INFORMATICA, 1991, Vol.2, No.1, 100–116

HIERARCHICAL 2D SHAPE REPRESENTATION AND COMPRESSION BY RECTANGLES

Jonas PUNYS

Kaunas Polytechnic Institute,
233028, Kaunas, V.Juro St.50, Lithuania

Abstract. The hierarchical principle of video-information analysis (progressive detalisation) is one from the set of principles which are implemented in the living vision system. The reflection of this principle in the techniques of the representation of a shape of the region, occupied by binary image, allows us to find a solution of two tasks simultaneously: data compression and data structure, which suits for geometric transformations of the image. This report includes operations which are performed on the hierarchical list of rectangles. The latter is built up by using intermediate pyramidal representation.

Key words: image compression, data structures, progressive detalization, pyramids, geometric transformations.

1. Introduction. The hierarchy of 2D and 3D shape representation means that a rough representation of the main part of the object comes first and then progressively come the details of it. Perhaps everyone will agree that at first we percept the general shape of the object and only later its details. The experimental data of eye-movement measurements confirm it. For example, in the report of Palmieri, Oliva and Scotto (1971) there is presented on Fig.8b and Fig.9 the record of the eye-movement when the subject was shown a womanly face, without instructions. We can see on Fig.9a that the subject's eye turns towards the centre of the face with the help of saccadic movements and only then it runs around the boundary of the face progressively returning to and from the centre. The given example is not a strict proof of the hierarchical representation of the shape in the human brain. It is only an illustration of the hypothesis of hierarchical information representation in our brain. Up-to-date science is far away from the exact knowledge about the human brain and its action, though the data are progressively accumulating in this field (Glezer, 1985). The principle of hierarchy reveals itself in different visions stages. Hubel and Wiesel (1962) have proposed hierarchical model of receptive fields of the visual cortex. The pmodels of Essen and Maunsell (1983) confirm this fact. The impressive confirmation of hierarchical shape representation in the brain is given in experiments performed by Podvigin and others (1975). These results are represented by Podvigin (1979) as the hypothesis about image information transmition by lateral geniculate body (LGB). We explain it in the language of specialists in signal processing (terms in neurosciences see in Kunt's, Ikonomopoulos's and Kocher's (1985) article.) The systems of receptive fields of LGB can be considered as a filter of the spatial frequencies (Podvigin, 1979).

The frequency band of this filter depends on the ratio of diameters of receptive field summation zone and inhibition zone, and on the ratio of weight functions of exciting and inhibiting processes in the receptive fields. By increasing the diameter of the summation zone, the system of such receptive fields transfer the low spatial frequencies better, and by decreasing the diameter of the summation zone – the high spatial frequencies better. After the retina having accepted an image, the diameter of the summation zone of the receptive field is big at the first moment. Consequently the system of the receptive fields acts as a low spatial frequency filter. The resolution of the system is low. The size of the summation zone of the receptive field decreases with time. This is equivalent to the expanding of the frequency band of the filter, and to the resolution increasing. Then ascent of the frequency characteristic values comes on high frequency area, and descent on low frequencies. That is the final state of the adaptive filter of the spatial frequencies, which is implemented in a brain by nature. In this final state the edges of objects are extracted in the image.

On the basis of the above given feature the hypothesis claims, that the visual cortex uses information which is transfered from receptive fields of LGB with resolution, varying with time.

The above described hypothesis of hierarchical representation of video-data in the visual cortex has not been confirmed by neurophysiological experiment yet. If it is not so, then only signals, which are transferred to visual cortex in origin and final states of receptive fields of LGB carry information, i.e. only texture and edges. We do not know exactly in what way the hierarchical representation is used in our visual system. Nevertheless such video-data representation in technical systems impart new good features. The flow of investigations in this direction confirms the fact (see Rosenfeld (Ed.), 1984). It should be noted that to simulate exactly real visual system on technical devices not always is necessary. Solely approximate reflection of any principle of vision system mechanism in a computational technique (taking into account exploiting technical means) may effectively enhance the features of the technique. Such an approximate reflection of the hierarchical principle of information processing is the data structure in the form of a list of rectangles or rectangular parallelepipedes (ordered according to the size) in cases 2D or 3D shape representation respectively. These data structures are nonrecursive ones, but they have hierarchical property as well as in recursive trend (Klinger, 1971; Rosenfeld (Ed.), 1984; Samet, 1984; in russian: Punys, 1980; Aleksandrov and Gorsky, 1985). Moreover, such data structure is built with the help of intermediate recursive representation. The ordered lists of rectangles allow us not only to achieve image data compression, but they are well suited for image geometric operations. The latter are carried out on such data structure without regeneration into image original matrix form (canonical form according to Kunt et al., 1985).

2. A hierarchical list of rectangles. Coding with distortion. In general an image represents a 2D signal. For this reason covering the image by a set of 2D shapes (e.g., rectangles) gives an efficient data compression ratio (Aoki, 1979; Ambraziene and Punys, 1985; in russian: Novikov and Punys, 1980). "Covering" here is understood as follows; image black regions are covered by black efficient rectangles, and image white regions – by rectangles of white colour. Any rectangle may be represented by the coordinates of the left top corner and sizes of two edges. The efficient rectangle is the one, for which quadruple (x, y, l_x, l_y) occupies less storage space than the pixels of the rectangle on the original image. The pyramidal data structure is used for efficient rectangle building (Aoki, 1979).

The values from the set $\{0, 1\}$ are assigned to the elements of the plane No.0, i.e., the bottom plane contains the original image $2^k * 2^k$ pixels and the values from the set $\{0, 1, G, L\}$ – to the elements of other k planes. Here G – the grey value and L – labeling value of the appropriate pixel.

The pyramidal data structure reflects nearly exactly the hierarchical principle of image description, used by vision system. On the top of the pyramid (k-th plane) we have the least resolution. The plane consists of one element only, which contains information (rough if the value is equal to G and exact if it is equal to 1 or 0) about the whole original image.

The resolution increases progressively going down the pyramid and the image is detalized up to the smallest details on the zero plane.

The pyramidal data structure is an intermediate one which allows us approximately to "inspect" the coarse parts of the image without multiple rescanning of its elements. This reduces the time of analysis. Though, such representation occupies a large space in memory, and it is not suitable for creating arxives of images (video-data bases). Ordered lists of efficient rectangles mentioned above are much better for this aim.

On analyzing the pyramid from top to bottom the automatically ordered description of efficient rectangles included into the list according to its size, i.e., according to the area occupied, is obtained. At the beginning of the list big rectangles, which represent the rough description of an image, are placed. Following them smaller rectangles, enabling a progressive detalization, are stored. The principle of hierarchical representation of the object shape is observed here again. But the latter representation is only an approximate reflection of the image description hierarchical principle carried out by living vision system. On the other hand, such a list is suitable for storing and processing by a computer of von Neumann's architecture.

The threshold value for rectangle discrimination into efficient and nonefficient ones depends on the size of the original image. It equals to 32 for the image of size $256 * 256$ pixels, and to 36 for the image of $512 * 512$ pixels. We used format of $512 * 512$ pixels, but for the sake of programming convenience we used to encode separately four quadrants of the image and appropriate threshold value 32. The compression ratio of the whole image is achieved of a little lower value comparing with the values which might be obtained according to this technique.

The example of the image coding by a set of black and white efficient rectangles is shown in Fig.1b. Experiments were carried out with a help of a system of programs (Ambrazienė and Punys, 1985) on the base of the modified algorithm of Aoki (1979).

NOTE. The image regenerated from the sets of efficient rectangles (b) and from the complement (c) coincide with the original images (a).

Compression ratio of the image with a fragment of the text by Lithuanian poet J. Maironis and a girl's portrait achieved by image covering with black and white efficient rectangles (Fig.1b) are equal to 0.0956 and 0.0508 bit/pixel respectively. It may be seen from Fig.1b, that the quality of the images regenerated from the set of efficient rectangles is not very good, but the shape of objects is expressed quite well.

3. Coding errors. Coding without distortion. The pixels outside the efficient rectangles represent the complement to the set of pixels, covered by efficient rectangles. The complement is an error of the coding described above. The error consists of black pixels outside the black efficient rectangles and white pixels outside the white efficient rectangles.

We have carried out quantitative and qualitative investigation of coding errors which were left without sufficient attention in Aoki's (1979) report. Some examples of coding errors are given in Fig.1c (only black region coding errors can be seen there). The error ratio equals to the complement pixels amount devided by the size of original image in pixels. This ratio is equal to 0.138 and 0.06 for the text and for the portrait of the Fig.1c respectively.

The given technique allows us to perform image coding without distortions. This is achieved by adding one more list

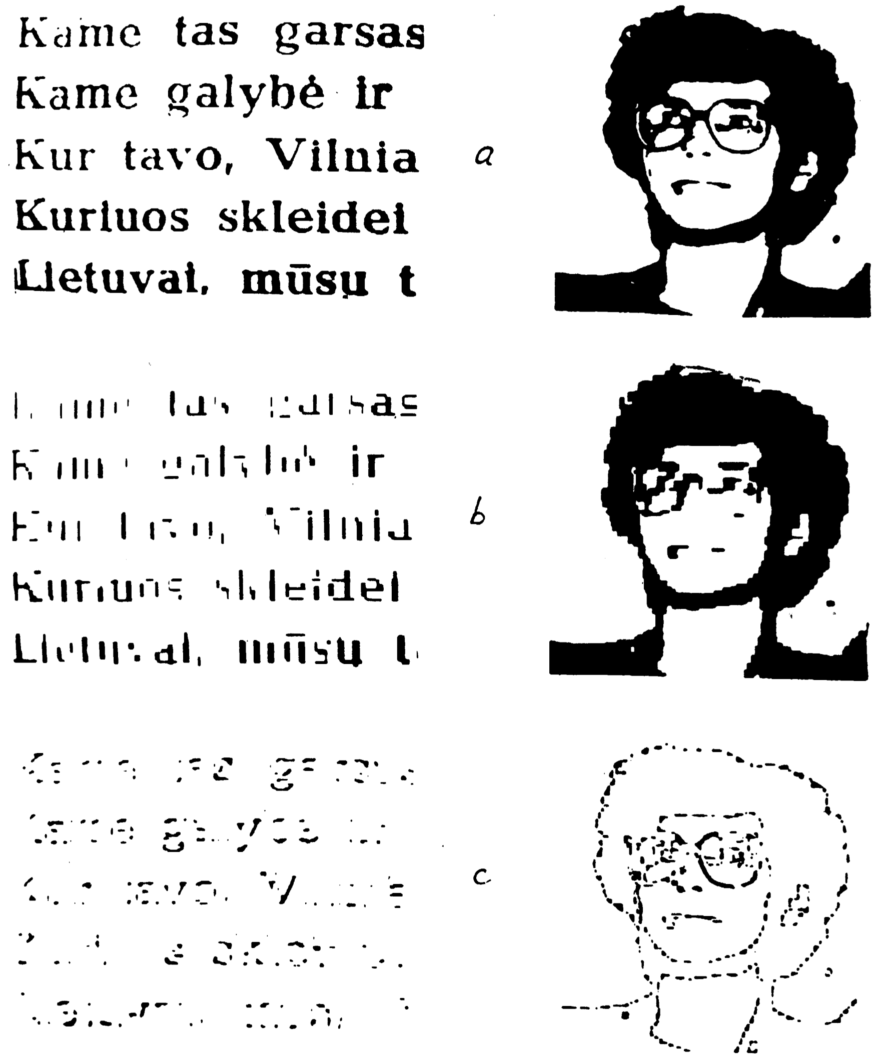

Fig.1. Examples of image coding by rectangles, the coding errors and coding without distortion: a – original images; b – regenerated images from a set of efficient rectangles; c – black pixels of the complement (coding errors).

Table 1. Coding results with distortion

Test image	Efficient rectangles			compression ratio bit/pixel	Errors ratio
	The amount of black B	The amount of white W	Total amount R		
File 0;1(Aoki)	257	252	509	0.062	0.055
" 2 "	183	285	468	0.057	0.063
" 4 "			593	0.072	0.083
" 5 "			799	0.098	0.145
" 6 "	215	384	599	0.073	0.095
" 7 "			351	0.043	0.045
" 8 "			872	0.106	0.138
text (our Fig.1)			783	0.096	0.138
portrait "	210	206	416	0.051	0.060

to the list of the efficient black and white rectangles, which contains ordered binary values of complement set. One binary digit is reserved for every value from the set $\{0, 1\}$ in the list of the complement. Compression ratio of coding without distortion is equal to 0.23 and 0.11 for the image of the text and for the portrait (Fig.1a) respectively. Image regeneration programs provide an image display without distortion according to the information stored in the lists of efficient rectangles (Fig.1b) and in the list of complement (Fig.1c).

Some experiments were carried out to compare the results of coding by using the image, selected from standard patterns of CCITT. The images were scanned by vidicon camera digitizer with $512 * 512$ pixels directly from the illustrations of the journal (Aoki, 1979). The scanned image approximately corresponds to half of the image used in the latter article. The obtained images were distorted because errors of the input device. But these differences are not essential and our coding results are close to those in Aoki's article (see Table 2).

Table 2. Coding results without distortion

Test image	Complement pixels			Compression ratio	
	The amount of black C_b	The amount of white C_ω	Total amount C	our	Aoki (1979)
File 0;1(Aoki)	7222	7218	14440	0.12	0.11
" 2 "	8845	7721	16566	0.12	0.18
" 4 "			21687	0.16	0.14
" 5 "			37940	0.24	0.46
" 6 "	13334	11532	24866	0.17	0.27
" 7 "			11686	0.09	0.13
" 8 "			36091	0.24	0.40
text (our Fig.1)			36535	0.23	–
portrait "	8067	7613	15680	0.11	–

The coding errors are mainly caused by the contours and the incline lines (see Fig.1). The coding errors form a relatively small part even for thin-detailed images. For example, the percent of the total amount of black and white pixels, representing errors, equals 9.5% and 13.8% for the images of the text (file 6 and file 8 in Aoki, 1979). The images with thin curves are the most difficult task for covering with efficient rectangles. Error percent for the map (file 5) equals 14.5% (see Table 1).

4. Coding for video-data base

4.1. Two trends in video-data coding. Recently two trends have been formed in the field of video-data coding: 1) coding with the purpose of maximum data compression; 2) coding for representing information in video-data base. The effort to built a data structure which allows us to carry out image processing not on its original pixels array, but on its representation in data structure, prevails in the 2nd trend. The compression ratio is not so important as in the 1st trend of coding.

Two generations in both trends can be distinguished (following Kunt's et al. (1985) methodology). The techniques of the 2nd generation (in both trends) differ from the first one in that they reflect the features of a living vision system.

Freeman's (1974) chain coding is an example of the 1st generation technique of the 2nd trend.

Video-information representation by the recursive structures (see Pavlidis, 1982; Samet, 1984; Punys, 1985; Aleksandrov and Gorsky, 1985) belongs to the 2nd generation of coding for video-data bases because of the hierarchical feature of these structures. This feature, however is characteristic to the non-recursive structure (hierarchical list of rectangles) too. Processing of the latter is discussed in the following chapter.

It should be noted that the 2nd generation techniques for both trends, which have developed independently, began to merge (Kunt, 1986, the split and merge procedure).

4.2. Image geometric transformations. The hierarchical non-recursive data structure is used in the form of lists of ordered rectangles and a complement. The transformations are mainly carried out with compressed image representation.

The algorithms of fundamental operations are given below. The image rotation requires a lot of calculations according to classical techniques (Castleman, 1979).

Rotated co-ordinates (x_r, y_r) with respect to the origin of axes x and y are defined as:

$$\begin{aligned} x_r &= x\cos q - y\sin q, \\ y_r &= y\cos q + x\sin q. \end{aligned} \tag{4.1}$$

According to the 1st classical technique the formula (4.1) should be used 2^{2k} times for images rotation of size $2^k \times 2^k$.

In the case of a binary image it is sufficient to rotate either black or white regions, i.e., the number of uses of the formula (4.1) would be equal to the number of black or white pixels on the original image. These numbers depend on the image content. The black regions of the text and the portrait (Fig.1a) occupy approximately half of the entire area of the image, i.e., the number of black pixels is equal to $2^{2k-1} = 131071$ (for $k = 9$).

If the image is represented by the list of rectangles (coding with distortion) the rotation operation is carrying out according to the **algorithm 1:** 1) Define the rotated co-ordinates (x_r, y_r) only of four vertices for every rectangle according to the formula (4.1); 2) Connect the resulting points on the rotated image with straight lines, and fill the area inside the rectangle with uniform elements according to the colour of the rectangle.

For a binary image it is enough to rotate either black or white areas. So, using rectangular coding with distortion, the formula (4.1) is applied only $4N$ times ($N := B$ or $N = W$, see Table 1). For realistic images $N \ll 2^{2k-1}$ in this case.

The examples of rotated images are given in Fig.2. The number $B = 210$ for this image (see Table 1). Consequently, the formula (4.1) has been applied only 840 times instead of 131071 in the classical case.

In the case of rotation of the image, encoded without distortion, **the algorithm 2** should be applied: carry out the 1st and the 2nd steps of the algorithm 1 for selected colour of rectangles; 3) Mark complement points on the original image; 4) Scan the marked points in the order they were included into the list of complement pixel values, and define the pixel co-ordinates (x, y) for the selected colour. Write the pixel colour (the value with co-ordinates (x, y)) on the rotated image with co-ordinates (x_r, y_r) determined by the formula (4.1).

The formula (4.1) is applied here $(4N + M)$ times. According to the selected colour, $N = B$ or $N = W$, and $M = C_b$ or $M = C_w$ (see Table 2). The experiments, given in the Table 1 and 2, show that for realistic graphical images $4N + M \ll 2^{2k-1}$ (for $k = 9$). For example, for the portrait (Fig.1a), encoded without a distortion, the number $B = 210$ and $C_b = 8667$. Thus, for its rotation it would be necessary to use (4.1) only 8907 times, instead of 131071 according to the 1st classical algorithm.

Fig.2. The examples of rotated images.

By using homogeneous co-ordinates of the point and the composition of two-dimensional geometric transformations (Foley and van Dam, 1982) scaling, rotating and shifting (or some other compositions of transformations) can be performed by multiplying every point $(x, y, 1)$ in homogeneous co-ordinates by an appropriate matrix of size 3×3. This increases the computational effectiveness as compared with the sequential application of scaling, rotating, and shifting of the image by transforming ordinary co-ordinates (x, y). In the case of applying homogeneous co-ordinates, however, data structure in the form of the lists of rectangles and a complement gives the possibility to increase the computational efficiency. This is achieved due to the fact that the multiplication of the point of homogeneous co-ordinates by the matrix of transformations composition is carried out not for every pixel of the original image but only for the vertices. The algorithm of such transformation is similar to algorithm 2. For compositional transformation the operation of multiplication of corresponding points in the homogeneous coordinates by the matrix of size 3×3 is used instead of formula (4.1).

By using the specific features of the data structure being analysed some operations of geometric transformations can be accomplished more efficiently than by formal application of the technique described above based on the homogeneous co-ordinates and algorithm 2. For example, the composition of shifting and scaling transformations (with respect to the point of co-ordinates origin) can be carried out by simple recalculation of the parameters in every quadruple according to the formula:

$$(s_x(x + x_T),\ s_y(y + y_T),\ s_x l_x,\ s_y, l_y), \tag{4.2}$$

where x_T, y_T are the shift values, s_x and s_y are the scaling coefficients along x and y directions respectively. If $s_x \cdot s_y < 0.5$ it is not necessary to estimate a list of complement pixels. In this case it is sufficient to use the formula (4.2) only for the list of rectangles the colour of which includes the information about the objects on the image. When the number of rectangles is N, the composition of shifting and scaling operations involves $N\ (a_2 + m_2)$ operations according to (4.2). Here $a_2 = 2$ (the number of addition operation), and $m_2 = 4$ (the number of multiplication operations), required for the modification of rectangle parameters according to (4.2).

If the length l_x and the width l_y are not taken into account, and the composition of transformations is carried out for every vertex of rectangle using the formula

$$\begin{aligned} x' &= s_x(x + x_T), \\ y' &= s_y(y + y_T), \end{aligned} \tag{4.3}$$

then the number of operations is equal to $4N\ (a_3 + m_3)$. Here $a_3 = 2$ (the number of addition operations), and $m_3 = 2$ (the number of multiplication operations), which are required for the definition co-ordinates of one vertex according to (4.3). Consequently, the advantage in the number of operations is defined by the ratio $4 * N * (a_3 + m_3)/N * (a_2 + m_2) = 2.67$. If $s_x > 1$ and $s_y > 1$, the compositional transformation according to the formula (4.2) becomes more complicated. The simplest way for solving this problem (but not the most efficient with respect to compression ratio) includes the formation of an additional list of rectangles (with a size $s_x \times s_y$) representing enlarged pixels (the squares of size 1×1), stored in the complement list.

The examples of the shifting and scaling operations for the portrait are given in Fig.3.

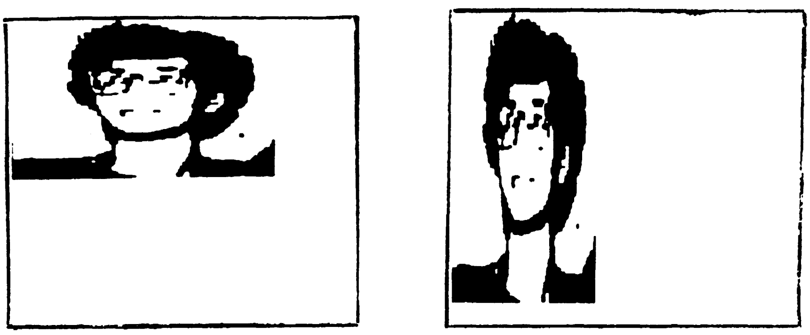

Fig.3. The examples of shifting and scaling operations.

5. Conclusions. The pyramidal data structure, which reflects the hierarchical principle of video-data representation in a living brain quite exactly, allows as to built another hierarchical structure: ordered sets of rectangles and a complement. The latter structure imitates the action of a living vision system worse than a recursive structure. However, it is better suited for video-data processing on a usual computer. Hierarchical data structure in the form of lists of rectangles and a complement allows us to achieve high compression ratio (coding without distortion), and it is simultaneously well suited for carrying out image geometric operations. The latter feature meets the image coding requirement for video-data bases. The image geometric transformations are being carried out especially effectively on image approximate representation by a list of efficient rectangles. The exact representation allows us to carry out these operations more effectively too, than the carrying out of these operations according to classical algorithms on entire original image. The coding technique treated in this report, is well suited for: 1) recognition of rectangles on a layout of an integrated circuit; 2) editing (simultaneous composition) of graphical and textual information obtained by a video sensor.

Acknowledgments. The author is gratefully indebted to his young colleagues A. Ambrazas and D. Ambrazienė for helping in the skilful programming and carrying out of the experiments.

REFERENCES

Aleksandrov, V.V., and N.D.Gorsky (1985). *Image Representation and Processing: The Recursive Approach.* Nauka, Leningrad. 192pp. (in Russian).

Ambrazienė, D., and J.Punys (1985). Image coding by a list of rectangles and the errors of coding. In *Automatische Bildverarbeitung (Kurzreferate 1.Internationale Fachtagung,* Berlin, 17 und 18 Okt. 1985). Kammer der Technik, Berlin. pp.C14/1–4.

Aoki, M. (1979). Rectangular region coding for image data compression. *Pattern Recognition,* **11**, 297–322.

Castleman, K.R. (1979). *Digital Image Processing.* Prentice-Hall Inc., New Jersey. 429pp.

Essen, D.C., and J.H.R. Maunsell (1983). Hierarchical organization and functional streems in the visual cortex. *Trends in Neurosciences,* **6**(9), 370–375.

Foley, J.D., and A.van Dam (1982). *Fundamentals of Interactive Computer Graphics.* Addison-Wesley Publ. Comp., Massachusetts.

Freeman, H. (1974). Computer processing of the line drawing images. *Computing Surveys,* **6**(1), 57–97.

Glezer, V.D. (1985). *Vision and Thought.* Nauka, Leningrad. 246pp. (in Russian).

Hubel, D.H., and T.N.Wiesel (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J.Physiol,* **160**(1), 106–154.

Klinger, A. (1971). Patterns and search statistics. In J.S. Rustagi (Ed.), *Optimizing Methods in Statistics.* Academic Press, New York. pp. 303–337.

Kunt, M., A.Ikonomopoulos, and M.Kocher (1985). Second-generation image coding techniques. *Proc. IEEE,* **73**(4), 549–574.

Kunt, M. (1986). Recent progress in high compression image coding. In *Pattern Recognition (Proceedings of the 8th ICPR,* Paris, 27-31 Oct. 1986). pp. 1018–1025

Novikov, V.M., and J.K.Punys (1980). The algorithm of covering of black-white image by rectangles. In P.Kemėšis (Ed.), *Techninė kibernetika,* Proceedings of Conf. of the Republic, Kaunas, Lithuania, 1980. Council of Editors of the Higher Education Ministery of Lithuanian SSR, Vilnius. pp. 23–24. (in Russian).

Palmieri, G., G.A.Oliva and M.Scotto (1971). C.R.T. Spot-follower device for eye-movement measurements. *Kybernetik,* **3**, 23–30.

Pavlidis, T. (1982). *Algorithms for Graphics and Image Processing.* Springer-Verlag, Berlin. 446pp.

Podvigin, N.F., A.M.Kuperman, V.K.Mirtov, and I.V.Chueva

(1975). On image description on the level of lateral geniculate body. *Neurophisiologia,* **7**, 35–40. (in Russian).

Podvigin, N.F. (1979). *Dynamical Properties of Neurons Structures of a Vision System.* Nauka, Leningrad. 157pp. (in Russian).

Punys, J. (1985). Images representation by quadrees and operations on them (Survey). In *Automatische Bildverarbeitung (Kurzreferate 1.Internaationale Fachtagung, Berlin, 17 und 18 Okt. 1985).* Kammer der Technik, Berlin. pp. T20/1–4.

Punys, J.K. (1980). Data compression using a texture scanning by rectangles. In P.Kemėšis (Ed.), *Techninė kibernetika,* Proceedings of Conf. of the Republic, Kaunas, Lithuania, 1980. Council of Editors of the Higher Education Ministery of Lithuanian SSR, Vilnius. pp. 21–22 (in Russian).

Rosenfeld, A. Ed. (1984). *Multiresolution Image Processing and Analysis.* Springer, Berlin (W). 385pp.

Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Comp. Surveys,* **16**(2), 187–260.

Received March 1990

J. Punys is a senior research Scientist of Automation and Telemechanics department at the Kaunas Polytechnic Institute. He received the Degree of Candidate of Technical Sciences from Kaunas Polytechnic Institute, Kaunas, Lithuania, in 1974. He was an Associate Professor in the departments of Control Systems and Automation and Telemechanics up to 1989. His research interests include the areas of digital image processing, 2D and 3D shape representation using progressive refinement in the machine vision and computer graphics, video-data structures and video-data bases.