A Parallel Conveyer Algorithm for the Recursive Method of Scanning Mask for Primary Images Processing

Naiden Vasilev
Atanaska Bosakova-Ardenska

Abstract: *In this paper the recursive method of scanning mask is analysed and one new parallel conveyer algorithm is proposed based on the parallel algorithm "conveyer processing"- simultaneous conveyer processing (SCP).*

Key words: *Parallel Algorithms, Recursive Method of Scanning Mask, Conveyer Processing.*

INTRODUCTION

The images processing is important field of the information processing in the contemporary computer systems. Some of the fields where the image processing is essential are robotics, criminology, text recognition, recognition of the targets in military actions, medicine, mineralogy, cartography and etc. [1].

The method of the scanning mask is one of the methods for primary images processing [2]. It is founded on every-element visitation of the image matrix with a mask which includes the surround area of pixels of the treated element (or pixel). The masks are square matrixes with an odd number of the raster (3x3, 5x5, 7x7, 9x9). The central element of the mask coincides with the treated element of the image. Every element of the mask contains a coefficient (fig.1). The image is represented by one or more matrixes in dependency of if it is colour picture or not. The disposition of the pixels in the matrix is defined by the indexes in the matrix.

k_1	k_4	k_7
k_2	k_5	k_8
k_3	k_6	k_9

Fig.1 Scanning Mask 3x3

The value of the gradation of the treated element situated under the central element of the mask is replaced with a value received from the function f(kl,bl) where kl, l=1,2,…,t – are the coefficients of the mask and bl are the values of bij, l = 1,2,…,m, j=1,2,…,n of the gradations of the pixels fallen under the mask. The values of t are usually 9, 25, 49 and 81. The type of the function f (kl,bl) specifies the filter. One of the most popular filters is '1/9' where t=9 and all the coefficients have value 1. The function is:

$$f(k_l b_l) = \frac{1}{9}\sum_{l=1}^{9} k_l b_l = \frac{1}{9}\sum_{l=1}^{9} b_l \qquad (1)$$

Calculated by this way value becomes the new value of gradation of the processed element. During this treatment on every next step moving the mask **the new values are used which are obtained from the preceding steps**. For this processing the outlying rows and columns couldn't be filtered because they couldn't become central elements.

AIMS

The tasks of images processing are characterized by operations "of the same kind" executed over large data massives. In some cases the requirements to the computer systems for the image processing are too high (processing moving objects for example). This impels the quest of algorithms insuring an acceleration of the processing. One effective method for reaching a high capacity in image processing is the usage of

parallel algorithms. The characteristics of primary image processing are favourable for invention and usage of parallel algorithms.

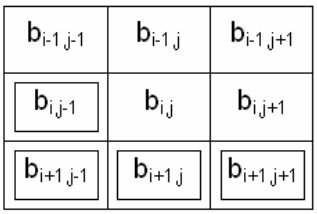
The aim of the present work out is to be proposed a parallel algorithm for primary image processing using the recursive method of a scanning mask which develops and improves the idea in [6].

ALGORITHM "SIMULTANEOUS CONVEYER PROCESSING" (SPC)

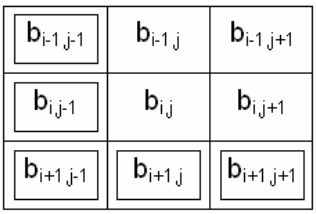
We will propose a parallel algorithm based on the proposed algorithm in [6] «conveyer processing» (CP). The new points are:

the image processing starts from more than one place simultaneously;

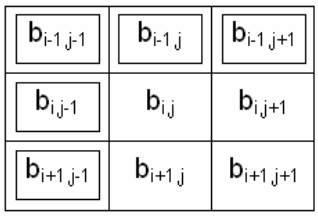
the new values of some elements of the image could be calculated not only with the new values of four neighbouring elements but with the new values of more neighbouring elements.



Фиг. 2



Фиг. 3



Фиг. 4

Let the processing starts simultaneously from **two places: from the first element of the first row and from the first element of the last row. The upper half and the lower half of the image are processed simultaneously like two different images by the algorithm "conveyer processing" [6]. But the moving of the mask in the lower half is from left to right and from below upwards**. That's why the new neighbouring elements taking part in the calculation of the current one are as shown on fig. 2.

Let we are given an image sized 14x18. On fig. 5 and fig. 6 are shown two different ways of distributing its elements for processing to 16 processors: 8 for the upper half and 8 for the lower one. Every processor calculates the upper or the lower half of elements in two neighbouring columns. All the elements with equal number could be calculated simultaneously. The number of parallel steps is at a minimum for every half of the image, e. g. the number is equal to the number of parallel steps in PCC (p=2) 18+2x8-8=26 (when sized 8x18) (fig.5).

It is possible another distribution of elements for processing to be allocated to the processors. For example, if the number of processors is 8 then the number of processed halves of neighbouring columns is 4. In this case the improvement of the algorithm compared with algorithm "conveyer processing" is minimal (see table 1).

If this image is processed by the algorithm "conveyer processing" [6], then the number of parallel steps with 8 processors would be at minimum, e. g. equal to the number of parallel steps in PCC: 18+2.14-8=38.

During the simultaneous conveyer processing of the image the elements of the finally processed rows in the upper and the lower half (except the first and the last element of the rows) can be calculated with 5 new values of the neighbouring elements. On fig. 5 the elements of these rows are coloured in gray. The neighbour elements in the upper half with new values for the elements of the finally processed row (except the first and the last element) are shown on fig. 4, and for the elements of the last processed row (except the first and the last element) in the lower half – on fig. 3.

We will propose the following distribution of output data among the processors when using local memories:

- in processor P_i i = 1,2,3,…,k/2 are stored the values of the elements of i-number p columns of the upper half of the image where the counting of the columns starts from 2. We will consider that k is an even number;
- in processor P_i i = κ/2+1,…,k are stored the values of elements of i-number p columns of the lower half of the image where the counting of the columns starts from 2;
- in every processor P_i i = 2,3,…,k/2-1 are stored also the values of elements of the last column of processor P_{i-1} and the first column of processor P_{i+1} of the upper half of the image;
- in every processor P_i i=κ/2+2,…,k-1 are stored also the values of elements of the last column of processor P_{i-1} and the first column of processor P_{i+1} of the lower half of the image;
- in processor P_1 are also stored the first column of the upper half of the image, the first column of processor P_2 and the first three elements of the firs row of the lower half of the image;
- in processor $P_{κ/2+1}$ are stored the first column of the lower half of the image, the first column of processor $P_{κ/2+2}$ and the first three elements of the last row of the upper half of the image also;
- in processor $P_{k/2}$ are stored the last column of the image, the last column of processor $P_{κ/2-1}$ and the last two elements of the first row of the lower half of the image also;
- in processor P_k are stored the last column of the image, the last column of processor $P_{κ-1}$ and the last two elements of the last row of the upper half of the image also.

		P1		P2		P3		P4		P5		P6		P7		P8		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
3		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
4		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
5		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
6		9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
7		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
8		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
9		9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
10		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
11		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
12		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
14																		
		P9		P10		P11		P12		P13		P14		P15		P16		

Fig.5 Ddistribution of elements for processing to 16 processors

Every processor calculates consecutively the values of elements of its column (fig.6a). The number of operations completed by every processor for finding one element is: 9 in the odd-number parallel steps (8 summing-up and 1 division), and 7 in the even parallel steps (6 summing-up and 1 division). The difference between the numbers of operations is due to the possibility in the even parallel steps a partial sum to be used which is received in an odd parallel step (on fig.6a – the vertical arrows). For example, the partial sum 25 (which is a new value) + 35 + 45 (calculated when finding the new value of element 34) is used for the finding the new value of element 35 (fig. 5).

In the common case the number of completed operations is 9 for the elements of every first column of a processor and 7 for the elements of every next column. (The number of columns processed by every processor could be 2, 3, …etc.)

The exchange interrelations between processors (when the distribution is 2 columns for processing in a processor) are shown on fig.6a. The exchange interrelations are defined by the explanations given above for finding the processed elements of the image. It can be seen that on every even parallel step the new values of elements are transferred to the next processor on the right and on every odd step – to the processor on the left. On the even parallel steps (for every row except the last one) every transferred element is necessary for finding the new value of the neighbouring element on the right side of the same row of the image. On the odd parallel steps (for every row except the last one) every transferred element is necessary for finding the new value of the neighbouring element on the left side of the next row of the image.

In the common case, if the number of columns processed in every processor is p = 3,4, … , then the new value of every processed element of the first column of processor P_i (i = 2,3,… k) is transferred to processor P_{i-1}, and the new value of every processed element of the last column of processor P_i (i = 1, 2,3,…k-1) is transferred to processor P_{i+1}.

When reaching the last row for processing in every half of the image there are exchange interrelation between the pairs opposite processors also. Opposite processors are those which process the same columns. Besides this, every processor (except the last pair opposite processors: $P_{k/2}$, P_k) transfers to the next processor on the right side of the opposite one as well (fig.6b).

The exchange interrelations considered above are defining the requirements of the topology of interconnection network of parallel architecture: it can be Two-dimensional mesh [4].

The number of steps until the last two processors start is:

$$r = p(k/2 - 1) \quad (2)$$

where p is the number of columns processed by every processor.

The number of parallel steps where all the processor elements take part is:

$$q = p(m/2 + 1 - k/2 - 1) = p(m-k)/2 \quad (3)$$

The formulae (2) and (3) are obtained from the formulae for r and q [6] for image sized (m/2+1)xn. (The two halves of the image are processed simultaneously every of which by k/2 processors.)

In the common case the number of parallel steps for resolve the problem s_e is:

$$s_e = 2r + q \quad (4)$$

The number of parallel steps se for resolve the problem, when p=2, will be equal to the number of steps of PCC [6].

$$s_e = s = n + 2(m/2 + 1) - 8 = n + m - 6 \quad (5)$$

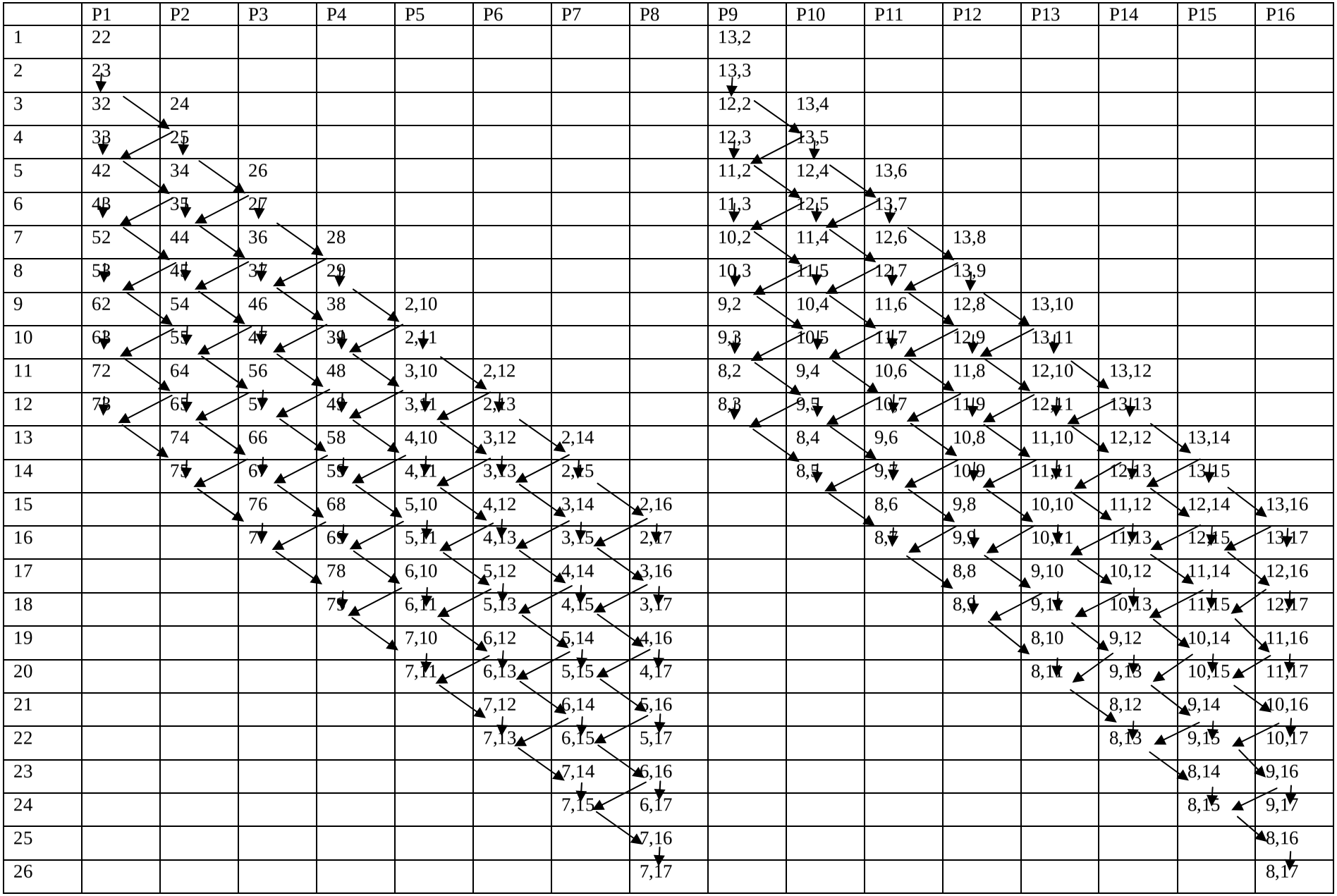


Fig. 6 (a) Ddistribution of elements for processing to 16 processors

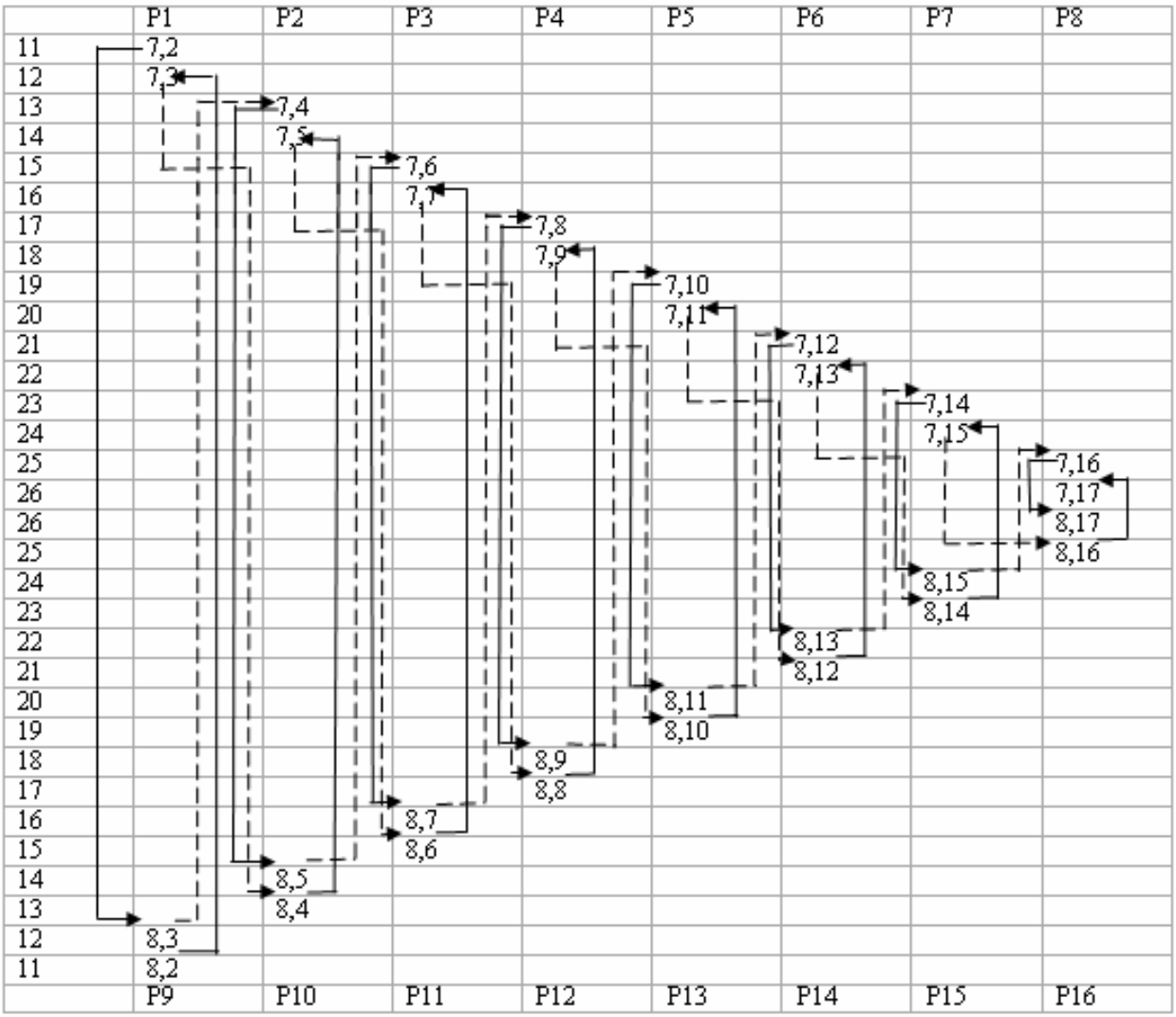


Fig. 6 (b) Ddistribution of elements for processing to 16 processors

In table 1 and 2 are shown the values of parameters for images sized 14x18 and 768x1024.

Table 1 Comparison between algorithms CP and SCP for image sized 14x18

Algorithm	s	r	q	k	p	Elements per processor
CP	38	14	10	8	2	24
SCP	26	14	-2	8(16)	2	12
SCP	36	12	12	4(8)	4	24

Table 2 Comparison between algorithms CP and SCP for image sized 768x1024

Algorithm	s	r	q	k	p
CP	2552	1020	512	511	2
SCP	1786	1020	-254	(511)1022	2

CONCLUSIONS AND FUTURE WORK

The present work out is a continuation and a development of the proposed algorithm in [6] for «conveyer processing» (CP). One parallel algorithm of the recursive method of scanning mask is proposed- SCP.

Tables 1 and 2 show that the proposed algorithm have better parameters than the algorithm CP. In the algorithm SCP there are the least number of parallel steps.

Note that in proposed algorithm the two rows in the middle are calculated with 5 new values (but not with 4 as it is for the other elements).

The future directions for development are related with a search of other parallel algorithms with improved parameters of the considered problem and with a search of parallel algorithms for other similar tasks as well.

REFERENCES

[1] Gonzalez, "Digital image processing" second edition, 1987г.

[2] A. Bosakova-Ardenska, N. Vasilev, Parallel algorithms of the scanning mask method for primary images processing, CompSysTech'04, Rousse, Bulgaria

[3]Э.В.Евреинов, Ю.Г.Косарев, "Однородные универсальные вычислительные системы высокой производительности", Издательство "Наука" Новосибирск, 1966г.

[4] Seyed H. Roosta, Parallel Processing and Parallel Algorithms: theory and computation, 2000г.

[5] Vasilev N., Main Principles for Searching and Creating Parallel Algorithms, Information Technologies and Control, 2004, 2, ISSN 1312-2622.

[6] Vasilev N., Bosakova-Ardenska A., One Parallel Algorithm of the Recursive Method of Scanning Mask for Primary Images Processing, Computer Scince'04, Sofia, Bulgaria

ABOUT THE AUTHOR

Assoc.Prof. Naiden Vasilev, PhD, Department of Computer Systems and Technologies, Technical University Sofia, Branch in Plovdiv, Phone: +359 659 705, E-mail: mnvasilev@yahoo.com.

PhD student Atanaska Bosakova-Ardenska, Department of Computer Systems and Technologies, Technical University Sofia, Branch in Plovdiv, Phone: +359 659 704, E-mail: abosakova@yahoo.com.