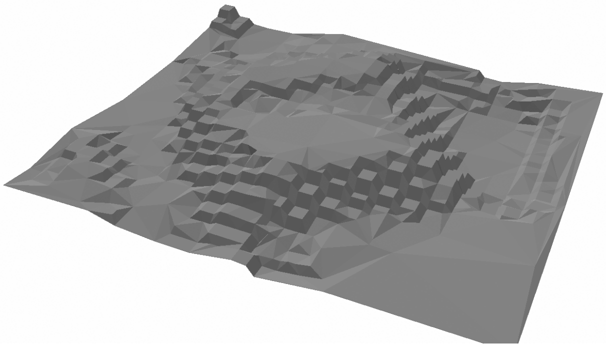


Fig.2. 3D view of the image

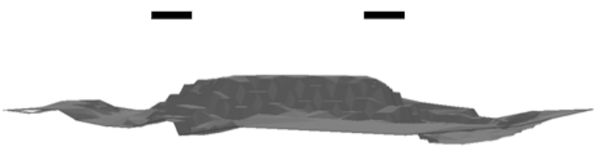


Fig.3. Possible boundaries of the object

Let's consider some local image area containing a straight-line part of an edge (further segment). Usually image intensity varies smoothly along a line orthogonal to the segment (Fig.3). The real segment position can be placed everywhere along the line. We see an imprecision of the image.

The methods considered above obtain edges using some local information. Often obtained edges don't correspond real edges. The image on Fig. 1 has been processed by the method of Canny J.F. widely known and applied [2]. The result is Fig.3. A part of edges has not been obtained. Moreover there are many wrong edges (Fig.4). Local information is not enough for edge detection.

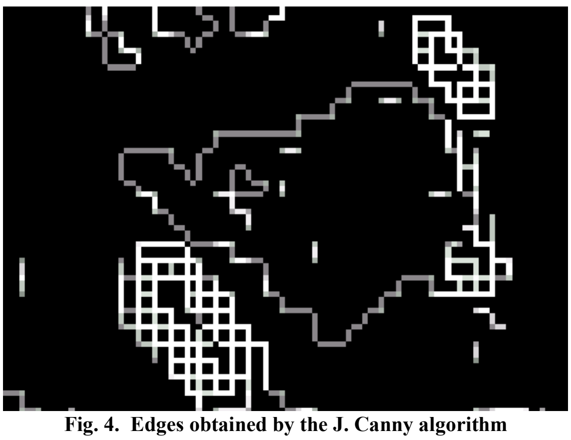


Fig. 4. Edges obtained by the J. Canny algorithm

Since we humans can sense and interpret imprecise and vague visual characteristics of the world around us, we may conclude that our vision system is mainly fuzzy. In 1965 L. A. Zadeh introduced the fuzzy set theory which was a milestone for the area of visual and cognitive science. It provides a proper framework for dealing with vague and uncertain events.

As stated in [6] the imprecision in image patterns is derived from several factors including: ambiguity in gray levels of an image (describing whether a pixel is bright or dark), spatial ambiguity (imprecision in object boundaries or edges), imprecision in knowledge base (scene description, object recognition, region segmentation), and several combinations of these. The membership function is useful in modelling and quantifying several imprecise linguistic or ambiguous terms.

Considering existence of other objects on a raster image and imprecise there is a large set of possible object segments on image which are characterized by a membership function. A part of these segments is wrong. Using relative position of segments can be useful for rejecting wrong segments. It's required a grammar to define relative position of segments. A grammar is applied in the syntactic pattern recognition.

The syntactic pattern recognition has been being developed since the 1960's [4]. Its feature is the use of structure of an object to be recognized. Using structural information can help to discard evidently unsuitable segments on image. For example, the number of alternatives of a segment position can be greatly restricted by searching only segments with a limited length. Because a length of an object segment is usually longer than the length of a segment obtained as a result of an image noise.

The authors of the paper [4] point that using of an object structure at recognition algorithms requires formalization of the structure by some formal language defined by the grammar:

$$G = (V_n,\ V_t,\ P,\ Q,\ S),$$

where V_n, V_t, P and S are finite sets of nonterminal and terminal symbols, substitution rules, start symbol, Q is a set of probability measures defined on the set of substitution rules P. Considering a noise and a loss of information it is suggested to use an approach where substitution rules are regarded as nondeterministic and corresponding probability measures are set. The substitution rules can base on spatial descriptors. For example they can be "Left", "Right", "Up" and etc [6].

Fuzzy set and methods have been successfully been integrated into primitive extraction, production rules, collection of data from uncertain sources and parsing activities carried out during classification [7,8,9,10,11,12,13,14].

Let's introduce the following notation for clear understanding:

$S \subset \mathbb{N}_0^{\,2}$ is a set of points on the plane;

$L \subset \mathbb{N}_0^{\,m}$ is a set of pixel values (intensity of the gray, color in RGB or CMYK and etc.);

$f : S \to V$ is a raster image.

III. Problem solution

An approach is offered within the syntactic pattern recognition in which the object recognition is based on a logical inference using structural information about an object. Matching segments of an object on raster image is performed as a unification of the built-in predicate line consistently with a set of constraints defined by structural information.

Object boundaries are modeled only by segments at our work. Spatial descriptors of segment position can be formalized by various ways [15]. In order to reduce number of alternatives to the segment position we formulate the following spatial descriptors.

1) One of the ways to substantially reduce the size of the set of alternatives to the position of segments is to limit their lengths. To calculate the segments length we'll use the following auxiliary function, which computes Euclidean distance between two points:

$\mathrm{len}: S^2 \rightarrow \mathbb{R}$

2) Another way to cut down some segments is to set a constraint on the relative position of object segments, which is defined by computing the angle between two segments with a common end point:

$\mathrm{angle}: S^3 \rightarrow A$, where

$A = \{a : a \in \mathbb{R} \text{ and } 0 \leq a < 360\}$.

The spatial descriptors of segment position can be defined as a set of constraints with using these functions. Next, we call them spatial constraints. An object structure can be defined by a set of spatial constraints describing relations among segments, i.e. conjunction of constraints. A segment set corresponds to an object structure if all of constraints are realized.

Often, a part of a real object on an image can be optional. For example a building can have a porch or not. Therefore using disjunction in a description of an object structure increases an expressiveness of the description language.

Considering constraints, disjunctive and conjunctive constraints, a need of flexible exhaustive search the most suitable mechanism is “Prolog III” (CLP, Constraint Logic Programming) published in [16]. CLP is closely connected with traditional logic programming. Constraint logic programming is a form of constraint programming, in which logic programming is extended to include concepts from constraint satisfaction. A constraint logic program is a logic program that contains constraints in the body of clauses. As in regular logic programming, programs are queried about the provability of a goal, which may contain constraints in addition to literals. A proof for a goal is composed of clauses whose bodies are satisfiable constraints and literals that can in turn be proved using other clauses. Execution is performed by an interpreter, which starts from the goal and recursively scans the clauses trying to prove the goal. Constraints encountered during this scan are placed in a set called constraint store. If this set is found out to be unsatisfiable, the interpreter backtracks, trying to use other clauses for proving the goal. In practice, satisfiability of the constraint store may be checked using an incomplete algorithm, which does not always detect inconsistency.

Constraints can be expressed over a variety of different domains [17]. As a general rule, two conditions should be satisfied for a CLP language over a computation domain:

• Constraints can be handled in a deterministic way without loosing completeness.

• Efficient constraint solving methods exist for the domain.

A number of computation domains have been identified which satisfy these conditions, and which are now used in one or several CLP systems. They are

1. Finite domains. Finite domain constraints are expressed over variables which range over a finite set of possible values. Often, the domains are expressed as bounded subsets of the natural numbers. Constraints may be arithmetic or symbolic. Complex, global constraints have been added in the last years.
2. Linear arithmetic terms. Constraints over linear arithmetic terms can be handled efficiently by Gaussian elimination and the Simplex algorithm. Several systems allow the use of rationals, while other handle floating point numbers.
3. Boolean terms. Equality constraints over Boolean terms can be handled by Boolean unification. Since any complete solver has worst case exponential complexity, other, incomplete solvers are popular for certain applications.

Some other computation domains have been proposed, but are not yet widely used for problem solving. Therefore to propose computation domains for image recognition we should consider problem solving. The most appropriate computation domains are finite domains. Position of edge points is limited by image size and coordinates are set by integer values. That is coordinates have finite domain. Constraints to relative segment position can be given as non-linear.

To implement the proposed approach is necessary to supplement the following entities:

1) a set of variables describing point positions on image (the domain of definition of these variables is limited by the size of an image);

2) a set of spatial constraints;

3) special predicates.

Let’s define the possible spatial constraints. Before we introduce a set of character of comparisons (a set of binary relations):

$\Theta = \{=, >, <, \neq, \leq, \geq\}$

Spatial constraints may be as follows:

$\mathrm{len}(s_1, s_2)\theta c$,

$\mathrm{angle}(s_1, s_2, s_3)\theta c$,

$x(s)\theta c$,

$y(s)\theta c$,

$x(s_1)\theta y(s_2)$,

$q\theta c$,

Where $\theta \in \Theta$ is a character of comparison, s is a variable, describing a point on plane, c is a numeric constant, q is a real variable. For example the followed constraints can be expressed:

$\text{angle}(s_1, s_2, s_3) = 90$,

$\text{len}(s_1, s_2) > 20$,

$\text{len}(s_1, s_2) \le 30$.

A. The unification of spatial predicates

The unification of predicates is an important point of the approach. Special predicates are required for matching object boundaries on an image. The unification of the special predicates is required to obtain all of various alternatives of an object boundary satisfying constraints. There may be several such predicates corresponding to a type of object boundary. For example predicates can correspond to segment, arc of a circle or other curves. All of them will have a different set of parameters. In our paper we consider only predicate matching segments:

$\text{line}(s_1, s_2, q)$,

where s_1, s_2 are points on plane, q is a value of interval $[0,1]$. It should be noted that the unification algorithm does not use only these parameters. It has to account for spatial constraints. Therefore spatial constraints and variables used in constraints are implicit parameters of the predicate. Let's consider more detailed the real inputs and outputs of the predicate.

Inputs:

$s_1, s_2, \ldots, s_n$ are points on plane. Some points can be undefined.

q is a variable for storing some estimation of boundary segment passing between points s_1, s_2.

Cs is a constraint store (a set of spatial constraints) on variables $s_1, s_2, \ldots, s_n, q$ defined above.

Outputs:

s_1, s_2 - if variables are undefined in the inputs.

q - is an estimation of membership function which the predicate calculates.

The variables s_1, s_2 can be defined. In this case the predicate validates spatial constraints with defined values of s_1, s_2, calculates an estimation of boundary segment passing for pair s_1, s_2 and finishes its work. Otherwise the predicate algorithm must enumerate all possible pair of points. For every pair the algorithm must validate constraints and calculate an estimation and next deliver control back to the inference machine.

It is required that a set Cs has constraints as $q\theta c$. Because in this case of absents of such constraints the predicate must make complete enumeration of pairs satisfying spatial constraints. This can lead to large computational complexity.

Calls of the predicate can exchange results each other for increasing performance.

B. Segment membership function

Consider a function of membership estimation. First of all let's introduce the function notation:

$\text{line} : S^2 \to B$,

$B = \{b : b \in \mathbb{R} \text{ and } 0 \le b \le 1\}$.

Value 1 corresponds to the segment, which is a sharp border between two parts of its neighborhood, value 0 denotes the absence of difference between the parts.

One can distinguish two types of boundaries on an image:

1) lines separating two different areas;

2) lines separating two similar areas, but having intensity change along the line itself (boundary line).

It is possible to suggest various estimations of segment passing. We'll use a heuristic function, which allows extracting boundaries of the first type. The given function analyses some neighborhood of a straight-line segment between two points of the image. More exactly the function is based on an analysis of two rectangles along a segment (Fig. 5). The width of rectangles is defined by the method parameter.

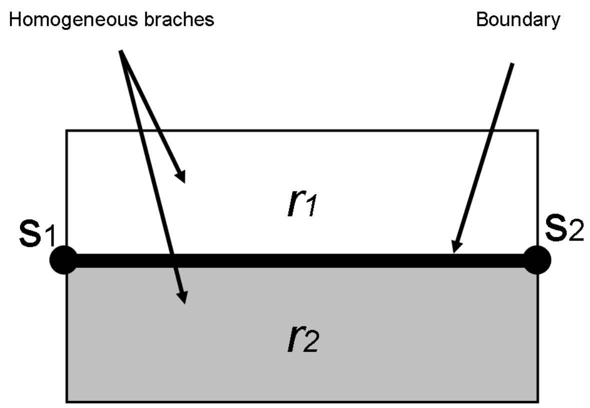
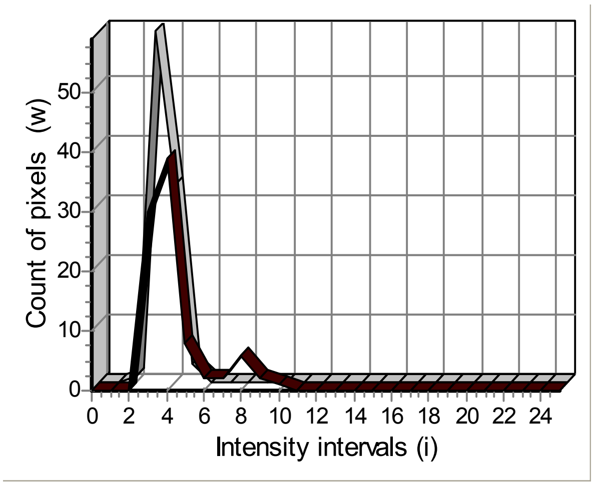
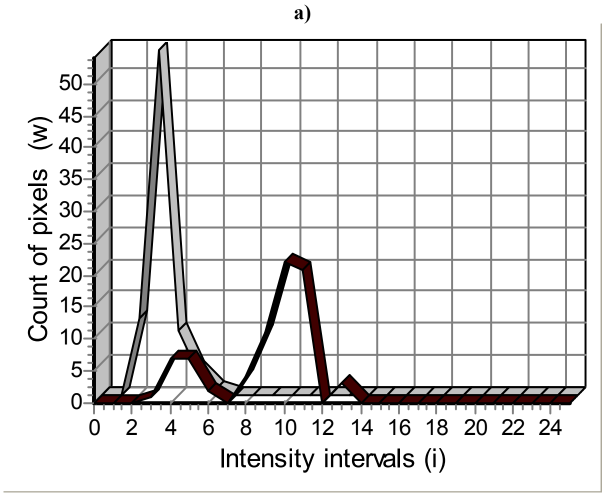


Fig. 5. Analysis area of the line function



a)



b)

Fig. 6. The histograms are likely (a) and not (b)

For each rectangle a histogram on n equal intervals is computed. Let's denote by w_i^r the count of pixels from the intensity interval i in the rectangle r. The line estimation function is defined as:

$$\mathrm{line}(s_1, s_2) = 1 - \frac{\sum_1^n \min(w_i^1, w_i^2)}{\sum_1^n \max(w_i^1, w_i^2)}$$

The function returns a value close to 0 if histograms of rectangles are similar (Fig. 6a) and returns a value close to 1 if histograms of rectangles differ (Fig. 6b).

Using the function line we can choose between the segments, which are compatible with the position of the whole object and the separate parts of its contour.

C. The recognition procedure

The procedure of matching a recognition object can be built as logical inference of a goal. During logical inference the unification of segments set satisfying some constraints

$$\mathrm{obj} = \left\{ \left(s_1^1, s_1^2\right), \ldots, \left(s_k^1, s_k^2\right) \right\}$$

is being performed. A recognized object must have the maximum of a general estimation $est = h(b_1, ..., b_k)$, where $b_i = line(s_i^1, s_i^2)$. The general estimation $h(b_1, ..., b_k)$ can be computed as a linear convolution $h(b_1, ..., b_k) = \sum_{i=1}^{k} m_i b_i$.

The convolution factors m_i can be considered as a measure of importance of the corresponding segments of object contour. Therefore this general estimation should be set separately for each object class.

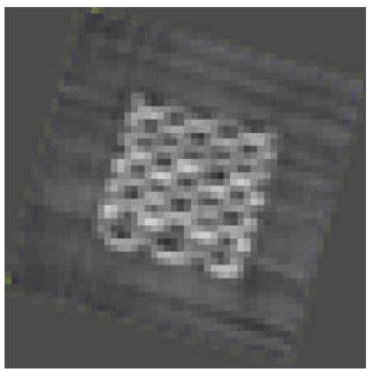
IV. Approbation

For check the approach the search of objects has been implemented with "Borland Delphi". The algorithm has found several positions of object satisfying constraints. The selection among found positions has not been performed. The mechanism of search corresponds to the object description (Fig.7).

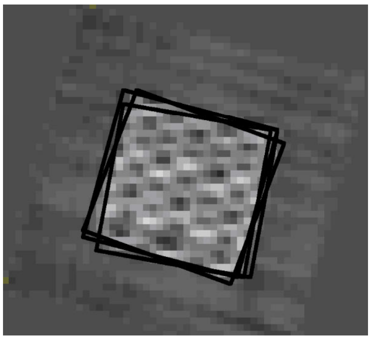
```
house(s1,s2,s3,s4) :-
line(s1,s2,b1),
b1>0.8,
len(s1,s2)>10,
angle(s1,s2,s3)=90,
line(s2,s3,b2),
b2>0.8,
len(s2,s3)>10,
angle(s2,s3,s4)=90,
line(s3,s4,b3),
b3>0.8,
len(s3,s4)>10,
line(s4,s1,b4).
```

Fig. 7. Description of an object structure on Prolog

The description corresponds to rectangular objects. The experiment was spent on a set of images, including Irkutsk city image fragments with the resolution of 0.7 meters on pixel. The results of experiment with this rule are shown on Figures 8, 9, 10.



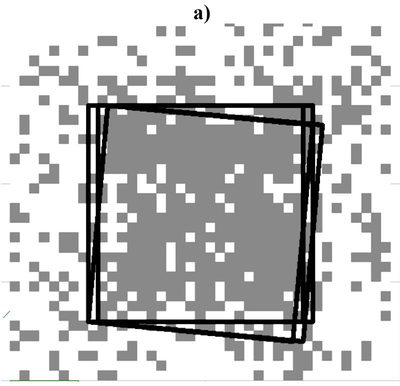
a)



b)

Fig. 8. Test image. a) is a source, b) is a result

a)



b)

Fig. 9. Test image. a) is a source, b) is a result.

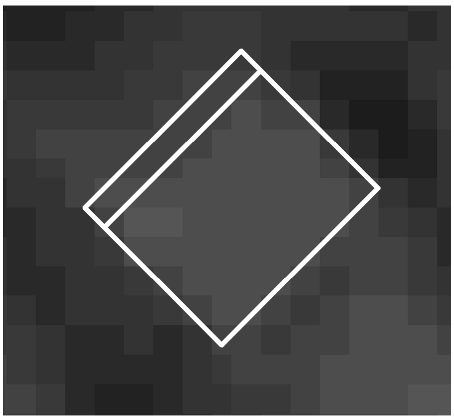
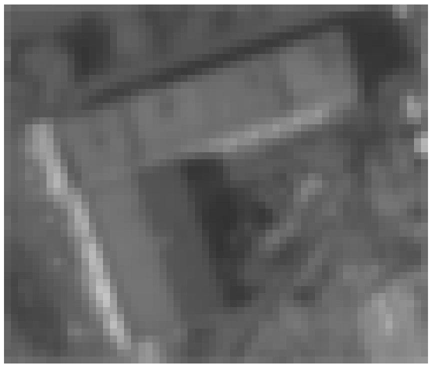


Fig. 10. Test image

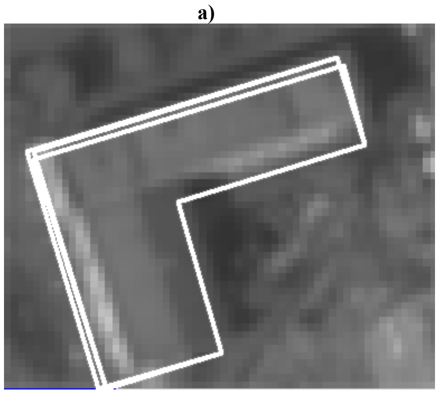
Another object description corresponds to a house which consists of two parts placed orthogonally (Fig. 11). The rule corresponding to the description defines segments next by next. Constraints on segments length allows to reduce the execution time. The results of experiment with this rule are shown on Figures 11, 12, 13.

```
house(s0,s1,s2,s3,s4,s5) :-
line(s0,s1,b1),
b1>0.8,
len(s0,s1)>10,
angle(s0,s1,s2)=270,
line(s1,s2,b2),
b2>0.8,
len(s1,s2)>10,
angle(s1,s2,s3)=90,
line(s2,s3,b3),
b3>0.8,
len(s2,s3)>10,
angle(s2,s3,s4)=90,
line(s3,s4,b4),
b4>0.8,
len(s3,s4)>10,
angle(s3,s4,s5)=90,
line(s4,s5,b5),
b5>0.8,
len(s4,s5)>10,
angle(s4,s5,s6)=90,
line(s5,s6,b6),
b6>0.8,
len(s5,s6)>10,
angle(s5,s6,s5)=90.
```

Fig. 11. Description of an object structure on Prolog



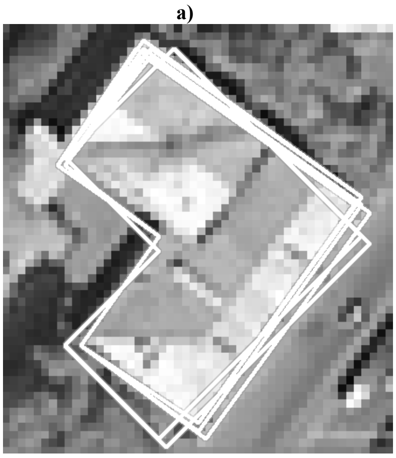
a)



b)

Fig. 12. Test image. a) is a source, b) is a result

a)



b)

Fig. 13. Test image. a) is source, b) is result.

It should be noted that the restructuring of the recognized object description leads to significant changes in the algorithm code. Therefore, description of the object structure with help of some language will allow more flexibility to apply new knowledge in image recognition.

V. Conclusion

The approbation results confirm efficiency of the approach: use of the additional structural information during segmentation and recognition of objects on the image. A lack is computing complexity. Further it is planed to develop convenient language of the object form description and to develop the interpreter of language which would be effectively on time and to quality to find recognized objects.

The unification algorithm uses the method of branch and bound. Branch and bound is an algorithm for finding optimal solutions. We have modified the method for finding permissible solution. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded, by using upper estimated bounds of the value of membership function. Now we use simply calculates upper estimated bounds.

Usually recognition objects don't overlap each other. It is planed to introduce implicit constraints to line estimation which the recognition algorithm will be dynamically changing during inference.

References

[1] D. Ziou, S. Tabbone, Edge Detection Techniques An Overview, International Journal of Pattern Recognition and Image Analysis, Vol 8, No 4, 1998, pp 537--559, http://citeseer.ist.psu.edu/ziou97edge.htm

[2] J. F. Canny, A computational approach to edge detection, IEEE Transactions. on Pattern Analysis and Machine Intelligence, 8, 1986, pp. 679-714.

[3] W. Green, Edge Detection Tutorial, 2002, [Online]. Available: http://www.pages.drexel.edu/~weg22/edge.html

[4] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Reading, Massachusetts: Addison-Wesley,1992

[5] P. Tzekis A. Papastergiou, A. Hatzigaidas, A. Cheva, A sophisticated edge detection method for muscle biopsy image analysis, Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing, Beijing, China, September 15-17, 2007, pp. 118-123, http://www.wseas.us/e-library/conferences/2007beijing/papers/554-388.pdf

[6] Tzung-Pei Hong, Szu-Po Wang, Tien-Chin Wang, Been-Chian Chien A Fuzzy Image Matching Algorithm with Linguistic Spatial Queries http://myweb.nutn.edu.tw/~bcchien/Papers/C_WSEAS2003.pdf

[7] K. S. Fu., Syntactic Pattern Recognition and applications, NJ, Prentice-Hall, 1982.

[8] W. Siler, J. J. Buckley - Fuzzy Expert Systems and Fuzzy Reasoning, Published by John Wiley & Sons, Inc., Canada, 2005.

[9] J. P. Marques de Sa, Pattern recognition, Concepts Methods and Applications, Springer, 2001.

[10] R. Fabian, V. Crăciunean, E. M. Popa, Intelligent system modeling with total fuzzy grammars, Proc. of the 8th WSEAS Int. Conf. on Mathematical Methods and Computational Techniques in Electrical Engineering, Bucharest, October 16-17, 2006, pp. 82-87, http://www.wseas.us/e-library/conferences/2006bucharest/papers/518-285.pdf

[11] Ralf Fabian, Vasile Crăciunean, Emil Marin Popa, Syntactic pattern model classification with total fuzzy grammars, 12th WSEAS International Conference on COMPUTERS, Heraklion, Greece, July 23-25, 2008, pp.317-321, http://www.wseas.us/e-library/conferences/2008/crete/Computers/048-computers.pdf

[12] L. A. Zadeh, Fuzzy sets. Information and Control, 8, 1965, 338-353.

[13] L. A. Zadeh, Similarity relations and fuzzy orderings. Information Sciences, 3, 1971., 177-200.

[14] L. A. Zadeh, Knowledge representation in fuzzy logic. In An introduction to fuzzy logic applications in intelligent systems. Kluwer Academic, 1992.

[15] R.O. Duda, P.E. Hart, Pattern classification and scene analysis, A Wiley-Interscience Publication, 1973.

[16] A. Colmerauer, An introduction to Prolog III, Communications of the ACM, Vol. 33 (7), 1990, pp. 69-90.

[17] H. Simonis, Constraint Logic Programming, [Online]. Available: http://www.4c.ucc.ie/~hsimonis/Constraint%20Logic%20Programming.pdf

Roman K. Fedorov works at Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences as a senior research fellow. He defended his Ph.D. in 2005. The theme of the Ph.D. work is software for graph and logical presentation and analysis of space-distributed data.

Alexei O. Shigarov works at Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences as a researcher. He defended his Ph.D. in 2009.