John von Neumann Society for Computing Sciences
Computer and Automation Institute, Hungarian Academy of Sciences
SZKI Pixel Kft.

PROCEEDINGS OF THE THIRD HUNGARIAN WORKSHOP ON IMAGE ANALYSIS

Budapest, April 29-30, 1991

Edited by: Dmitry Chetverikov and Géza Álló

Table of Contents

COMPUTER VISION: GOALS, MEANS, AND PROBLEMS[1]

Dmitry Chetverikov

Computer and Automation Institute,
Hungarian Academy of Sciences
Budapest, P.O.Box 63
H-1518 Hungary

Abstract: In this short note, the author presents in an informal way his personal views on some general problems facing computer vision. The main issues discussed are the ultimate goal of computer vision, computer versus human vision, heuristic versus theory, and significance and verification of experiments. The author responds to the recent papers by R.Haralick [1] and K.Price [2] who dwell on similar problems, agrees with them on some points and argues about others.

1. Computer vision as a science.

According to Haralick [1], the theory of computer vision is expected to provide "laws and principles by which computer algorithms can be designed to solve a variety of vision tasks", including general 3D scene understanding. These laws and principles are to be mathematically derived from the initial problem statement, assumptions about the particular vision phenomenon considered, and the general laws of computer vision. The aim of the experiment is either to obtain data that would hopefully facilitate the derivation of a theory concerning the phenomenon being studied, or to test an existing theory. A report on an experimental study in computer vision is expected to contain "clear de-

[1] Reprinted from Proc. 3rd Int. Conf. on Automated Analysis of Images and Patterns, Leipzig, GDR, 1989.

scriptions of controlled situations under which the experiments are performed, a precise statement of the algorithm being used, and a statement of the results which includes some measure of the certainty of the stated results".

To this clear statement of requirements we can only add that like any serious theory, the theory of computer vision must have a predictive power, i.e. be able to predict the behavior of a vision algorithm in given circumstances. Experimental results must be general enough, statistically significant, and reproducible.

In his paper [1], Haralick concludes that presently computer vision fails to meet these basic requirements any hard science is expected to meet. Price [2] comes to a similar conclusion. Here are some of the main problems facing computer vision.

- Computer vision lacks clear and consistent conceptual basis and terminology.

- There is a gap between the theory and the practice of computer vision; heuristic and ad hoc methods dominate.

- In most cases, no theoretical criteria exist to evaluate the performance of vision algorithms.

- Often, there is no way to automatically set the optimal values for the parameters of the algorithms.

- A lot of experimental results are statistically insignificant and irreproducible.

- It is very difficult to share computer vision software; most of us prefer implementing our own programs instead of using the existing ones.

- In general, we pay too little attention to each other's results; our efforts are separate and unrelated.

Both Haralick and Price are dissatisfied with the present situation and suggest certain remedies. Before any attempt to change the situation is made, it is necessary to agree upon the ultimate goal of computer vision, define the object of research, and clarify basic concepts and terminology. In the rest of this note, we will discuss the first two of these issues. We will also touch some of the above mentioned problems.

2. The ultimate goal of computer vision.

Computer vision aims at studying and creating artificial vision systems. "Studying" means understanding the mechanisms of vision, while "creating" means developing systems for solving particular vision tasks. These two sorts of scientific activity must cohere. Currently, they seem to be pursuing different goals. Those who "study" often do not bother with the feasibility of their methods, while those who "create" pay little attention to understanding. There are several reasons for this. One of them is the engineering versus scientific mentality dualism typical for research people in general. Another one is the "application pressure" and the extensive character of the development of this immature field of science.

Still there is a deeper reason lying in different attitudes toward the aim of computer vision. It is usually believed that the ultimate goal of computer vision is to duplicate the vision capabilities of humans. This pursuit is different from the one that aims at developing a theory to support the design of systems that solve vision tasks. The key question is whether one agrees that human vision is superior in all cases and that duplicating such sophisticated capabilities as vision, intuition etc. is possible. If not, emulating human vision is not, in the long run, a valid pursuit. We must keep in mind that any electronic device, software etc. aimed at artificial vision will differ in operation and performance from human vision.

Personally, I do not believe in the possibility of the duplication, and I do not think that the object of the computer vision research is human vision. I do not share the purely pragmatic, application-centered attitude either. Most of us want to build systems that solve real problems with real input data, but also want to understand things. I do not believe we want computer vision to be a collection of ad hoc tricks (kept in secret by companies, the military, etc.)

We study the physical reality with the help of vision sensors, devices and mathematical structures capable of processing and interpreting, on a high level of abstraction, information provided by the sensors. The aim is to teach such devices to "see", i.e. solve sophisticated vision problems. Trying to reach this aim we learn from human vision. The computer vision research and the human vision research profit from each other. However, there is no guarantee that the mechanisms that are used by human vision are useful for computer vision also. For example, it has been discovered that in the spontaneous texture perception by humans, the second order image statistics play a dominant role [3]. This mechanism is effectively used in computer vision also. On the other hand, the discovery of textons [4] - intensity features which are the structural elements of human texture perception - seems to have no impact on texture analysis by computer. The crucial point here is the possibility of extracting the necessary features from the image.

The ultimate goal of computer vision is <u>not</u> to compete with human vision, but to develop a coherent theory facilitating the design of hardware and software to solve vision tasks. Pursuing a wrong goal we will sooner or later find ourself in the situation familiar to the artificial intelligence community: the "theoreticians" will reject any working vision system as not being "the true artificial vision", while the "pragmatists" will do the same because the system does not solve their particular tasks.

3. On some problems.

What tasks do we mean when we speak about a theory supporting the solution of vision tasks ? Haralick [1] emphasizes that more effort should be put into "the definition of canonical computer vision subproblems and ... their optimal solution". Without the definition of canonical subproblems developing a computer vision theory is hardly possible. The set of canonical subproblems would reflect the present state-of-the-art. It could be updated when new ideas enter the field.

Some of the canonical subproblems are well-defined, some (e.g. edge detection) are not. It may be argued that any well-defined subproblem has nothing to do with image analysis unless an adequate image model is given. (Which rarely happens.) Or the statement of a subproblem assumes that certain information is available. (To provide this information is another, perhaps even harder problem.) I agree that life is not easy. However, at the moment I see no other way.

Haralick [1] insists that computer vision problems must be solved under some criteria of optimality. He argues that unless a problem is stated as some kind of optimization problem one does not really know how robust a solution to this problem is. I think that in computer vision research, this aim is extremely difficult to reach since quite a few problems seem to loose their "original" meaning when expressed as optimization problem. Optimality is very much goal dependent. This may be misleading, especially when the underlying assumptions are irrealistic. What shall I do with, say, those "optimal" threshold selection techniques that fail in practice ? Anyway, we should try to give clear statement of the problem, the assumptions, the proposed methods, and the results. Stating the problem in an optimization manner would help one judge upon the applicability of a solution. It could also facilitate the automatic setting of parameters of vision algorithms which I consider to be one of the most important goals.

Price [2] argues that simple problems must not be solved with too complex methods. If a practical solution exists outside the realm of computer vision, this solution should be considered. A classic example of such a situation is the so called "bin picking" task in the industrial machine vision. Why to throw the parts into a bin before picking them by a robot ? Obviously, as an industrial application problem this task is nonsense since the industrial environment is highly organized. (Well, not everywhere.) However, it could make sense in a different environment. Designing a vision system capable of handling multiple overlapping 3D objects is a challenging scientific problem.

It is common knowledge that the generality and the certainty of experimental results in computer vision suffer very often from insufficient image data and the lack of common data sets for the experiments. Apparently, everyone is happy with his/her own algorithm as if it were sufficient in itself. If the algorithm is intended for some particular application and produces the desired results for that application, I would not object unless the algorithm is published as a **scientific** result. But in most cases it is. A few "typical" images of a certain kind are selected and the parameters are properly tuned for these images. Price [2] suggests that vision algorithms be tested on more (at least six) **real-world images of different origin** in order to make the limits of the algorithm clearer. He implicitly objects using **synthesized images** because "natural" images are usually subject to noise, distortions etc. in a very much different and often unpredictable way. However, using synthesized images appears to be the only direct way to control the data set. Another approach is to start with simple images of a very good quality moving gradually toward more realistic input data. In any case, an explicit description of the procedure used to set the program parameters should be given.

Finally, I would like to emphasize the importance of detailed theoretical and experimental analysis and comparison of different approaches to the same computer vision problem. The theoretical analysis is at least as difficult as desirable since the computer vision algorithms are very complicated. The experimental comparison is an excellent way of sharing experience about the **real** strength and limits of various methods - the experience you will hardly find in books.

Conclusion.

Computer vision is a young and immature science currently being in the period of extensive development: accumulating knowledge, trying to work out its conceptual basis and terminology, designing heuristic tools, extending the application area. I hope we will put more effort to go to the intensive period, the period of the development of coherent and unifying theories, the period of understanding. Otherwise, it will become clear that as a hard science computer vision is an "ill-posed problem".

References.

[1] R.M.Haralick, "Computer Vision Theory: The Lack Thereof", Computer Vision, Graphics, and Image Processing **36**, pp.372-386, 1986.

[2] K.Price, "Anything You Can Do, I Can Do Better (No You Can't)", ibid., pp.387-391.

[3] B.Julesz, "Experiments in the Visual Perception of Texture", Scientific American, Vol.232, No.4, pp.34-43, 1975.

[4] B.Julesz, "Textons, the elements of texture perception, and their interaction", Nature, Vol.290, pp.91-97, March 1981.

Image analysis research and development in MTA SzTAKI

Dmitry Chetverikov

Computer and Automation Institute,
Hungarian Academy of Sciences
Budapest, P.O.Box 63
H-1518 Hungary

Abstract: The author is with Computer Vision and Robotics Lab
Computer and Automation Institute (MTA SzTAKI). In this paper,
gives a brief survey of the recent activities of the lab in dig
tal image analysis, pattern recognition, and their application.

1. Introduction.

Image analysis research in Hungary dates back to the earl
seventies when a few laboratories were created in research insti
tutes and universities, mostly in Budapest. Currently, the acti
vities of the Hungarian image processing community are still con
centrated in the capital of the country, although considerabl
work is done in some other towns as well, e.g. at the Universit
of Szeged. Usually, these activities are dominated by the applie
research and development. This shift toward applications might b
considered to reflect an international trend in computer visio
which is particularly typical for small countries. To a certai
extent, this trend is explained by the recognition of the inhe
rent problems the scientists face in their attempts to lay th
theoretical foundations of computer vision [1]. However, in Hun
gary the advanced image analysis research suffers also from th
increasing financial pressure and the low, by the Western stan
dards, technological level of the industry that has neither mone

to support advanced research, nor capabilities to adapt its results.

Despite these problems, Hungary was the first East European country to become a member of IAPR, the International Association for Pattern Recognition. The results of the research and development activities of the Hungarian image processing community are regularly published and presented at international conferences and the Hungarian Workshop on Image Analysis [2,3]. Some of these results were discussed in a previous survey [4] devoted to the industrial machine vision in the East European countries. The present paper is a brief overview of the recent work done in one of the leading Hungarian research laboratories, the Computer Vision and Robotics Lab of Computer and Automation Institute (MTA SzTAKI).

2. Research and development in Computer and Automation Institute.

Computer Vision and Robotics Laboratory was organized in this academic research institute about fifteen years ago by Prof.Tibor Vámos. Today, it has 12 members of staff, most of them being research engineers. The lab is involved in research and development in image analysis, pattern recognition, computer graphics, and robotics. Applied machine vision research and development is presently one of its main activities, but some fundamental research is done as well. As far as image processing and analysis is concerned, the lab has gained considerable experience in 3D object recognition, 2D shape analysis and recognition, image enhancement, texture analysis, motion analysis, and applications, mostly in industry. Usually, the team concentrates on software, although hardware is also developed when necessary.

Here is a brief account of the recent activities related to image analysis and recognition.

By the end of the seventies, a 3D object recognition software system was developed that was capable of recognizing polyhedral objects [5,6]. This experimental system for 3D scene analysis incorporates the well-known Mérô-Vassy edge detector [7] designed in the framework of this study, a contour following and approximation algorithm, a syntactic recognition program using pattern grammars, and a geometrical modelling and animation package. Experiments with programmed illumination [8] have also been carried out.

Later on, a software system for flat workpiece identification from gray-scale images has been created which exploited some original ideas and used a novel contour matching procedure [9,10]. This research resulted in the design and production of VM-02, a microcomputer-based stand-alone industrial image analysis and object recognition module [11] which was one of the first machine vision modules in Eastern Europe to be produced. The module was used for real-time recognition, inspection and gauging of industrial objects which can be distinguished by their outer and inner contours. Also, it was coupled with a vibratory bowl feeder to determine the type and position of shaft-shaped industrial parts and sort out faulty parts [12].

At the same time, a research project was launched whose aim was to investigate some fundamental properties of image textures. A series of papers was published devoted to various aspects of analysis of homogeneous textures, such as anisotropy [13] and regularity [14,15]. The texture anisotropy features proposed in [13] were successfully used for rotation-invariant texture discrimination [16]. Another study of this project resulted in the development of a new approach to defect detection in textures [17,18].

Although there was no persistent research in low-level image processing, some methods were studied or developed when necessary. For example, the recursive version of the Canny edge detector proposed by Deriche [19] was implemented and analyzed [20] as a tool that is particularly suitable for multiresolution image analysis.

The edge detector was applied to image sequences showing the motion of a spermatozoa population. This particular motion estimation problem triggered a more general study in motion analysis of dense ensembles of small-size, oriented objects. A novel technique was developed and implemented for matching two frames of a motion sequence of an ensemble [21,22,23]. A feasibility study was conducted to investigate the application of this technique to motion estimation of a spermatozoa population [24].

Presently, Computer Vision and Robotics Laboratory is involved in the development of a prototype machine vision system for the shoe and upholstery industry. The first version of the high resolution software and hardware system for automatic segmentation of images of marked hides was presented in [25,26]. The system is aimed at the automation of layout design and cutting of hides.

3. Summary.

We have presented some of our recent activities in image processing and pattern recognition. This brief survey is by no means comprehensive. However, we hope that it gives the reader a correct impression of the scope and the scientific and engineering level of our research and development.

References.

[1] D.Chetverikov, "Computer vision: goals, means, and problems", Proc. 3rd International Conference on Computer Analysis of Images and Pattern, Leipzig, GDR, 1989, pp.11-14.

[2] Proc. of the 1st Hungarian Meeting of the Researchers in Image Processing and Pattern Recognition, MTA SzTAKI Studies 176/1985, Budapest, 1985.

[3] Proc. of the 2nd Hungarian Workshop on Image Analysis, SzTAKI Studies 206/1988, Budapest, 1988.

[4] D.Chetverikov, "Industrial computer vision in the East Eu pean countries", Proc. AMPS-COMPCONTROL 85, Budapest, 19 Vol.III, pp.560-569.

[5] T.Vámos, M.Báthor, L.Mérô, "A knowledge-based robot visi system", Proc. 6th International Joint Conference on Artifici Intelligence, Tokyo, 1979, pp.920-922.

[6] L.Mérô, "An optimal line following algorithm", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.PAMI-3, No. pp.593-598, 1981.

[7] L.Mérô and Z.Vassy, "A simplified and fast version of th Hueckel operator for finding edges in picture", Proc. 4th Inte national Joint Conference on Artificial Intelligence, Tbilis 1975, pp.650-655.

[8] T.Vámos, M.Báthor, "3D complex object recognition using pro grammed illumination", Proc. 5th International Conference on Pat tern Recognition, Miami Beach, 1980, pp.1091-1093.

[9] L.Mérô, D.Chetverikov, M.Báthor, "Bus-body identificatio oriented two-dimensional recognition system", Proc. Manufa cont'80, Budapest, 1980, pp. 233-240.

[10] L.Mérô, "An algorithm for scale- and rotation-invariant re cognition of two-dimensional objects", Computer Graphics and Im age Processing, 15, pp.279-287, 1981.

[11] P.Solt, "Industrial vision module and its application fo automatic painting", Proc. 1st KDT International Conference o Automated Image Processing, Berlin, GDR, 1985, pp.B6/1-5.

[12] D.Chetverikov, "Coupling a vision system with a bowl feeder", Proc. 1st KDT International Conference on Automated Image Processing, Berlin, GDR, 1985, pp.B1/1-4.

[13] D.Chetverikov, "Textural anisotropy features for texture analysis", Proc. IEEE Comp. Soc. Conference on Pattern Recognition and Image Processing, Dallas, 1981, pp.583-588.

[14] D.Chetverikov, "Measuring the degree of texture regularity", Proc. 7th International Conference on Pattern Recognition, Montreal, 1984, pp.80-82.

[15] D.Chetverikov, "Generating contrast curves for texture regularity analysis", submitted to Pattern Recognition Letters, 1990.

[16] D.Chetverikov, "Experiments in the rotation-invariant texture discrimination using anisotropy features", Proc. 6th International Conference on Pattern Recognition, Munich, 1982, pp.1071-1073.

[17] D.Chetverikov, "Texture imperfections", Pattern Recognition Letters, Vol.6., pp.45-50, 1987.

[18] D.Chetverikov, "Detecting defects in texture", Proc. 5th Scandinavian Conference on Image Analysis, Stockholm, 1987, pp.427-433.

[19] R.Deriche, "Fast algorithms for low-level vision", Proc. 9th International Conference on Pattern Recognition, Rome, pp.434-438.

[20] A.Lerch and J.Fiser, "Experiments with the recursively implemented Canny edge detector", Proc. 3rd International Conference on Computer Analysis of Images and Pattern, Leipzig, 1989, pp.168-171.

[21] D.Chetverikov, A.Lerch and C.Magyar, "Model-based track
of ensembles of small-size objects", Proc. 13th Meeting of
Austrian Working Group for Pattern Recognition, pp.65-81, 1989

[22] D.Chetverikov, A.Lerch and C.Magyar, "Tracking multiple
jects in digital images", Proc. 11th IFAC World Congress, Ta
linn, 1990.

[23] D.Chetverikov, "A data structure for fast neighborh
search in point lists", Proc. 6th Annual Israel Conference
Pattern Recognition and Computer Vision, Ramat Gan, Israel, 198
pp.549-556.

[24] D.Chetverikov and A.Lerch, "A matching algorithm for moti
analysis of dense populations", Pattern Recognition Letter
Vol.11, pp.743-749, 1990.

[25] D.Chetverikov and A.Lerch, "A prototype machine vision sy
tem for segmentation of hide images", submitted to Internation
Journal of Imaging Systems and Technology, 1991. See also the
Proceedings.

[26] A.Lerch and D.Chetverikov, "Detection of turning points
contours approximated by polygons", submitted to 7th Scandinavia
Conference on Image Analysis, Aalborg, 1991.

Automatic analysis of marked hide images for leather industry

Dmitry Chetverikov and Attila Lerch

Computer and Automation Institute,
Hungarian Academy of Sciences
Budapest, P.O.Box 63
H-1518 Hungary

Abstract: In the modern shoe and upholstery industry, the automation of the design and cutting procedures is very desirable. This paper gives a brief description of the operation principles of LeaVis, a prototype machine vision system aimed at processing and segmentation of the images of large hides marked by lines and other symbols that show defects and areas of different quality. The goal of LeaVis is to provide a visual input for an automatic design system which sets trajectories for the knife cutting the hides. This task is significantly complicated by a very high resolution required, the poor quality of the hand-drawn lines and the limited computing resources selected to decrease the price of the future product. We discuss the basic motivations and principles of the vision system design, present some of its algorithms and show examples of their operation.

1. Introduction: the task.

In the shoe and upholstery industry, large hides that may reach a few meters in diameter are usually cut into parts of desired shape (e.g. parts of shoes) by a human operator. Normally, a hide has defects and areas of different quality (quality areas) suitable for different parts. Using his skill and experience, the operator defines the quality areas, detects the defects and controls the cutting trying to optimize the output. Today, a very limited number of expensive computer systems exist that help automatize the visual input and the planning of the cutting process.

In the Computer Vision and Robotics Laboratory, the first version of LeaVis, a prototype machine vision system aimed at analysis of the hide images, has been recently developed in cooperation with a leading Austrian shoemaking company. LeaVis is implemented on a PC AT equipped with two high-resolution line-scan CCD cameras and a general-purpose image processing plug-in card. In the current setup, a human operator draws on the hides the borders of the quality areas, marks the areas by the specially designed stamps and indicates the defects. The vision system scans a marked hide, recognizes the quality areas and the defects and produces a segmentation of the hide image which is passed to the computer-aided design system.

The second version of the system is planned to partly automatize the process of defect tion. In this paper, we discuss the main ideas and algorithms of the present version of I

2. General assumptions and constraints.

The hides have different color. The color of the drawing tool (chalk, etc.) is pr selected for each hide color. An image may contain several hides of the same color. To hides are treated as one piece. The outer contours of hides including possible holes are found.

Fig.1 shows a binarized hide image. Each hide is partitioned by hand-drawn line regions of uniform quality called quality areas (Q-areas), including defect areas. A pa quality area border may formed by the outer contour. Q-areas may be nested in an arb depth.

Each Q-area except the defect (zero quality) areas is marked by a quality mark (Q-r A Q-mark is a specially designed, pre-manufactured standard stamp which is placed insic appropriate Q-area. Areas without Q-marks are zero quality, i.e. unusable areas.

Linear defects are marked by short lines. Pointwise defects are currently indicat small hand-drawn circles, although alternative markers are also possible.

The following constraints are valid for the lines and symbols that may appear on l

- Quality areas are topologically connected domains. Their borders do not inte Hand-drawn contours showing Q-areas are normally closed, but small gaps ma appear.

- The shape of quality marks is designed so as to facilitate their recognition. Q-mar much smaller than Q-areas. They do not touch any other symbol.

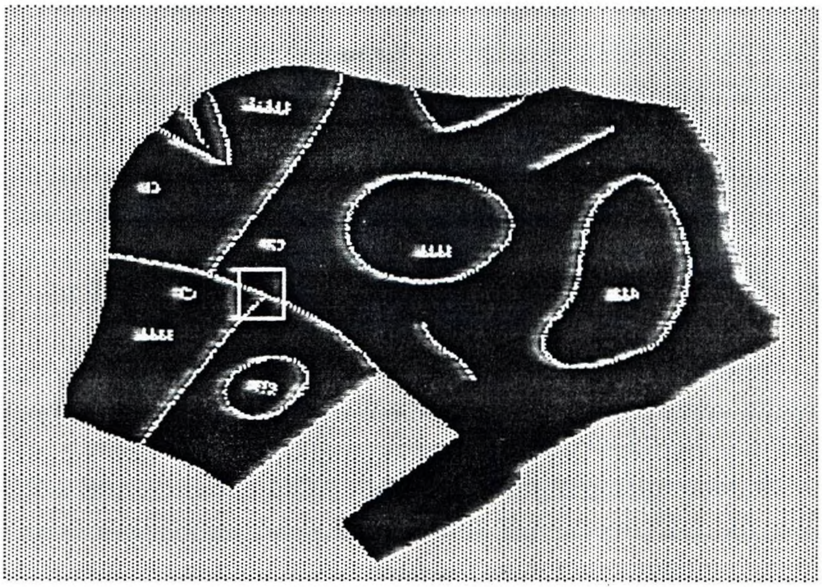


Fig.1. A binarized hide image with Q-marks

- The length of local symbols indicating defects is considerably smaller then that of the Q-area borders. These symbols must differ from Q-marks.

3. Technical considerations.

In principle, a hide image has three basic intensities (colors): the scanner table (background), the hide (leather) and the symbol (lines, marks, etc.) intensities. However, we experienced that, given a large variety of hides, it is very difficult to select the colors so that the three intensities are distinct and the image is reliably thresholded. We decided not to distinguish between the table and the symbol colors, i.e. we used a bilevel approach. The binary approach has the following advantages:

- Reliability

- Easier installation and teaching

- Due to the specifics of the image processor used, more of the time-consuming image processing operations can be executed in hardware. For example, in case of line breaks larger gaps can be filled in; noise removal is more effective, etc.

- Everything that appears on hides (lines, symbols, holes, etc.) can be treated in a uniform way. The recognition system becomes clearer and better structured.

The binary approach has also a number of potential drawbacks:

- There is no straightforward way to distinguish between the lines and the background (scanner table). In the thresholded binary image, each Q-area looks like a separate piece of leather (an "island", see Fig.1).

- Extracting the outer contours of hides is somewhat cumbersome because these contours are composed of pieces of the "island" contours.

- Q-marks look like holes. To differ from the holes, they must have a specific shape.

4. Principles of operation.

The main hardware components of the LeaVis system are the mechanical scanning table, the frame with a fixture for the cameras, two top-view high-resolution line-scan CCD cameras and a PC equipped with the image digitizer and processor card. We use two cameras to obtain the necessary resolution of about 8000 pixels per line. (This resolution corresponds to the precision of the cutting tool.)

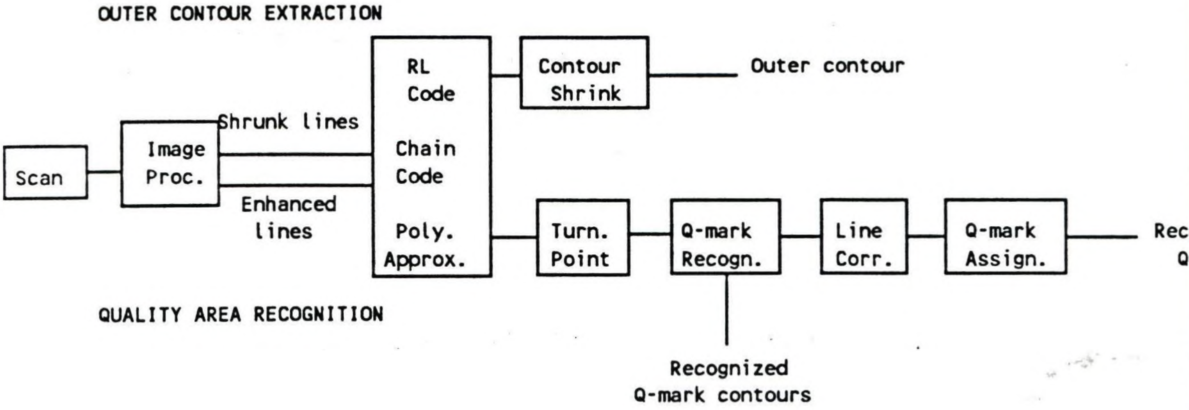


Fig.2. LeaVis: principles of operation.

Fig.2 illustrates the principles of the system operation. LeaVis scans a hide, pro and encodes its digital image and approximates the contours. The main branch of the s recognizes the quality marks, corrects the line contours, assigns the Q-marks to the pro areas and compiles a description of the hide that is suitable for computer-aided design. A rate branch of the program extracts the outer contours of hides.

The **Image Digitizer and Processor** has a number channels that simultaneously p several binary images stored in different bits of the image matrix. A shading correction dure is applied. The original image serves for the system calibration. The processed i result from two different processing chains, the line shrinking and the line enhancemen first one is used for the outer contour extraction while the second one serves for the recog purposes. (The modified, topology-preserving median filter used in the enhancement will be discussed in Section 5.)

The **Run-Length Coder** [1] computes the run-length code of the image. This is do by line during the mechanical scanning. Storing the original image matrix (1 byte per would have been impossible because of the huge memory required, therefore the imag coded on-line and stored in the run-length format.

The **Chain Coder** [2] transforms the run-length code into a sequence of chain describing the contours of the topologically connected components of the image. This c particularly suitable for contour processing and analysis [1]. We preferred it to the ske representation because our images show partitions of hides, i.e. they are intrinsically do oriented. After the Chain Coder, all objects in the image are represented by closed con

The **Polygonal Approximator** [3] yields a polygonal approximation of chain-coded co Apart from the necessary contour format, this program provides a further significant data co sion which is necessary for fast processing.

In the main branch of the program, the **Turnpoint Finder** [4] detects drastic dire changes (turning points) on the polygonal contours. The turnpoints are feature points cating the ends of lines. These points play the key role in line correction and quality mar ognition, hence their precise detection is of crucial importance.

The **Quality Mark Recognizer** processes and recognizes quality marks of specified s This module selects the candidate contours, finds their turnpoints and matches them ag the corresponding points of the standard Q-marks. The recognized marks are then excluded from further contour processing.

The **Line Corrector** tries to fix the line defects resulting from poor drawing, image noise, or illumination effects. Its purpose is to close gaps in lines or line junctions, including joining the lines to the outer contours. (This is also partly achieved in hardware by line enhancement.)

The **Quality Assignment Routine** assigns quality marks to the proper quality areas that may be nested. In case of inconsistency (two or more different marks assigned to the same contour), an assignment error is reported for the given contour. Fig.3 demonstrates the recognized quality areas of the hide shown in Fig.1.

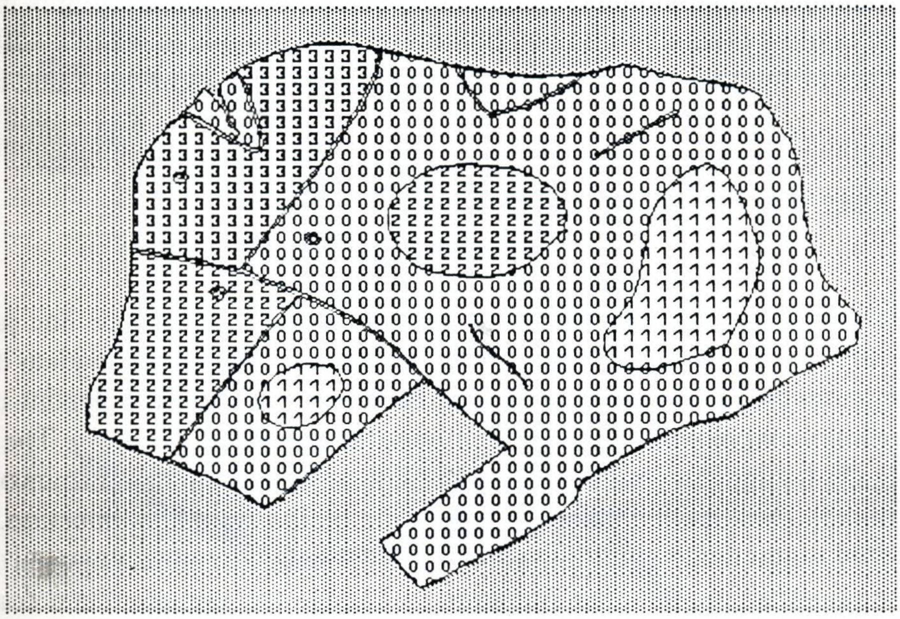


Fig.3. The recognized quality areas of the hide shown in Fig.1.

In the other branch of the program, the **Outer Contour Extractor** uses a shrunk-line image to extract all significant outer contours of hides, including major holes. (Very small holes are presented as zero quality areas.) Since the total elimination of hand-drawn lines and quality marks during the line shrinking (hardware-limited to 3 passes) is not guaranteed, the run-length and the chain codes of the corresponding binary image are filtered in order to remove the remaining parts of lines. The line shrinking leads to minor expansion of the outer contours which is later compensated for by a special routine. This measure ensures the coincidence of the outer contours and the corresponding parts of the quality area contours except for the negligible deviations originating from image filtering and contour approximation.

5. A few algorithms in brief.

In this section, we briefly present three of the algorithms developed for LeaVis.

1	1	1

	1	
	1	
	1	

	1	
1	1	

	1	
	1	1

1	1	
	1	

	1	1
	1	

Fig.4. To the definition of the topology-preserving median filter. Line pixels are indicated by 1's.

A topology-preserving median filter

This modification of the 3x3 median filter for binary images was designed to enlpo lines without breaking them when they are very narrow. The modified filter preserves connected lines passing through the central pixel. If the line pixel count N in a 3x3 windo or more, the central pixel C is set to 1. If N<3 or C=0, then C is set to 0. Otherwise, C is 0 unless either of the line pixel configurations shown in Fig.4 can be found in the window

An algorithm for line correction

This sophisticated algorithm fixes gap-like defects in lines or line junctions, inclu joining the lines to the outer contours. Here, we discuss only the main ideas of the method. lines are represented by oriented polygonal contours, i.e. closed polygons that are assign tracking direction, clockwise ("object") or counter-clockwise ("hole"). The correction is by detecting the turnpoints (ends of lines) and connecting them to contours or other turnpo The contours are then split or merged depending on the type of connection. The Line Co tor uses a set of syntactical rules that reflect the physical constraints imposed on the hand-dr lines. These constraints allow the program to handle the contours in a consistent way. F demonstrates the result of the correction for the lines shown in the box in Fig.1.

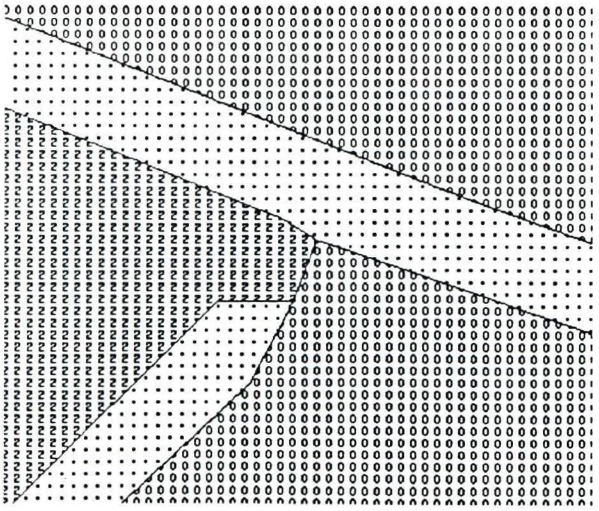


Fig.5. Result of correction for lines shown in in Fig.1.

Shrinking contours represented by polygons

The shrinking procedure compensates for a minor hide contour expansion introduced by the line shrinking (i.e. hide expansion) operator. The same procedure is used for hole expansion ("negative shrinking"). The procedure assumes that the expanded hide contour is relatively smooth. Let D be the shrinking distance which is in fact the number of passes in the hide expansion operator. Fig.6 shows a vertex V of a closed polygon representing the contour. B is the bisector of the angle composed by the adjacent edges in the given vertex. V_s is the resulting position of the vertex. To compensate for the expansion, V is shifted inside the polygon by

$$d_x = -\frac{D\cos(\alpha+\beta)}{\sin\alpha}$$

$$d_y = \frac{D\sin(\alpha+\beta)}{\sin\alpha}$$

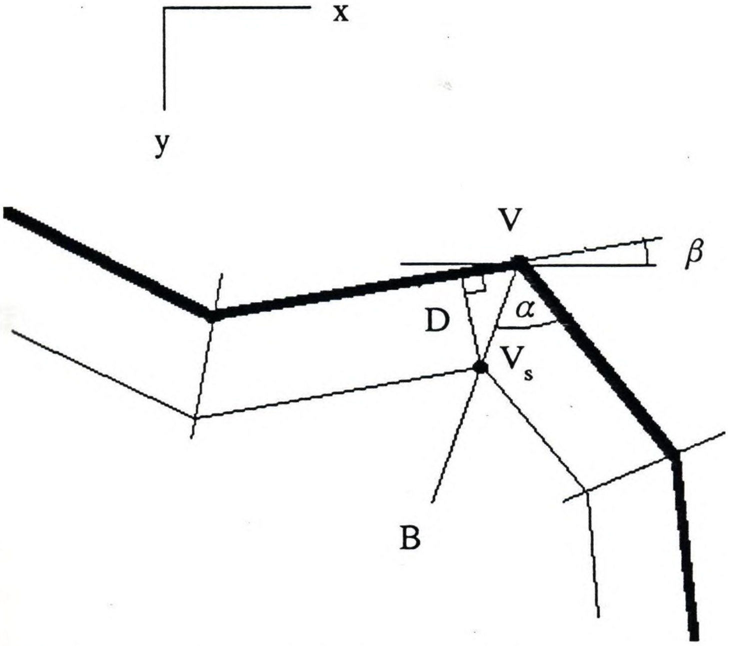


Fig.6. The polygon shrinking algorithm.

6. Conclusion.

We have briefly presented the basic principles and a few algorithms of LeaVis, a prototype machine vision system for leather industry. The first version of the system has been intensively tested. The tests show that the system is fast and, by virtue of the line correction procedure, quite reliable. With the scanning table size of approximately 1.5x3 meters, the scanning takes about 20 seconds. The execution time of the image processing and analysis program is tipically 10 to 50 seconds depending on the complexity of the image. The camera calibration procedure used provides a precision of about 1 millimeter. Further improvement in the reliability, which is of crucial importance in the industrial environment, will be achieved by using a better source of illumination and a more precise mechanism of installation. Later on, w to add algorithms for automatic defect detection.

References.

[1] T.Pavlidis, Algorithms for graphics and image processing, Springer-Verlag, Berlin-Heide 1982.

[2] A.Sube, Binary image coding, diploma work, Technical University of Budapest, 19 Hungarian)

[3] K.Wall and P.-E.Danielsson, A fast sequential method for polygonal approximation o tal curves, Computer Vision, Graphics, and Image Processing, vol.28, 1984, pp.220-227.

[4] A.Lerch and D.Chetverikov, Finding turning points on contours approximated by pol submitted to the 7th Scandinavian Conference on Image Analysis, 1991.

Calibration of a high resolution multicamera scanner

Peter Solt
Miklós Báthor

Computer and Automation Institute,
Hungarian Academy of Sciences
Budapest, P.O.Box 63
H-1518 Hungary

Abstract: A scanner of resolution approx. 8000x10000 pixels was developed in the Computer Vision and Robotics Laboratory of Computer and Automation Institute. It is used in the leather and upholstery industry. The camera signals have distortions because of the misalignment of the optical axes. Also, the lenses have imaging error. The distortion was compensated for by using a LUT technique to achieve the required precision of 3 pixels. The signals coming from the two cameras are adjusted using the method of multiple calibration points.

Keywords: scanning, leather industry, calibration

1. Introduction

In the leather and upholstery industry, it is essential for CAD and CAM users to obtain geometric information about pieces of leather (hides). Recently, our group has developed a scanning station for this application. It's main task is to provide a polygonal description of the quality areas and the outer contour of a hide. There are different defects and faults on the hide surface. These are identified by leather experts and marked by simple chalk-marks. In this way different quality areas are drawn and marked. The contours of these areas are described in a polygonal way. Together with the outer contours and possible holes they form a tree-like complete description of the hide.

The scanner uses two line-scan cameras with 4096 pixel resolution each. The cameras are mounted above the scanning table that holds a moving palette of size 3000x3500 mm. These palette dimensions are required for a full-size hide. The palette moves in Y direction (3500 mm) at a constant, manually tunable speed.

The table sends a START signal when the palette has reached the final speed after acceleration phase, and an END signal at the end of the table.

The cameras are positioned at approx. 1500 mm high from the palette's surfa The distance between the cameras is approx. 1500 mm. The illumination is provided light tubes fixed on both sides of the viewing line. The tubes are supplied from high f quency power sources.

The scanner is based on a 386 AT equipped with two special boards for interf ing the cameras. The boards can provide either a 8 bit/pixel gray-scale image or a p processed binary image with programmable parameters. The scanning software v written in C. The user interface is based on MS Windows 3.0. The output format for t CAD systems is an internal standard of ATOM+VICAM GmbH.

This paper discusses some issues related to the calibration of the scanner syste The image analysis and recognition software of the system is presented in [1].

2. Background compensation

The aim of this compensation phase is to compensate for the spatial variation of light i tensity I which is the function of the position x:

$$I = I(x) \qquad (1)$$

The camera interface board includes a look-up-table RAM to compensate for this ki distortions. This LUT is addressed by the input signal S and position:

$$P = P(x,S) \qquad (2)$$

P: pixel value.

The scanning system needs constant lighting. This can be achieved by well configur light-tubes and the LUT programming.

The CCD interface board provides a facility for the compensation. The compe sation is additive and depends on position and intensity. The compensation method is follows. The cameras scan a fully white surface several times. The accumulated imag gives the intensity profile of the lighting source. (See Fig.1.) The compensation means variable amplification which is in inverse ratio to the normalized intensity.

$$O = P\left[\frac{I(x)}{I_{max}}\right]^{-1} \qquad (3)$$

O: compensated pixel value
P: input pixel value
I: intensity profile
I_{max}: maximum intensity

Because of certain hardware constraints, an additive formula is used to obtain the corrected pixel value O:

$$O = P + A(x,P) \qquad (4)$$

where

$$A(x,P) = P\left[\frac{I_{max}}{I(x)} - 1\right] \qquad (5)$$

is the additive compensation function.

The latter formula was used to program the LUT RAMs with some limitations introduced to avoid data overflow.

The changing illumination was successfully compensated using a prototype intensity profile. The best results were achieved when I_{max} was approx. 80-90% of the total range of intensity. If there were pixels of very low intensity, the excessive amplification made the image too noisy.

3. Fitting images from two cameras

The fitting of the two images is based on six calibration points. The camera images are parallelograms since the scanlines are not perpendicular to the direction of the palette motion.

The calibration points are all included in the resulting image. The points in the middle are in the overlapping region of the two cameras. In this way, the calibration points define a clipping as well.

Clipping in X

The leftmost pixel used from camera #0 is the minimum X coordinate of (l_{00},l_{10}). (See Fig.2.) The rightmost pixel used from camera #0 is the average of l_{01} and l_{11}. The leftmost pixel used from camera #1 is the average of r_{00} and r_{10}. The rightmost pixel used from camera #1 is the maximum of r_{01} and r_{11}.

Clipping in the X direction is programmable, i.e. only the required line segments are read from the CCD sensor.

Clipping in Y

The coordinates of the calibration points and the size of the calibration image are stored. The number of image lines Y depends on two parameters, the palette speed S_p and the integration time of the CCD sensor T_{int} (which equals the line acquisition time):

$$Y = Y(S_p,T_{int}) \tag{6}$$

The image capturing and processing subroutines were designed in such a way that they are independent of the actual image size. The integration time depends on the intensity of the illumination, the background and hide colors, etc. The palette speed is not controlled by software in this version of the scanning system. Consequently, the Y size of the images is not programmable. To perform the required clipping the calibration image has to be scaled to the actual image. The coordinates are transformed using the following expression:

$$y' = y\frac{Y_a}{Y_{ç}} \tag{7}$$

Y_a : actual image Y size
Y_c : calibration image Y size

The first line used from the image of camera #0 is the minimum of (l'_{00},l'_{01}), the last is the maximum of (l'_{10},l'_{11}). The first line used from the image of camera #1 is the minimum of (r'_{00},r'_{01}), the last is the maximum of (r'_{10},r'_{11}). This would be two parallelograms of different length, hence one has to calculate the differences on both ends and correct the parameters to obtain the common length Y:

$$i = \min(\ \min(\ l'_{00},l'_{01}\ )-l'_{01},\ \min(\ r'_{00},r'_{01}\ )-r'_{00}\ ) \qquad (8)$$

$$Y^{0}_{start} = l'_{01} + i \qquad (9)$$

$$Y^{1}_{start} = r'_{00} + i \qquad (10)$$

$$Y = \max(\ \max(\ l'_{10},l'_{11}\ )-Y^{0}_{start},\ \max(\ r'_{10},r'_{11}\ )-Y^{1}_{start}\ ) \qquad (11)$$

where

Y^{0}_{start} ,Y^{1}_{start} : the indices of the first line from camera #0 and camera #1

The image lines are stored in the computer memory in a coded way during scanning. Based on the above mentioned calculations, a LUT can be created to access the left and right half-images taking the discarded parts and the offset into account. This LUT makes it possible to handle missing lines on too noisy images as well. Fig.3 illustrates clipping in X and Y.

A full image was taken from the two-camera setup for calibration purposes. The coordinates of the six calibration points and the image size were registered. The clipping of the active region and the fitting of the half-images was done in both directions. In X, it is possible to perform clipping in advance. In Y, it is done after scaling the calibration image to the actual image, and a LUT is created to access the image lines.

4. Lens compensation

The distance between the palette surface and the camera is approx. 1500 mm, while the required precision is less than 1 mm. A pixel is approx. 0.35 mm wide. The viewing angles from the camera are as follows:

precision: 2'17.51"
pixel : 0'48.13"

It is practically impossible to fix the camera absolutely perpendicular to the palette plane. This leads to a perspective distortion of the projection.

The best quality optics is used for the scanning system but it is still producing nonlinear distortions. Equal distances on the palette become different in the image:

$$x' \neq px + a$$
$$x' = p(x+D(x)) + a \qquad (12)$$

p: projection coefficient
a: offset coefficient
D: distortion function

The compensation of this error is based on measuring as well. A regular grid of bars is scanned for calculating the error. A compensation vector is created in such a way that the ends of the viewing line have zero distortions:

$$c_i = i\frac{p_N}{N} - p_i \qquad (13)$$

p: position of a bar in X
N: # of bars of the grid
c: compensation vector element
i: 0,1,..,N-1

A compensation vector is created for both cameras separately. The discrete vector elements or linear interpolation of vector element pairs can be used for this compensation, resulting in a piece-wise linear approximation of the compensation function.

The compensation of the perspective and non-linear lens distortions was required to achieve a precision better than 1 mm. A grid of approx. 240 bars was scanned on a 3000 mm wide palette to create a compensation vector. The compensation is used as a part of the coordinate transform only, otherwise a pixel image transform applying the lens compensation would be extremely time consuming.

5. Combining different types of compensation

Mapping parallelograms to rectangles is a linear transform. The calibration points are defined in the CAD system coordinate system and in the pixel coordinates as well. Using the equations of LSM, a transform matrix is calculated for both cameras. First, the lens compensation is performed in the pixel space, then it is determined which half of the image is involved and the output is calculated from the linear transform.

Reference

[1] D.Chetverikov and A.Lerch, "Automatic analysis of marked hide images for leather industry", see this volume.

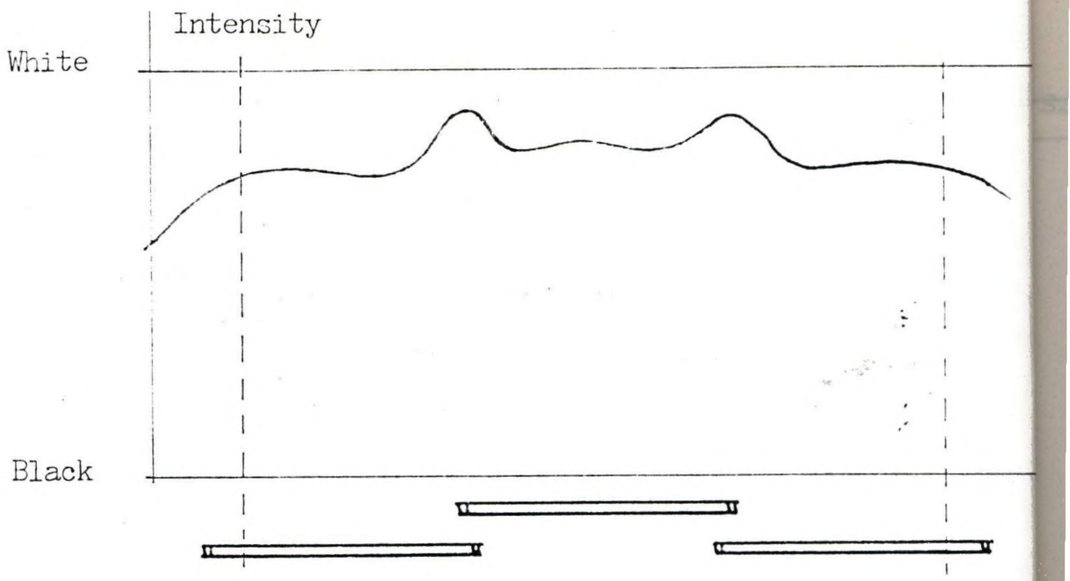


fig. 1

Typical intensity profile

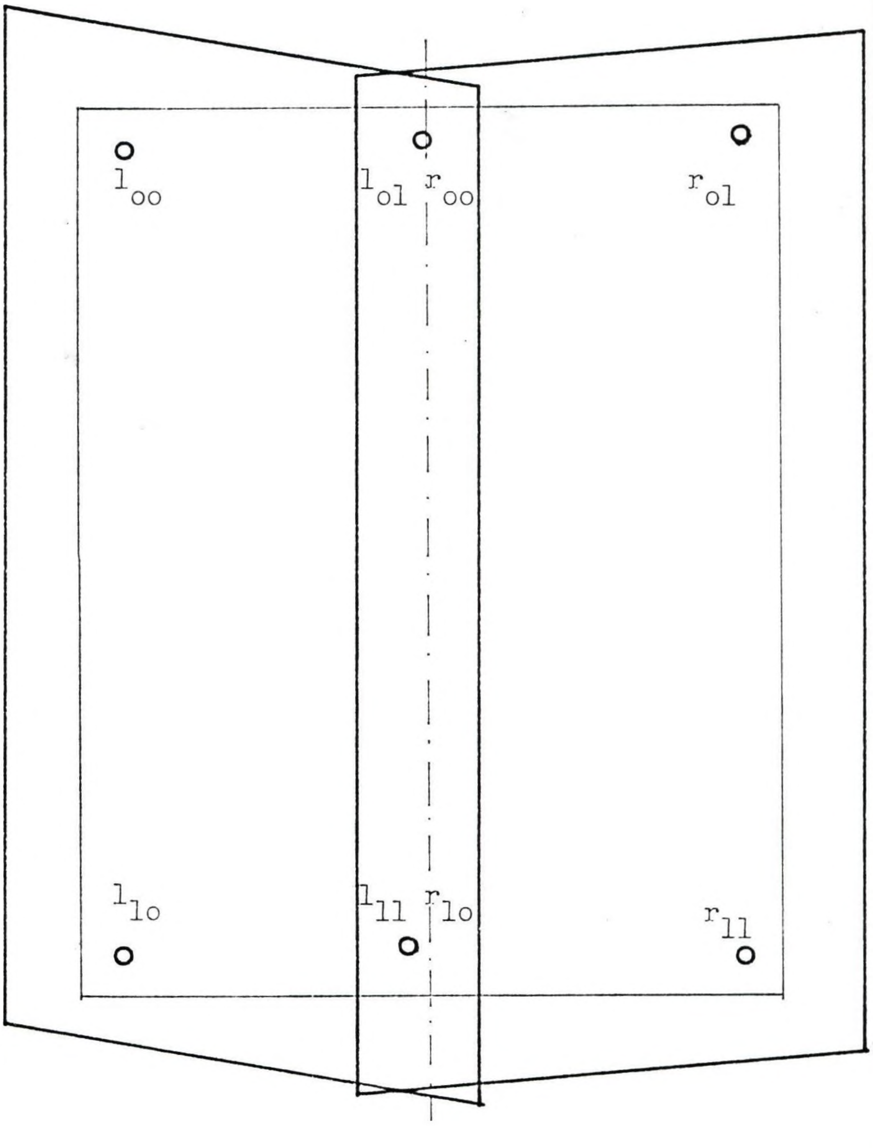


fig. 2.

The arrangement of the calibration points.

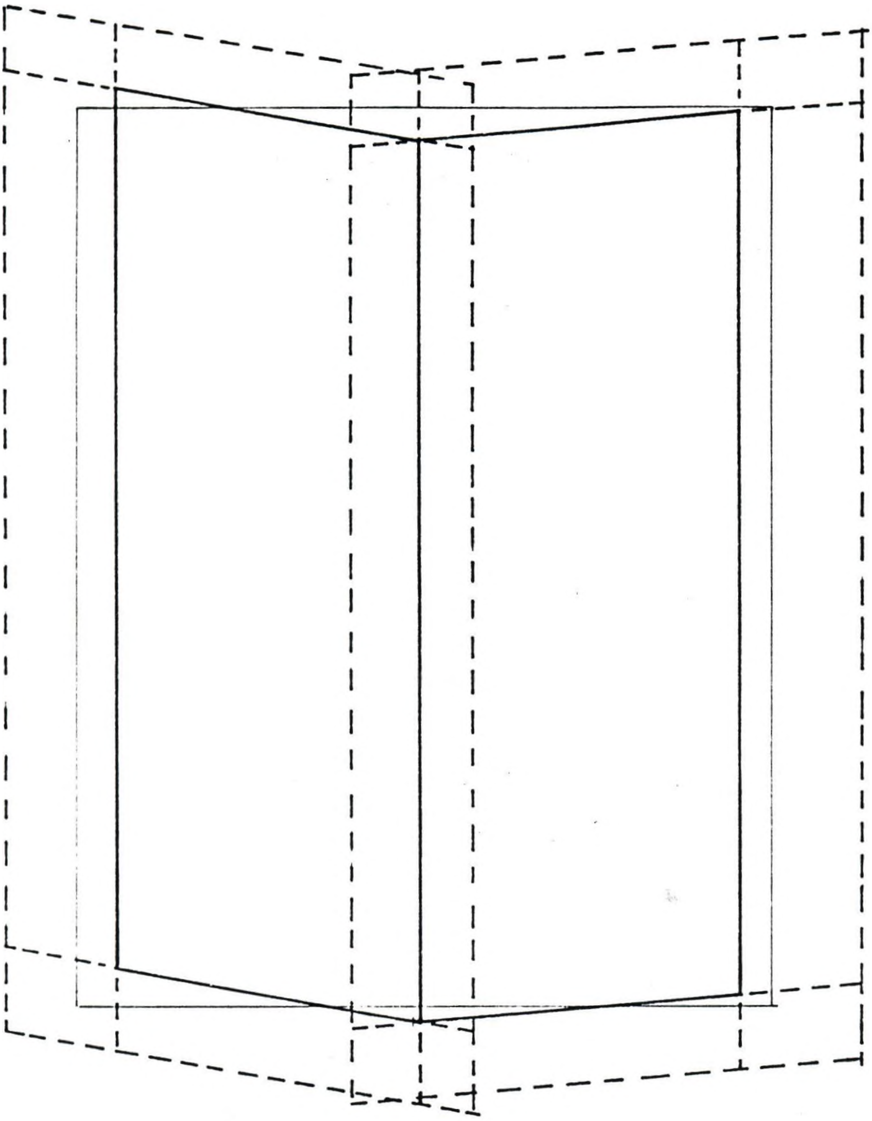


fig. 3.
The result of the clipping in X and Y

Tamás SZIRÁNYI and József CSICSVÁRI

SUBPIXEL TECHNOLOGY FOR PATTERN RECOGNITION AND MEASUREMENT OF KNOWN SHAPES

Subpixel technology is a quite new part of digital image processing. Several measurements [8,9,10] was developed to get some extraction of specific features. In these measurements the accuracy is better than the final resolution of the scanning tools.

Now, a new statistical method has been developed for detection, recognition and measurement of objects which are (much) smaller than the diameter of the elementary scanning windows. Originally, this method was developed for one dimensional purposes [Szirányi, 1,2].

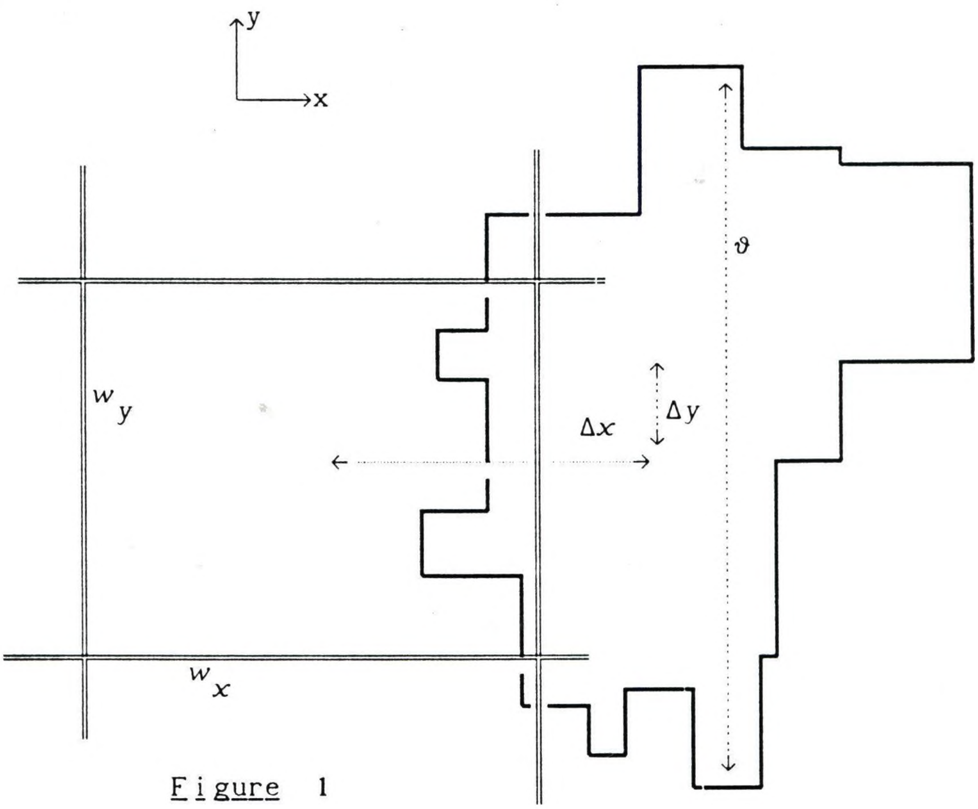


Figure 1

A sensor window of scanner (camera) and the measured object

Computer and Automation Institute of the Hungarian Academy of Sciences
(MTA SzTaKI), TGE Group, H-1014 Budapest, Úri utca 49.

Because of the statistical approach, a greater number of the same object (separately) is needed. Practically, in this measurement the pixel points covered (partly) by these similar objects on the screen must be seperated from the background and a gray-level histogram is made from the separated picture. Then this measured histogram is compared to a set of histograms calculated earlier on the basis of the known shapes of objects and known sensitivity function of the windows of the scannner (camera).

The theoretical grounds

In the first step the theoretical distribution of possible gray-levels (histograms) is calculated for the objects to be detected.

If S(x,y) is the fotosensitivity-distribution of an elementary sensory window (See <u>Figure</u> 1.), $A(x-\Delta x,y-\Delta y,\vartheta)$ is the light-intensity distribution of the measured object at $(\Delta x,\Delta y)$ characteristic distance from the window's reference point, and ϑ is the class of the different objects, the photo-exposition for an elementary sensor-window (the pixel gray level):

$$E(\Delta x,\Delta y,\vartheta) = \int_{\{x\}}\int_{\{y\}} S(x,y).A(x-\Delta x,y-\Delta y,\vartheta)\ dy\ dx \qquad (1)$$

If both the Δx and the Δy have the uniform probability distribution (Δx, Δy and the position of the exposition are really indipendent), then the $p(E|\vartheta)$ probability density of $E(\vartheta)$ can be calculated in the $(E,\Delta x,\Delta y)$ space:

The $E(\Delta x,\Delta y,\vartheta)$ function is cut by the $E(\Delta x,\Delta y,\vartheta) = E_o$ and the $E(\Delta x,\Delta y,\vartheta) = E_o + \Delta E$ planes. The projections of this section of the $E(\Delta x,\Delta y,\vartheta)$ surface to the $(\Delta x,\Delta y)$ plane is proportional to the probability of $E \in (E_o,E_o+\Delta E)$. When ΔE -> 0, we get the

$p(E|\vartheta)$ density (Figure 2).

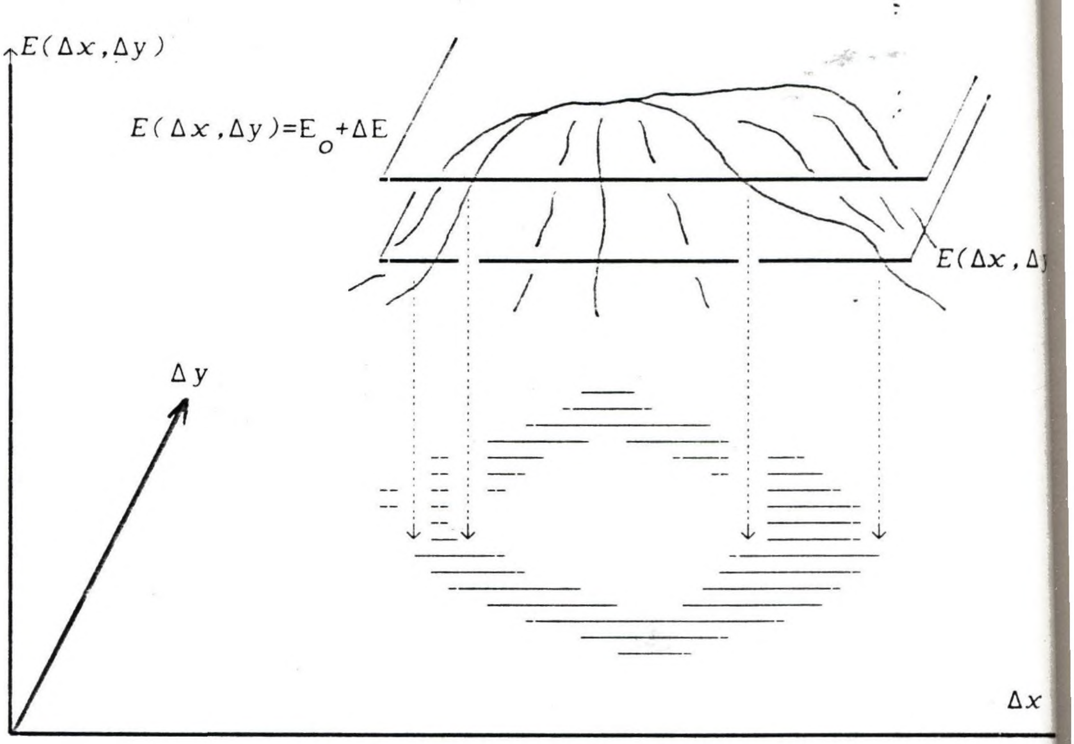


Figure 2

The probability calculation of $P(E \in [E_o,E_o+\Delta E] \mid \vartheta)$ fro $E(\Delta x,\Delta y,\vartheta)$ function in the $(E,\Delta x,\Delta y)$ space. The probability calculated from the projection of the $[E_o,E_o+\Delta E]$ interval to $(\Delta x,\Delta y)$ plane.

Practically, in the case of every calculated ϑ, the $(\Delta x,\Delta y)$ area is divided into small equal parts, and the E direction also divided into quantized values. For every $(\Delta x,\Delta y)$ area the quantized E_i is calculated from (1) and the counter of the calculated E_i value is increased by one. Because there are positions, where an object is placed on two or more pixel-windows at the same time (it is on the border among windows), some neighbourhood of the windows should be taken into consideration.

Not only $(\Delta x, \Delta y)$, but the rotation angle of the object can be calculated in this way, too.

In the measurement we take the probability density of grey levels of the measured similar objects, and the measured $p(E)$ is compared with the simulated $p(E|\vartheta)$ distributions for every ϑ.

Experimental example

In our experimental example the measured object was a curving line with stable width of ϑ (Figure 3, 6). The elementary objects were line-segments crossing the pixel window.

The task was the recognition of curves of different ϑ widths.

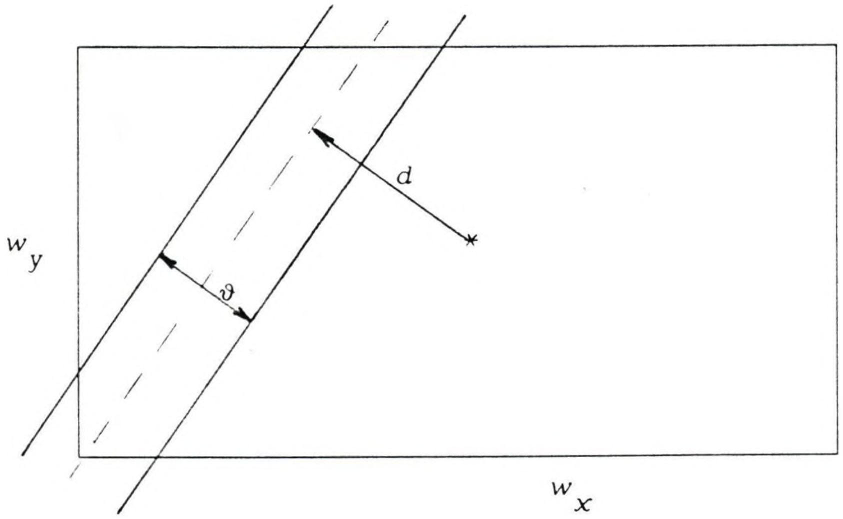


Figure 3

An elementary sensor window crossed by a line-segment of ϑ *width and d distance*

The simulation was made for uniform distribution of α and d, $w_x = w_y$.

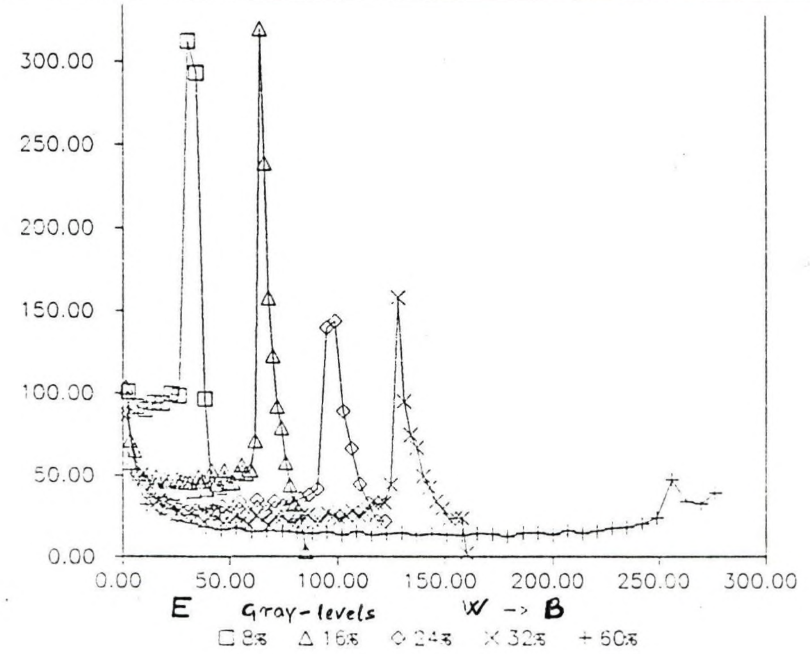


Figure 4 *The simulated histograms in the function of line-width of the % of an elementary pixel-width, $w_{..} = w_{..}$*

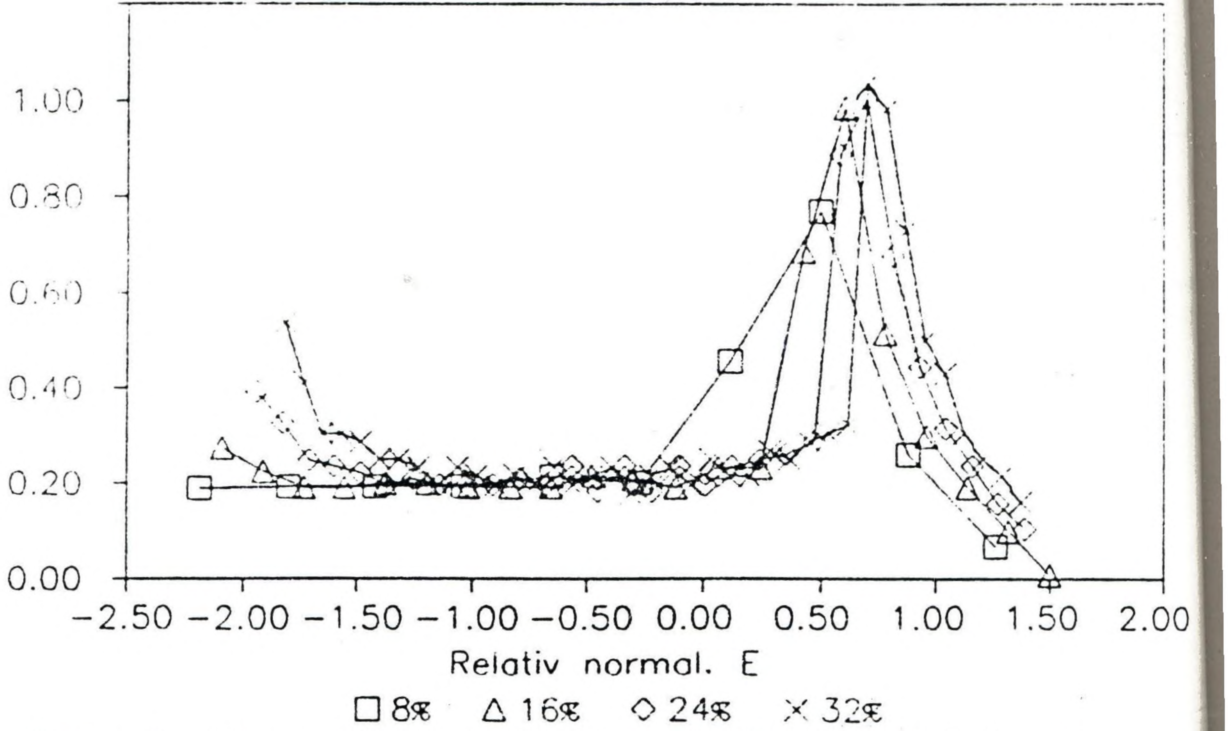


Figure 5 *The normalized simulation results (The integral is 1 and the E direction is normalized around its average by its standard deviation value)*

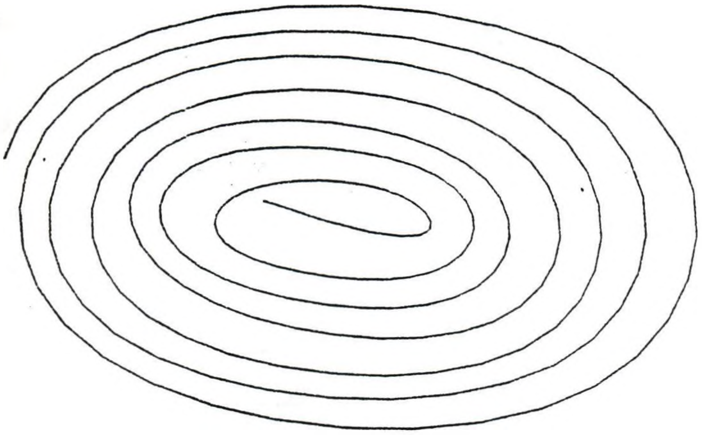


Figure 6 *The measured curve-line. This picture was scanned by an image scanner at 50dpi, and lower resolution was got by software*

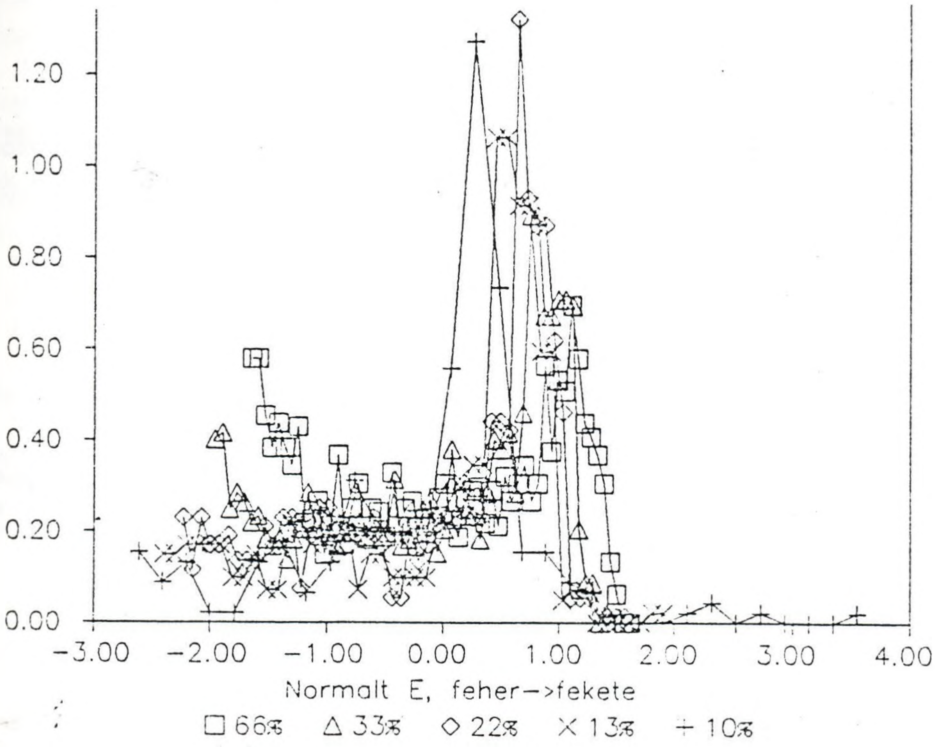


Figure 7 *The normalized histograms of the measurement*

Simulations

M	ϑ%	8	16	24	32	44	52	68	76
e	10	**88**	53	54	51	55	54	54	58
a	13	*91*	*87*	85	72	65	57	55	57
s	25	87	*94*	**96**	87	80	71	64	66
u	33	81	84	*91*	**93**	*88*	83	73	72
r	40	78	82	89	*94*	*91*	85	76	73
e	50	79	79	85	87	*94*	**95**	*91*	86
d	67	77	74	81	86	*96*	*97*	*94*	*93*

Correlation x100%

Simultions

M	ϑ%	8	16	24	32	44	52	68	76
e	10	**47**	83	82	85	82	82	80	79
a	13	*43*	*44*	*44*	65	72	80	80	81
s	25	41	29	**24**	39	49	60	63	64
u	33	46	34	*30*	*30*	36	44	51	56
r	40	51	39	30	*25*	*29*	38	45	51
e	50	50	44	36	32	*24*	**22**	29	35
d	67	53	49	41	31	*18*	*17*	*21*	*25*

Integral of the ab
difference x100%

Simulations

M	ϑ%	8	16	24	32	44	52	68	76
e	10	**14**	24	26	27	28	29	29	27
a	13	*13*	*13*	*14*	21	25	28	29	28
s	25	12	*08*	**07**	13	17	21	23	22
u	33	14	*10*	*10*	*10*	13	15	18	19
r	40	16	12	*10*	*08*	*10*	13	16	18
e	50	16	13	11	10	*08*	**07**	10	12
d	67	18	16	14	10	*06*	*06*	*08*	*08*

Average absolut
difference x100

Simultions

M	ϑ%	8	16	24	32	44	52	68	76
e	10	**04**	14	14	15	14	15	16	14
a	13	*03*	*04*	05	09	11	14	15	14
s	25	03	02	**01**	04	06	08	11	09
u	33	05	04	03	**02**	03	05	08	08
r	40	06	05	03	*02*	*02*	04	07	07
e	50	05	05	04	04	*02*	**01**	*03*	03
d	67	06	07	05	04	*01*	*01*	*02*	*02*

Average square
difference x100

Table 1 *Comparison of the normalized simulations and measurements at ϑ% (of sensor-window width) relative line-width*

CHARACTER RECOGNITION IN A HYBRID OF CELLULAR NEURAL NETWORK AND DIGITAL LOGIC (CNND)

T. Szirányi[+], J. Csicsvári[++]

Computer and Automation Institute of the Hungarian Academy of Sciences
(MTA SzTAKI), Úri-u.49, Budapest, H-1014, Hungary
[+]*part time associate from VIDEOTON Co., Budapest*
[++]*student at Technical University of Budapest*

INTRODUCTION

In many tasks (picture processing, feature extraction) the analog neural netwo are much faster than the conventional digital-logical methods. It can be a good opti when analog neural networks make the basic picture processing functions and the mi and higher level decisions are due to some digital-logical system [Roska, **3**].

There is the so called **Dual structure, CNND** (Figure **1**) [Roska **3,13**]. It contains an analog *Cellular Neural Network* (**CNN** [**1,2**]) and some *digital decision* units. Here the input picture is preprocessed by the *CNN* by templates as edge-detectors, hole-detect corner-detectors, etc., and the *Connected Component Detector* (**CCD**) is the final step the preprocessing. From the result of the CCD templates the parts **a** and **b** calculate semifinal results and the part **c** calculates the final result.

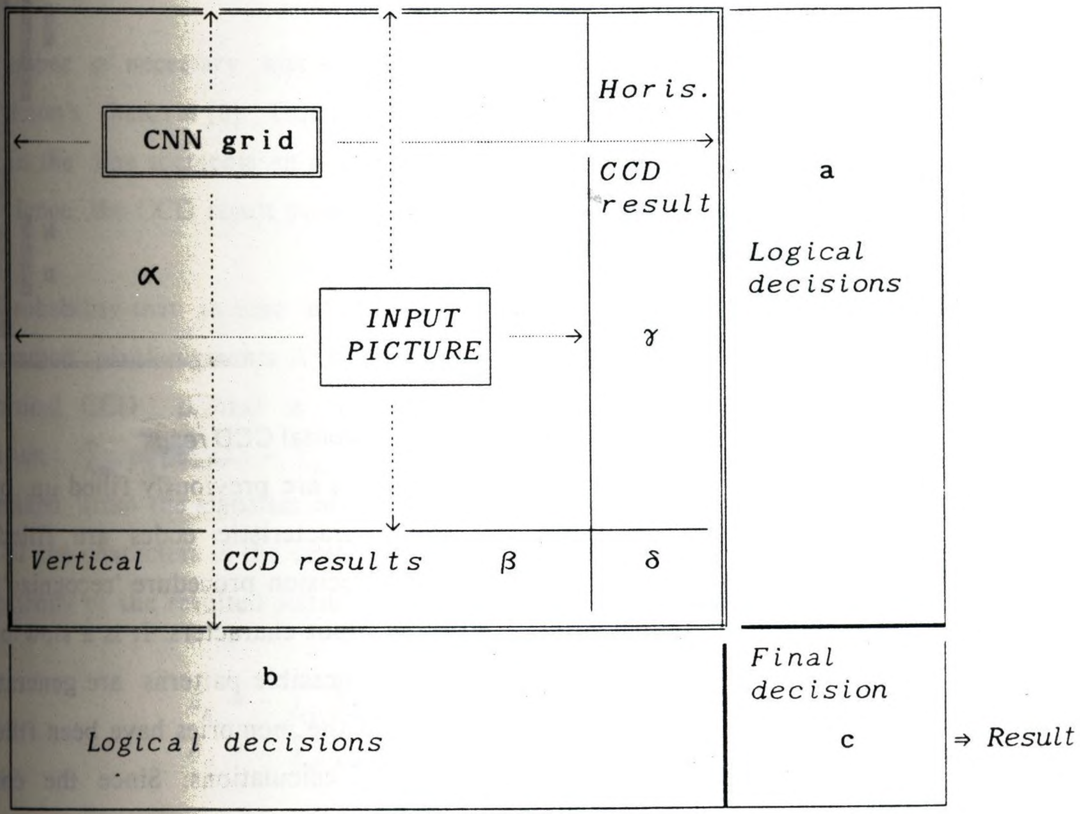
This system is ideal for such purposes as recognition of printed characters. T goal of this paper is to show the reality of this system as a usable hardware.

It will be shown that the **CNND** system can work as a character recogni system at an efficiency of more than **96%**, and at the recognition rate o **2**μsec/character. The hardware can be built from achievable elements.

1. THE CNND ARCHITECTURE

The CNND architecture is shown in Figure **1**. The "eye" of this system is a CN array shown in a double frame. The original image is fed into this α area. Next, the C array with a sequence of various templates perform noise removal and feature extraxti The final CNN template pair is a horizontal and a vertical connected component detect (CCD) [4]. These CCD-s as "analog shift registers" are doing an information compressio of the whole area. The CCD results say that in the same row/column there are *m* groups of connected pixels. The results are appearing in the corner bands which are marked by the areas β and γ. Here the cell outputs are representing black (+1) and white (-1) values which will be converted into logical ones (1 and 0). Due to the nature of the CCD only every second row/column has useful information in these corner bands. So the original 40x40 pixels are converted into two stripes of about 4x40 dotted lines (Figure 2). These useful logical values of the CNN corner bands are the input to the digital decision parts **a** and **b**. Any operator with digital output can be defined here. The final decision (pattern classification) needs generally the outputs of both units.

The digital decision functions can be realized by memories, PLA-s, EPLD-s etc. Even statistical parameters with a few bit precision can be calculated. Moreover, any information compression scheme can be realized in this way.



The structure of Dual **CNND** *structure. The low-level processing is done in the Cellular Neural Network (CNN). The final results are computed from the* **CCD** *results of the CNN in the logical* **a,b,c** *parts of the system. The memories of these logical parts are filled up by the sets of teaching samples.*

Figure 1. *The* **CNND** *architecture*

2. CHARACTER RECOGNITION IN THE DUAL STRUCTURE

The CNN structure itself can be very fast comparing to logical elements, so i no sense to use a difficult logical evaluating and decision system in the recognition operation. It is much better to use a fast logical circuit, which compress CCD results into usable short codewords, and the compressed data -as input- a recognition memories (**a** and **b** in Figure 1.). The output of the memories horizontal and vertical CCD results are used as input to the final decision logic (**c**). The content of the memories are previously filled up. In this the teaching samples are analysed and the characteristic codes are filled in. simulated input patterns are generated and a real decision procedure 'recognize' (or n the simulated patterns as the features of possible input characters. It is a slow procedu but it is made only once. In this way all of the possible patterns are generated and recognition results are filled in the memories. After the memories have been filled in, real recognition works perfectly without additional calculations. Since the compress logic can be realized with fast PLD devices, the resulted recognition procedure can w real-time.

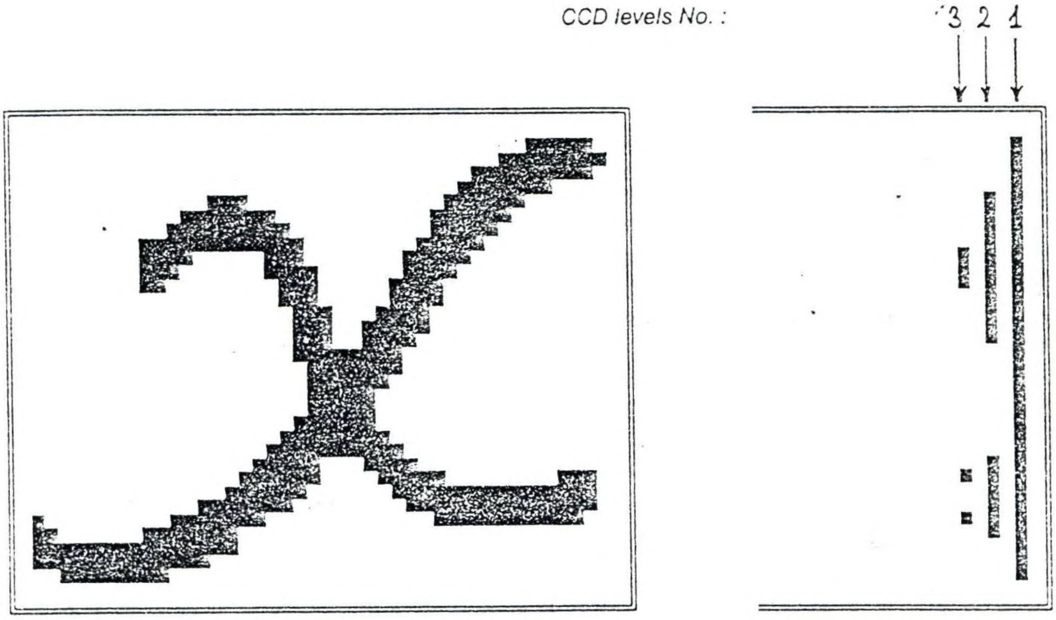


Figure 2. *Input image and its* **horizontal CCD** *result*

2.1. Realizability of the digital part

Realizability means, whether the coding (compressing) step, the recognition (memor step and the final decision step can be realized by achievable hardware components not.

The bottle-neck of the whole system is the number of input points (address field size) of the recognition memories.

The question is: Can we compress the CCD data into a memory address size of not more than some MB or not ?

The character image

It is known from the practice [5,7] that the well acceptable recognition rate for normal printed characters needs at least 300dpi input resolution. It means that a sampled picture of a printed character contains about 40 x 40 binary picture points (pixels). The character lines of the picture and the characters of the lines are separated by a searching algorithm (image selector) that uses local histograms. The final results of the CNN picture processing are generated by a Connected Component Detector (CCD).

Compressing

The number of necessary bits to characterize and code the CCD result can be calculated by Shannon's theorem [8]. Due to the nature of the CCD results (Figure 2.) every black pixel in the line segments on a higher (2..5) level has a black pixel neighbour in the lower level. Hence, the CCD result pictures are very redundant.

The probability that in case of a horizontal CCD at a given y position the number of unconnected pixel-segments is m (m = 0...5) is denoted by $p_h(m,y)$. Similarly, in case of vertical CCD $p_v(m,x)$ is the probability of m segments at the x coordinate. By definition $\sum_m p_h(m,y) = \sum_m p_v(m,x) = 1.0$. The probability can be determined from the statistics of a large set of characters. In this procedure the reference point of the characters is the centre of gravity of the input character picture.
The entropy of the resulted statistical distribution is calculated as follows [8] :

$$E_h = - \sum_y \sum_m p_h(m,y) \log_2 p_h(m,y)$$
$$E_v = - \sum_x \sum_m p_v(m,x) \log_2 p_v(m,x)$$

From the calculations of our character set $E_h = 29.7$, $E_v = 33.2$. It was 630 for the whole original image. In this calculation the probability values of an x or y position were considered as independent variables. But it is not valid at all. We can define conditional probabilities (similarly to Markov chains) for every m at any x and y positions. In this case the entropy will be lowered. From the calculations above, it can be shown that, as address inputs of available memories, the CCD images can be used as input in the decision memory.

Designing the coding system we have examined several methods. Finally, the o described below proved to be optimum.

The coding system is based on the *(m,h)* code words of CCD results, where m is the number of connected pixel-segments, *h* is the length of the section of this *m* va With Shannon's theorem, it can be calculated that (in the independent case) these *(m,h)* codes can be compressed into about *3.9 bits/codeword*. This compressing procedure can be realized by PLDs and/or analog circuits in an easy way. In Figure 2. there horizontal CCD result. It can be coded in the form (from the bottom-right point) :

(1,3)(2,1)(3,1)(2,2)(3,1)(2,1)(1,8)(2,4)(3,3)(2,4)(1,4)

Calculating the average number of codewords at one character (it is *6.0* f horizontal and *7.4* for vertical CCD results), the average number of necessary bits of t compressed codes is *23.8* for the horizontal and *27.9* for the vertical cases.
The entropy (the necessary bits for the average code-words) should be lowered calculating the dependence of the codewords on each other. Since the recogniti memories are very redundant (there are several input patterns without meaningful output character), it is possible to decompose a large memory into some smaller ones in more cascade layers. If the content of the memory can be expressed in Boolean functions (truth table), this decomposition could be executed by some known algorithms [**11**].

It can be shown that the horizontal and vertical CCD results are independent, since the original characters' pixel probability densities are also independent in the sense of *x* and *y* directions [**7**]. *So, the whole original image is compressed into about 52 nearly independent bits.*

2.2 The character recognition procedure

Filling in the recognition memory

Filling in the decision memories is a time consuming procedure. Doing this, the next steps are the same for both the vertical and horizontal CCD parts.

From the teaching sample set we define a 3 dimensional *(k,m,h)* space (Figure **3**), where *k* is the starting position (*x* or *y*) of an *(m,h)* segment. The *(k,m,h)* codewords of every known characters are captured in the *(k,m,h)* space. Generating all samples for every possible input variations of the recognition memory, a Nearest Neighbour (NN) [**10**] method is used to find the appropriate (nearest in some sense) teaching character. This system can be interactive (asking human decision in some cases).

Finally, the decision memory is filled up for the possible input address patterns with the proposed decision result(s).

The final decision logic (part **c** in Figure 1.) makes the final decision about the expected characters (it can also be a memory in the more difficult case).

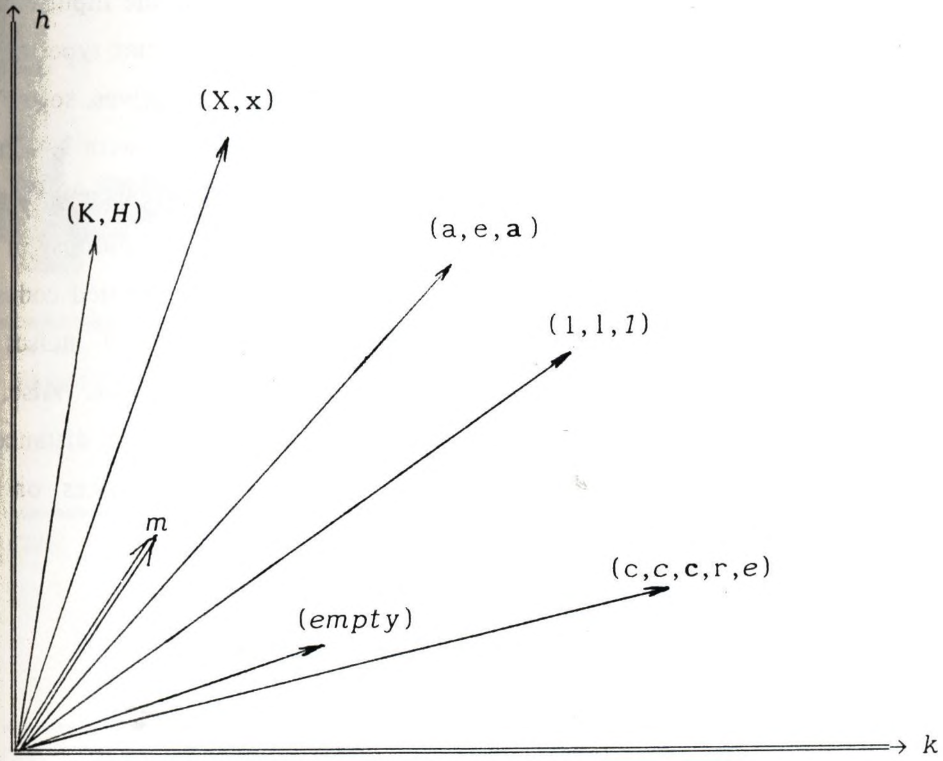


Figure **3**

The predicted character is 'sitting' in the position or in its neighbourhood shown by the (k,m,h) code-vectors in the (k,m,h) space, where k is the starting positions of a CCD segment, m is the number of unconnected parts in a column/row and h is the length of a segment.

<u>*Experimental results for the recognition procedure using the CNND* model</u>

Now, experiments of recognition of character sets of 3 fonts are shown. Training and test sets were chosen from the same universe of samples, but of course they were disjoint.

The possible content of the filled up <u>memory was simulated as follows</u>. The classifier method was a *modified NN* procedure :

With the training characters a *(k,m,h) index*ed character matrix was filled in (the matrix value is the character itself). Each input *(k,m,h)* feature code selects a character a the *(k,m,h) indexes* in the matrix or at some neighbouring index place. Then the m frequent character type is chosen as the probable result.

If the occurrence rate of a chosen character is not dominating, then for the most probable characters some distance is calculated from the input pattern feature vector and the decision is made based on the minimal distance. At this step the input codes ar compared to the code-sets of the predicted characters. At one character type the code words of the different training feature vectors are mixed among themselves, so the code words of the input feature vectors are compared to the training vectors with NN method In this distance calculation first Nearest Neighbour is calculated separetely for eve **(k,m,h)** code-word in the *(k,m,h)* space. The **3n** dimensional (**n** is the number of code-words) *NN* distance is calculated from the Nearest Neighbour of the separated codes, so it is not an absolute minimal distance. In this way the strong effects of noises can avoided, because there is not a fixed order of codes in comparison. Also, we recognize characters that are different from the teaching font-sets. The *NN* distances are calculated as the sum of the absolute differences of the *k,m,h* coordinates on all th input vectors.

The first recognition results

If no preprocessing or feature extraction templates are used, then for **150** characters of the **3 mixed fonts** the recognition efficiency was more than **90%**. Using NN distance calculation, the efficiency was more than **92%**.

As a new dimension, using the results of the diagonal CCD of 45 degre projection (Figure **4**), without any NN distance calculations, the recognition efficiency was better than **96%**.

In Figure **5** a possible realization of the CNND hardware is seen. Using a 2 μm technology it takes about 2 μsec (1 μsec is the CNN procedure time for 40x40 points and less than 1 μsec is the access time of the 3 memory areas).

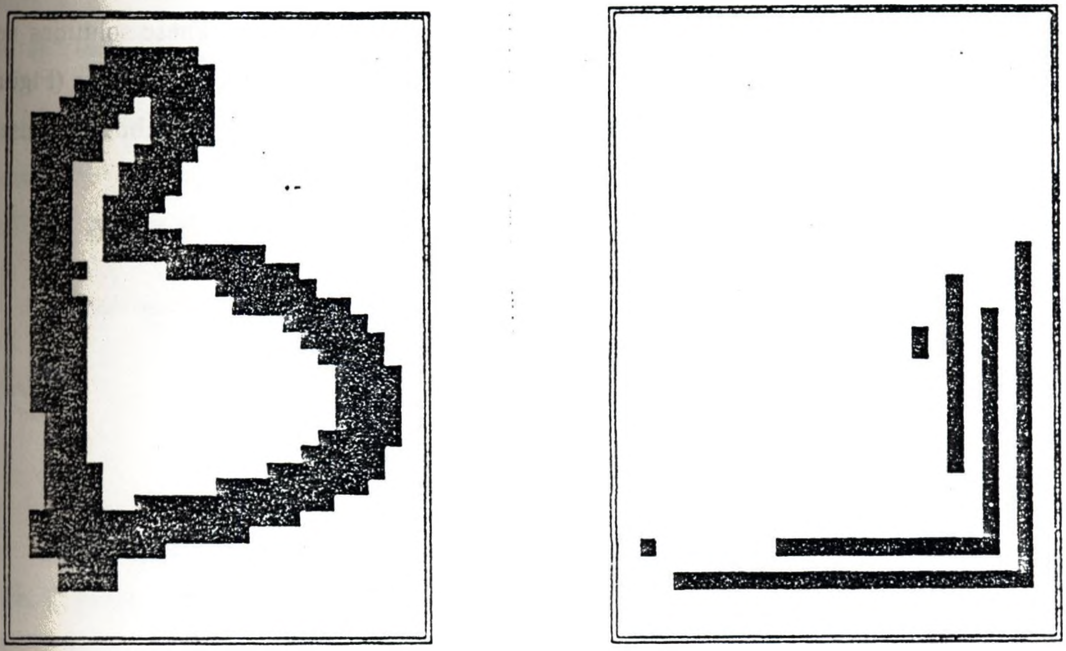


Figure 4 *An input picture and its* Diagonal CCD *result*

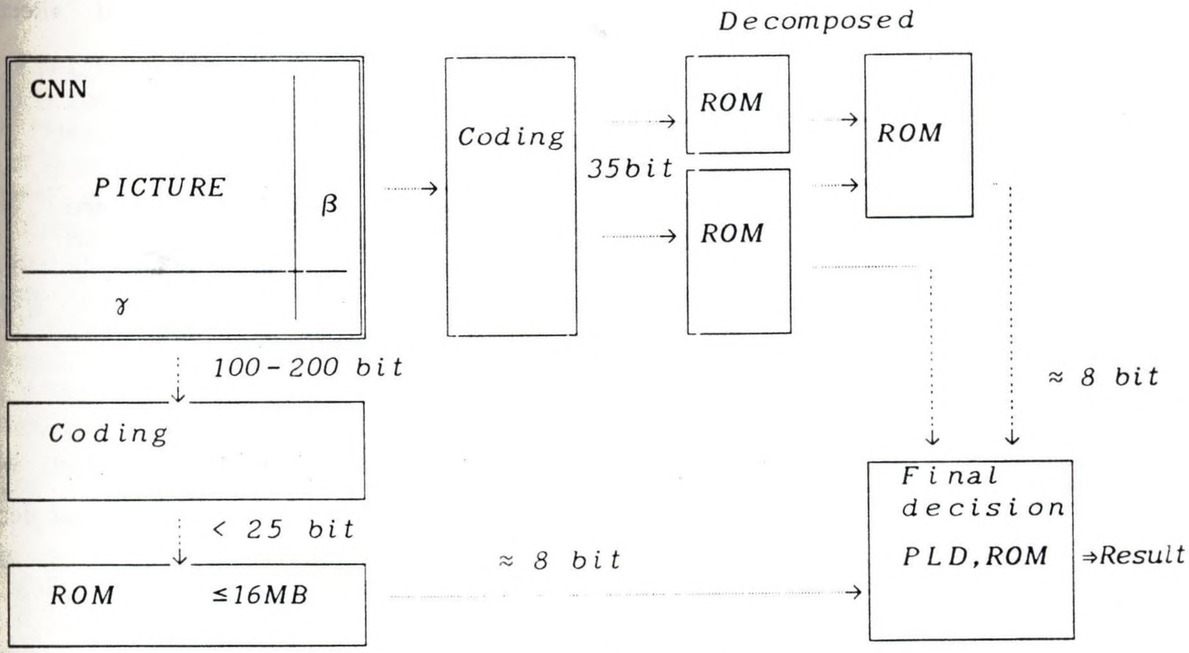


Figure 5
The proposed hardware system

The reduced hardware method

On the base of the above calculations a little slower sequential recognition method can be realized with much smaller memory space. Here the memory of CCD results has only about 10-12 inputs (1-4 kword). Each *compressed k,m,h* feature code addresses this memory, and the output (predicted character type on the basis on one codeword) is loaded sequentially into the final **c** decision logic. Loading every feat code in the decision logic, the most frequent character is chosen. These solutions ne two 4 kword ROMs and a 256 word of 4 bit RAM with some logical elements (Figur **6**). The execution time is 5–10 times longer than the former solution (about 10 μsec The recognition rate was more than **95%** for the characters of **3** fonts.

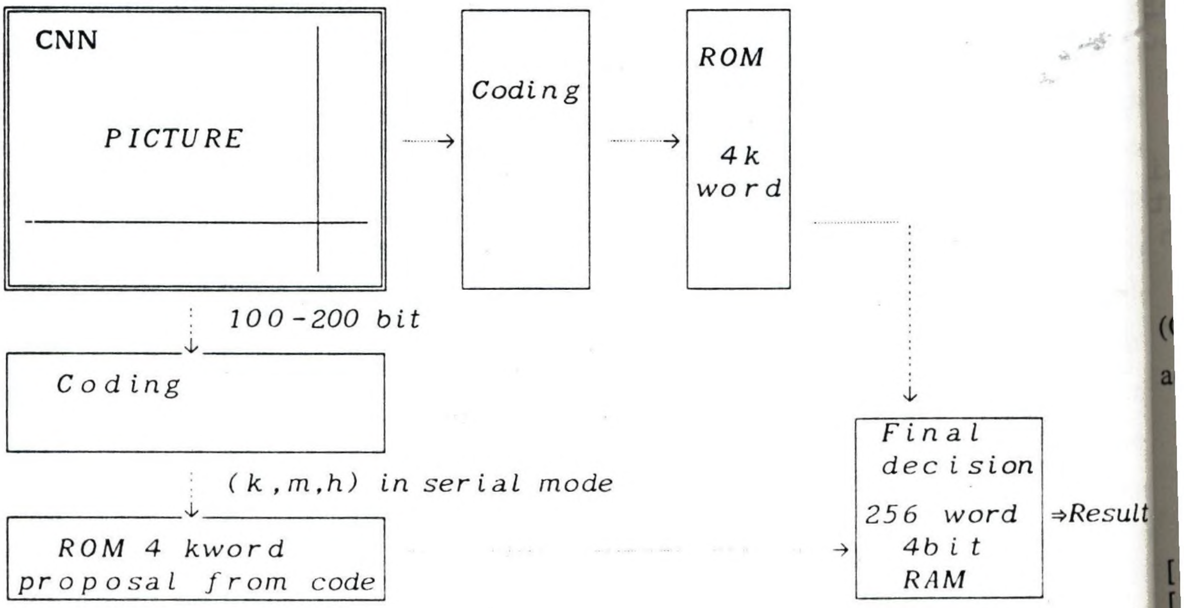


Figure 6
The simplified hardware solution. The (k,m,h) CCD codewords are fed in serial mode into the decision logics. Making a proposal on every separated code, the final decision makes the result from the most frequent prediction.

Scale invariation

In the present experiment character fonts of similar outline dimensions were used. When it is not so, additional *inputs of 2* or *3* bits should be used in the decisio memory as the measure of scale.

CONCLUSIONS

We have found that the Dual structure CNND can be used in the practice for character recognition. It is obvious from our first experiments that this method can be realized by available hardware elements.

Our preliminary experiments show that this dual CNND architecture represents a promising solution for some pattern recognition tasks. The crucial point is t combination of analog dynamics (CNN) and logical decision made directly by the appropriate VLSI devices. Hence, the algorithmic and the device level is verified.

The recognition error can be lowered by using more CNN templates in the procedure (corner detection, hole detection, etc.). In this case more decision levels and a feedback system should be used.

These encouraging first results justify our further work in this research field.

ACKNOWLEDGEMENTS

The work has been supported by the National Research Fund of Hungary (OTKA) grant No. 1212/1989. The many helps and advises of prof. T. Roska, and the active assistance of P. Szolgay, A. Radvânyi and T. Boros is gratefully acknowledged.

REFERENCES

[1.a.] *L.O. Chua, L. Yang*,"Cellular Neural Networks : Theory",
[1.b.] *L.O. Chua, L. Yang*,"Cellular Neural Networks : Application",
IEEE Transactions on Circuit and Systems, Vol. 35, pp.1257-1272 and pp.1273-1290, **1988**
[2.] *T. Roska, L.O. Chua*,"Cellular neural networks with nonlinear and delay-type elements", in this Proceedings.
[3.] *T. Roska*,"Analog events and a dual computing structure using analog and digital circuits and operators", in Discrete Event Systems: Models and Applications, ed. by P. Varaiya and A.B. Kurzhanski, Berlin: Springer Verlag, **1987**
[4.] *K. Krieg, L.O. Chua, L. Yang*,"Analog signal processing using cellular neural networks", Proc. IEEE ISCAS-90, May, **1990**
[5.] *S. Kahan, T. Pavlidis* , H. Baird," On the recognition of printed characters of any font and size", IEEE Trans. PAMI, Vol. 9, pp. 274-287, **1987**
[6.] *T. Roska, A. Radvânyi*, "CNND simulator, Version 3.01, Users' guide", Report 37(1990), Comp. Aut. Inst., Hungarian Academy of Sciences (MTA SZTAKI), **1990**
[7.] T. *Szirânyi, Á. Böröczky, T. Kovâcs*," Character recognition and optical characteristics of image scanners", Proc. of SPIE, Vol. 1153, pp. 485-493, **1989**
[8.] *C. E. Shannon, W. Weawer* : The mathematical theory of communication, **1972**
[9.] *T. Szirânyi*: Statistical pattern recognition of low resolution pictures, Pattern Recognition Letters, Vol. 8., pp. 221-228, **1988**
[10.] *G. Loizou, S.J. Maybank*," The nearest neighbour and the Bayes error rates", IEEE Trans. PAMI, Vol. 9, No. 2, **1987**
[11.] *P. Szolgay*,"Computer assisted cut-and-try design of the logic and the layout of combinational circuits", Proc. of ECCTD87, pp 345-350, Paris, **1987**
[12.] *A. Rényi*,"Probability calculation", Tankönyvkiadó, Budapest, **1981**
[13.] *T. Szirânyi, T. Roska, P. Szolgay* : A CNN neural network embedded in a dual computing (ÇNND) structure and its use for character recognition, **IEEE Proc.** of **CNNA'90**, 90TH0312-9, **pp. 92-99**, **1990**

The Framebase pictorial database management system

István Szabó
SZKI Pixel Ltd.

Summary : The article gives an introduction to the Frame picture database management system and the conne assisting procedures. This new system was designed on basis of our experiences with the Pigalle [1] system. main characteristics of the system are summarized in general introduction section, and the subsequent sect contain a detailed description of its opera principles, the file formats used, and the solutio problems associated with the displaying of images.

Keywords: pictorial database, hypertext, hypermap

General Introduction

The FrameBase is a programm for the archiving and trieval of image and text files, and the storage of user fined relationships between images and texts (in the foll ing : items). The system can use a VGA graphic adapter for displaying of images, with not only the standard, also the enhanced resolution modes. Pictorial informatio stored in TIFF, text in ASCII file format. The system is pable of processing palette-colour (256 colour), two t (binary) and toned (64 gray scale) black-and-white (furt on b&w) images.

Before a more detailed description, the following to be noted on the structure of the database. All items marked with an identifying label (name), through which stored information on the item can be retrieved, and item may be displayed. By giving a label the item is ente into the database, where subsequently it may be linked w other items.

What is meant by linkage ? To each item we can ass an arbitrary number of keywords, which may either describe the item, or may give a reminder on some of its characteristics. In fact by giving keywords we define non-disjunct equivalence classes: those items belong to any one class, which the given keyword was linked to. Hence by linkage, we understand the joining of an item through the assignment of a keyword to other items with the same previously assigned keyword. Any one class constitutes a larger segment of the database, which the elements can easily be reached of.

A very useful feature of FrameBase is its ability to pass to and receive parameters from other database management systems. Thus FrameBase allows for the supplementing of existing databases with pictorial and text information.

Image management in the FrameBase system

FrameBase is capable of displaying standard international TIFF format binary or toned (64 scale) b&w, and 256 colour palette-coloured images. This requires a VGA adapter card, whose the features are fully utilized. It is to be noted, that the system is capable of handling not only the IBM VGA standard 256 colour graphic mode, but the markedly finer (640x480 and 800x600 pixel) geometric resolution 256 colour modes, too.

This mode of displaying images naturally requires the conversion of the three-band (RGB) images by an appropriate method to a 256 colour palette-image. This problem was solved by a special programm package joined to FrameBase. By its use it is the user who may control the process, which starts from a digitized RGB image, and results in the palette - and sometimes compressed - image accepted by FrameBase. Later on, more details will be given on this package.

The essence of the palette conversion [2] is a di
ion of the RGB colour domain, on the basis of a histogr
through the calculation of medians, into 256 disjunct par
lelepiped regions, i.e. boxes. The colours of individ
points within each box are substituted with the colour
fined by the centre of the box. According to our experien
besides its high quality colour reproduction, the other
vantage of the algorithm is in its high speed.

Another important element of the FrameBase is the r
tine which enables flexible changing of the displayed ima
size. To reach an acceptable speed, the image must be fu
stored in the random-access memory. For this purpose, Fra
Base uses the extended memory. (For large image files,
suitable extension of main memory may be required.)

In initial phase the image - in most cases - appea
on the screen in shrinked form. Through a practically unli
ited number of similarity transforms the user may magni
the image until 4 pixels fully fill the screen. The essen
of the procedure is that - depending on the similari
transform ratios - we define those rows and columns of t
image, which will be visible after magnification, and on
these are moved in main memory. Naturally, magnificati
only provides new information until the limit when furthe
magnifying can yet be performed only by multiplying existi
pixels. As a visible window, the screen may be scrolled a
panned above the image, thus all sections can be viewed
This flexible handling of images makes the system user
friendly.

The data

A very important practical problem was the solution of the efficient storage of pictorial and text information. With the latter, the ASCII standard was obviously given. The image format was not so self-evident, however, since there were a number of possibilities to choose from. The choice of the TIFF 5.0 Standard was motivated by the fact that the relevant literature anticipates its future proliferation. Even presently it is a basic requirement for all image management programms to read TIFF files. The tendencies indicate, that major digitizing equipment manufacturers also prefer the TIFF format.

An important element of the mentioned data conversion utility is the practically complete TIFF reading-writing programm. During its development, not only the mandatory rules, but also the recommendations of the standard were taken into consideration. It must be noted, that the system is capable of reading Intel and Motorola format image files, as well.

Naturally, FrameBase has a complete built-in TIFF reading routine, thus the system is capable of working with imported image files, without any costly image digitizing equipment being mandatory.

The next unit in the package contains the compression routines recommended in the TIFF standard. It must be emphasized, that our compression methods work without any loss of information. Colour and toned images are compressed with the LZW coding procedure [3], while binary images are compressed with the PackBit routine. The efficiency of the compression is between 30-70%, depending on the complexity of the image.

Attention should be brought to the fact, that due to the large volume of data, a sufficiently large background storage device is required. FrameBase supports the use of optical discs.

Database construction principles

In the general introduction part we have already (scribed the structure of the database. Through the classi cation of images by keywords, the items of the database ; grouped into larger units. Among the retrieval methods have already mentioned the direct search by labels.

Besides this, we may also perform a search in the sy tem based on keywords: by free combination of logical AND/ relations between them, we can determine an arbitrary c and/or union of the classes assigned to them.

A more visual approach to this method is the hyperma hypertext search mode in the database. In the general intr duction we have mentioned the assignment of keywords items. In practice this means, that the user may either s lect an arbitrary part of the text or depict an arbitra rectangle on the image, and through the assignment of a ke word in fact a name is given to this selected item-part. (default, the first maximum 20 characters of the select text define a keyword, which may be overwritten, of course After this the essence of the hypermap search can easily comprehended: by pointing the cursor to a depicted area (an image or text we get a list of labels, which contains a the elements of the class defined by the selected keywo (-part). After selecting one label from this list, the pr cedure may be continued.

The system uses the keywords - in order to be able t perform the hypermap search - as character strings. Thus n direct search by numeric keys can be performed. To elimi nate this drawback, building of connections with other tra ditional database managers is also possible: the FrameBas is capable of passing and recieving parameters through file. Thus traditional (eg. numeric) search procedures ca be performed by an other database manager.

Here we have to mention, that one of the developmen directions for FrameBase is the elimination of this tradeoff. Speaking of development directions, we have to mention two of our further basic plans:

- first of all we want to expand the TIFF handling unit, whose an important element is the joining of facsimile coders to the system;
- furthermore, we wish to increase the efficiency of the single monitor system by the simultaneous displaying of images and texts on the screen.

Application areas

FrameBase is a general purpose image processing system. However, there are a few application areas where its efficiency is specially advantageous. Beyond the construction of museum-, store- and spare-part-catalogues, the organisation of room reservations in the tourism industry or the sorting of medical x-ray, ultrasound-, microscopic etc. photographs are greatly facilitated for its user. In education, it can be utilised with a previously prepared image and text database.

References

[1] J. Kópházi - Cs. Molnár: Austrian-Hungarian Conference on "Beyond Number Chrunching", Graz (1989)

[2] P. Heckbert: Color image quantization for frame buffer display
Proc. Siggraph (1982) pp. 297-307

[3] Tag Image File Format Specification Revision 5.0. (1987,1988) Aldus Corporation, Microsoft Corporation

Very high precision digit recognition

Géza Álló - László Feró - József Langer
SZKI Pixel Ltd.

Summary: A camera input digit recognition system (DIANA), is described, which we have accomplished a d measuring classification method in. For this we measure 9 features for each digit, and we compute the distanc resulting feature vector from the feature vectors of previously given samples with a special distance fu Recognition is based on the nearest neighbour rule. With appropriately selected features we have achieved reco rates better than 99.9 %, with 0,8 s average computing time per digit.

Keywords: distance measuring classification, nearest neighbour rule, digit recognition

1. Introduction

Pattern recognition plays an important – and according to our estimates an ever increas role in the practical applications of digital image processing. By that we mean the classificati evaluable objects in an image, in an abstract *n* - dimension feature space, where *n* is the numb features taken into consideration during classification. Where the features describe the geom characteristics of objects, we speak of *shape recognition.*

We restrict ourselves to solve problems only in two dimensional case. In an article overvie the subject [1], Pavlidis describes four groups of methods, based on wether direct or derived, wether internal (those globally describing the object) or external (those describing the bo domains of the object) features are used. In the following we mention a few examples:

- **direct features:**
  - *internal features:* skeleton lines, set of diameters fitted on the centroid of the object, sim shapes coming from decomposition, etc.;
  - *external features*: arcs got from the decomposition of the contour, breakpoints of the con hull, radius function (Fig.1.), normal direction contour distance function [2] (Fig.2.), cur ture function of the perimeter, etc;

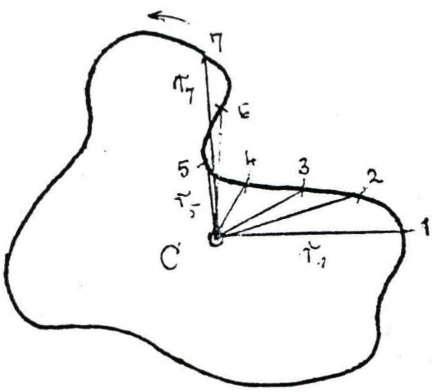


Figure 1.: The radius function of an object

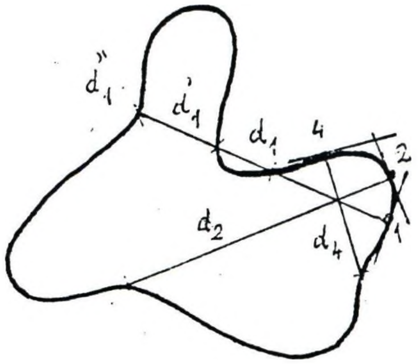


Figure 2.: The normal direction contour distance functio

- **derived features:**
  - *internal features:* moments and central moments, shape factor, convexity etc.;
  - *external features:* elongation, filledness, Freeman-code series of the perimeter, Fourier-transform of the radius function, etc.

In principle we may say, that methods using direct features require less computing, and preserve the geometric characteristics, but are sensitive to noise. In contrast, derived features require much computing, and – after transformations – the geometrical shape is no more reconstructable, however, they are size-, shift- and rotation-invariant.

2. The DIANA digit recognition system

The DIANA (DIgit ANAlysis) special application system was derived from a previously developed multifunction general purpose image processing system [3], which had as one of its functions shape recognition. In the DIANA system optimised exclusively for the recognition of digits, we have retained the *nearest neighbour* rule, but of the about 40 features originally used for as components of feature vectors, we have retained only nine.

2.1. Classification method

For classification we usually presume, that there exists a

$$D = \{d_1, d_2, ..., d_h\}$$

decision set, that defines the classes. In the teaching phase the operator provides digit samples for each class. The programme determines their $\underline{w}$ sample vectors and from their assigned δ teachings creates ($\underline{w}$, δ) teaching pairs:

$$T_i = \{(\underline{w}_{ij}) \mid \underline{w}_{ij} \varepsilon \mathbf{R}^n,\ \delta_i \varepsilon D,\ i = 1, 2, ..., h;\ j = 1, 2, ..., t_i\}$$

(in our case n = 9, h = 10, t_i = 10-20). After this the average and standard deviation vectors of the sample vectors of every class are computed

$$\underline{\mathbf{w}}_i = 1/t_i \sum_{j=1}^{t_i} \underline{w}_{ij}; \quad \underline{\sigma}_i^2 = 1/t_i \sum_{j=1}^{t_i} (\underline{\mathbf{w}}_i - \underline{w}_{ij})^2$$

Finally joioning of the results, the complete *teaching set*

$$T = \{(\underline{\mathbf{w}}_i, \delta_i) \mid \underline{\mathbf{w}}_i \varepsilon \mathbf{R}^n,\ \delta_i \varepsilon D,\ i = 1, 2, ..., h\}$$

will be constructed.

2.2. The structure of the feature vectors

A feature vector of a number to be recognised is determined by the measured 1 direct in and 8 derived, partly internal, partly external features. To enable uniform comparabilit computed features are always linearised.

During evaluation, we frequently use the moments (M_{ik}) and central moments (I_{ik}) objects, which are computed according to the following well known formula:

$$M_{ik}=\sum_{A} x^{i} y^{k};\ \text{ and }\ I_{ik}=\sum_{A} (x-x_{0})^{i}\ (y-y_{0})^{k};$$

where $x_0 = M_{10}/M_{00}$, $y_0 = M_{01}/M_{00}$, and the summing is to be extended to the minimal encl rectangle (whose the area was marked with A). Here we have supposed, that image were previ binarized, and 0 and 1 graycodes were assigned to the background and object pixels, respecti

The features calculated for each evaluable object from the measurements are the following:

- **Convexity (C)**

  The square root ratio of the object area (M_{00}) to the area of its minimal enclosing rectangle

$$C=\sqrt{M_{00}/A}\ .$$

- **Elongation (E)**

  The square root ratio of the maximal and minimal second order central moments of the obje

$$E=\sqrt{I_{2max}/I_{2min}}\ ;$$

  where

$$I_{2max}=I_{20}\cos^{2}\alpha+I_{11}\sin\alpha\cos\alpha+I_{02}\sin^{2}\alpha\ ,$$

$$I_{2min}=I_{20}\sin^{2}\alpha-I_{11}\sin\alpha\cos\alpha+I_{02}\cos^{2}\alpha\ ,$$

  are the second order central moments to the main- and secondary inertiaaxes of the obj respectively; is the angle between the main and the +x axis (for its computing, see later.)

- **Number of holes (H)**

  The number of holes within the object with a size above a given threshold.

- **Filledness (F)**

  The square root ratio of the total area of holes considered in an object to that summed with object area:

$$F=\sqrt{\sum_{H} A_{h}\Big/\left(M_{00}+\sum_{H} A_{h}\right)}\ .$$

- **X-slimness (S_x)**

The ratio of the cubic root of third order x-direction central moment to the x-dimension of the minimal enclosing rectangle:

$$S_x = \sqrt[3]{I_{30}} \,/\, x_m \ .$$

- **Y-slimness (S_y)**

The ratio of the cubic root of third order y-direction central moment to the y-dimension of the minimal enclosing rectangle:

$$S_y = \sqrt[3]{I_{03}} \,/\, y_m \ .$$

- **Left asymmetry (Θ_l)**

The cubic root ratio of the y-direction third order central moment (L_{03}) of the *background* area within the minimal enclosing rectangle to the left of the object and related to its centroid, to the y-dimension of the enclosing rectangle (more precisely only the areas in the central zone are considered (see fig. 3.), since in the border zones disturbing effects may appear.):

$$\Theta_l = \sqrt[3]{L_{03}} \,/\, y_m \ .$$

- **Right asymmetry (Θ_r)**

The cubic root ratio of the y-direction third order central moment (R_{03}) of the *background* area within the enclosing minimal rectangle to the right of the object and related to its centroid, to the y-dimension of the enclosing rectangle (see fig. 3.):

$$\Theta_r = \sqrt[3]{R_{03}} \,/\, y_m \ .$$

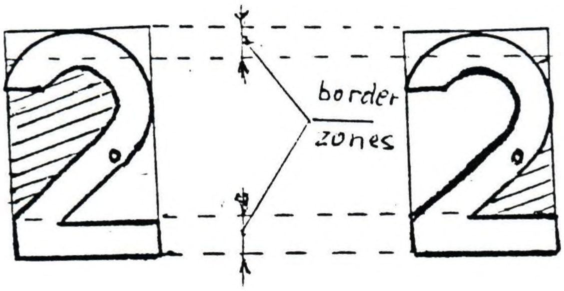


Figure 3.: Left and right asymmetry of an object

- **Main axis angle (α)**

The angle between the main inertia axis of the object and the +x axis:

$$\alpha = \mathrm{arctg}(\, I_{11} \,/\, |\, I_{20} - I_{02} \,|\,) \ ;$$

if $I_{20} - I_{02} < 0$, then $\alpha = \alpha + \pi/2$ correction is necessary.

The measured values of the listed 9 features give the 9 components of the *m*-th object feature $\underline{z}_m$; the average and standard deviation of features measured on given samples for a class p the average and standard deviation vectors for that class, $\underline{w}_i$ and $\underline{\sigma}_i$ respectively. Formally

$$\underline{w}_i = \left(w_i^{(1)}, w_i^{(2)}, \ldots, w_i^{(9)}\right), \text{ ill. } \underline{\sigma}_i = \left(\sigma_i^{(1)}, \sigma_i^{(2)}, \ldots, \sigma_i^{(9)}\right);$$

where for eg. $w_i^{(1)}$ is the *average* of the convexity of samples for class i, $w_i^{(2)}$ is that of elong $\sigma_i^{(1)}, \sigma_i^{(2)}, \ldots$ are their *standard deviations* etc.

2.3. Distance measurement

As it is well known, the key question of the distance measuring classification techniq finding the appopriate distance function (metric); an illfitted metric may result in a completely classification.

For the computing of the distance function we commence on three assumptions:

- the probabilities (more precisely the relative frequencies) of the occurrence of classe equal;
- the measured features follow a normal distribution and
- are independent of each other.

In our example, the first assumption is quite obvious, since we had to perform a sequential cour of events; thus all digits appeared with approximately equal frequency. As a result in the classi tion, we did not take into consideration the probabilities of occurrence of classes.

The second assumption cannot be accepted so readily, but since we had no statistical models could provide a better approximation, we have retained it.

The third one is obviously not true in our case; later we will discuss the methods used for correc the resultant errors.

According to the accepted criteria each object is assigned to the class, which has the hig density function in the given point (as determined by the feature vector of the object). The situa is illustrated in Fig. 4. with one dimensional feature vectors and two classes.

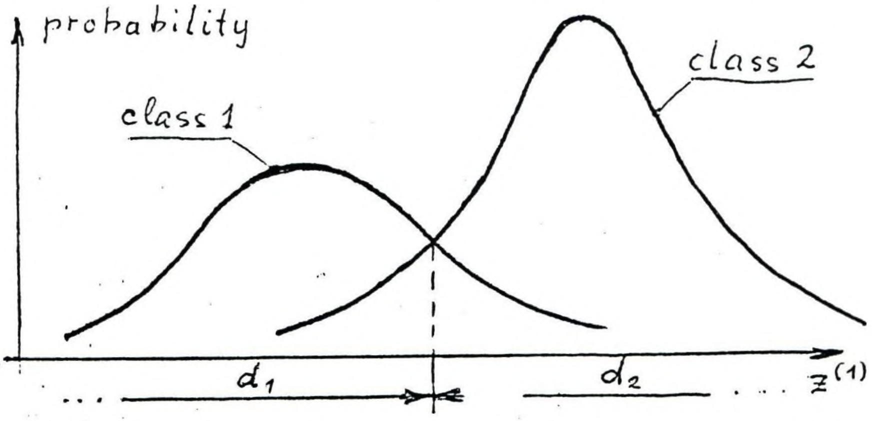


Figure 4.: Decision rule in case of one-dimensional density functions of two classes

For the determination of the distance function we should start with the formula for an *n*-dimension density function. With the above notations:

$$s_i\,(\underline{z}) = (\sqrt{2\pi}\,\sigma_i^{\,(1)} \ldots \sigma_i^{\,(n)}\,)^{-1}\ \exp(-\sum_{j=1}^{n}((\,z^{(j)} - w_i^{\,(j)}\,)/\sigma_i^{\,(j)}\,)^2)\,;$$

where i = 1, 2, ..., h (=10) is the serial number, $\underline{\mathbf{w}}_i$ and $\underline{\sigma}_i$ are the sample- and standard deviation vectors of the classes, respectively, and n (=9) is the dimension number. After this, the "distance" of the object described with the $\underline{\mathbf{z}}_m$ feature vector from class *i* is s_i ($\underline{\mathbf{z}}_m$). Instead of this - for increasing the computing speed and regarded the monotonity of the logarithmic function - we have introduced the following distance function:

$$d_i\,(\,\underline{\mathbf{z}}_m\,) = -\sum_{j=1}^{n} \ln \sigma_i^{\,(j)} - \sum_{j=1}^{n}((\,z_m^{\,(j)} - w_i^{\,(j)}\,)/\sigma_i^{\,(j)}\,)^2\,.$$

It can be proven, that the above defined function is a metric, thus it may justifiably be used for distance measurement.

Note, that the first sum is constant for any single class, thus it needs to be computed only once, after the entry of the sample vectors.

2.4. The operation of the DIANA system

The DIANA system works in two phases, based on the previously explained methods:

Teaching phase

- image capturing from camera or scanning;
- noise filtering with 3x3 median filter;
- edge enhancement with 3x3 rank filter [4];
- selection of field(s) of interest (in the example to be solved, one field consisted of 4 digits.);
- cutting to two levels using the histogram of the field of interest;
- selection of sample digits (minimum 10 for each class) and the designation of their classes;
- determination of the average sample vectors ($\underline{\mathbf{w}}_i$) and standard deviation vectors ($\underline{\sigma}_i$) of the classes.

Recognition phase

In this phase the programme continuously repeats the following steps:

- image capturing from camera or scanning;
- noise filtering with 3x3 median filter;
- edge enhancement with 3x3 rank filter;
- selection of next field of interest;
- cutting to two levels using the histogram of the field of interest;

- calculating the feature vector ($\underline{\mathbf{z}}_m$) of the next digit;
- computing d_k ($\underline{\mathbf{z}}_m$) distances;
- recognition of the digit, that is its classification into the class *i* for which

$$d_i(\underline{\mathbf{z}}_m) \le d_k(\underline{\mathbf{z}}_m) \quad (i,k=1,2,\dots,h).$$

3. Summary

The DIANA digit recognition system was developed for IBM AT 286/386 personal comp from the [3] system, with purposeful experimenting. The 10 possible classes were determine the averages of the feature vectors of the digits given during the teaching phase. We have cre the feature vectors through the measurement of only 9 features, instead of the over 40 allowe [3]. The main aim in the reduction of the number of features was the decreasing of run times the experiments indicated, the dropped features did not significantly increase the order of reco tion, or did not give stable results.

The digits were recognised on their direct and derived features. For this we have modified distance function: instead of the euclidian distance of the feature vectors in the feature space, have used the maximum of the density function (or rather to save time, its natural logarithm). resulting *hybrid* system enabled very high recognition rates (> 99.9%) with acceptable run ti (≈ 0.8 s/digit).

As a further development, we want to take into consideration that the measured features are independent. For this reason, we determine the $\underline{\Omega}$ standard deviation matrix and, solving th

$$\underline{\Omega}\,\underline{\mathbf{u}} = \mu\,\underline{\mathbf{u}}$$

eigenvalue equation, we calculate *n* uncorrelated $\underline{\mathbf{u}}_j$ base vectors. By transforming the feat vectors into the *n*-dimensional space spanned by the base vectors, their components will beco uncorrelated. Thus always one density function will be dominant in every point of the feature spa

Another of our plans is to use a transputer card for the further reduction of recognition tim According to our preliminary measurements, we expect a 8-10 fold increase in speed.

References

[1] T. Pavlidis: A review of algorithms for shape analysis
CGIP-7 (1978) pp. 243-258.

[2] D. Vernon: Two-dimensional object recognition using partial contours
Image Vision and Computing - 5/1 (1987) pp. 21-27.

[3] L. Fer6: The SEAKING image processing system - Users Manual (in hungarian)
SzKI Pixel Documentation 001/91 (1991) Budapest.

[4] R. M. Hodgson et al.: Properties, implementation and applications of rank filters
Image Vision and Computing-3/1 (1985) pp. 3-14.

An efficient image compression programm for personal computers

Gy. Csaba Hegedüs
SZKI-PIXEL Ltd.

Introduction

In SZKI-PIXEL Ltd. (and in its predecessor, the Mathematical Laboratory of SzKI) during the near 15 year work in image processing it was a reoccurring basic task to compress images. Some actual implementations were created, like the CCITT for binary facsimile coding, and the coding of toned images in the PIGALLE (PIcture and text GALLEry) pictorial database management system.

This presentation focuses on a project with the aim of developing a high efficiency image compression programm for the IBM PC.

In terms of efficiency both the large compression ratio and the high speed is important. Naturally a basic requirement is to minimise information loss, or to make it user selectable.

In the project we work on the compression of toned black-and-white (further on: b&w) images, and information loss is connected to the very difficultly definable feature of how visible is the change of quality.

The first part of the presentation gives a short overview of the image compression methods. The second part gives a briefing on the 'TDCT V1.0' programm (the original aims, and the applied special solutions), while the third part is on the possibilities of further development. Finally we list the more important articles and brochures used during development.

1. On image compression in general

The aim of image compression is to achieve the largest possible saving in space (or during image transfers: the saving of cost/time), with procedures allowing for the reconstruction of the image with a quality meeting the demands, that may be implemented efficiently.

The image compression procedures may be classified into two groups, procedures with:

1) no information loss, and

2) with information loss allowed

In the following, only procedures of the second class will be discussed.

Among these procedures a decade ago only a 1:10 compression ratio was the attainable maximum, which is very far from the theoretical limit expressed by the entropy of the source. (Also the information theory approach does not take into consideration the laws of human vision,

provide for further neglections.)

The new research results have far surpassed this limit. (Eg. the separated contour and te coding procedure based on the direction sensitivity of the neurons have resulted in compre rates of 1:70 [1].)

Image compression is usually done in two steps:

1) the pictorial data are converted into 'sets' of messages, which store the original pictoi information very truly. These messages may be the original pixel values, groups of pixels (blocks), or function values calculated on pixel groups.

2) Of the messages the less important ones are deleted, with the resulting ones containin most possible pictorial information. These are coded efficiently.

While the early (first generation) image compression procedures stressed the second step new (second generation) procedures equally emphasize both steps.

Among the first generation systems the so called transformation methods are the most pro ing. The basic principle is to produce instead of the strongly correlated pixel values less correla more independent coefficients, and have them described with an efficient coding. By some asp the Karhunen-Loeve transform (KLT) is optimal, since the measure of correlation between resulting coefficients is zero, but it has no known algorithm that may be implemented reasonable efficiency. Among other more rapidly computable ortogonal transforms (eg. Four Haar, Hadamard [2]) the discrete cosine transform (in the following: DCT) nears most advantageous features of KLT. Instead of a detailed description of the first generation procedu we refer to the ample literature (eg. [3])

An example for a second generation system is the description of the image with spots hav an efficiently coded shape and texture. (For more see [1].)

Among the slowly developed standards in image compression methods the DCT method beca accepted for the coding of toned b&w/colour images. The need for standardization did not o originate from software developers, but also it became necessary for the development of equipm for the planned HDTV picture transferring on existing broadcasting channels. The compressi requirement for this latter is 1:10, but for eg. in video telephone systems the possible rate of trans is 8Mbit/s which, assuming the CCIR Rec.601 216 Mbit/s dataflow (with an 8 bit A/D) require compression ratio of 1:27. The 'International Standards Organization' (ISO) and the 'Consultati Committee on International Telephone and Telegraph' (CCITT) recommendations ('JPEG': Joi Photographic Experts Group , [4]) may be considered currently accepted. It was by using the standards, that for eg. the A121 circuit was designed, or also the CCUBE Microsystems CL-5 circuit (see also: [5], [6]). The image compression cards, operating at video speeds, based on the chips have entered the market at a price of a few thousand german marks. An additional informatio that the Nextdimension graphic card for the Next machines (with an I860 RISC processor) contai a built-in CL-550 image compressing processor - at a price of about 6000 DEM.

Standardization was predated by a considerable amount of research work, taking into consider tion the largest possible (selectable) compression ratio, and the highest possible speeds wh implementing the software. As a result, when starting the project we have in principle used the recommendations of the JPEG.

2. The 'TDCT V1.0' programm

When developing the 'TDCT' (TIFF-DCT) programm we had the following aims:

- the programm should employ DCT coding, and should deviate from JPEG recommendations in exceptional cases only;
- the 'V1.0' (first) version should be capable of the handling of toned b&w TIFF images;
- an efficient implementation (assembly) should ensure acceptable execution times;
- the compression ratio should be user selectable;
- during image reconstruction the output image should be of arbitrary size;
- operation should be with both menu and command line;
- the important modules should exist in a separate library (with a high level language interface).

The final reason for starting the project was the increasing demand for image compression, and the very high price of image compressing hardware devices.

2.1 Compression

The basis for the DCT image compression procedure is the following transformation [7]:

$$\mathrm{DCT}(k_1, k_2) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x(n_1, n_2) \cos\left(\frac{2\pi(2n_1+1)k_1}{4N}\right) \cos\left(\frac{2\pi(2n_2+1)k_2}{4N}\right)$$

where N is the block size; x(,) is the pixel gradation code; n_1, n_2, k_1, k_2 are block relative positions.

The above transformation is usually performed on blocks containing 8x8 pixels, although there are examples for larger or varying block sizes [8]. The correlation between the received coefficients is much smaller than the correlation between individual pixel values, thus the redundancy of the data is smaller. In practical implementation the RVFFT (real value of fast Fourier- transform) is more advantageous. On the reorganised data, after rotation, with an 8x8 block size it requires 32 multiplications and 320 additions, in contrast to the 112 multiplications and 470 additions of the traditional DCT [10].

The first possibility for compression is the fact that different coefficients have different visual meaning, and a large part of them may be dropped without a deterioration in image quality, or may be displayed with a lower precision. The change of coefficients can be performed efficiently e.g. by a LUT (look-up table) changing from position to position. (We should note, that there are other approaches for decreasing the number of coefficients. As per [9] by sorting the coefficients according to their absolute values, we receive the order of their importance, and it is sufficient to retain only the first few elements of this set. The JPEG recommendations contain a different solution.)

The most efficient coding of the remaining coefficients forms the next step of compressio This may be achieved with a zig-zag scanning of the coefficients, and with some kind of (variab code length) entropy coding, for eg. with adaptive Huffman-coding [10], or with arithmetic codir [11]. The efficiency of the coding may be enhanced through preprocessing with for eg. medi filtering (offering a 10% advantage), or the preceding uniformisation of the blocks with rot tion/mirroring, which naturally should be considered during image reconstruction. (This latter not among the JPEG recommendations.)

The user, when using the TDCT programm, may select the rate of compression.

2.2 Reconstruction

The reconstruction of the images differs in principle from the compression procedure only b the fact that there are only a few data items per block, since the rest have been neglected. Thus th speed of reconstruction may be much greater than that of compression.

Reconstructing compressed images, usually a size transformation is necessary, so we have buil such functions into the TDCT programm. For this the original image function is to be reconstructe in (real coordinate) positions depending on the ratio of size change, and the pixel values of th output image are received from their quantization.

In general, reconstruction can be performed by using an interpolation function. In one dimen sional case:

$$P'(x) = \sum_{n} P(n)\, f(x-n) \ ,$$

where P(n): original pixel value in the position *n*; f (x-n): interpolation function; P'(n): output pixel value (at the place of the new sampling).

It can be proved, that the ideal interpolation function is sin(x)/x, which should be applied to al input image pixels. Aside from its practical impossibility, this transformation has no real meaning, since the attainable precision is strongly limited by the quantization errors of output pixel values.

The simplest interpolation function designates the pixel value of the closest neighbouring pixel for resampling:

$$f(x) = \begin{cases} 1, & \text{if } 0 \le x < 0.5; \\ 0, & \text{otherwise.} \end{cases}$$

For a high level work, this solution is unacceptable.

The first degree interpolation function:

$$f(x) = \begin{cases} 1-x, & \text{if } 0 \le x < 0.5; \\ 0, & \text{otherwise.} \end{cases}$$

In two dimensional case, this results the new pixel value $P'(x,y)$ as the weighed sum of $P([x],[y])$, $P([x],\lceil y\rceil)$, $P(\lceil x\rceil,[y])$ and $P(\lceil x\rceil,\lceil y\rceil)$ pixel values (the so called bilinear interpolation). (Here $[x]$ and $\lceil x\rceil$ represent the integer part of the number *x*, and the next greater integer, respectively.)

Reordering of the formula, one can reduce the number of operations to four multiplications and four additions [12]. What is more, comparision of integer numbers is enough in most cases instead of multiplications (if the change in the neighbouring pixel values is below a certain threshold, the result of the multiplication is always 0, thus it is unnecessary to perform it).

A further increase in efficiency can be attained if instead of multiplications we use previously filled LUT-s with the possible intermediate results. Then the computing – apart from address calculations – is reduced to four additions, and the number of operations decreases with the increasing magnification ratio. The disadvantage of this method is the increase in required storage space.

The bilinear interpolation creates a somewhat blurred image, but gives results far exceeding the previously described one.

The third degree spline interpolation has even better characteristics [13]:

$$f(x) = \begin{cases} x - 2x + 1, & \text{ha } x \leq 1; \\ -x^2 + 5x - 8x + 4, & \text{ha } 1 < x \leq 2; \\ 0, & \text{otherwise.} \end{cases}$$

The 16 neighbour connections in two-dimensional case can be solved with the above tabular method by using 16 additions/subtractions, and the image quality meets all possible practical requirements.

In the TDCT programm the user may choose between the above three solutions.

3. The near future

According to feedbacks and development experiences we want to solve the management of different popular file formats, and the compression of colour images. Using the efficient core modules of the system, we find practical to form later on a standalone programm product working in novel systems-programming environments (like WINDOWS 3.0 etc.).

References

[1] M. Kunt - A. Ikonomopoulos - M. Kocher: Second-Generation Image-Coding Techniques
Proc. IEEE - 73/4 (1985) pp. 549-574.

[2] G. Álló - Gy. Cs. Hegedüs - D. Kelemen - J. Szabó: Basic problems of digital image processing (in hungarian)
Academic Press, Budapest (1989) pp.475.

[3] A. K. Jain: Image data compression: a Review
Proc. IEEE - 69/3 (1981) pp.349-389.

[4] Initial draft for adaptive discrete cosine transform technique for still picture data compr
sion standard
ISO document (1988) ISO/JTC1/SC2/WG8 N800.

[5] INMOS takes on image compression with discrete cosine transform processor
Product inform in COMPUTER DESIGN (Aug. 1.,1989) p. 86.

[6] W. Liebsch: Parallel architecture for VLSI implementation of a 2-dimensional discrete c
transform for image coding
3-rd Int. Conf. on Image Processing (IEEE-ICIP) (1989) pp. 609-612.

[7] W. H. Chen - W. K. Pratt: Scene adaptive coder
IEEE Trans. on Comp. (March, 1984)

[8] I. Dinstein - K. Rose - A. Heiman: Variable block-size transform image coder
3-rd Int. Conf. on Image Processing (IEEE-ICIP) (1989) pp. 613-617.

[9] E. Meyer - S. Gaertner: An intrafield DCT-codec for consumer application
3-rd Int. Conf. on Image Processing (IEEE-ICIP) (1989) pp. 553-557.

[10] Ch. Guillemot - P. Duhamel: A new transform for image coding with reduced complexity
and same performance as DCT
3-rd Int. Conf. on Image Processing (IEEE-ICIP) (1989) pp. 576-580.

[11] G. R. Seabrook: Arithmetic coding - an alternative VLC strategy for video coding
3-rd Int. Conf. on Image Processing (IEEE-ICIP) (1989) pp. 613-617

[12] Gy. Cs. Hegedüs: Fast geometric correction of digital images Computer analysis of Imag
and Patterns
Proc. 2-nd Int. Conf. on Automatic image processing, Wismar (1987) pp.62-69

[13] P. Stucki: Image processing for document reproduction
in: Advances in digital image processing, Plenum Press, N.Y. (1978) pp.332

Porosity Determination of Paper Surface Based on Image Analysis

János Győri and Unto Pulkkinen

Technical Research Cetre of Finland, Graphic Arts Laboratory

1. Problem statment

The importance of porosity on paper is high, because the ink partly penetrates into the paper pores. If the porosity of the paper surface variates, the print quality variates too. It causes mottle effects degradating the visual print quality. Papers can divide into too main classes: coated and uncoated papers. The pores on coated papers are small in diameter, about 0.1 - 2.0 µm. For example the pore size on a newspaper surface is about 0.1 - 5.0 µm. There are smaller pores on the surface of fibres and larger ones between fibres. Because the pore sizes can be small, about the same size than the visible light wavelength, it is difficult to measure the pores by optical or optoelectronical methods. For measuring pore characteristics the possibilities of image processing methods are obvious. By the help of image analysis methods it became possible to measure and compute the teritories and the shape of the pores on the coated paper surface. From these datas it will be made statistics of different kind of papers and these will be used in quality evaluation.

2. Main approach, solution

The main problem and goal was to develope a method and a system which is able to create porosity statistics of sample paper surface. The paper porosity analysis system consists of :

- Microcomputer (AT) equipped normally
- Image bard (Matrox PIP 1024 or Tagips)
- CCD B&W video camera
- Microscope
    - Optical microscope: magnification ranging 100x...2000x, 2...3 lights
    - Electron microscope: magnification ranging 100x...5000x

In case of optical microscope the lights must be positioned as near as it is posiible to the horisontal plane direction. The inclination can be maximum 20 degrees. The openings of the pores this way seems to be like dark spots imaged from vertical direction. Fig 1. shows schematically the pore measurement system hardware configuration.

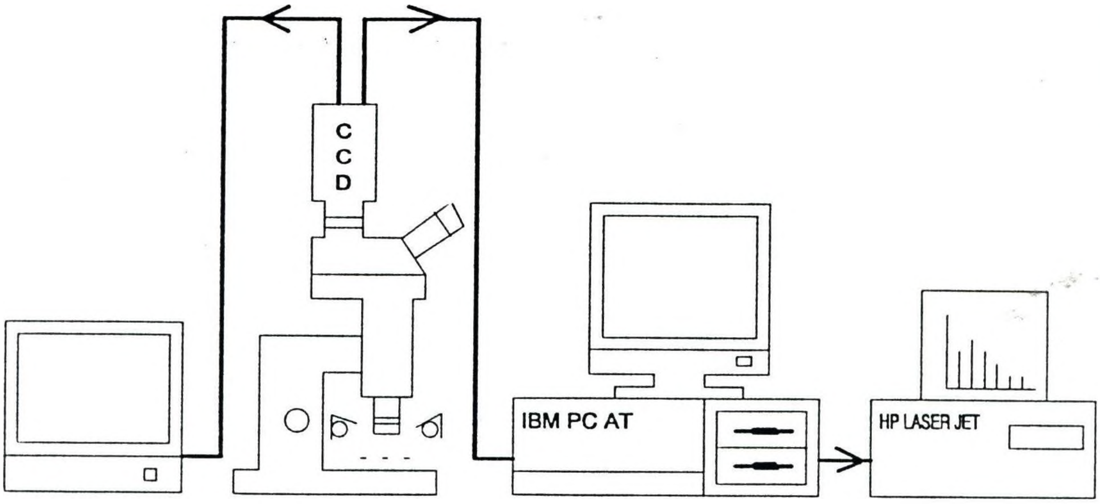


Fig 1. Paper surface measurement system

3. Data structures

The most important algorithm of this system is the *Segmentation algorithm.* In that section of the program are computed the various segment attributes, which are collected into structures. The attribute fields of that structures are the following:

- area
- perimeter
- weight point x coordinate and weight point y coordinate
- minimum moment
- minimum moment angle
- orthogonal moment
- form complexity: (perimeter* perimeter)/area.

One of this structure is typical only one of the pores. These structures are linked to each other by a double linked data structure method.

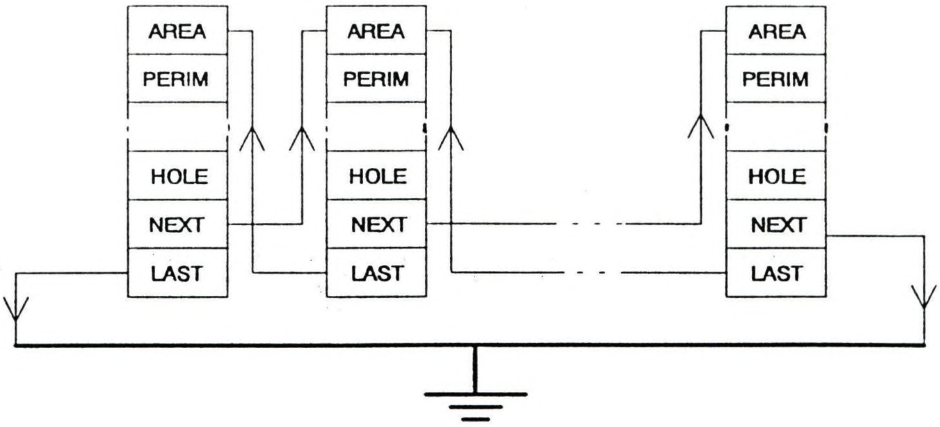


Fig 2. The data structure of the segmentation algorithm

This data organisation method is very flexible easy to merge a new object and easy to delete an already existing object.

4. Picture analysis methods

The picture coming from the microscope is digitised to 256 gray level picture in the memory of the frame grabber. The pores on the grabbed picture look like unregular shaped dark spots. The main picture analysis problem is to select these dark spots from the uneven lighter background.

4.1 Segmentation by polygonal extraction of boundaries from a raster scan

Before this algorithm it is nesessary to do some preprocessing procedures, e.g. filtering and thresholding. The filtering is taken place by a simple avereage and one isolate point detection convolution matrixes,

$$A=\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad P=\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad (1)$$

By these filters it is possible to eliminate the discontinuities of the segments belonging together, and it is possible to eliminate the isolate point "pores" too. The thresholding method is a global thresholding using the histogram of the gray level picture. After making the histogram the program scans it by cutting it into two parts and computes how much point is lighter and how much is darker than the given point. From that values than it makes two distributions one for the brighter and one for the darker levels. The threshold value will be that value where the two kinds of distributions are equal. If there are more, the firts one is taken into account. The result will be a binary image where the pores are black and the background is withe.

The segmentation algorithm shortly described here uses dynamic data structure. The frame may include any number of objects, each of which may contain any number of holes and there is no restriction on complexity. There is no need to store the whole picture in the memory, only the transition points of the raster line currently beeing processed are required. Of course, storage must be provided for the above mentioned dynamic data structure.

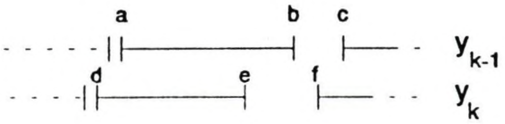


Fig 3. Current segment on previous line and new line

On Tab 1. we can follow how the the edge point connections (a,b,c,d,e,f) are made. Th notation is corresponding to the Fig 3.

Condition	Interpretation
e < a	Segment [d,e] is the begining fo a new object in the frame. Connect e to d.
d > b	Segment [a,b] is the termination of an existing object in the frame. Connect a to b.
neither above	Segment [d,e] overlaps with segment [a,b]. Connect a to d.
f < b	Segment [a,b] has split and [e,f] could be the begining of a hole. Connect e to f.
e > c	Segment [d,e] is merging with segment [a,b] and the following segment and [b,c] could be the termination of a hole. Connect c to b.
neither above	Connect e to b.

Tab 1.

The computation of the segment attributes mostly takes place during the segmentation algorithm. After the segmentation some statistical datas are computed, for example average, variance. By the help of these statistical datas there is the possibility to do some filtering. Depending on the microscope magnification the user can deside what is the less pore area to be accepted by the measurement system. Segments under this limitation will not be taken into account, and their influence will not appear in the final pore report.

4.2 Segmentation by directed trees method

This other segmentation method needs other kind of preprocessing. This requires a negated edge gradient picture. Before creating the edge gradient picture it is useful to do an edge sharpening. There are some convolution matrix solutions but these were not flexible enough, they produced a lot of uncontinous segments which would have belonged together originally. To avoid this problem we used a gradient determined gray level morphological edge sharpening procedure. This is a two passes operation which makes first an eorosion than makes on the result picture a dilatation. There were a lot of possibilities make do from the sharpened picture a negated edge gradient picture. The Prewitt filter seemed to be the best.

The segmentation algorithm detects and labels homogeneous areas in an image using directed trees for labeling. These directed trees devide the image into disjoint segments. The number of the detected segments depends on a before given parameter so called sensetivity parameter. This sensetivity parameter distinguishes between the edge gradient on slowly varying plateau regions from that at the valley regions. Scanning the picture the program creates directed trees by linking the image poins to each other taking care of not to do any directed circles. That way the parent of each point is determined so it is enough to label only the root of each directed trees.

5. Summary

We have presented a not too expensive hi-tech paper surface and paper printability measurement system based on image processing methods. There were implemented two kinds of segmentation algorithms and several filters and other picture analysis programs for preprocessing. Several segment attributes are computed and the final results are printed in the Porosity report.

6. References

D. V. Capson
An Improved Algorithm for the Sequential Extraction of Boundaries from a Raster Scan
Computer Vision, Graphics, and Image Processing 28, 109-125 (1984)

M. C. Morgan
A Morphological Transformation for Sharpening Edges of Features Before Segmentation
Computer Vision, Graphics, and Image Processing 49, 95-94 (1990)

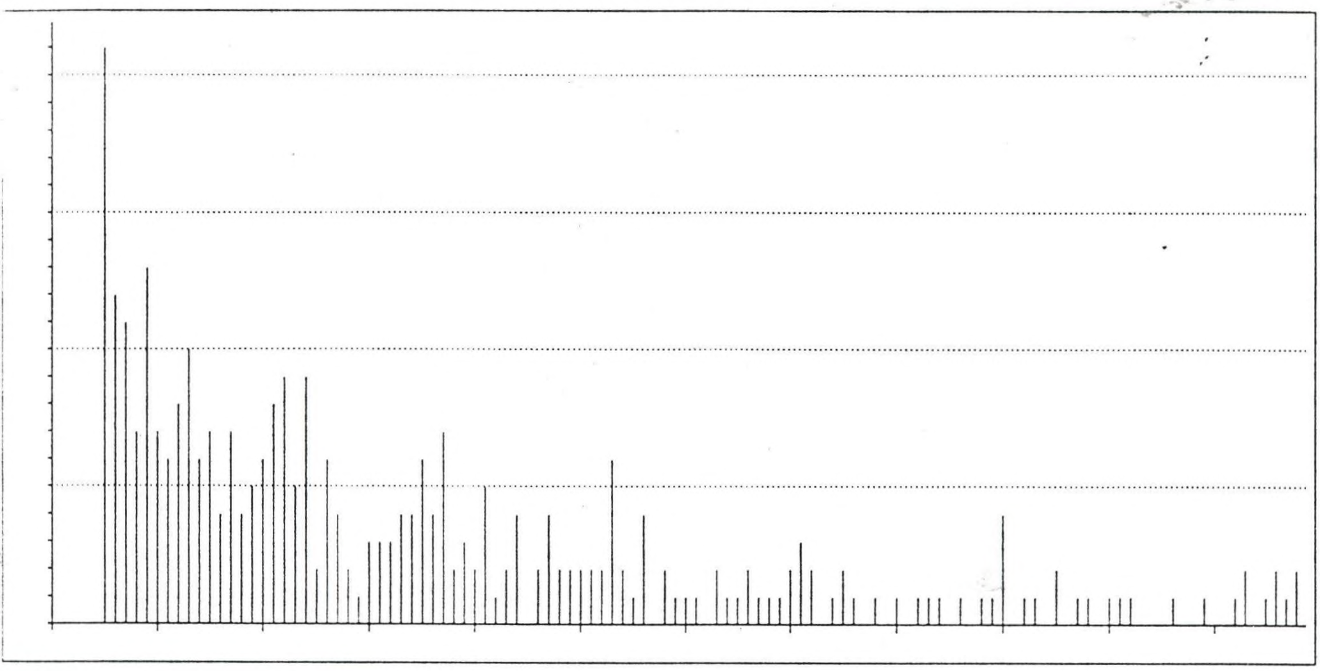
P. M. Narendra and M. Goldberg
Image Segmentation with Directed Trees
IEEE Transactions on Pattern Analysis and Machine Intelligence
vol. PAMI-2, No. 2, March 1980

R. C. Gonzalez P. Wintz
Digital Image Processing
Addision-Wesley Publishing Company, 1987

Paper surface measurement report: AREA disrtribution

```
Statistical datas
     Minimum pore AREA...........: 5
     Maximum pore AREA...........: 1258
     Average pore AREA...........: 77
     Variance of pore AREA.......: 2916
     Minimum number of given AREA: 0
     Maximum number of given AREA: 21
```

AREA distribution diagram



The datas of the diagram numerically:

```
   5:  21        30:   3        56:   4        89:   1       153:   1       283:   1
   6:  12        31:   3        58:   2        90:   4       157:   1       285:   1
   7:  11        32:   3        59:   1        92:   1       159:   1       288:   1
   8:   7        33:   4        60:   1        93:   1       162:   1       304:   1
   9:  13        34:   4        61:   1        95:   2       168:   1       309:   1
  10:   7        35:   6        63:   2        97:   1       170:   1       312:   1
  11:   6        36:   4        64:   1        98:   1       176:   2       318:   2
  12:   8        37:   7        65:   1       100:   1       177:   1       319:   1
  13:  10        38:   2        66:   2       101:   1       179:   2       324:   1
  14:   6        39:   3        67:   1       102:   1       183:   1       356:   1
  15:   7        40:   2        68:   1       106:   1       186:   1       357:   1
  16:   4        41:   5        69:   1       109:   1       187:   2       358:   1
  17:   7        42:   1        70:   2       112:   1       188:   1       363:   1
  18:   4        43:   2        71:   3       113:   2       194:   1       384:   1
  19:   5        44:   4        72:   2       115:   1       197:   1       406:   1
  20:   6        46:   2        74:   1       116:   2       199:   1       441:   1
  21:   8        47:   4        75:   2       117:   1       204:   1       456:   1
  22:   9        48:   2        76:   1       118:   2       212:   1       460:   1
  23:   5        49:   2        78:   1       120:   1       213:   1       479:   1
  24:   9        50:   2        80:   1       124:   2       231:   1       537:   1
  25:   2        51:   2        82:   1       137:   1       246:   1       586:   1
  26:   6        52:   2        83:   1       138:   1       258:   1       706:   1
  27:   4        53:   6        84:   1       144:   2       266:   1       856:   1
  28:   2        54:   2        86:   1       147:   1       267:   1       910:   1
  29:   1        55:   1        88:   1       149:   1       270:   1      1258:   1
```

ON PAPER ANALYSIS AND SIGNAL PROCESSING ALGORITHMS

Unto Pulkkinen

Technical Research Centre of Finland, Graphic Arts Laboratory

0. SUMMARY

We present here paper printability measures describing their nature, how they are produced by different process variables , what kind of effect they have on print quality and how they can be characterized. The printability factors are for example: roughness, porosity, formation, marking, fibre orientation and texture properties. The second emphasis area in this paper is the signal processing methods needed in determining the measures mentioned above. The emphasis is on fast methods suitable for online implementation on digital signal processors. The interested reader can dive deeper into these topics with help of the included reference list.

1. PAPER PRINTABILITY MEASURES

The production on paper machines and printers is massproduction and the web velocities are 5-24 m/s. These things set high requirements for online measurements and signal processing. Typical important factors affecting the paper printability are: roughness of paper surface, porosity, strength of marking (regular texture from wires and felts), paper formation, fibre flocculation and the orientation of fibres and texture patterns.

Roughness of paper affects adhesion of ink on paper surface. If the paper is very rough, raster dots may adhere only partially. The topography values characterize the roughness of paper. The surface contains small hills and walleys, height about 1...5 μm. The number of hills/walleys pro unit area and their width are measures of roughness. The variance of topography values is also a measure characterizing the roughness. The roughness may variate in two principal directions: the web direction and the cross web direction. Thus it is necessary to evaluate some roughness measures separately in both directions. Some measures are two dimensional by nature, for example the number of walleys/hills pro unit area. The power spectrum computed from topography data is a compact and illustrative way to present the character of the roughness. It shows the roughness 'energy' distribution as a function of spatial roughness wavelength. The power spectrum may be evaluated either in truly two dimensions or in two principal directions. The spatial wavelength area is 10 μm ... 1 mm. The determination of roughness statistics presumes a topography scanner, which allows to scan small spatial surface area.

Porosity of surface affects the ink absorption on paper. If the porosity on paper surface variates, it causes mottle on even coloured regions. The distributions of pore size and spatial pore density should be even and on suitable level, not too high and not too low. In case of uncoated papers the pores are formed between wood fibres, when many fibres are crossing each other. The width of pores are 1...10 μm. In case of coated papers the pores are formed between particles of coating material. In this case the width of pores is 0.1 ... 2.0 μm. The pores may reach from surface into the paper as caves. Also in the paper there are some pores, which do not reach the surface. We can characterize the porosity by the area distribution of pore openings and by the spatial density of pores.

Marking means those mechanical impressions on paper surface sometimes caused by wire or felt of a paper machine. The marking forms regular grid like figure, where the base threads in horisontal and vertical directions cause the strongest marking components. Marking is a regular texture pattern. The wavelengths of a marking texture are typically 0.1 ... 1.0 mm. If the wire or felt is very worn or otherwise deteriorated, the marking is stronger. Today the printing is normally raster printing, where ink dots form dot matrices. The width of dots is 50...200 μm and the gap between dots is about the same size. If these two regular texture patterns interfere now mutually, there may be brought about a negative visual effect decreasing the quality of printed matter. We can characterize the marking determining the principal wavelengths and their strength. This must be done separately for web direction and cross web direction. It is possible to carry out the marking analysis based on optical transilluminated image of paper, but it will be done better based on topography image, because the marking is a mechanical impression on paper.

Formation describes the spatial variation of grammage of paper. It is very difficult to achieve a completely even fibre layering on paper machine wire despite of the fact that the fibres will be spouted onto the wire dispersed in the 2...3 % pulp solution. Critical moment of formation generation is, when the pulp spouts out from headbox. The spatial wavelength area of formation is 1 mm ... 5 cm. We can evaluate visually the paper formation by looking through it against the light. There we see a cloudy picture, where the cloudiness is caused mainly by the grammage variation. Formation can be measured indirectly by analysing the optical transilluminated image. More accurate method is to use beta radiogram image, where the graytone values correspond directly to the grammage values. We can characterize formation, like roughness, by variance of graytones and by power spectrum. Bad formation in paper may cause unevenness on printing. Especially, if the paper is calendered, the thicker areas have become denser and thus the porousity has become lower. This causes mottle on printing via porousity changes.

Flocculation means the bundling of fibres, so that they can be seen on paper as some kind of graininess. It causes visual unevenness on paper and prints. The geometrical dimensions of flocculation are 0.01 ... 1.0 mm, in other words below the scale of roughness and formation. The fibre bundles may arise from statistical reasons, when fibres are settling down on the wire, or from chemical reasons, when fibres are not completely dissolved during pulp production. We can characterize the flocculation by size distribution of flocs and by their number pro unit area.

Fibre orientation tells us, in which directions the fibres are oriented and in which relative amounts. Fibre orientation affects the tensile strength of paper. If the fibres are mainly oriented for example in the web direction, the tensile strength is also higher in this direction. Additionally, the paper shrinks differently depending on the fibre orientation. This may cause difficulties in printing machines, if geometrical dimensions of paper are changed. The distribution of fibres in different angles is as follows (/1/Rance):

$$A(\phi) = \frac{1}{\pi} + e\cos(2\phi), \qquad \text{where } e = \frac{3}{\pi}\,\frac{N(\pi/2) - N(0)}{N(\pi/2) + N(0)} \qquad (1)$$

$N(\pi/2)$ = the number of fibres in cross web direction
$N(0)$ = the number of fibres in web direction

The preceding formula presumes, that we first determine the number of fibres in the two main directions on the area under study. After that we can determine the parameter e and we can draw the distribution. In reality the fibre orientation will today be determined by measuring the tensile strength in two main directions and using these measures in place of real fibre numbers in each direction. Second possibility is to capture a magnified optical image from the paper surface and anayse it. In analysis we can compute autocorrelation function in two main directions or truly in two dimensions simultaneously. After this we can threshold the autocorrelation functions, using as threshold for example value 0.25, starting from zero lag (and thus from value 1.0). We stop, when the function value drops first time under the threshold. We have now the surrogate values for the number of fibres in each main direction for N(0) and N(π/2).

Texture properties mean the visual patterns of paper surface and their characteristic features. The texture may be regular or random. Marking is a typical regular texture. The spatial location and orientation generated by random layering process gives a more or less random texture. The goal is to make the paper surface texture as fine graded, random and even as possible. In that case the visual impression is best and the texture do not disturb the print. The texture measures may be as follows (/15/Conners):

Energy $\Sigma_i\Sigma_j M(i,j|d,\Theta)^2$ (2)

Entropy $-\Sigma_i\Sigma_j M(i,j|d,\Theta)\log M(i,j|d,\Theta)$ (3)

Contrast $\Sigma_i\Sigma_j (i-j)^2 M(i,j|d,\Theta)$ (4)

Correlation $\dfrac{\Sigma_i\Sigma_j (i-\mu_x)(j-\mu_y)M(i,j|d,\Theta)}{\sigma_x\sigma_y}$ (5)

Homogeneity $\dfrac{\Sigma_i\Sigma_j M(i,j|d,\Theta)}{1+|i-j|}$ (6)

where M = co-occurrence matrix

d = intersample space in matrix M

Θ = direction in matrix M

σ_x, σ_y = standard deviations in x and y directions in M

μ_x, μ_y = means in x and y directions in M

The second possibility, instead of or in addition to preceding measures, is to use Markov model (MRF=Markov Random Field) when characterising the texture properties. Based on the estimated MRF parameters we can deduce the attraction, repulsion and orientation properties.

2. BASIC SIGNAL PROCESSING METHODS

When computing different measures it is possible to use often the same basic DSP methods, like FFT, filters, computation of power spectrum and autocorrelation. In the next chapter we shall discuss basic methods mentioned above.

2.1 Fast Fourier Transform (FFT)

The transform coding (most often FFT) of time domain or spatial domain data vector is one of basic algorithms in many digital signal processing methods and applications. Thus it is important to compute the FFT as fast as possible, and this way the total computation time will be smaller. This requirement of efficiency is essential especially by online applications. In its basic form in (7) the FFT requires a huge amount of computation. The FFT is of very central importance, when fastening many computation algorithms, and because of that it should be computed as fast as possible.

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk} \qquad (7)$$

where N = number of data elements, here 2^n, n>0 is integer

x(n) = n:th data element

X(k) = k:th frequency component

n = time domain (spatial domain) index 0...N-1

k = frequency domain index 0...N-1

W_N = exp(-j2π/N)

According to the formula (7) the amount of computation steps is proportional to N^2. When N is power of two, we can divide the FFT to two half size subtasks, and continue this way recursively up to the level, where FFT contains only two elements. This is the basic technique to decrease computation steps (Figure 1.).

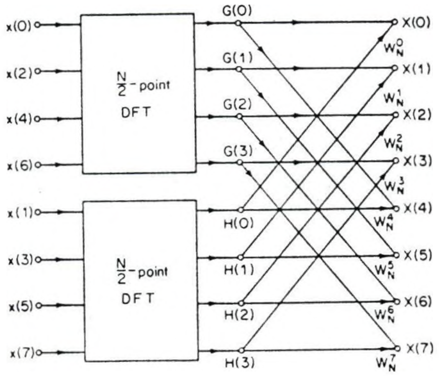


Figure 1. Dividing the FFT into two half size subcomputations (/2/Oppenheim).

2.1.1 Bit reversing

In the preceding scheme (Figure 1.) the original data sequence is arranged so that after computation the FFT components are in the right order. We can see that corresponding indices are related by value so, that the one is bit reversed value of the other. For eight point FFT data the correspondence is as follows:

original position	--->	new position
0 (000)	--->	0 (000)
4 (100)	--->	1 (001)
2 (010)	--->	2 (010)
6 (110)	--->	3 (011)
1 (001)	--->	4 (100)
5 (101)	--->	5 (101)
3 (011)	--->	6 (110)
7 (111)	--->	7 (111)

When the FFT algorithm is optimized in butterfly computation, it is additionally possible to shorten the computation time by optimizing the bit reversing operation. If there is a 1024 points long FFT, the bit reversing takes about 20 % of computation time. This contains the motive for optimization. One algorithm for bit reversing is presented in reference (/3/Rodriquez). It is an enhancement of conventional widely used bit reversing algorithm (/2/Oppenheim). Rodriquez's algorithm makes the operation about 1...10 % faster depending on the size of FFT. For example, if N=128, the improvement is about 8.5 % and if N=512, the improvement is about 4.2 %. The advantage of this algorithm is its simplicity.

There is even faster algorithms for bit reversing. Some of them are described in references (/12/Buneman), (/4/Evans) and (/5/Walker). For example Walker's algorithm computes bit reversal about three times faster than Rodriquez's algorithm. The program needed is about three times longer. If we can use the same permanent length of FFT, then the fastest way to do bit reversing, is to do it beforehand only once into index vector and use it by pivoting the data elements.

A different way to go around the bit reversing is to compute each time two butterflies at the same time before restoring the data elements back to memory. This is possible by choosing appropriate pairs of butterflies, where two of the data elements exchange place. This method is described in reference (/12/Buneman), where the method is applied to Fast Hartley Transform.

The normal additional way to reduce computation effort is to compute the sine and cosine terms needed beforehand and store them into table. When needed, the value will be fetched from the table using the index computed from sine/cosine argument.

2.1.2 Real input data

If we can assume the input data to be real, we can reduce computations remarkably (/11/Sorensen). In that case there in transform are

$X(0)$ and $X(N/2)$ real valued and
$X(k) = X^*(N-k) \quad 1 <= k <= N/2-1.$

These symmetries reduce the computation work to about half, because components $X(k)$ $N/2+1 <= k <= N-1$ and imaginary parts of $X(0)$ and $X(N/2)$ need not to be computed and stored. As a consequence the remaining FFT components with their imaginary parts can be stored in place of the original data. The computation steps needed are in case of radix-2 FFT:

$(3/4)MN-(5/2)N+4$ multiplications and
$(7/4)MN-(7/2)N+6$ additions,

where $N=2^M$. This kind of practice makes the computation more effective, than if we just remove those operations from computation, where the imaginary part of input data is zero.

Normally used and efficient FFT algorithm is radix-2 FFT, where the original and intermediate FFT's will recursively be divided into two new and half size FFT's such, that the one contains all the even indexed points and the other the odd indexed points. If we divide the odd indedex FFT of length N/2 again into two N/4 length FFT's, we obtain the following radix-2/4 (or mixed radix) decimation in time recursion formula:

$$X(k)=\sum_{r=0}^{N/2-1} x(2r)W_{N/2}^{rk}+W_N^{k}\sum_{r=0}^{N/4-1} x(4r+1)W_{N/4}^{rk}+W_N^{3k}\sum_{r=0}^{N/4-1} x(4r+3)W_{N/4}^{rk} \qquad (8)$$

$k = 0, 1, \dots, N-1$

Also in this expansion formula the conjugate symmetry property for real input data is applicable. This allows to reduce computation as in case of radix-2 algorithm. The necessary computation steps are:

$(1/2)MN-(3/2)N+2$	multiplications and
$(3/2)MN-(5/2)N+4$	additions,

where $N=2^M$ .If N=512 and M=9, then radix 2 FFT and radix 2/4 FFT require the following numbers of computation steps:

	Radix 2 FFT	Radix 2/4 FFT
Additions	6278	5636
Multiplications	2180	1538

Because in one dimensional signal processing the input data is often real, we can use successfully the radix-2/4 FFT algorithm shortly described above. If we additionally use Walker's algorithm or precomputed bit reversed index vector in bit reversing, we have a very efficient FFT algorithm in hands.

In practice we often need also the inverse Fourier Transform (IFFT). According to the radix-2/4 expansion method we can develope similarly a very efficient IFFT (/11/Sorensen). Instead of decimation in time principle it is better for us to use decimation in frequency, because then we can reduce more easily all the unnecessary operations.

Two dimensional FFT is computable in mainly three different way. The conventional way is to compute for N*N size input matrix N FFT's row wise and after that N FFT's columnwise. Additionally there is needed to transpose the data matrix. One such method is described in reference (/16/Eklundh). In this computation we can naturally use optimized one dimensional FFT algorithm. One disadvantage of this method is, that we need in addition to the real input FFT version an imaginary input FFT algorithm, because the data will be transformed into complex form during computation.

The second way is to use for 2-D FFT radix-2 or radix-2/4 expansions in two dimensions at the same time. One such method is described in reference (/10/Mou). The program code becomes very long, but we need only this one program. By this method the amount of computation steps is lower, than by using highly optimized 1-D FFT's in row and columnwise computation. In two dimensional case it is also possible to reduce operations, when the input data is real valued.

The third way to tackle the computation is to use parallel processors and parallelized FFT algorithms, like in reference (/9/Gertner).

2.2 Fast Hartley Transform (FHT)

Fast Hartley Transform is a real transform. It means, that both input and output are real valued. The inverse transform of FHT will be carried out with the same algorithm as the forward transform. This is an advantage, if there is a limited memory space to use. The FHT is defined as follows:

$$H(k) = \sum_{n=0}^{N-1} x(n)\mathrm{cas}(2\pi kn/N) \qquad 0<=k<=N-1 \qquad (9)$$

where $\mathrm{cas}(x) = \cos(x) + \sin(x)$.

Using FHT we can compute the same things than using FFT, because these two transforms are mathematically closely related with each other. We can use the same means than by FFT to make the FHT algorithm more efficient. There are only small differences in computation steps needed.

2.3 Autocorrelation and power spectrum

Autocorrelation function is a versatile tool in image and signal analysis. We can compute the ACF via convolution in frequency domain and then transforming back again into time or spatial domain, using FFT or FHT. Using convolution theorem in computations is relatively the more advantageous the longer the data vector is. When computing the ACF piece by piece and summing the results componentwise, we can compute shorter FFT's. At the end we take an inverse transform only once. Two dimensional ACF can be computed for example via 2-D FFT.

Power spectrum, whether it is one or two dimensional, can be computed for example by taking FFT for ACF, by definition. Care have to be taken, especially in case of two dimensional spectrum, that there is enough data to obtain stable enough estimate. In the figure 2. (/18/Bäckström) a one dimensional power spectrum computed from a transilluminated paper sample image is presented. In the spectrum there are some clearly visible peaks caused by wire or felt marking.

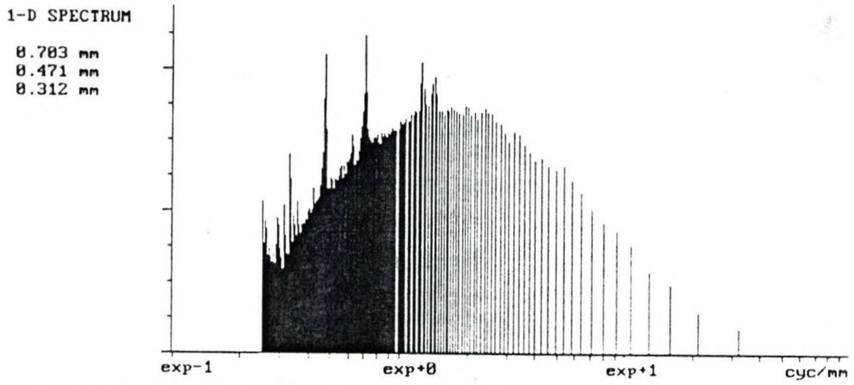


Figure 2. One dimensional power spectrum.

2.4 Filters

We can use for example a median filter to remove random noise from data. The median filter may be one or two dimensional. It removes efficiently random noise peaks so, that the natural signal progression will be maintained. A very efficient median filter algorithm can be based on the double heap data structure and appropriate primitive functions. Inserting a new measurement into double heap causes an amount of computation steps proportional to O(nlog(n)). The double heap maintains the median value in the common root of two heap parts. This algorithm is described in the reference (/17/Astola).This algorithm is not relatively as efficient for two dimensional median filter. Two dimensional filter can be implemented by mainatining the histogram for filtering window. When moving right, we need to subtract the values of the left column and add the values of the right column. When moving down, just subtract upper row and add lower row. After that we can adjust slightly the preceding median value moving in histogram. This method is described in the reference (/8/Huang).

The traditional Wiener filter is widely used in signal processing. The filter coefficients will be estimated from data by using LMS criterium. Wiener filter is well suited for stationary processes. If the process is not completely stationary or the process is not well known, we can use the adaptive version of Wiener filter. There the filter coefficients are determined by approaching step by step the optimum values in the sense of LMS. This adaptive version allows us to monitor changes in process behaviour, and thus also in output variables. We can use as monitoring time series either the residual RMS error or a monitoring variable combined from filter coefficients or both of them. Of course this kind of monitoring does not tell the reason for changes.

One step further in filtering/monitoring is to connect the output estimates and the process state estimation by Kalman algorithm. This is well argumented, because the quality parameters are dependent of paper machine state.

2.5 KALMAN ALGORITHM

There is an elegant algorithm called Kalman algorithm for estimation of dynamic process state and process output (/7/Haykin). This method is applicable to many different purposes as well, and they are: filtering, prediction and smoothing. The Kalman algorithm contains separate models for the process and the measurement.

The process model $x(n+1) = \Phi(n+1, n)x(n) + v(n)$ (10)

The measurement model $y(n) = C(n)x(n) + w(n)$ (11)

n = discrete time index
x = state vector of the process
Φ = state transition matrix of the process
v = process error vector
C = measurement matrix
w = measurement error vector

We assume, that the error vectors v and w are white noise with zero mean . The measurement error and process error are statistically independent. The state transition matrix Φ and the measurement matrix C are both assumed known. We are also given the observed data consisting of the vectors y(1), y(2),...,y(n).

Using the LMS (Least Mean Squares) method optimizing the estimates we obtain the model for filtering by Kalman algorithm as follows:

$$\Sigma(n) = C(n)K(n, n-1)C^{T}(n) + R(n) \qquad (12)$$
$$H(n) = K(n, n-1)C^{T}(n)\Sigma^{-1}(n)$$
$$\alpha(n) = y(n) - C(n)\Phi(n, n-1)X(n-1)$$
$$X(n|Yn) = \Phi(n, n-1)X(n-1) + H(n)\alpha(n)$$
$$K(n) = K(n, n-1) - H(n)C(n)K(n, n-1)$$
$$K(n+1, n) = \Phi(n+1, n)K(n)\Phi^{T}(n+1, n) + Q(n)$$

Σ = Correlation matrix of the innovations process α
R = Correlation matrix of the measurement error
C = Measurement matrix
H = Filter gain matrix
K = Predicted state-error correlation matrix
α = Innovations process
y = Measurement vector
X = Predicted state vector
Φ = State transition matrix

The time complexity of Kalman algorithm is $O(N^2)$, because the size of multiplier matrices is N*N, and they will be updated during every iteration. That's why the algorithm needs heavy computations in case of large N. The so called Fast Kalman Algorithm exists, where the computation steps needed are O(N). We refer in this case to (/13/Shet) and (/14/Shet).

Because the filter based on Kalman algorithm is applicable in dynamic systems, it is in many real situations more practical than the Wiener filter, which presumes a stationary process. This happens because the Kalman algorithm determines first a state estimate for dynamic system. After that , taken into account the state transition, the modell parameters will be updated so that again in the next step the optimal output estimates will be achieved.

The Kalman algorithm needs the current state vector for computing the succeeding state. The current state vector contains or represents all the state information, no past state values are needed. The state vector contains M state values. Additionally it is needed to maintain from measurement data a moving data window of size N. Thus the Kalman algorithm is well suited to implement in the signal processors and consequently in different sensors, where only a limited memory space is often available.

REFERENCES

/1/ H.F. Rance (Editor): Handbook of Paper Science: 2 The Structure and Physical Properties of Paper. Elsevier 1982.

/2/ A.V. Oppenheim and R.W. Schafer: Digital Signal Processing. Prentice-Hall International Editions 1975.

/3/ Jeffrey J. Rodriguez: An Improved FFT Digit-Reversal Algorithm. IEEE Transactions of ASSP, Vol. 37. No. 8, August 1989.

/4/ David M. W. Evans: An Improved Digit-Reversal Permutation Algorithm for the Fast Fourier and Hartley Transforms. IEEE Transactions of ASSP, Vol. 35. No. 8, August 1987.

/5/ James S. Walker: A New Bit Reversal Algorithm. IEEE Transactions of ASSP, Vol. 38. No. 8, August 1990.

/6/ Pierre Duhamel: A Connection Between Bit Reversal and Matrix Transposition: Hardware and Software Consequences. IEEE Transactions on ASSP, Vol. 38. No. 11, August 1990.

/7/ Simon Haykin: Modern Filters. MacMillan Publishing Company, 1989

/8/ T.S.Huang, G.J. Yang and G.Y. Tang: A Fast Two-Dimensional Median Filtering Algorithm. IEEE Transactions on ASSP, Vol. 27, No. 1, February 1979.

/9/ I. Gertner and M. Rofheart: A Parallel Algorithm for 2-D DFT Computation with No Interprocessor Communication. IEEE Transactions on Parallel and Distributed Systems, Vol. 1, No. 3, July 1990.

/10/ Z. Mou and P. Duhamel: In-Place Butterfly-Style FFT of 2-D Real Sequences. IEEE Transactions of ASSP, Vol. 36, No. 10, October 1988.

/11/ H.V. Sorensen, D.L. Jones, M.T. Heideman and C.S. Burrus: Real-Valued Fast Fourier Transform Algorithms. IEEE Transactions on ASSP, Vol. 35, N0. 6, June 1987.

/12/ O. Buneman: In-Situ Bit-Reversed Ordering for Hartley Transforms. IEEE Transactions on ASSP, Vol. 37, No. 4, April 1989.

/13/ K.C. Shet and B.V. Rao: An overview of the Kalman algorithm. Int. J. Electronics, 1985, Vol. 59, No. 6.

/14/ K.C. Shet and B.V. Rao: Modified Fast Kalman Algorithm. IEEE Transactions on ASSP, Vol. 35, No. 7, July 1987.

/15/ R. Conners and C. Harlow: A Theoretical Comparison of Texture Algorithms. IEEE Transactions on PAMI, Vol. 2, No. 3, May 1980.

/16/ J.O. Eklundh: A Fast Computer Method for Matrix Transposing. IEEE Transactions on Comput., Vol. 21, pp. 801-803, July 1972.

/17/ J.T. Astola and T.G. Campbell: On Computation of the Running Median. IEEE Transactions on ASSP, Vol. 37, No. 4, April 1989.

/18/ C. Bäckström, U. Pulkkinen and T. Mansten: The Measurement of the Structure of Paper Surface. Project Report, Technical Research Centre of Finland, 1990. In Finnish.

HARDWARE STRUCTURE OF ARGUS IMAGE PROCESSING WORKSTATION.

Gy. Bangó, M. Gárdos, J. Miskolczi, I. Rényi, I. Szabó.

Central Research Institute for Physics
P.O.Box 49, Budapest 1525, Hungary

A large number of the picture processing algorithms demand intensive computation. In real applications, response time is a very important feature and many applications require real-time response time. This demand leads to application of special architectures (different and well-chosen architectures for different algorithms) and application of high speed (algorithm specific) LSI circuits: including application of digital signal processors (DSP). The relatively powerful and inexpensive professional personal computers extended by these high speed LSI circuits give new possibilities in the field of picture processing.

The concepts behind the design of the immage processing workstation (ARGUS) were the following:

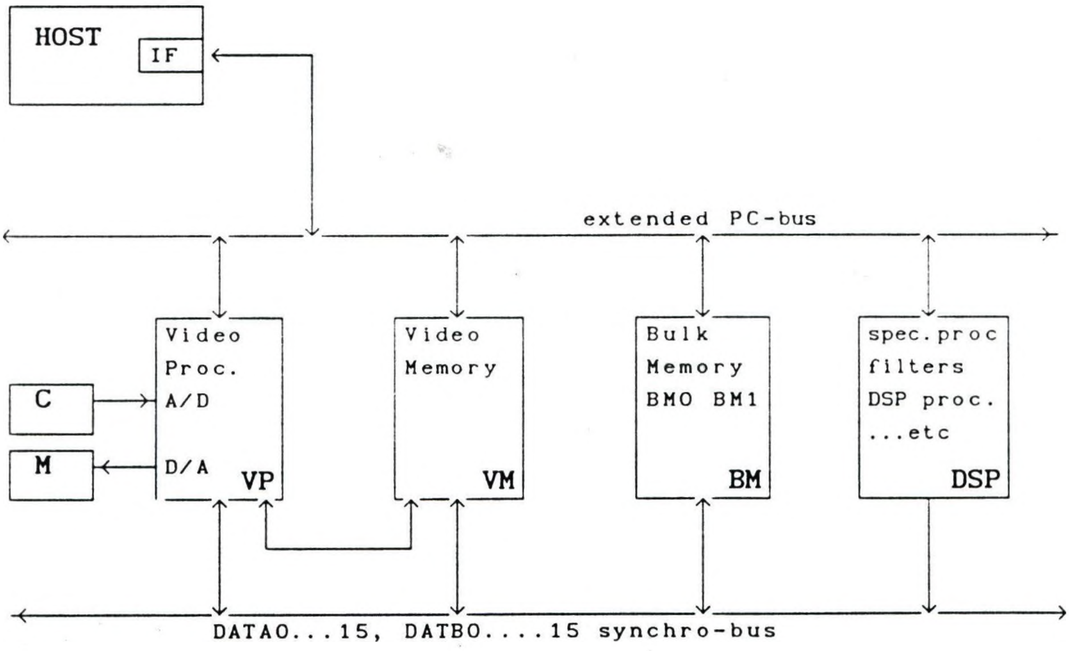
- Cost effective, small and desktop size equipment for different applications.
- Easy adaptivity to specific application requirements by open ended architecture.
- The use of local parallelism and pipe-line method among the processor modules.
- The use of up-to-date LSI components or processors and improving efficiency by using microprogramming.
- The use of a PC (for example IBM PC/AT) as HOST computer to control the system.
- Convenient menu driven and command language user interface.

The architecture of the ARGUS is shown in **Fig. 1.** The modules are connected to two bus systems:

- **Extended bus system of HOST computer** (in this variation it is a extended IBM PC/AT-bus).

  Both programmed and DMA data transfers may take place via this bus for controlling the ARGUS modules (e.g. downloading microprograms, look-up tables, reading /writing image memories by the PC etc.). One simple interface module (**Fig. 2**) is for transferring the internal PC-bus to external PC-bus. This module encloses receivers and drivers and a simple address mapper circuit. 64Kbytes of memory are reserved for addressing the ARGUS internal registers and some control memories, other 64Kbytes of memory are reserved for addressing the ARGUS picture memories. Addressing the (16Mbytes) picture memory is carried out by a mapper circuit (on the interface board) through the 64Kbytes memory windows.

- **Synchro-bus system** contains four independent 8-bit wide data bus and each of them has two control signals. Two of them can be connected to form a 16-bit wide data bus. The buses do not carry address information, the transfers (8-bits, 16-bits, 24-bits or 32-bits long words) are carried out in every 100 nanosec synchronously with the master clock cycle.



C: camera, **M**: monitor

Fig. 1. IPW block diagram.

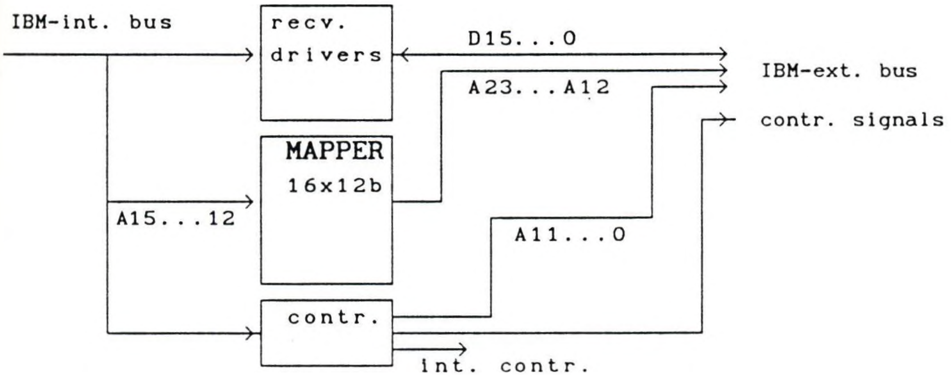


Fig. 2. HOST interface module.

The ARGUS includes two modules (VP, VM) in base configuration and four modules (VP, VM, BM, DSP) in medium sized configuration. The maximum number of modules may be eight in one ARGUS cabinet.

Video Processor (VP) module.

The VP (**Fig. 3.**) contains three analog/digital (A/D) converters for transferring the analog signal of camera or video recorder to digital signal and three digital/analog (D/A) converters for transferring digital signals to analog (which is connected to the monitor input). The contrast modification (and picture transformations) is performed by three look-up tables (LUT). The module has a direct (3x8-bit wide) data connection to video memory (VM) module, therefore displaying pictures from the VM module will not disturb neither communication on the synchro-bus nor image processing calculations. This module contains the synchron pulse generators for monitor and the clock generator for the whole ARGUS system. These synchron generators can be synchronized from external equipment, too.

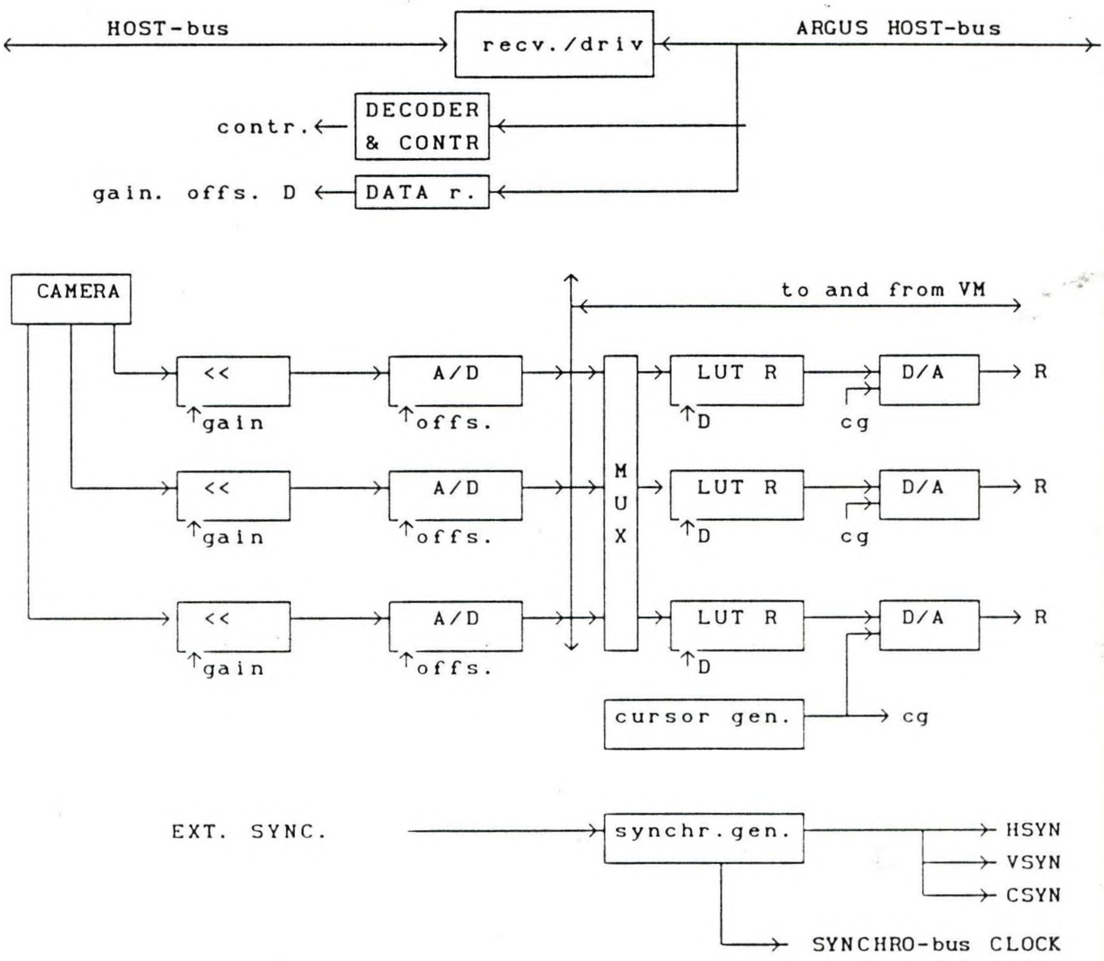


Fig. 3. Video processor module (**VP**).

Video memory (VM) module.

The VM (**Fig. 4.**) contains three RAM modules to store pictures (one picture plane contains 512x512 pixels, one pixel is 8bits). The memory modules are built up of dual-port video RAM modules for video rate acquisition (from the VP) and display (through the VP). The module is controlled by video processor (TEXAS type: TMS34061) for displaying pictures (zoom, roam etc.) and for different data transfers through the synchro-bus. The module is controlled by the HOST through the extended PC-bus. The HOST has random excess to the three memory modules through the HOST extended-bus system. Synchronous data transfers can be carried out on the synchro-bus among the VM and BM and other modules.

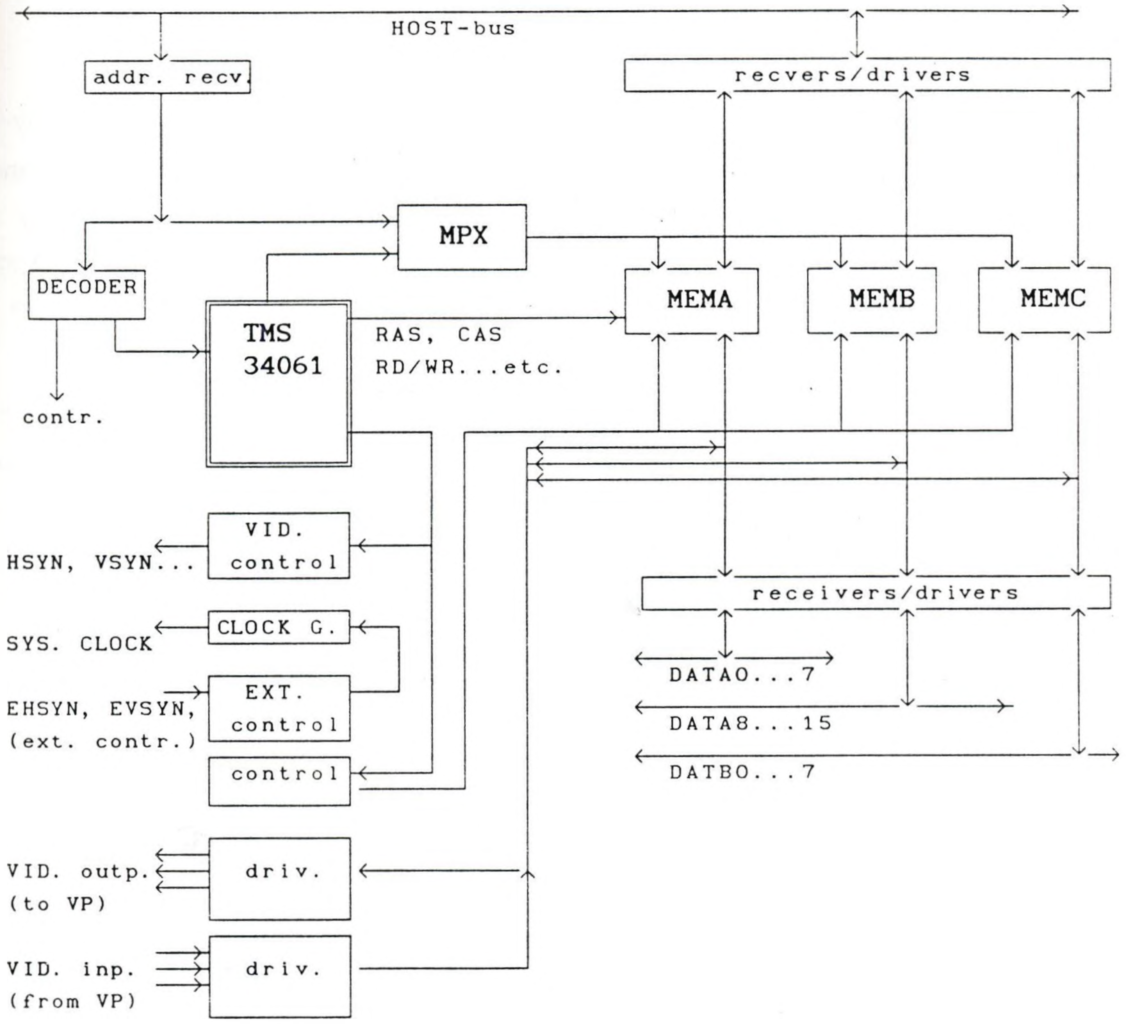


Fig. 4. Video memory **(VM)** module.

Bulk Memory (BM) module.

The BM (**Fig.** 5) uses high capacity (1Mbytes) DRAM chips to store eight 16-bits (or sixteen 8-bits) image planes. These modules also are connected to the extended PC-bus (for random or DMA data transfer) and to the synchro-bus system. Data transfer rate may be 8-, or 16-, or 24- or 32-bit wide word in each 100nanosec. The memory blocks are controlled by DSP1410 address processor (**Fig.** 6.) and the whole BM module is controlled by microprogram and by microprogram address controller (type: Am2910). Various microprograms for various applications are downloaded to the 58-bit long word writable control store (WCS) memory. The basic functions of these microprograms are:

- execute random access (reading/writing data) to/from HOST extended bus system
- reading and writing data to/from synchro-bus system.
  The transfer can be carried out line-by-line or column-by-column (between BM and VM, BM and BM or BM and DSP). The start and stop address of the transfer, the transfer mode and the memory module numbers are stored in the internal registers of the ADSP-1410 address generator.

The ADSP-1410 address generator is a fast, flexible circuit optimized for digital signal processors. It contains a 16-bit ALU, a comparator and 16-bit register files. The registers are organized into four files: sixteen address (R) registers, six offset (B) registers, four compare (C) registers and four initialization (I) registers. It has a 16-bit address (Y) port for outputting addresses and a 16-bit data (D) port for I/O between internal and external registers.

The ADSP-1410's 10-bit microcode instructions include commands (increment or decrement address registers (R), add or subtract address registers (R) and offset registers (B)) for looping operations, internal data transfer, register read/write and logical/shift operations.

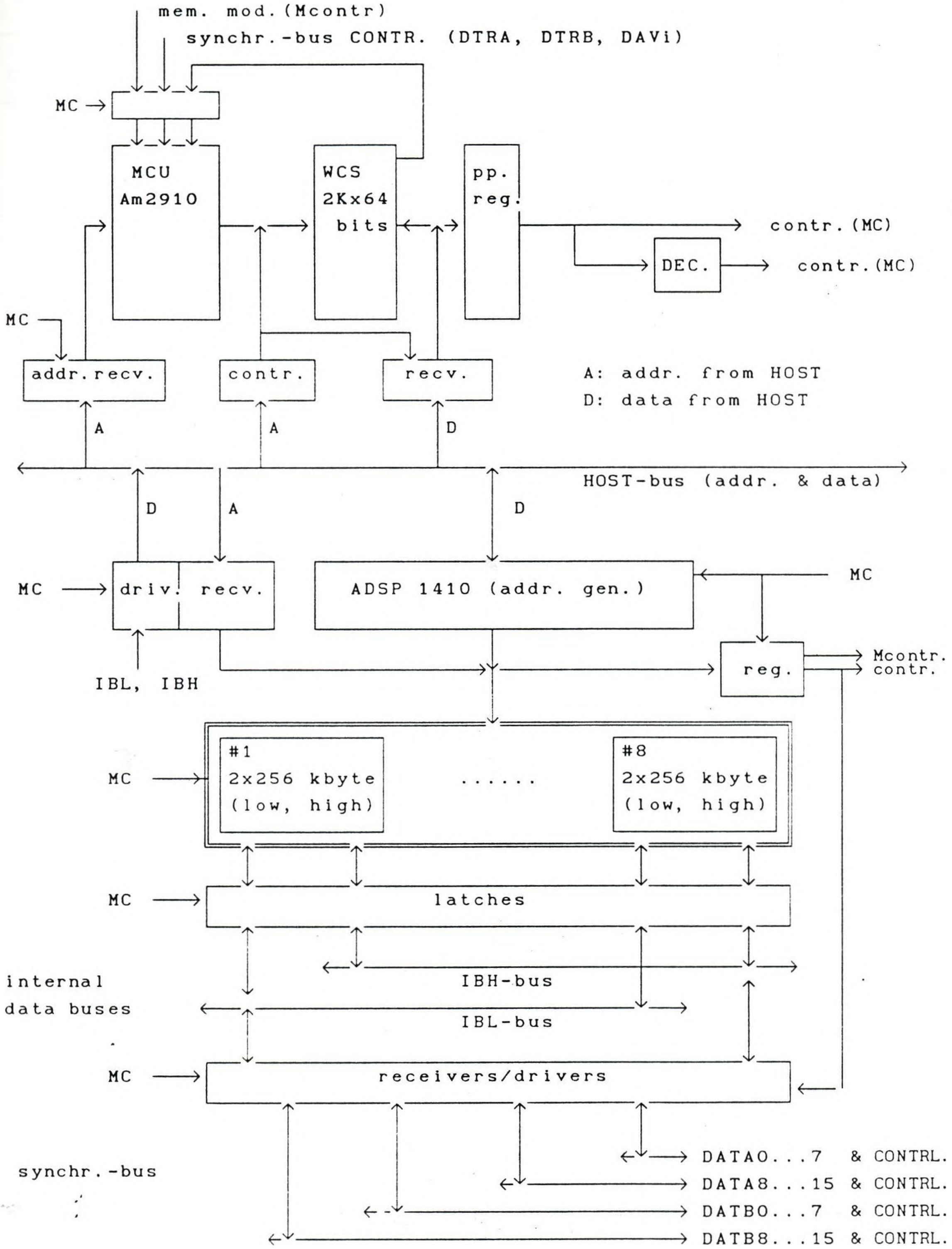


Fig. 5. Bulk Memory Module (BM).

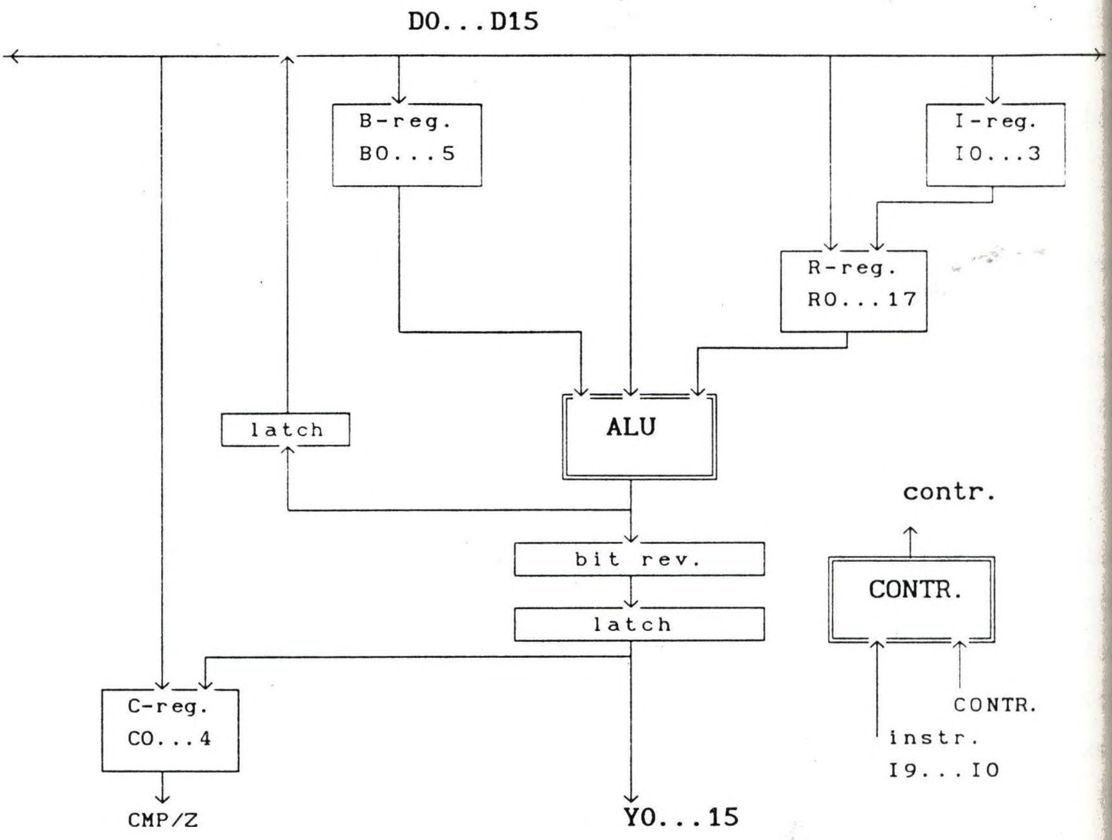


Fig. 6. Address processor (ADSP1410)

Digital Signal Processor (DSP) module.

The DSP module (**Fig . 7.**) contains a simple fix-point DSP processor (Philips: PCB-5011) for fix-point picture processing calculations. The module has two interfaces:

- the first is the HOST interface:
  - for write and read the writable control store (WCS)
  - and for controlling the module (read and write the module status register)
- the second is the synchronous interface. The module is connected through the DATA15...0 or DATB15...0 synchro-bus to the VM or BM modules.

The module has four 4kx16-bits RAM memories (A0, A1, B0, B1) as working buffer memory. The correct organization of these memories will increase the DSP calculation efficiency: in the meantime, when the DSP uses one memory module for calculation, the other modules may be switched to reading or writing from or to the synchro-bus. Addressing these buses can be carried out by the DSP processor or counters during the synchronous data transfer.

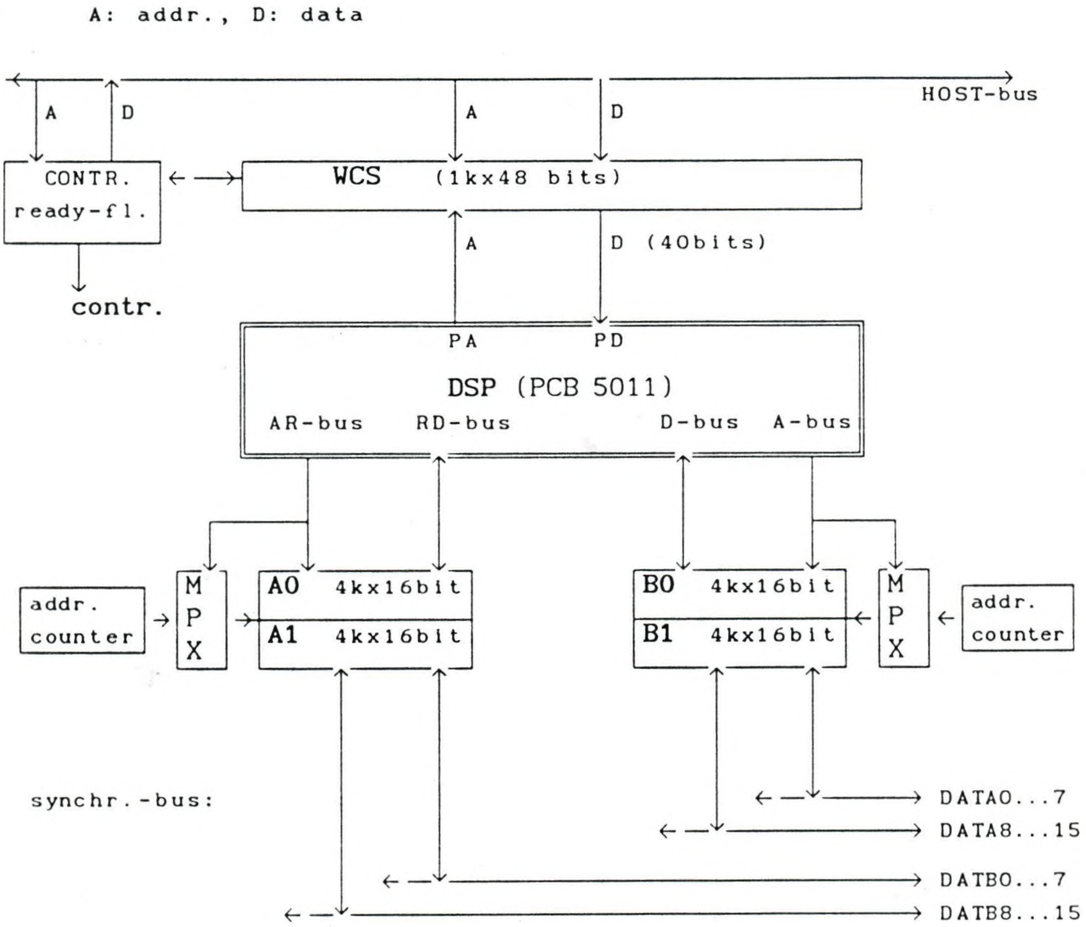


Fig. 7. Digital Signal Processor module (**DSP**).

The PCB-5011 processor (**Fig. 8.**) has a very clear parallel (Harvard) architecture with independent program and data buses. It has 10-bit wide address-bus for addressing the 40-bit wide external program memory (WCS in the DSP-module), and two address buses (ARR8...0 and A15...12, A11...0 for addressing external memories) and two 16-bits wide data buses (RD15...0, and D15...0). It has three independent addressing units (ACUR, ACUA and ACUB) for addressing the two external and two internal data memories. The ACUR and ACUA is used for addressing the A0, A1 and B0, B1 memories of the DSP module. The two internal 16-bit wide data buses (X and Y) are for data transfers among memories and arithmetic units. The arithmetic unit includes a 16x16 multiplier, ALU, barel-shifter and a register file. The processor uses a pipelining mechanism to increase the processor performance. The pipeline depth depends on the mode of the operation: the basic instructions have pipeline depth of 0 or 1 mode depending on the instruction. Compound operations (a sequence of basic instructions) may have deeper pipelines. This feature will increase programming difficulties.

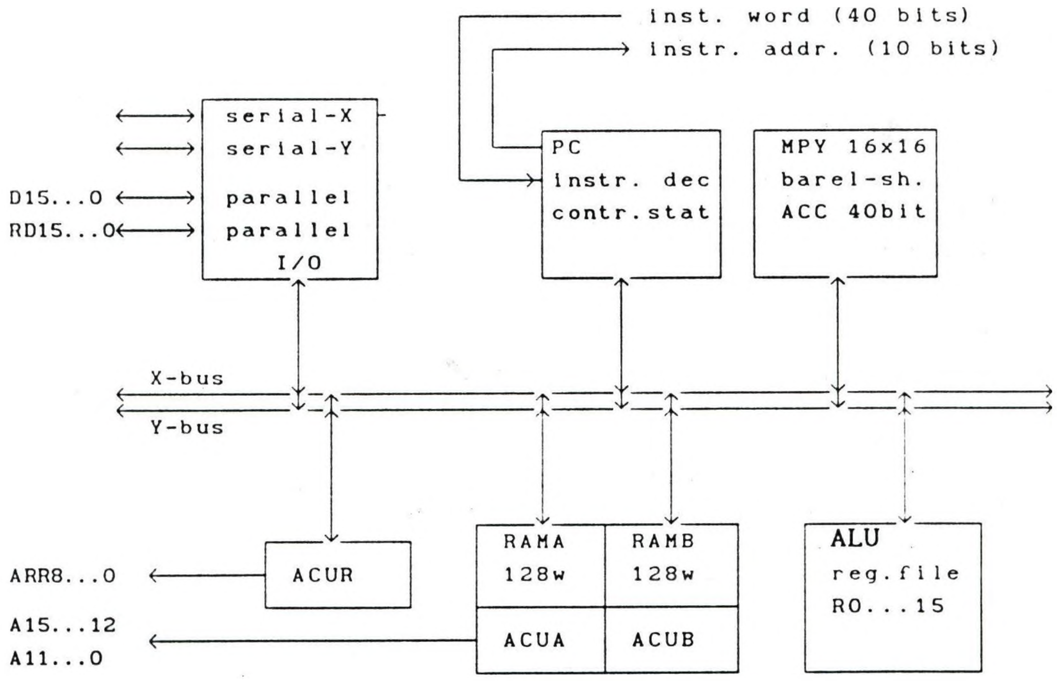


Fig. 8. Digital Signal Processor (PCB-5011).

The VP, VM, BM and DSP modules are the basic modules of the ARGUS. One feature of ARGUS is the open-ended architecture, therefore it is easy to extend by other modules for other specific picture processing application requirements.

THE ARGUS IMAGE PROCESSING SOFTWARE

Gy. Ambrózy, G. Bóna, I. Cseke,
Z. Fazekas, J. Pongrácz, S. Zöld
Central Research Institute for Physics,
P.O.Box 49, Budapest 1525, Hungary

1. Introduction

ARGUS is a PC-based, general purpose image processing workstation. This paper gives a short overview of ARGUS' software architecture. The software structure of ARGUS is hierarchical and is composed of three layers:

- the user interface layer
- the program library layer
- the function library layer.

All of these layers are implemented in Microsoft C language under MS DOS 3.3 operating system.

2. User Interface Layer

The user interface layer consists of an MS DOS based menu system supporting commonly used menu techniques such as menu bars, pull-down menu windows, help facilities, etc. Instead of typing long command lines, the user can easily access the provisions of ARGUS. The menu system of ARGUS is fairly effective for both skilled and inexperienced users. This is a generative menu system being flexible enough to allow easy creation and modification of menus. It is necessary because the software architecture of ARGUS is

an open-ended one, it contains a growing set of low-level and high-level programs and utilities.
The menu system consists of two parts:

- an application-independent menu controller
- an application-dependent menu database.

In this way (e.g.: in order to add a new modul into the menu) only the menu database must be updated. The menu database consists of some binary files containing all information about ARGUS (or other possible systems, respectively). These files are created by the menu generator, which is a simple compiler that translates a menu description file (a structured text-file) into the menu database. The menu designer can easily construct this description file with the help of a simple definition language. The syntax of the definition language consists of simple data structures, keywords and separating characters. The menu generator realizes an effective syntactical and semantical error detection, thus one can avoid most of the definition errors during the coding phase.

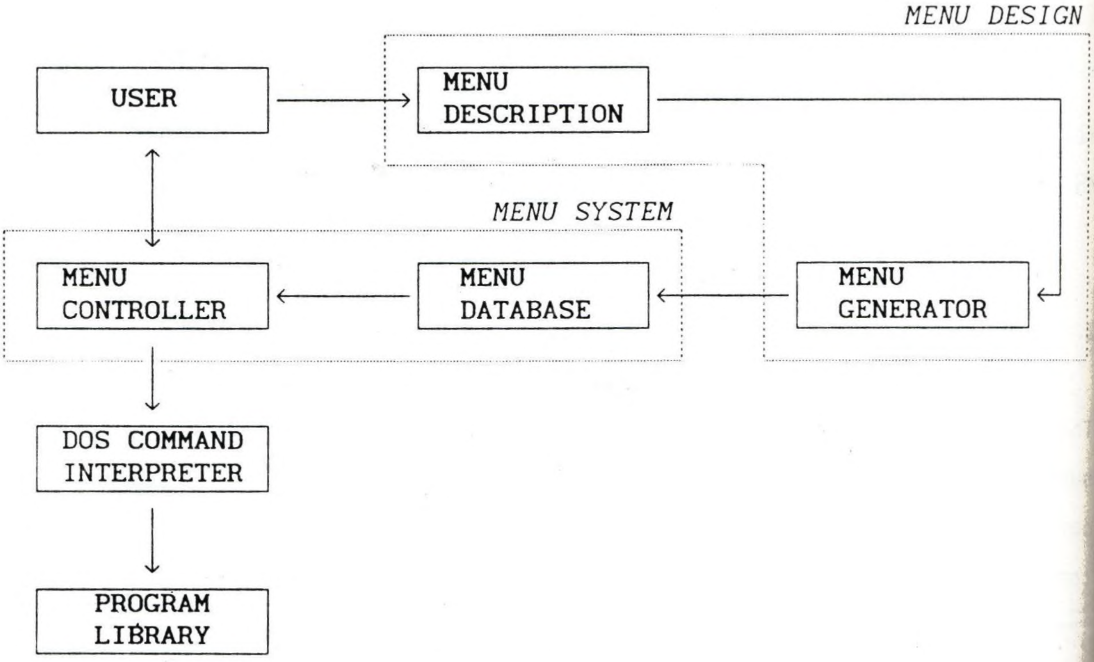


Figure 1.
The menu-driven user interface of ARGUS

The menu controller makes different provisions possible. These facilities are the following: tree-structured menu system, menu bars, pull-down menus, help texts about moduls and about the menu, pointing input devices (mouse), setting attributes of menus, direct program execution, activating the DOS command interpreter, parameter collection through dialogue boxes, setting preselected default values and history generating mechanism. With the help of the history generating mechanism the user can store long command line sequences in a history (batch) file and afterwards he can modify and execute its contents. Another important facility of the menu controller is the effective dialogue boxes. Sometimes quite a great number of parameters do belong to image processing commands. The user can easily set these parameter values using dialogue boxes. A dialogue box may contain different fields, such as text-fields (wherein the parameters must be typed in), list-fields (wherein the parameters are chosen from a list), option-fields (wherein the options are selected) and different command-fields. Figure 1 shows the simple arrangement of the menu-driven user interface layer in ARGUS.

3. Program Library Layer

The components of this layer make up the ever growing set of image processing and computer vision programs and different system utilities. To date only the most commonly used algorithms and methods are realized. Some time-consuming algorithms are sped up using the general-purpose digital signal processor (DSP). In this way one can obtain at least one order of magnitude increase in speed for many algorithms (in comparison with the software solutions). We mention now some of the image processing and image analysis programs realized to date:

- gray scale transformations (contrast stretching, binarizing, trinarizing and other monotonic and non-monotonic transformations)

- histogram modification (histogram equalization, histogram hyperbolization, transformation on the base of an ordinary histogram)
- gray level correction (1-dimensional and 2-dimensional shading correction)
- convolution (spatial) filtering (edge and line enhancement, sharpening, smoothing, user-defined masks)
- nonlinear local filtering (median filtering, min/max filtering, out-of-pixel filtering)
- orthogonal transformations (FFT and inverse FFT, Walsh-Hadamard transformation, power spectrum)
- frequency filtering (ideal-, trapezoidal-, Butterworth-, exponential lowpass and highpass filtering)
- image segmentation (global and local thresholding, automatic threshold selection)
- geometrical transformations (expansion, reduction, shifting, rotation by bilinear interpolation)
- gray level statistics (histogram features, scattegram, cooccurence matrix)
- pixel operations (addition, subtraction, multiplication, division, logical operations)
- binary image operations (shrinking, expansion, skeletonizing, labelling, shape description)
- classification (k-nearest neighbour estimation, isodata clustering)

The other components of the program library layer provide for the system usage. Examples of the system utilities are the following:

- image file management (loading from disk, saving on disk, copying among memory blocks, test images and noise generation, compression)
- image displaying (displaying images and subimages, zoom and shift operations, colouring, displaying of pixel data)
- graphical services (overscripts, simple drawing, graphical sheet loading and saving, contour filling)

- image input (scanning and grabbing image or image sequences from camera, line scanner and video-player)
- others (test, initiating, demonstration)

4. Function Library Layer

Finally, the lowest layer is the function library. This layer can be regarded as an extension of the Microsoft C library, its components can be called from any C programs in form of an ordinary C routine. The function library layer includes a collection of general routines for keyboard, mouse and display management, interactive subimage selection, image processing primitives, etc. It also includes a collection of routines handling the different ARGUS hardware moduls. They provide for data and image transferring on the synchronous bus, controlling the digital signal processor, look-up tables and cursor management, camera control functions, image memory mapper handling, image input/output operations, graphical functions, etc.

5. Some Examples

In the first step two algorithms, the convolution filtering and fast Fourier and inverse Fourier transformation have been selected for the realization by using DSP, because they are mathematically very intensive algorithms that is they require considerable execution time (see Table 1) and these algorithms very often appear in various image processing applications, and processing them with high speed is essential to obtain high performance in the image processing systems.

The DSP works in a concurrent and autonomous manner, it is independent of the host processor (e.g. a conventional, 20 MHz AT). Using the DSP the algorithms run on two levels: on the micro-program-level the calculation is executed by the microprogrammed DSP and on the C-language-level the calculation and the data transfer are controlled by the host processor. From Table 1 we can see the performance of the convolution filtering and the Fourier transform in the case of the realization by software means (using the mentioned 20 MHz AT) and DSP, respectively. An essential speed increase can be found. For convolution filtering the processing time in Table 1 regard images consisting of 512·512 pixels.

PROCESSING	SOFTWARE REALIZATION (s)	DSP REALIZATION (s)
CONVOLVE by 3·3 MASK	25	1
CONVOLVE by 5·5 MASK	105	2
CONVOLVE by 7·7 MASK	190	3.5
FFT on 128·128 PIXELS	65	1
FFT on 256·256 PIXELS	270	4
FFT on 512·512 PIXELS	-	18
INVERSE FFT on 128·128	70	1
INVERSE FFT on 256·256	285	4
INVERSE FFT on 512·512	-	19

Table 1.
Processing time on ARGUS

Finally, some images demonstrating the ARGUS are appended. On image CONV4 the upper left image is convolved by a vertical edge enhancing mask (L=3), by a horizontal line enhancing mask (L=5) and by a smoothing mask (L=7), respectively (where L is the mask size). On image FFT2 the upper left image is transformed into the Fourier domain and its power spectrum can be seen on the upper right image. Then the image is filtered in the frequency domain by an ideal lowpass filter (where cut-off frequency is 64) and the

Image name: conv4

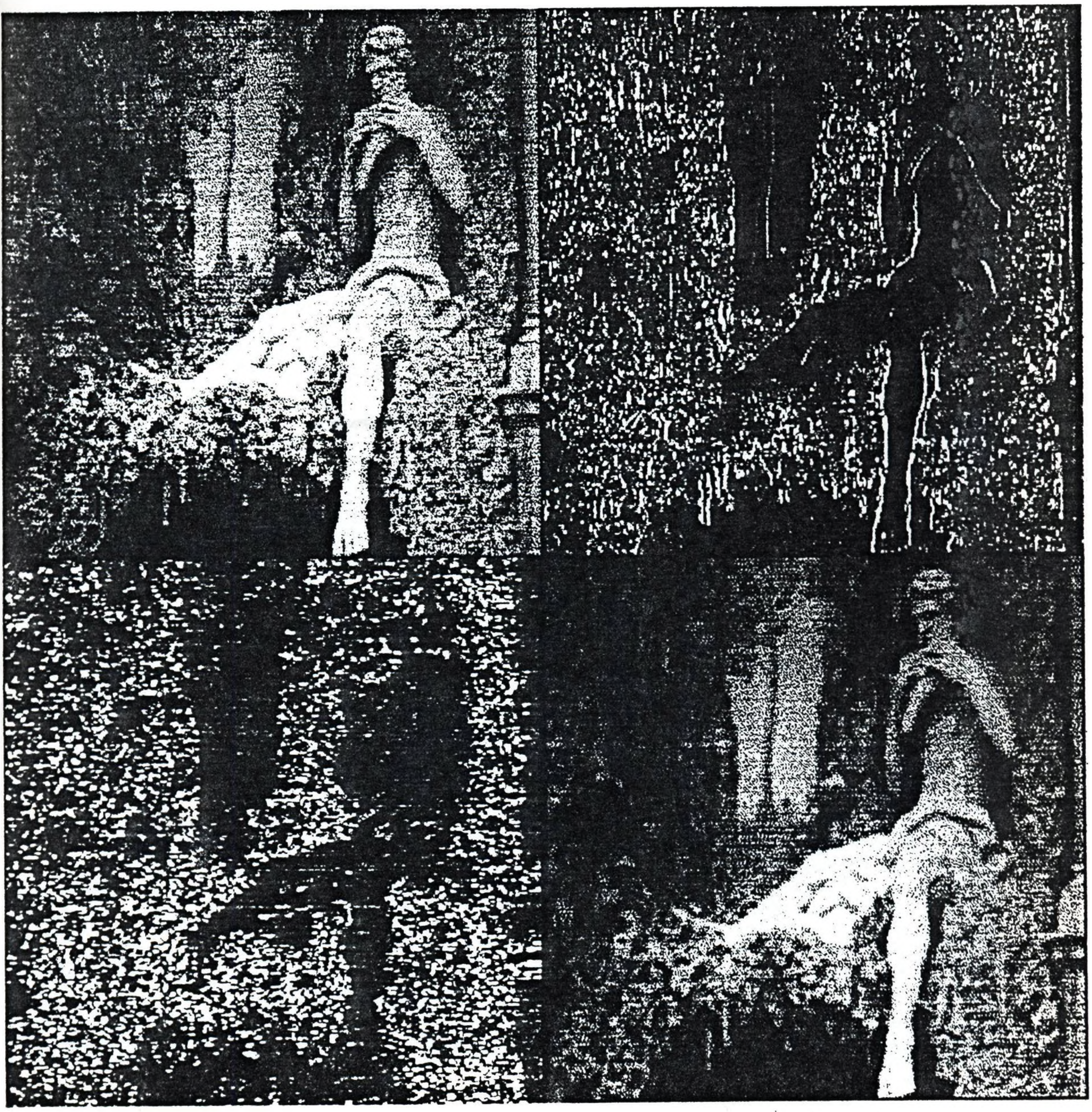
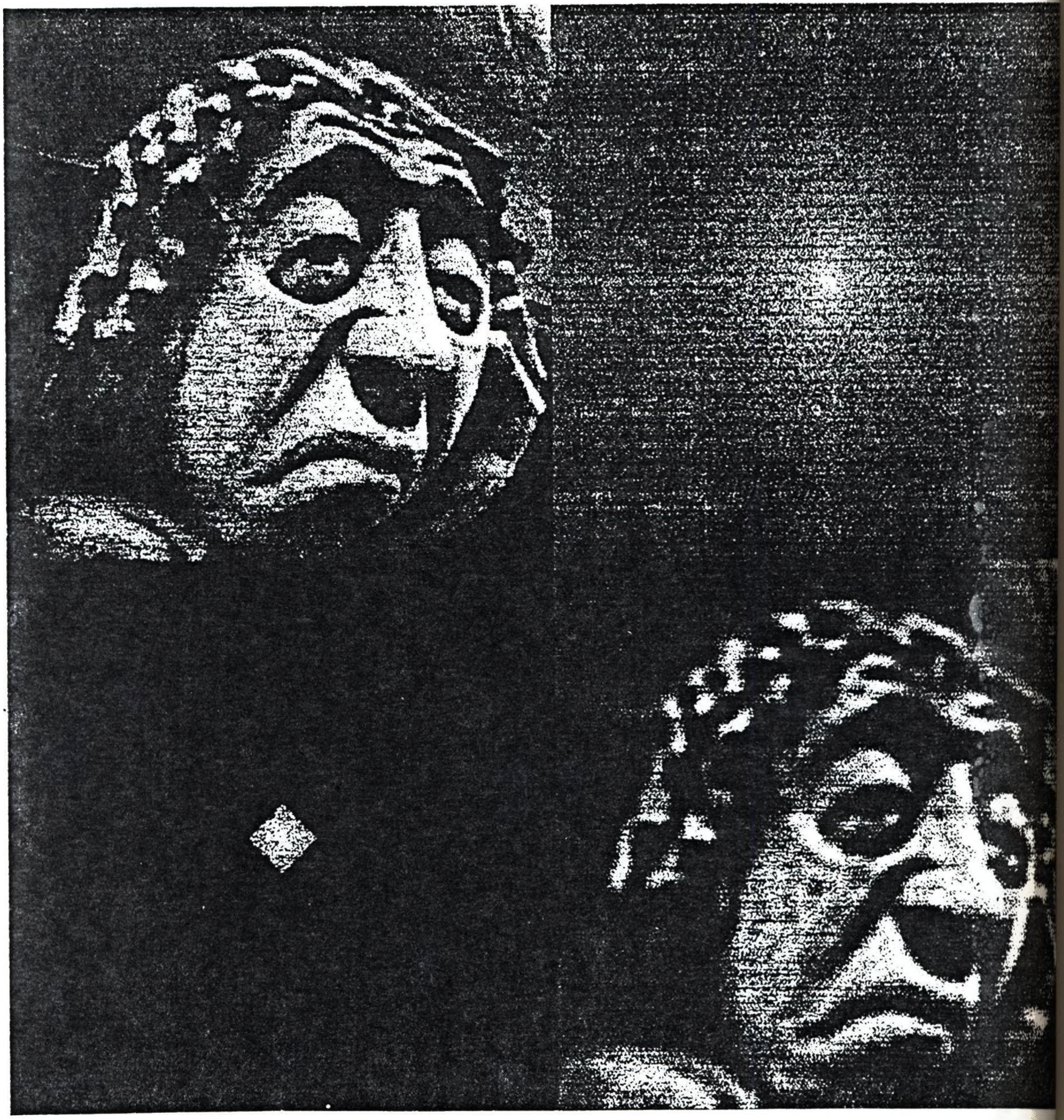


Image name: fft2



power spectrum and the inverse transform of the filtered image are shown.

6. References

Gy. Bangó et al.: IPW: an image processing workstation; Proc. of the 2nd Hungarian Workshop on Image Analysis, Budapest, June 7-9, 1988., MTA SZTAKI Tanulmányok 206., pp. 81-84.

The microSegams image processing system

A.Kuba,Á.Makay,E.Máté,M.Tóth-Abonyi
Kalmár Laboratory
József Attila University of Szeged

1. Introduction

A new image processing system (microSegams) is presented which was created for nuclear medicine data acquisition and evaluation (for a detailed description see e.g. [1]). We give the description of the whole system with emphasis on the image processing techniques used.

2. Some words about nuclear medicine.

A radioactive material, which takes part in physiological processes, is administered into the patient. The emission of this accumulated or transmitted material is detected by gamma camera within one or more time intervals and from one or more directions. After the analog-to-digital conversion the images are recorded on film or stored on magnetic disks. In some cases the acquired pictures themselves are very meaningful for the physician but in many cases complex procedures should be carried out to obtain the necessary information (for a survey see e.g. [2]).

3. Hardware

The CPU is MC68OOO with 1 MByte RAM. It has an extension card for the interface with the gamma camera and 2 MByte extra RAM for image acquisition and quick presentation of pages. The interface is able to collect images in a minimum of 1O msec with 64x64, 128x128, 256x256 matrix sizes, bytes or words. There is a possibility to acquire data only from a part of the field of view (zoom in 255 steps) and also to record ECG signals in parallel with image acquisition There are I/O ports for two cameras. The disk drivers are a 3,5" built-in floppy with 88OK formatted capacity and a 4O Mbyte harddisk. A high-resolution monitor and a colour matrixprinter are attached. The input devices are the keyboard and the mouse.

4. Software

The programs were written mainly in C language. Only few routines were written in assembly. The programs are organized into a three level depth tree-structure and may be entered through a mouse-handled menu system. The five main menu items are: Acquisition, Clinical, Individual, SPECT and System (Fig.1).

4.1. Acquisition

Every acquisition has two steps: first the data about the patient and the acquisition should be entered and after it the images should be acquired. The second step should not immediately follow the first one. It is possible to enter the data in advance. Later on these prepared studies may be acquired at any time.

In every nuclear medicine institute there are some basic types of investigations. The parameters of the acquisition are the same for every study of a given type. This led to the concept of the predefined study. It contains all the acquisition-specific data and only the identifier and the patient-dependent part should be entered during the preparation. But the acquisition-specific part may be overwritten too, if it is necessary.

4.2. Clinical

This part of the menu contains complete clinical programs classified according to the organs to be investigated. These programs perform the most frequent procedures of the everyday practice. Each of them begins with the selection of an image file. If the selected file is suitable (e.g. its type and size are the prescribed ones) then the processing starts. The steps of the processing follow each other automatically till they need some manipulation of the assistant. After the manipulation the procedure is continued till the next waiting state or the completion of its last step. The result of the execution is a picture on the screen, summarizing the most important images, curves and parameters generated. Some programs create new images which are stored for further evaluations.

4.3.Individual

The clinical programs of course do not cover all nuclear medicine methods. The user may want to create his own procedure. To assure this, the system provides more than two hundred macro entries. These macros are the elementary steps of the clinical programs too. So many items cannot be fitted into a menu system in usable way.

The most important ones are listed in this branch of the menu and may be entered directly.

The others may be accessed only via an interpreter called SLan. In the System branch (see: 4.5) of the main menu there is a node To SLan. By activating it at the first time, the interpreter is loaded and an environment is built to accept SLan commands. These commands form a subset of the C programming language, supplemented by the macros. When an instruction is sent to the interpreter it is executed interactively and its effect becomes visible if it has any on the display. The user may return to the menu simply by pushing the menu button on the mouse. By repeated activation of the To SLan node the system changes from menu to interpreter mode immediately.

There exist the necessary tools to archivate a block of instructions even if it was mixed with menu selections. Later these blocks may be executed in batch-like manner or used as a clinical program after compiling and linking.

4.4.SPECT (Single-Photon Emission Computed Tomography)

The functions of SPECT perform special operations on the projections themselves or the sections reconstructed from the projections. The reconstruction is executable if the projections have been acquired from a full or a half-circle rotation. The recontsructed sections have standard positioning (anterior-posterior, left-right, up-down) not depending on the way of the acquisition and the position of the patient. The items of this branch are directly enterable macros, similarly to those of Individual (see:4.3).

4.5. System

These functions are necessary to maintain the system and to create or modify the context of acquisitions, evaluations and presentations. Help text is also available here.

5. Image presentation

The display is divided into three parts. The first, 64 pixels wide area, is a control field. The next 256x256 pixels are used for mimage presentation. The third part is a narrow strip showing the colour scale used.

The images to be presented are stored on 15 pages in the memory. The page is a 256x256 sized word matrix. It is divided into frame areas, where the images should be copied for presentation. One page contains 26 64x64 sized frames or 4 128x128 sized frames or 1 256x256 sized frame. The 128x128 frames may be enlarged to double, the 64x64 frames to double or quadruple size.

The colour of a pixel depends on the value of the corresponding element of the actual page (i.e. the presented page). A minimum and a mayimum value is given for the presentation. The interval between these two values is divided into 32 equal parts and the colours are corresponded to these subintervals. Values under the minimum and above the maximum are presented with the lowest and highest colour, respectively. The minimum and maximum values may be set even with the mouse. Special hardware helps to map the subintervals to the colours, so the results of the changes are immediately visible.

The colours are composed of three components: red, green and blue.The intensity of each component may vary from O to 15. Consequently 4O96 different colours exist, out of which up to 32 form a colour scale.

The final view is covered by an overlay and a text bit map, produced by drawing and text writing operations, respectively. The presentation of these bit maps may beswitched off.

6. Image processing

A complete frame algebra is available to execute on frames. The user is able to select up to 8 areas on frames which ones he is interested in .These areas are called regions of interest or briefly ROIs. The borders of the ROIs can be drawn by the mouse. If the border is ready the system automatically fills the corresponding area inside the border. A ROI algebra is provided to perform operations on ROIs,if it is necessary. In the case of a SPECT study after the recontsruction it is possible to select three-dimensional ROIs, which consist of a sequence of two-dimensional bit matrices.

From the image processing point of view the investigations may be classified into two groups:

- pictures are acquired from on or more directions, the radioactivity is assumed to be the same for every picture
- pictures represent a process, the radioactivity changes at the time.

The heart studies are special ones: for a longer period they fall into the first group, but for the heart-cycle time they fall into the secong.

In the first group the most simple method is to draw some ROIs on different parts of the organ and to compare the detected counts (Fig.2). On the other hand, the reconstruction of SPECT studies is a very complex procedure. The transversal sections are reconstructed first and they are used to generate frontal, sagittal and oblique angle sections. To correct the absorption, the one-step version of Chang's method 3 is implemented. The contour of the body must be drawn by mouse on the transversal sections and then the correction matrices are computed for every section, assuming that the absorption at a gíven point is directly proportional to the distance of that point from the contour of the body at any direction. The sections may be smoothed in three dimension (Fig.3).

In the second group a ten-speed movie is available to trace the process.

Some technics produce a picture, containing counted parameter values. The values of a pixel the result frame is a parameter counted from the same position pixels of the picture-sequence. Such result frames are for example the maximum, Tmax, T 1/2 pictures (Fig. 4) or the amplitude,phase pictures (Fig. 5). The maximum picture is the overcover of the picture-sequence. In the Tmax picture the time of achieving the maximum, in the T 1/2 picture the half-period of the empyting process, beginning from the maximum, are peresented. The latter one is counted after fitting exponential curves. The amplitude and phase pictures are got from fitting cosine curves.

The other family of technics is performed on ROIs. Time activity curves are generated showing the distribution of the activity in time on the ROI areas. Regular curves (constant, linear, exponential, gamma) may be fitted to them on some intervals. The parameter values of the fitted curves are used in describing the state of the investigated organ (Fig. 6).A completes curve algebra is available to perform any manipulation on both the original and the fitted curves.

REFERENCES

[1] A.Kuba,Á.Makay,E.Mâté,L.Csernay: Data-processing system for nuclear medicine images
To appear in International Journal of Imaging Systems and Technology

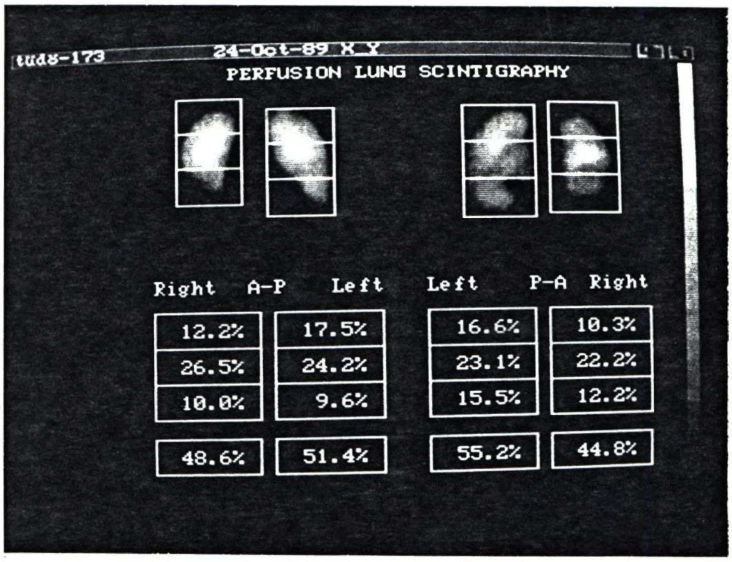
[2] A.Gottschalk,P.B Hoffer, E.J.Potchen: Diagnostic Nuclear Medicine Williams and Wilkins, Baltimore, 1988.

[3] Chang: A method for attenuation correction in radionuclide computed tomography
IEEE Trans. Nucl. Sci. NS-25, 638-643 (1978).

```
Acquisition                 Clinical             Individual  Spect          System
--------------------------- -------------------- ----------- -------------- ----------

Prepare and acquire study   Cardiac studies      Study       Corrections    Archivation
Prepare an individual study Respiratory studies  Information Reconstruction Reparation
Acquire a prepared study    Neurological studies Display     Presentation   Color scales
Construct predefined study  Gastroenter. studies ROI         3-D ROI        Quality control
Delete predefined study     Kidney studies       Curve       Atten. corr.   Patient class
Select Channel              Endocrin studies     Overlay     3-D Smoothing  InterFile
                            User programs        Text                       To SLan
                                                 Page                       Help
                                                 Slice
```

Figure 1.
The menu structure of microSegams without the third level.



```
tudy-173        24-Oct-89 X Y
           PERFUSION LUNG SCINTIGRAPHY

Right    A-P    Left       Left     P-A   Right
 12.2%       17.5%          16.6%         10.3%
 26.5%       24.2%          23.1%         22.2%
 10.0%        9.6%          15.5%         12.2%

 48.6%       51.4%          55.2%         44.8%
```

Figure 2.
The results of a lung study.

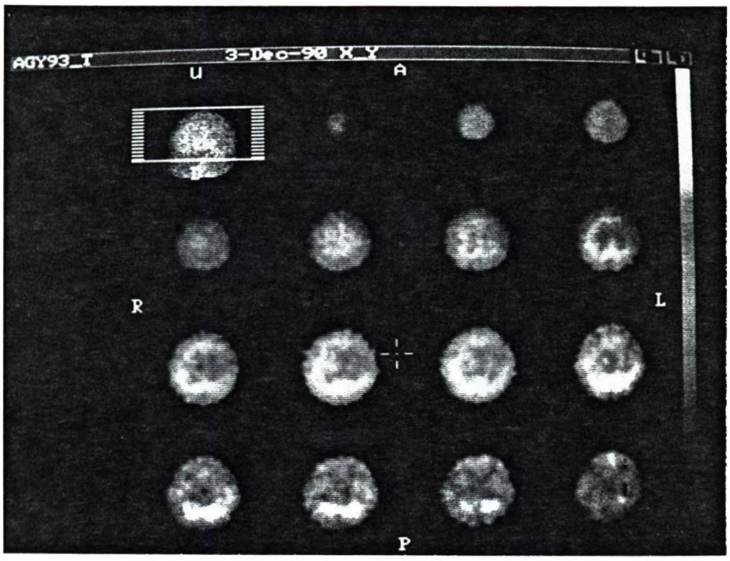


Figure 3.
Presentation of the reconstructed transversal sections of a brain SPECT study.

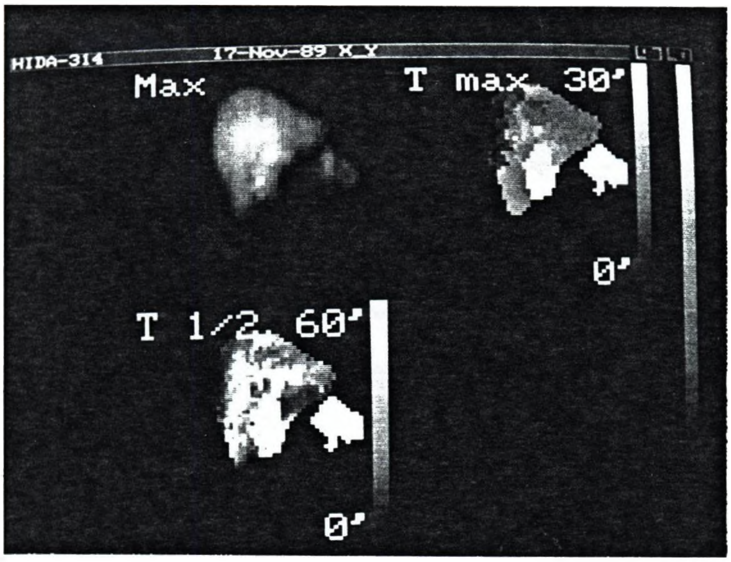


Figure 4.
The maximum, Tmax, T 1/2 pictures of a dynamic liver study.

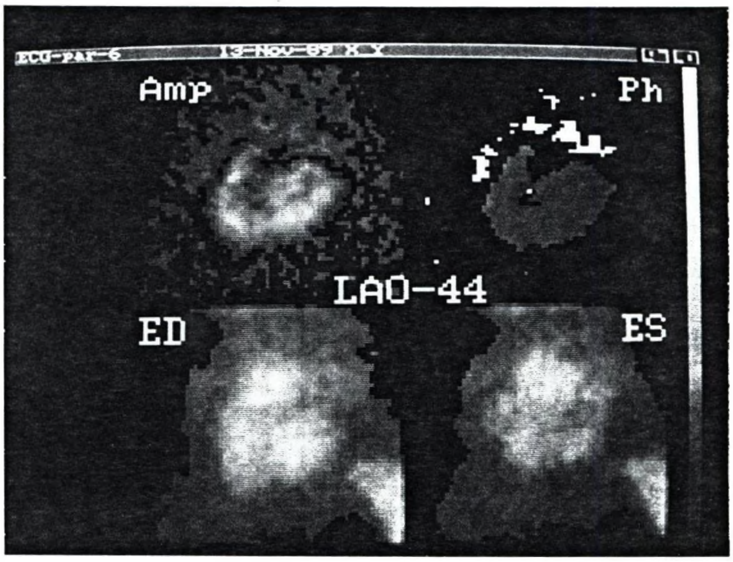
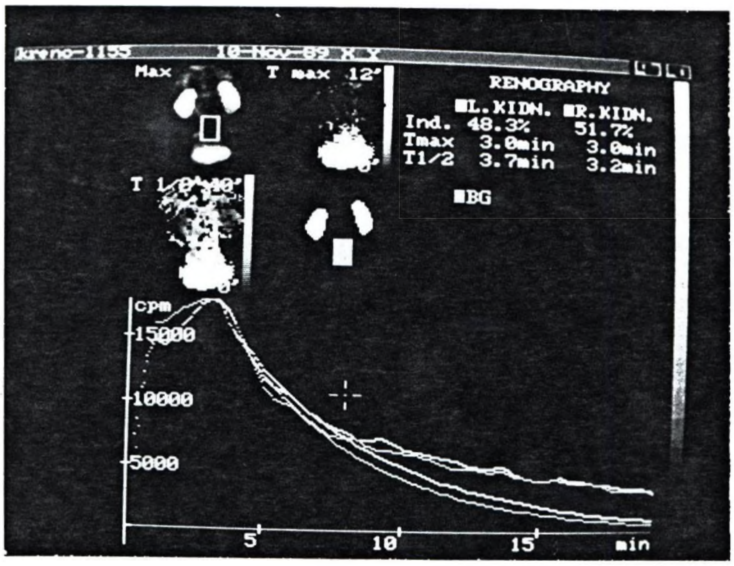


Figure 5.
The amplitude, phase, end-diastolic, end-systolic pictures of a gated heart study.



RENOGRAPHY

	■L.KIDN.	■R.KIDN.
Ind.	48.3%	51.7%
Tmax	3.0min	3.0min
T1/2	3.7min	3.2min

■BG

Figure 6.
The results of a kidney study.

ROTATIONAL-INVARIANT MOMENTS FOR 2-DIMENSIONAL SHAPE ANALYSIS

Tamás Réti, Imre Czinege
Bánki Donát Polytechnic
H - 1081 Budapest, Népszinház u. 8

ABSTRACT

Shape characterization of two-dimensional (2-D) objects using invariant moments is presented. These generalized trigonometric moments are defined on the basis of suitably selected weighting functions. From trigonometric moments shape descriptors are generated which are invariant to translation and rotation. The newly developed shape descriptors are less sensitive to the shape distorsion occuring under digital processing.

INTRODUCTION

In recent years shape analysis of two-dimensional (2-D) objects is becoming an increasingly important research area. It is not too surprising that the size of the relevant literature dealing with shape discrimination is immense.

In practice the methods based on the boundary scalar transformation techniques are of particular importance (see Exner 1987, Flook 1987, Pavlidis 1878). These involve mostly the Fourier transform (FT) of the object contour which can be expressed in terms of tangent angle versus arc length, or as a complex function describing the closed boundary curve in the complex plane. The coefficients generated by the FT are used for shape description. The major disadvantage of the techniques based on Fourier analysis is that they can not be applied for characterizing the shape of complicated objects which are represented by multiply connected regions (components) and where holes are permitted inside the regions. The new approach presented in this paper uses the concepts of the "generalized Fourier analysis" published recently (Reti and Czinege, 1989), it does not suffer from this limitation and is applicable to any type of regions.

The method reported here relies on the concepts of invariant moments and can be considered as a possible generalization of the techniques proposed by Hu (1962) and used by Hsia (1981) for the identification of 2-dimensional objects. After summarizing the mathematical bases of the suggested method we outline the computation of the trigonometric moments and some results of the preliminary experiments carried out to verify the theoretical results.

SHAPE DESCRIPTION USING CHARACTERISTIC FUNCTION

In what follows we suppose that the 2-D object A is multicolored and it is represented by closed, bounded and connected regions (components) of the Euclidean plane. Assuming that the object A is located in a polar system (s,r) such that its centroid coincides with the polar origin O, let us denote by Y_A the smallest circle with radius R_A which containes A as shown in Fig.1.

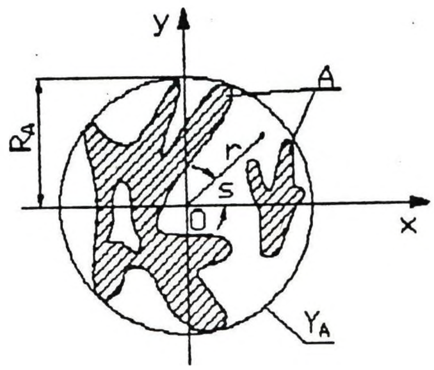


Fig.1. Illustration for defining the characteristic function $F_A(s,r)$

For the quantitative shape description we use the characteristic function $F_A(s,r)$ defined as

$$F_A(s,r) = \begin{cases} f(s,r) & \text{if} \quad (s,r) \in A \\ 0 & \text{if} \quad (s,r) \notin A \end{cases} \tag{1}$$

where the positive bounded function f(s,r) is the color of the object A. In the case of single-color objects, $f(s,r) = C_1$ by definition, where 1 = 1,2,...L, and L is the number of different colors. This implies that $F_A(s,r)$ is a binary function for 'monochrom' shapes. In the following we suppose that $F_A(s,r)$ is periodic with period 2π with respect to the rotation angle s in the interval $(-\infty,\infty)$.

Two objects A and B have the similar shape if object A can be obtained from the region B through translation, dilation and rotation. Using characteristic functions the definition of the shape similarity can easily be extended for multicolor objects. Two color objects A and B are considered as being of similar shape if the corresponding characteristic functions F_A and F_B satisfy the condition

$$F_B(s,r) = F_A(s + s_u ,r)$$

for all (s,r) except on a subset of measure zero, where $s_u \in [0,2\pi)$. A shape denoted by A is said to be

i. J-fold rotational symmetric if the relation

$$F_A(s,r) = F_A(s + s_J,r)$$

is valid, where $s_J = 2\pi/J$ and $J = 2, 3,...$

ii. axially symmetric, if there is an $s_y \in [0,2\pi)$ which satisfies the following equation:

$$F_A(s,r) = F_A(-s + s_y,r)$$

iii. multiplicative if its characteristic function can be written as

$$F_A(s,r) = F_1(s) F_2(r)$$

iv. circular symmetric if its characteristic function is independent of the variable s, that is $F_A(s,r) = F_C(r)$ is valid.

v. of radial type if its characteristic function does not depend on the variable r, consequently $F(s,r) = F_R(s)$ is fulfilled.

As follows from the definitions the circular symmetric and/or radial-type objects can be considered as special cases of the multiplicative objects and a multiplicative shape can be generated as a 'product' of a circular symmetric and a radial-type shape (see Fig. 2).

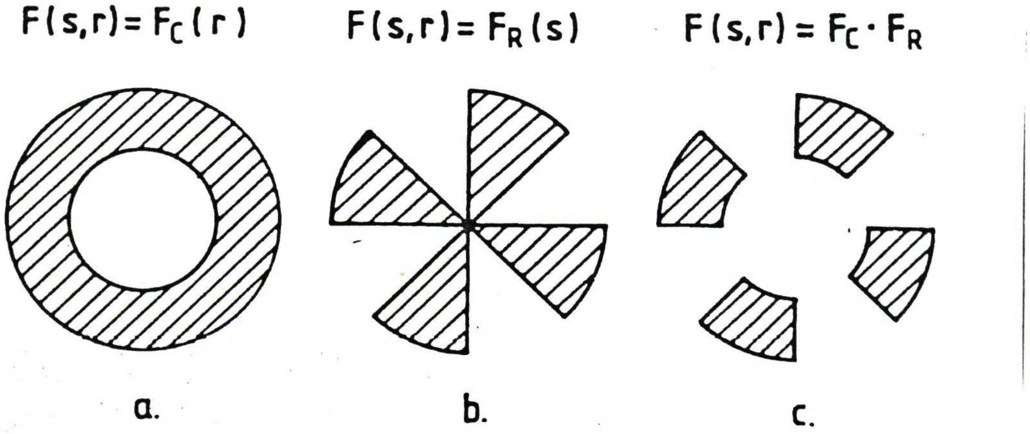


Fig.2. Special shapes: circular symmetric (a.), radial type (b.), multiplicative (c.)

TRIGONOMETRIC MOMENTS

Using the characteristic function we define the mth-order trigonometric moments $T_m^{(A)}[\eta,H]$ belonging to the object A as

$$T_m^{(A)}[\eta, H\ ] = \int_0^{2\pi}\int_0^{\infty} F_A(s,r)\sin(ms + \eta)H(s,r)drds$$

where $H(s,r)$ denotes a real weighting function, $\eta \in [0,\pi/2]$ is a scalar parameter and $m = 0,\ 1,\ 2\ \ldots$ Choosing $\eta = \pi/2$, $\eta = 0$, and $H_k = H_k(r)$ an infinite set of the mth-order trigonometric moments may be constructed:

$$T_m^{(A)}[\pi/2, H_k] = \int_0^{2\pi}\int_0^{\infty} F_A(s,r)\cos ms\ H_k(r)drds \tag{2}$$

$$T_m^{(A)}[0, H_k\ ] = \int_0^{2\pi}\int_0^{\infty} F_A(s,r)\sin ms\ H_k(r)drds \tag{3}$$

where $k = 1,\ 2, \ldots$

Based on the method for shape characterization published recently (Reti & Czinege, 1989), now we'll show that different shape descriptors which are invariant under translation, rotation and dilation can be easily generated from trigonometric moments. This novel technique uses a given set of appropriately selected 'shape functions' for generating Fourier shape descriptors. Defining the vector-valued function $U_0(s)$ which contains significant information about the shape, let us introduce the set $\mathbb{M}_A$ of K-component shape functions $U_A(s)$ given as

$$\mathbb{M}_A = \{\ U_A(s)\ |U_A(s) = U_0(s + s_x),\quad s_x \in [0,\ 2\pi)\ \}$$

where

$$U_A(s) = [U_{A,1}(s), \ldots\ U_{A,k}(s), \ldots U_{A,K}(s)]^T$$

$$U_{A,k}(s) = \pi\int_0^{\infty} F_A(s + s_x, r)H_k(r)dr \tag{4}$$

Using the Eqs. (2-4) it is easy to see that the trigonometric moments can be represented in the form of Fourier coefficients defined as

$$a_{m,k} = \frac{1}{\pi}\int_0^{2\pi} U_{A,k}(s)\cos(ms)ds = T_m^{(A)}[\pi/2, H_k] \tag{5}$$

$$b_{m,k} = \frac{1}{\pi}\int_0^{2\pi} U_{A,k}(s)\sin(ms)ds = T_m^{(A)}[0, H_k] \tag{6}$$

SHAPE DESCRIPTORS AND THEIR PROPERTIES

Starting with the Fourier coefficients obtained from the kth and nth shape function components the following shape descriptors are defined :

$$W^{(k)}_{A,m} = \sqrt{a^2_{m,k} + b^2_{m,k}} \quad (7)$$

$$W^{(k,n)}_{A,m} = a_{m,k}a_{m,n} + b_{m,k}b_{m,n} \quad (8)$$

$$Z^{(k,n)}_{A,m} = a_{m,k}b_{m,n} - b_{m,k}a_{m,n} \quad (9)$$

It should be pointed out that they are not independent of each other, that is they satisfy the following condition

$$[\ W^{(k)}_{A,m}\ W^{(n)}_{A,m}\ ]^2 = [W^{(k,n)}_{A,m}]^2 + \ [Z^{(k,n)}_{A,m}]^2$$

The main properties of the resulting Fourier descriptors $W^{(k)}_{A,m}$, $W^{(k,n)}_{A,m}$ and $Z^{(k,n)}_{A,m}$ $(m = 1, 2,\ldots)$ may be summarized as follows (Reti and Czinege, 1989):

a/ The shape descriptors $W^{(k)}_{A,m}$, $W^{(k,n)}_{A,m}$ and $Z^{(k,n)}_{A,m}$ are invariant to translation and rotation, and they can be made to invariant to size if the radius R_A of the enclosing circle with the minimum area is normalized or the weighting functions are appropriately selected.

b/ The shape factors $W^{(k)}_{A,m}$,$W^{(k,n)}_{A,m}$ are invariant under axial reflection of the object.

c/ In the case of axial reflexion of the object A the sign of $Z^{(k,n)}_{A,m}$ will be changed. Thus, if the shape is axially symmetric then $Z^{(k,n)}_{A,m}$ is equal to zero.

d/ If the object has J-fold rotational symmetry, then

$$W^{(k)}_{A,m} = W^{(k,n)}_{A,m} = Z^{(k,n)}_{A,m} = 0$$

for $m \neq 0(\mathrm{mod}\ J)$.

e/ If the object is multiplicative then $Z^{(k,n)}_{A,m} = 0$.

f/ It follows from the known property of the Fourier coefficients that the shape descriptors tend to zero with increasing m.

COMPUTATION OF TRIGONOMETRIC MOMENTS

Next, we show if the weighting functions are given in a special form and the integrals generating the trigonometric moments are expressed in terms of cartesian co-ordinates then the practical computation of the shape descriptors can be simplified considerably.

Let us assume that $h_k = h_k(u)$ is an appropriately selected real function, μ_A represents the area of the object A, and R_1 and R_2 are one- and two-dimensional scale parameters respectively. It is easy to see if the weighting functions are expressed in the form

$$H_k(r) = \frac{r}{R_2} h_k(r/R_1) \tag{10}$$

where R_1 is equal to $\sqrt{\mu}_A$ or to the radius R_A of the enclosing circle Y_A and $R_2 = \mu_A$ or $R_2 = R_A{}^2$, then the resulting trigonometric moments become scale-independent, and as a result, we can eliminate the size effects directly.

In what follows the generation of scale-independent trigonometric moments are illustrated by two concrete examples.

i. Assuming that the weigting functions are given as

$$H_k(r) = \frac{r}{\mu_A} [\, r/\mu_A^{1/2}]^{\,k-1} \tag{11}$$

the following scale-independent trigonometric moments

are obtained:

$$a_{m,k} = \frac{1}{\mu_A^{(k+1)/2}} \int_0^{2\pi} \int_0^{\infty} F_A(s,r)\cos ms\ r^k\ drds \tag{12}$$

$$b_{m,k} = \frac{1}{\mu_A^{(k+1)/2}} \int_0^{2\pi} \int_0^{\infty} F_A(s,r)\sin ms\ r^k\ drds \tag{13}$$

Using De Moivre's theorem for a positive integer m, we have

$$\exp(ims) = \cos(ms) + i\sin(ms) = (\cos s + i\sin s)^m$$

Because the right-hand side of this equation can be expanded by the binomial theorem the real and the imaginary parts of exp(ims) are written,

$$\cos(ms) = \cos^m s - \binom{m}{2}\cos^{m-2}s\ \sin^2 s + \ldots\binom{m}{4}\cos^{m-4}s\ \sin^4 s - \ldots$$

$$\sin(ms) = \binom{m}{1}\ \cos^{m-1}s\ \sin s - \binom{m}{3}\ \cos^{m-3}s\ \sin^3 s + \ldots$$

Changing the variables by substituting x = rcoss and y =rsins, the resulting integrals

$$E(m,k,j) = \frac{1}{\mu_A^{(k+1)/2}} \int_0^{2\pi} \int_0^{\infty} F_A(s,r)\cos^{m-j}s\ \sin^j s\ r^k drds$$

may be transformed to the form

$$E(m,k,j) = \frac{1}{\mu_A^{(k+1)/2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^{m-j}y^j[x^2 + y^2]^{(k-m-1)/2}F_A(x,y)dxdy$$

which is more suitable for practical computations, where $j = 0, 1, 2, \ldots$ It is obvious if $(k-m-1)/2$ is a nonnegative integer, then $E(m,k,j)$ can be computed as a simple linear combination of the $(p+q)$-order central moments defined as

$$\upsilon_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x - x_C)^p(y - y_C)^q F_A(x,y)dxdy \qquad (14)$$

$$x_C = \mu_{1,0}/\mu_{0,0}\ , \qquad y_C = \mu_{0,1}/\mu_{0,0}$$

and

$$\mu_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q \ F_A(x,y)dxdy$$

where p, q = 0, 1, 2,.... The central moments are location invariant and if the centroid of the object coincides with the polar origin, this implies that $x_C = y_C = 0$.

A finite set of scale-invariant shape descritors computed on the basis of the Eqs. (12-14) is given in Table 1.

Table 1.

Shape descriptors computed as a linear combination of central moments

$$\begin{aligned}
W^{(3,3)}_{A,0} &= [\upsilon_{2,0} + \upsilon_{0,2}] \ / \ \mu^2_A \\
W^{(5,5)}_{A,0} &= [\upsilon_{4,0} + 2\ \upsilon_{2,2} + \upsilon_{0,4}] \ / \ \mu^3_A \\
W^{(4,4)}_{A,1} &= [(\upsilon_{3,0} + \upsilon_{1,2})^2 + (\upsilon_{2,1} + \upsilon_{0,3})^2]^{1/2} \ / \ \mu^{5/2}_A \\
W^{(3,3)}_{A,2} &= [(\upsilon_{2,0} - \upsilon_{0,2})^2 + 4\ \upsilon^2_{1,1}]^{1/2} / \ \mu^2_A \\
W^{(5,5)}_{A,2} &= [(\upsilon_{4,0} - \upsilon_{0,4})^2 + 4\ (\upsilon_{3,1} + \upsilon_{1,3})^2]^{1/2} \ / \ \mu^3_A \\
W^{(3,5)}_{A,2} &= [(\upsilon_{2,0} - \upsilon_{0,2})\ (\upsilon_{4,0} - \upsilon_{0,4}) + 4\upsilon_{1,1}(\upsilon_{3,1} + \upsilon_{1,3})]/\mu^5_A \\
Z^{(3,5)}_{A,2} &= 2\ [(\upsilon_{2,0} - \upsilon_{0,2})(\upsilon_{3,1} + \upsilon_{1,3}) - \upsilon_{1,1}(\upsilon_{4,0} - \upsilon_{0,4})]/\mu^5_A \\
W^{(4,4)}_{A,3} &= [(\upsilon_{3,0} - 3\ \upsilon_{1,2})^2 + (3\ \upsilon_{2,1} - \upsilon_{0,3})^2]^{1/2} \ / \ \mu^{5/2}_A
\end{aligned}$$

Some of them are identical to those invariant moments which were proposed by Hu (1962) for the shape description of 2-dimensional objects.

If m and k are arbitrary positive integers, then $a_{m,k}$ and $b_{m,k}$ may be computed directly according to the relationships

$$a_{m,k} = \frac{1}{\mu_A^{(k+1)/2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \cos m[\tan^{-1}(y/x)]P_k(x,y)F_A(x,y)dxdy \qquad (15)$$

$$b_{m,k} = \frac{1}{\mu_A^{(k+1)/2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sin m[\tan^{-1}(y/x)]P_k(x,y)F_A(x,y)dxdy \qquad (16)$$

where $P_k(x,y)$ is given as

$$P_k(x,y) = [x^2 + y^2]^{(k-1)/2}$$

ii. An another weighting function we propose to use is defined by

$$H_k(r) = \frac{r}{\mu_A} \exp[-(k-1)r^2/\mu_A] \qquad (17)$$

In this case the resulting scale-independent trigonometric moments are

$$a_{m,k} = \frac{1}{\mu_A} \int_0^{2\pi} \int_0^{\infty} F_A(s,r)\cos ms \exp[-(k-1)r^2/\mu_A]rdrds$$

$$b_{m,k} = \frac{1}{\mu_A} \int_0^{2\pi} \int_0^{\infty} F_A(s,r)\sin ms \exp[-(k-1)r^2/\mu_A]rdrds$$

Changing the variables it can be seen that $a_{m,n}$ and $b_{m,n}$ can be expressed in terms of cartesian co-ordinates as

$$a_{m,k} = \frac{1}{\mu_A} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cos m[\tan^{-1}(y/x)]Q_k(x,y)F_A(x,y)dxdy \qquad (18)$$

$$b_{m,k} = \frac{1}{\mu_A} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sin m[\tan^{-1}(y/x)]Q_k(x,y)F_A(x,y)dxdy \qquad (19)$$

where $Q_k(x,y)$ is defined as

$$Q_k(x,y) = \exp[-(k-1)(x^2 + y^2)/\mu_A]$$

for k = 1, 2,...

EXPERIMENTAL STUDY

In order to illustrate some features of the method described we show here some results of the investigations performed on special test pictures.

For practical computations we supposed that all the 2-D object had the same single color. This implies directly that F_A can be regarded as a binary function which takes on the value 1 inside the object and 0 elsewhere. In the case of digital images the characteristic function $F_A(x,y)$ is represented by a picture matrix having 0- or 1- elements where x and y are the discrete locations of the image pixels.

In order to generate shape descriptors based on trigonometric moments a computer program written in QBASIC was developed and run an IBM AT microcomputer. For arbitrary digital objects this program computes a finite set of scale-independent trigonometric moments according to the Eqs. (15-16) and (18-19) and the corresponding shape descriptors $W^{(k)}_{A,m}$, $W^{(k,n)}_{A,m}$ and $Z^{(k,n)}_{A,m}$ for m = 0,1,2,3,4 and k,n =1,2,3,4,5.

For the investigation we choosed three digital test objects denoted by H2, U and C (see Fig.3).

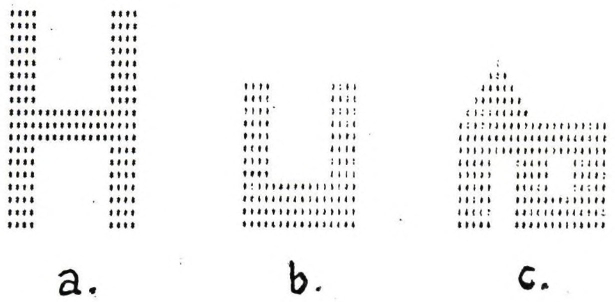


Fig. 3. Test objects used for shape analysis
Designation: H2 (a.), U (b.) and C (c.)

As an example, some shape descriptors computed on the basis of Eqs. (15-16) and Eqs. (18-19) are given in Tables 2 and 3 repectively. It can be seen that they are size- and rotation-invariant quantities. From the Eqs. (11) and (17) it follows that if k = 1 then the shape descriptors $W_{A,m}^{(1,1)}$ computed by Eqs. (15-16) and (18-19) are identical (see Tables 2 and 3).

Table 2. Shape descriptors of three test objects computed by Eqs. (15-16).

	Test object H2	 U	 C
$W_{A,0}^{(2,2)}$	0.5193 E+00	0.5448 E+00	0.4712 E+00
$W_{A,0}^{(2,3)}$	0.1618 E+00	0.1766 E+00	0.1232 E+00
$W_{A,0}^{(3,3)}$	0.3116 E+00	0.3242 E+00	0.2614 E+00
$W_{A,0}^{(5,5)}$	0.1366 E+00	0.1355 E+00	0.1010 E+00
$W_{A,1}^{(1,1)}$	0.2973 E-07	0.1103 E+00	0.1268 E-01
$W_{A,1}^{(2,2)}$	0.1491 E-07	0.2075 E-07	0.1637 E-07
$W_{A,1}^{(1,2)}$	0.4383 E-15	0.2267 E-08	-0.1882 E-09
$Z_{A,1}^{(1,2)}$	-0.6590 E-16	-0.3210 E-09	0.1343 E-09
$W_{A,2}^{(1,1)}$	0.2494 E+00	0.3239 E+00	0.3838 E+00
$W_{A,2}^{(2,2)}$	0.8135 E-01	0.2096 E+00	0.2406 E+00
$W_{A,2}^{(1,2)}$	0.2029 E-01	0.6787 E-01	0.9187 E-01
$Z_{A,2}^{(1,2)}$	0.7268 E-09	-0.4829 E-09	0.9387 E-02
Area (pixel)	192	128	187

Table 3.
Shape descriptors of three test objects computed by Eqs. (18-19)

	Test object H2	 U	 C
$W^{(2,2)}_{A,0}$	0.7465 E+00	0.7339 E+00	0.7823 E+00
$W^{(2,3)}_{A,0}$	0.4315 E+00	0.4068 E+00	0.4934 E+00
$W^{(3,3)}_{A,0}$	0.2673 E+00	0.5542 E+00	0.6306 E+00
$W^{(5,5)}_{A,0}$	0.3809 E+00	0.3411 E+00	0.4427 E+00
$W^{(1,1)}_{A,1}$	0.2973 E-07	0.1103 E+00	0.1268 E-01
$W^{(2,2)}_{A,1}$	0.2281 E-07	0.1260 E+00	0.1539 E-01
$W^{(1,2)}_{A,1}$	0.6763 E-15	0.1390 E-01	0.1944 E-03
$Z^{(1,2)}_{A,1}$	0.4918 E-16	0.2483 E-09	0.1588 E-04
$W^{(1,1)}_{A,2}$	0.2494 E+00	0.3239 E+00	0.3838 E+00
$W^{(2,2)}_{A,2}$	0.2255 E+00	0.2150 E+00	0.2602 E+00
$W^{(1,2)}_{A,2}$	0.5623 E-01	0.6963 E-01	0.9962 E-01
$Z^{(1,2)}_{A,2}$	-0.5493 E-09	0.2880 E-09	-0.7197 E-02
Area (pixel)	192	128	187

In the case of m = 1 the shape descriptors $W^{(1,2)}_{A,1}$ and $Z^{(1,2)}_{A,1}$ belonging to the object H are equal to zero. This can be explained by the fact that H is axially and centrally symmetric.

In order to evaluate the reliability of the method we analysed the computational errors which occur during the rotation of the digital objects about the origin through different angles α, where $\alpha = 0, 2\pi/9, 4\pi/9, 6\pi/9, 8\pi/9$. Some results related to the test objects denoted by H1, H2, H3 and H4 are summarized in Table 4. These test objects have the same shape (see Fig 3.a) with increasing digital area.

Table 4 includs the shape factors $W^{(2,2)}_{A,2}$ calculated by the Eqs. (15-16) and contains the means (ψ) and standard deviations (σ) of shape descriptors calculated from 5 samples. The results demonstrate that the information content in shape factors listed in Table 4 is affected by rotation and scale changes. As is seen from Table 4, the relative error ($100\sigma/\psi$) related to the computed shape descriptor is a decreasing function of the area of test object. It should be noted, that the error caused by rotation and scale changes cannot be avoid because the digital image is represented only by a set of pixels of finite size.

Table 4.

Shape descriptors $W_{A,2}^{(1,1)}$ computed by Eqs. (15-16) as a function of the rotation angle, α

Rotation angle α	Test object H1	H2	H3	H4
0	0.2557	0.2494	0.2482	0.2478
$2\pi/9$	0.2968	0.2363	0.2528	0.2496
$4\pi/9$	0.2578	0.2342	0.2502	0.2487
$6\pi/9$	0.2292	0.2534	0.2519	0.2529
$8\pi/9$	0.2196	0.2520	0.2504	0.2541
Mean, (ψ)	0.2518	0.2451	0.2507	0.2506
St.dev. (σ)	0.0269	0.0081	0.0016	0.0025
Error, $100\sigma/\psi$ %	10.69	3.32	0.629	0.977
Area (pixel)	48	192	432	768
Magnification	1:1	2:1	3:1	4:1

CONCLUSIONS

A new method for shape characterization of 2-D objects has been presented. In order to verify the validity of the proposed method some experiments have been carried out. From the results the following conclusions can be drawn.

1. The method based on the generation of trigonometric moments seems to be a very useful tool for morphological characterization of geometrical objects of arbitrary shape.
2. Depending upon the particular choice of the weighting functions, scale- and rotation-invariant shape descriptors of different type can be constructed. We have suggested two alternative ways of defining shape factors satisfying the above mentioned invariance requirements.
3. It appears that the shape descriptors computed from scale independent trigonometric moments are not very sensitive to the shape distorsion occuring under digital processing. The

investigation performed on digital test images suggests that it is possible to improve the discriminating power of the proposed procedure if we use the mean value of the computed shape descriptors obtained by rotating the digital object with different angles.

4. Further studies are obviously needed to evaluate the performance of the trigonometric moments technique in digital image analysis.

REFERENCES

Exner H. E. Shape - A key problem in quantifying microstructure. Acta Stereologica, 1987; 6/III: 1023-1028.

Flook A. The quantitative measurement of particle shape. Acta Stereologica, 1987; 6/III: 1009-1021.

Hsia T. C. A note on Invariant Moments in Image Processing. IEEE Trans. Syst., Man, Cybern., 1981; vol. SCM-11 831-834.

Hu M. K. Visual pattern recognition by moment invariants. IRE Trans. Information Theory IT-8, 1962; 179-187.

Pavlidis T. Survey. A review of algorithms for shape analysis. Computer Graphics and Image Processing, 1978; 7: 243-258.

Réti T, Czinege I. Shape characterization of particles via generalized Fourier analysis. Journal of Microscopy 1989; vol. 156: 15-32.

Radar Image and Precipitation Data Processing System at the Upper Tisza Valley Water Management Instution

György Matavovszky

Upper Tisza Valley Water
Management Institution

Nyíregyháza
Széchenyi u 19.
4400 Hungary

Introduction

Among the basic meterological elements the precipitation measurement is the most problematic because of its great variability in time and space. Nevertheless good quality, real-time areal precipitation information is highly needed by local Water Management Institutions in flood dangerous situations. Through its important role in meterological nowcasting (i.e. making extra short time weather forecasts) and through the fact that it serves as input for hydrological flood forecasting modes, it yields considerable gain in time for planning the protection.

This reason led to the development of new methods for measuring precipitation. One of them is based on the application of a radar. Though in general a meterological radar itself cannot work with sufficient accuracy, it is still often used, because it gives a quickly available, spatially and time-wide continous assesment of the precipitation field, and its uncertainties can be limited in several ways. More meterological radars, or even radar networks, are installed in Western Europe for operative usage.

In Hungary there is a network of three (MRL-5) radars. At all sites the data is processed manually which reduces the efficiency of the exploitation of the radar potential and increases the processing time very much. In Napkor (north-east of Hungary, 6 miles from Nyiregyháza), an Automatic Radar Control and Signal Processing system is now under development at one of the three radar observatories. This system will preprocess radar signals, generate precipitation intensity images in real-time mode and send them (via telephone line, at 1200 baud rate) to the center of Upper Tisza Valley Water Management Institution at Nyíregyháza. Here the images will be stored in the directories of the Radar Image and Preciptation Data Processing System designed to display images and process them for hydrological and hydrometerological purposes. The first version of the system is now ready and presented below.

Description of the system

The basic concept we followed when planning the system profile was that it should provide a tool for meteorologists to easily and quickly survey the current rain pattern and its changes thus helping them to keep track of clouds of heavy rainfall. We wanted to maximize the amount of up-to-date rainfall information in the system, so besides the radar images, the files of data produced by traditional telemetering rain gauges are also a part of system input.

Both radar images and data files (transformed by the system to pictures of izohieta lines) can be displayed in a pictorial form on a color monitor in several display modes, such as a full image display, zoomed display of user-defined subimages, and animation, i.e. a sequence of consecutive images displayed as a time-laps movie. Radar images appear projected onto a map consisting of characteristic georgaphic lines (rivers, catchment-area boundaries, etc.) using a 16 colour palette that can be composed by the user. The latter results in a loss of information because the image files give a 8 bit value for each pixel, but the colour distribution (i.e. the function defining value intervals for each of the 16 colours) can be modified flexibly. Any number of palettes and colour distributions can be saved on disk for later retrieval.

The most important step of image processing is the so called calibration ("correction") of radar images using paralell rain gauge data for integrating two kinds of precipitation information and obtaining a better assesment of the rain field than any of them can give alone. At the present stage of development, the calibration is based on a simple two dimensional interpolation, but a more realistic method is to be found after some examination of real data which is not available yet. The calibrated images can be saved and handled in the same form as the radar images.

Another image processing function of the system is to compute the difference of two consecutiv images for displaying trends and to compute the sum of several images to show rain total field for a certain period of time.

Apart from facilitating the meterological work by the enhanced graphic operations on the basis of any of the calibrated, summed, or radar images, the system can calculate areal precipitation averages of predefined subcatchment-areas for a given time period. These values will form the input data for hydrological flood forecast models.

As for the structute of the system, we regarded two basic point of view. On one hand, we used a directory-handling concept that is flexible enough to match many kinds of image storing principles not exactly specified at the moment. Also, this concept helps to organize projects of images, i.e. to enter a set of directories in which images belong together and are visible for the system. On the other hand, we tried to design a comfortable user interface to the system for easy operation. All functions can be reached through a multi-level menu with minimal number of key strokes.

The programs written in Turbo C v. 2.0 work under MS-DOS operating system on any IBM PC/AT (or compatible) computer equipped with a VGA graphics display adapter. No special image processing hardware is needed. Using a 8087/80287 or compatible math coprocessor chip means higher speed at some of the functions.

Conclusion

It is difficult to draw conclusions now when the system has not worked operativly yet and real images for tests are not available. In general, it can still be declared that the system fulfils all the original requirements and there are several reasonalble ways for further development. For example, with some more functions built into the system, it could be utilized not only in occasional flood dangerous situations, but in everyday meterological observation and in radar meteorological research as well. However, this is not only a question of programming techniques, since the struggle of different opinions about the future of the Hungarian radar meteorology will either give a chance to continue the present developmen, or put an end to it.

Low Cost IBM PC-based Image Processing Workstation

Péter Csillag

Technical University of Budapest
Department of Microwave Telecommunications
H-1111 Budapest, Goldmann György tér 3.

Introduction

Nowadays the development in hardware technology: the increasing speed of processors and the increasing amount of memory of the computers makes image processing available in many scientific and industrial applications using professional personal computers. The easiest way of bringing an image into a computer for processing is to use a video camera as an input device, and to digitise its video signal. In the different applications of video digitising there are different requirements regarding the speed of the digitisation and the quality of the digitised picture. Usually when real time digitising is required (e.g. industrial applications, production line, pattern recognition, machine vision, robotics) low quality (2-level) digitising is sufficient, and when high quality is required (e.g. image processing, picture archiving) the real time mode is not necessary. There are only a few applications which need real time and high quality at the same time (e.g. real time image coding) but these applications require special image processing hardwares and cannot be utilised on commercial personal computers. The so called "frame grabber" cards, which are produced by many firms, can capture one frame of a video signal

(usually in 512 x 512 resolution and 256 gray levels). These cards contain fast A/D and D/A converters, large amount of memory (usually 256 kbyte to 4 Mbyte) and their prices usually exceed the price of the whole computer system. As the image memory is on the frame grabber card, these digitisers usually require an additional analog monitor for the operation, and they can produce real time digitising only on that.

My aim was to design a cheap video digitiser board which can turn any IBM PC-XT/AT/386/486 or compatible computer into an image processing workstation. This digitiser can provide both of the above mentioned requirements (real time operation and high image quality) separately, and doesn't need an additional monitor even for real time digitising, so it can be used in almost all possible applications. The operation modes (real time or high quality) are software selectable, so the same card can be used in the different applications. Another advantage of my card is that while the frame grabbers usually cannot record a sequence of frames because of the big amount of memory used to store one single picture, my digitiser can store any length of moving picture (25 pictures / second), where the recorded time is limited only by the amount of memory placed in the computer. This capability can be used e.g. in making animations using a recorded motion or analysing a motion.

I. Video Digitiser Board

The operation of the digitiser

Fig. 1. contains the block diagram of the digitiser.

The video signal first goes into the input circuit where it is terminated and clamped. The clamped video signal then goes into a fast comparator, where it is compared to a DC voltage level, coming from a software controlled D/A converter.

The synchron separator generates the horizontal and vertical synchron pulses. The phases of these signals are adjustable, making it possible to adjust the horizontal an vertical position of the digitised area. The horizontal synchron pulse starts the oscillator, which produces the sampling rate. The width of the digitised area is determined by the frequency of this oscillator. When the counter reaches

320 (or 640 or 768) it stops the oscillator. The oscillator is started again at the beginning of the next row.

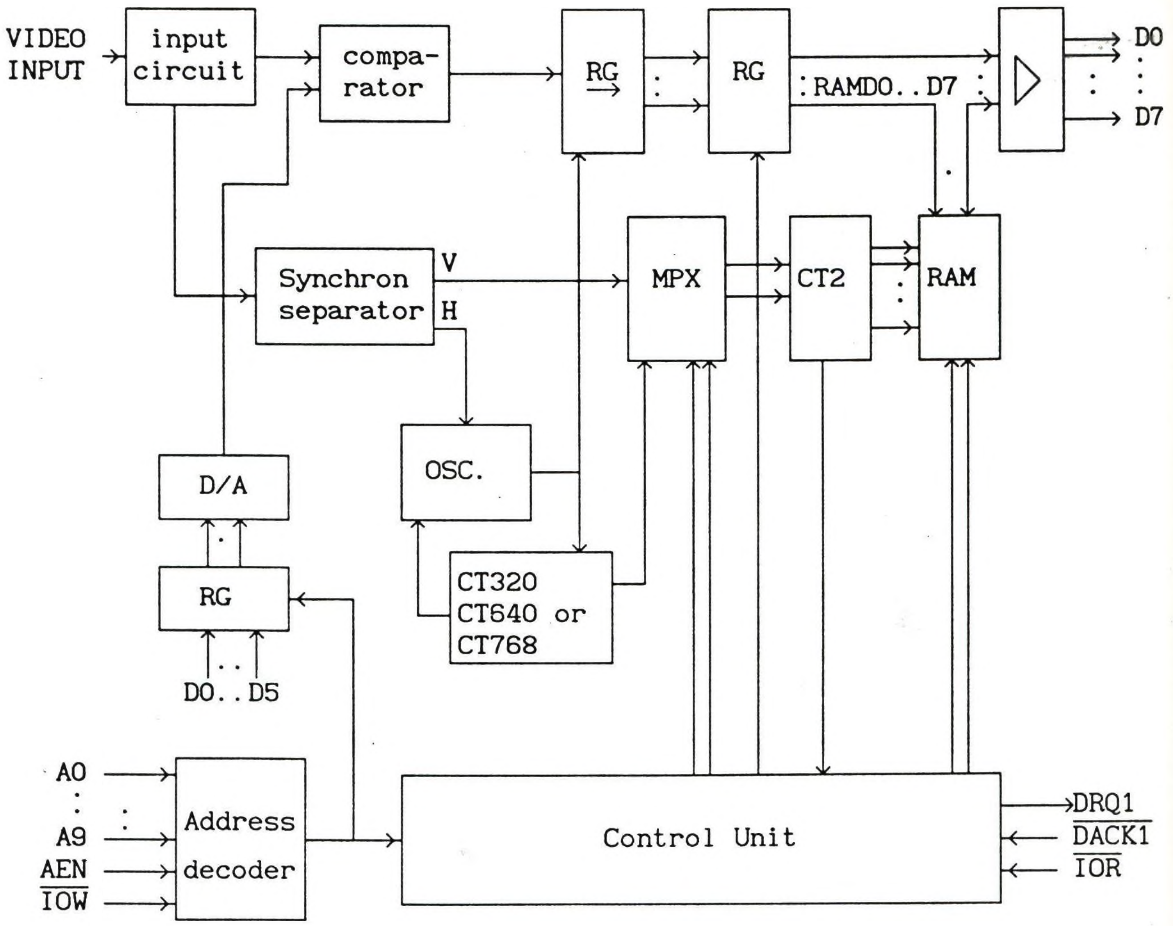


Fig. 1.
The block diagram of the digitiser

The output signal of the comparator is shifted into a shift register by the frequency of the start/stop oscillator. After every 8th shift the values of the last 8 neighbouring pixels are transferred into the internal RAM of the digitiser through a register. The addresses of the RAM are generated using a binary counter. The RAM can store one digitised bitplane (1 bit / pixel). Using 320 x 200 resolution the size of the RAM is 8 kbytes.

During digitisation the vertical synchron pulse is the reset signal of the binary counter, and the clock of the counter is the output signal of the oscillator divided by 8. When the RAM is full, the digitised picture is transferred from the RAM of the digitiser board into the RAM of the computer using a DMA (direct memory access) channel. During the DMA transfer the reset and the clock of the binary counter are generated by the control unit.

The role of the control unit is to provide all of the control signals for the digitising and for the DMA transfer.

The digitising is started by executing an OUT instruction using the peripheral address of the digitiser board. In the same out instruction a 6 (or 8) bit value is transferred to the register of the D/A converter, and 2 bits are sent to the expansion connector for the external colour decoder unit or video multiplexer.

The parameters of the digitiser

input: standard video signal ($1V_{pp}$, 75 Ω) from any video source

resolution:
320 x 200 recommended for CGA, MCGA and VGA
640 x 288 recommended for Hercules graphics card
640 x 480 recommended for VGA and super VGA (640x480/256)
768 x 576 recommended for super VGA (800x600)

gray levels: 2 to 64, software selectable, expandable to 256

colours: up to 262144 colours, with an external colour decoder unit or using a black & white camera and colour filters

speed: real time in 2-levels mode; using more gray levels or colours the speed of digitising mainly depends on the speed of the computer (using a 16 MHz 80286 AT the time of digitising a 320 x 200 picture in 64 shades of gray is around 6 seconds)

memory on the board: 8 kbyte to 64 kbyte (depending on the resolution) expansions: external colour decoder or video multiplexer

real time digitising is possible using any PC graphics card

Basic software, written for the digitiser:

DIGIT: This program enables real time digitising on any PC graphics card (CGA, Hercules, EGA, VGA, MCGA). Digitising is possible using the whole screen or only a part of it. The comparing level can be adjusted during the digitisation. The digitised picture can be stopped, saved or printed out.

PHOTO: This program digitises a picture in 64 shades of gray or 262144 colours into a file.

SHOW: This program displays a picture digitised by the PHOTO program.

Other programs: for converting the pictures into different standard file formats, displaying the pictures with different display adapters, printing the pictures on different printers in different sizes, filtering and transforming the pictures, calculating histograms etc.

The advantages of the above described video digitiser compared to the commercial frame grabber cards:

- low price
- no additional monitor required
- the 2-level real time digitising capability enables storing short image sequences

II. VGA - Video Converter

In some applications like producing video animations, making demonstration video cassettes of software products, or presenting computer generated or processed images in front of large public using video projectors or closed circuit television systems, it is necessary to convert the video signals of computers into standard video signal. It can be done using the VGA-Video Converter, which is a combination of hardware and software. It converts the output signals of a VGA card into standard PAL video signal.

Conclusions

The video digitiser and the VGA-video converter described in this article are the cheapest equipments that can upgrade an IBM PC or compatible computer into an image processing workstation. They can be used efficiently in scientific, educational or industrial applications. At the Technical University of Budapest this video workstation is used for carrying out image coding experiments.

Digitális képfeldolgozás mezőgazdasági alkalmazásának eredményei a Pannnon Agrártudományi Egyetem Számítóközpontjában

Berke József - Kárpáti László-Kárpátiné Győrffy Katalin

A PATE Georgikon Mezőgazdaságtudományi Kar Számítóközpontjában az OMFB által támogatott kísérleti fejlesztési program keretében egy személyi számítógépre alapozott képfeldolgozó rendszer került kialakításra. A kísérleti -fejlesztési tevékenység célja a mezőgazdasági területeken jelentkező inhomogenitások (talaj, növényállomány) behatárolása, okainak és mértékének meghatározása volt.

Az előadás egy rövid összefoglalása az eddig elvégzett feladatoknak, ismertetve a képfeldolgozó rendszer hardware elemeit és azok felhasználásával elért alkalmazási eredményeket.

A képfeldolgozó rendszer hardware bázisát egy IBM AT kompatibilis számítógép és a hozzá illesztett, a képek megjelenítését, és a digitalizált képi információk tárolását szolgáló perifériák képezik (1. ábra).

A tárolt képpontoknak a számítógép központi egysége felőli elérését, a képdigitalizálást és a képmegjelenítést egy képműkártya (Frame Grabber) biztosítja.

A megjelenítő eszközök közül figyelmet érdemel a színes nyomtató, grafikus plotter és a laser printer használata. A digitalizált felvételek mellett lehetőség van analóg felvételek digitalizálására is.

A feldolgozáshoz szükséges tárigény kialakításakor elsődleges szempont volt nagy mennyiségű adatok viszonylag gyors elérési lehetősége.

A feladat megoldását egy Vax 3000-el kompatibilis MicroVax és a PC hálózati összeköttetése jelentette. Természetesen a hagyományos tárak (HDD, FDD, Streamer) is megtalálhatók.

A képfeldolgozó eszközök hatékony működtetését az **SZKI* Pixel Kft. által kifejlesztett PRIMA (PRoper IMage Analysis) elnevezésű általános célú képfeldolgozó rendszer végzi.

* *Számítástechnikai Kutató Intézet és Innovációs Központ*

A programrendszer néhány jellemző vonása:

- nagyfelbontású képek Input/Output műveletei,
- digitalizálási lehetőség videokamerával,
- képbevitel videomagnóról,
- MS-DOS rutinok futtatása a programon belül,
- feldolgozási műveletek programcsomagba rendezése és futtatása (macrók),
- képek közötti aritmetikai és logikai műveletek,
- gyors, adaptiv feldolgozási algoritmusok,
- grafikai műveletek,
- statisztikai kiértékelések,
- széleskörű képmódosító eljárások,
- speciális számítási szolgáltatások,
- interaktív vagy parancsmegadásos vezérlés.

A Georgikon Kar kutatási témái közé tartozik a dombvidéki meliorációval kapcsolatos vizsgálatok. Légifelvételek kamerával történő digitalizálásával feldolgozást végeztünk a talajviz mozgására vonatkozólag. LANDSAT TM felvételek alapján meliorációelötti és utáni talajnedvesség viszonyainak feltérképezését végeztük a Magyar Tenger MGTSZ Zánka meloirált területein. Az űrfelvételeket a Földmérési és Távérzékelési Intézettől szereztük be, amely hazánk ürfelvétel adatbázisát gondozza. A talajnedvességgel kapcsolatos vizsgálatok egyben a rendszer próbáját is képezték.

Műholdfelvételek (LANDSAT TM, SPOT P) alapján egy kiválasztott mintaterületen elvégeztük:

- nagyüzemi táblák elkülönítését,
- a táblák területének meghatározását,
- a termesztett növények feltérképezését.

A szükséges referenciaadatokat az Óbudai Mezőgazdasági és Kertészeti TSZ táblatörzskönyve, a terület 1984. évi kiadású 1:10 000 méretarányú térképe és a gazdaság szakembereinek helyismerete szolgáltatta.

A 2. ábra a beazonosított mezőgazdasági táblákat szemlélteti.

Az egyes táblák területeinek meghatározása, tekintettel a domborzati viszonyokra, alapvetően a műholdfelvételek térbeli felbontóképességének és a táblákat alkotó elemi képpontok számának ismeretével történt. A felvételek kiértékelésének eredményei jó egyezést mutatnak az üzemi térképi adatokkal.

Néhány esetben mutatkozó eltérések elsősorban műveléstechnikai okokra vezethetők vissza, ilyen pl.: gépfordulások miatt műveletlenül hagyott területek.

A gyakorlati alkalmazhatóság szempontjából figyelemre méltó, hogy a módszer segítségével pontos és időszerű táblatérképi adatok nyerhetők és információkat kaphatunk az alkalmazott agrotechnika hatásairól.

Az egyes mezőgazdasági táblákon található növénykultúrák elkülönítése LANDSAT TM feltételek alapján történt. A feldolgozás során a 2., 4., 5., 7., sávú felvételeken átlagintenzítás értékek valamint inhomogenitás viszonyok meghatározásával lehetővé vált a mérési térben az egyes növénykultúrák elkülönítése.

Az általunk vizsgált mezőgazdasági növényi kultúrákkal végzett vizsgálati eredmények megbízhatóságát növelte az a tény is, hogy két software-vel (PRIMA, LANDSAT) végzett feldolgozás ugyanazt az eredményt szolgáltatta.

(A LANDSAT program az USA-ban készített szabadforgalmú software, amely alkalmas színkompozitok egyszerű statisztikai elemzésére. Magyarországon a FÖMI-nél szerezhető be.)

A képfeldolgozó rendszer alkalmas analóg felvételek digitalizálására is. Segítségével lehetőség nyílik nehezen felismerhető különbségek elkülönítésére és a különbségek számszerű adatokkal való jellemzésére. Pontosabban meghatározhatók a növényi kórokozók és kártevők által okozott struktúrális és felületi elváltozások mértéke.

A képfeldolgozó rendszer segítségével kidolgoztuk a növényi részek (levél, gyökér, burgonya gumó stb.) felület mérésének módszerét. Ezáltal eddig nem vagy csak nehezen mérhető növényi paraméterekről kaphatunk információt.

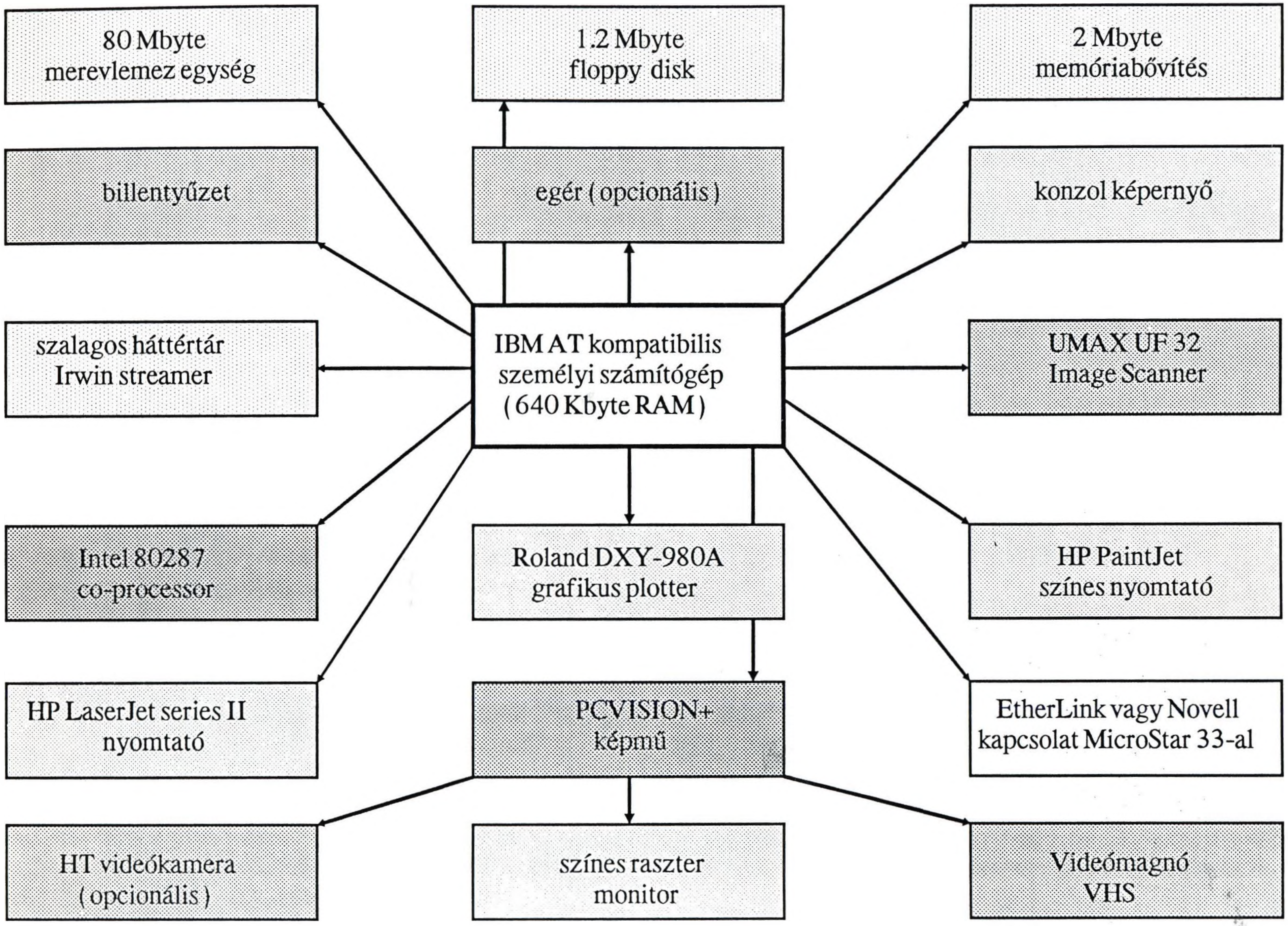
Ilyen paraméter a beteg és az egészséges növényi rész elkülönítése (3. ábra), felületük meghatározására.

A módszer alkalmazásával megállapíthatóvá vált:

- egyes herbicidek fitotoxikus hatásának mértéke,
- burgonyakárosodás számszerűsítése,
- az amerikai szövőlepke hernyójának falánksága (4. ábra).

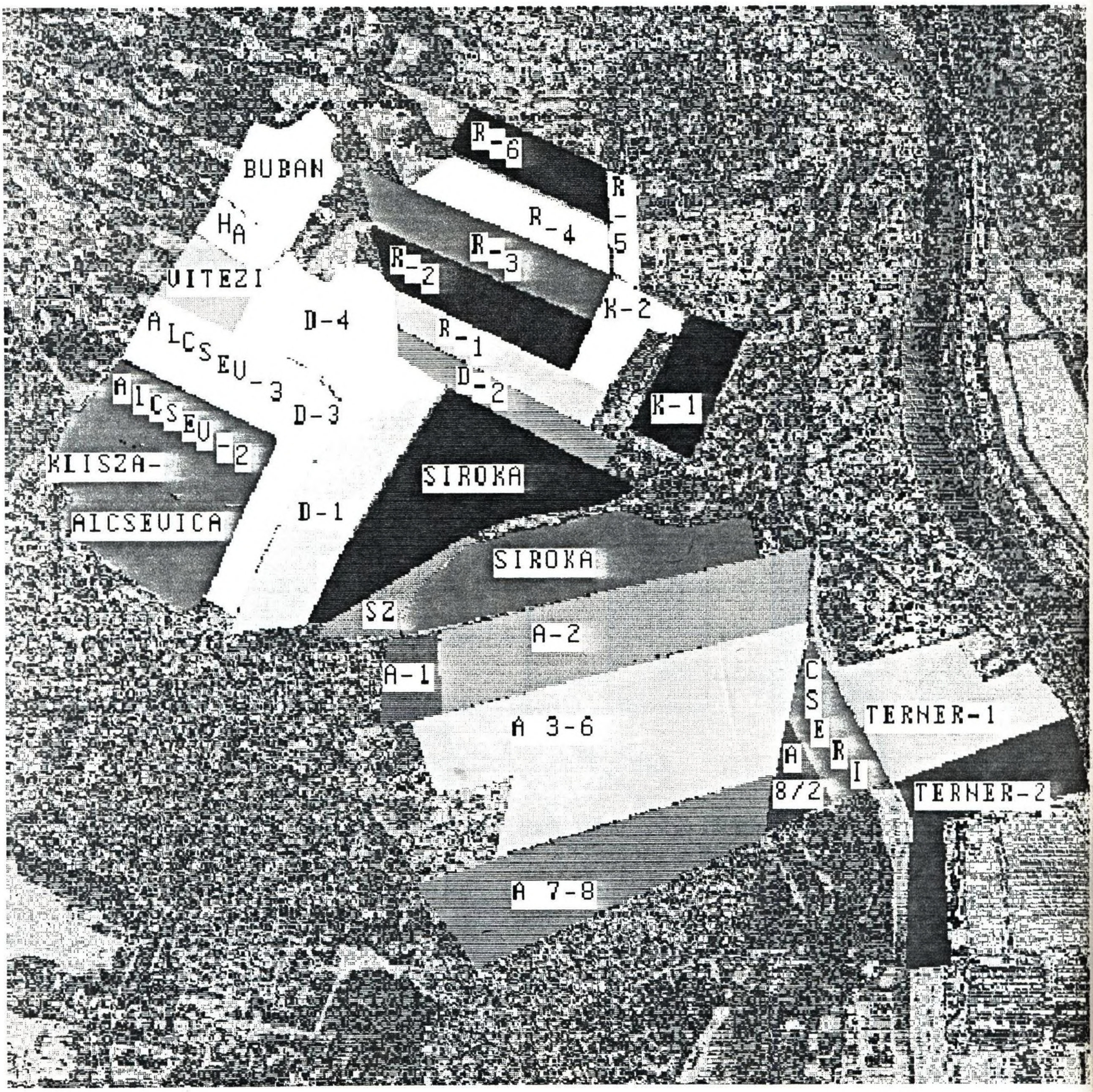
A képfeldolgozás mint vizsgálati módszer mezőgazdasági hasznosítása nagyon sokrétű lehet. Előnyei közé sorolhatjuk a gyorsaság mellett, a kinyerhető információk megbizhatóságát, és az újrafeldolgozás lehetőségét. Nagyon sok jellemző növényi paraméter hagyományos mérésénél, sokszor találkozunk az emberi szubjektivítással, mint az eredményeket befolyásoló tényezővel. A képfeldolgozási technika alkalmazásával e tényező hatása minimálisra szorítható, ami a kapott eredmények megbízhatóságát és pontossságát növeli.

PATE Számítóközpont képfeldolgozó rendszere


80 Mbyte merevlemez egység
1.2 Mbyte floppy disk
2 Mbyte memóriabővítés
billentyűzet
egér (opcionális)
konzol képernyő
szalagos háttértár Irwin streamer
IBM AT kompatibilis személyi számítógép (640 Kbyte RAM)
UMAX UF 32 Image Scanner
Intel 80287 co-processor
Roland DXY-980A grafikus plotter
HP PaintJet színes nyomtató
HP LaserJet series II nyomtató
PCVISION+ képmű
EtherLink vagy Novell kapcsolat MicroStar 33-al
HT videókamera (opcionális)
színes raszter monitor
Videómagnó VHS

1. ábra

```
WINDOW - TOP-LEFT OF CORNER:    0    0  ROW LENGTH: 600  COLUMN LENGTH:
```

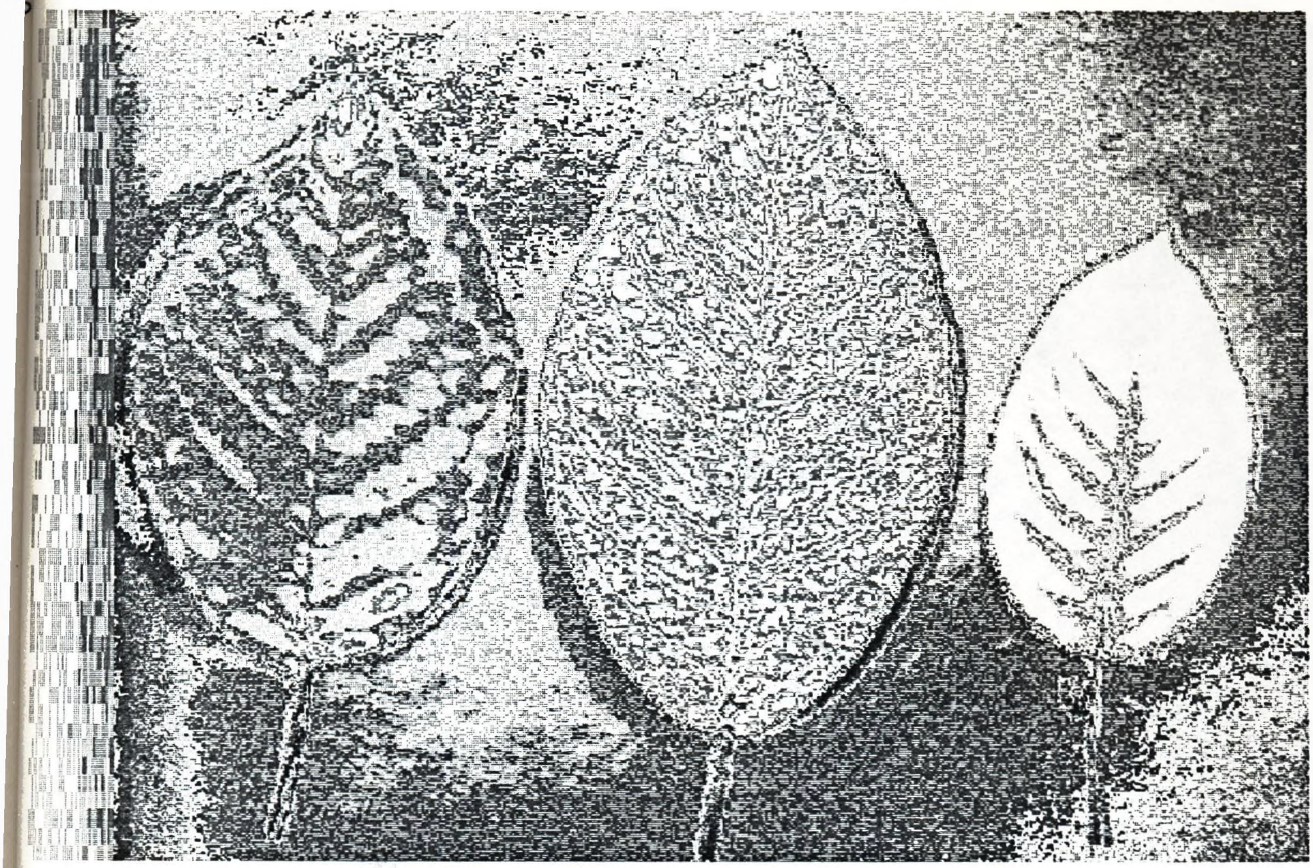


2.ábra

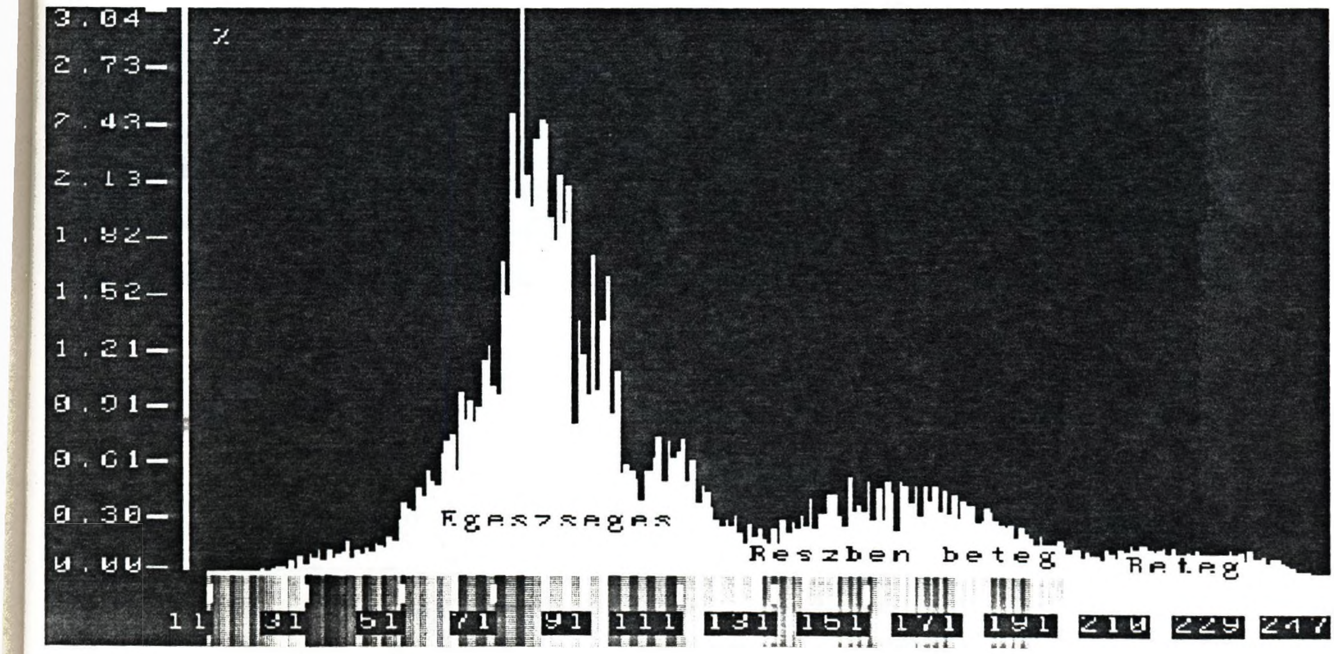
SPOT P űrfelvételen elkülönített mezőgazdasági táblák.

Egészséges Részben beteg levél Beteg

```
W - TOP-LEFT OF CORNER:   0    0  ROW LENGTH: 600  COLUMN LENGTH:  400
```



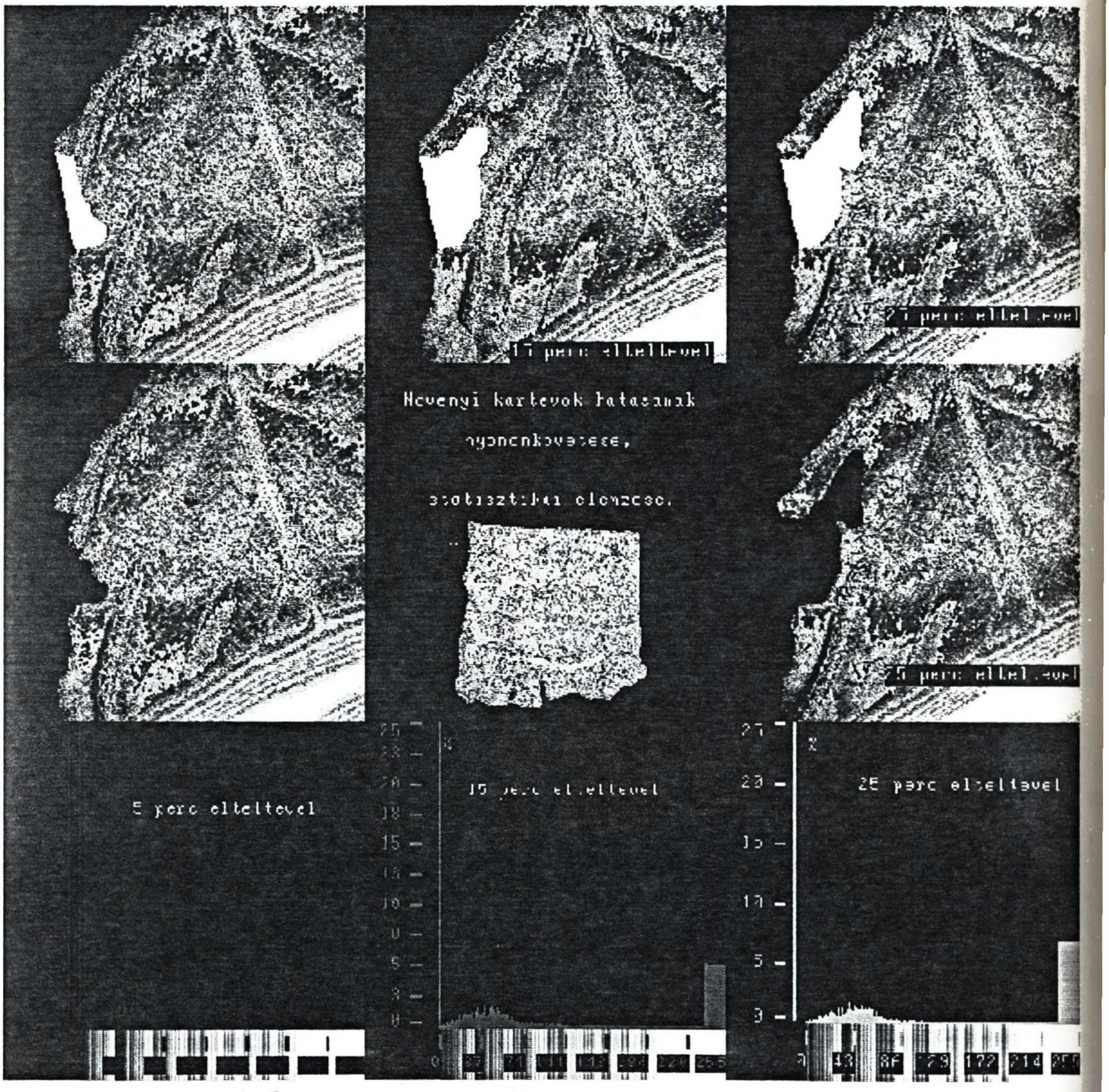
```
W - TOP-LEFT OF CORNER:   0    0  ROW LENGTH: 600  COLUMN LENGTH:  300
```



3.ábra

Növényi stresszállapotok elkülönítése.

```
WINDOW - TOP-LEFT OF CORNER:   0   0 ROW LENGTH: 600  COLUMN LENGTH: 6
```



4. ábra

Nehezen mérhető növényi paraméter (az amerikai szövőlepke "falánksága") meghatározásának nyomonkövetése.

AKTÍVFÉNYES VALÓSIDEJŰ ROBOT LÁTÓRENDSZER FEJLESZTÉSE

Dr. Loványi István - Nagy Ákos

BME Folyamatszabályozási Tanszék
1111 Budapest, Müegyetem rkp. 9.

A valósidejű képfeldolgozás igénye számos ipari alkalmazásnál felmerült. A kívánt felbontás és feldolgozási sebesség biztosítása elérhető áron - ez volt egyik fő célkitűzésünk. 1989/90-ben mutattunk be két olyan alkalmazást (ipari radiográfiai feltételek kiértékelése, felületi minőségellenőrző rendszer), ahol az extrém nagy felbontást és működési sebességet CCD vonalkamera, illetve célhardver biztosította [1], [2].

1990-ben a Folyamatszabályozási Tanszéken üzembehelyezett GT-6A oktatórobot intelligenciájának növelését biztosító látórendszer fejlesztése az említett problémákon túlmenően a 3-D információ valósidejű kezelését is felveti. A cikk ezen újabb - robot alkalmazástechnikai - példán keresztül kívánja az IBM PC AT bázisú, valósidejű képfeldolgozó rendszerek széleskörű ipari alkalmazhatóságát alátámasztani. A feldolgozás egyes lépéseit egy kisérleti robot "szem-kéz" rendszeren off-line módon próbáltuk ki, jelenleg a valósidejű realizáción dolgozunk. Egyes részek kapcsolódnak a Tanszék és több francia egyetem együttműködése keretében a BME-n diplomatervezési gyakorlatot végző francia hallgatók munkájához is [3], [4].

A robot látórendszer megvalósítása során a szakirodalomban [5] jól ismert módszerekkel (sztereo látás, 3-D információ mérése árnyékhatások alapján, aktívfényes eljárás) egyaránt próbálkoztunk, de végül - viszonylagos egyszerűsége illetve zavarérzéketlensége miatt - az aktivfényes megvilágítást választottuk.

Megemlítjük, hogy a szürkeségi szint különbségek kiértékelésén alapuló módszereket is sikerrel alkalmaztuk robottal szerelt gömbcsapok pozícionálására és néhány egyszerűbb hiba detektálására (furathiány, illeszkedési hiba, stb.), de a problémát csak 2 dimenziósként kezelve, referencia megvilágítási viszonyokat (erős átvilágítás) alkalmazva lehetett eredményt elérni [6].

I. KISÉRLETI ROBOT SZEM-KÉZ RENDSZER ELEMEI (1. ábra)

GT-6A robot

6 szabadságfokú eletromos oktatórobot. A vezérlést egy IBM PC-AT-be helyezett elektronika és külön erősáramú egység látja el. Standard alkalmazásnál e PC soros vonalon fogad parancsokat egy másik, felügyelő funkciókat ellátó IBM PC-AT-tól.

INSIGHT képfeldolgozó rendszer

Az IBM-PS/2 bázisú látórendszer felbontása 720x560 képpont, 256 szürkeségi szint. Az érzékelő CCD mátrixkamera. A feldolgozás eredményét soros vonalon továbbítja a felügyelő számítógépbe.

Lézer fénykéses megvilágítás

A 3D információ mérésének elvét az 2. ábrán láthatjuk. A fénykést lézerdióda, kollimátor és hengerlencse biztosítja. A kiértékelhetőség feltétele, hogy a kamera és a megvilágítás egyaránt kalibrált pozícióban legyen [3]. A GT-6A robotkarra szerelt lézerforrás adott mintát vetít a mérendő objektumra. A minta torzulása a 2D kameraképen hordozza a 3D információt. A fényforrás, a tárgy és a kamera relativ helyzetének változtatásával kapott metszékek képe komplex 3D látvány kiértékelését is lehetővé teszi. Ismert pozíciójú megvilágítás és kamera esetén a metszék 3D trajektóriája a 2D kameraképből triangulációs módszerrel meghatározható. A kisérletek során egy fix kamerát és egy, a manipulátorra szerelt mozgó megvilágító egységet használtunk, de a felismerendő tárgyak bonyolultságától függen az érzékelő, megvilágító egységek száma és elhelyezése is módosulhat.

TRANSPUTER bázisú aritmetika

Feladata a nagytömegű képi információ gyors feldolgozása. Jelenleg a felügyelő IBM-PC-AT része.

II. INTEGRÁLT "SZEM-KÉZ" RENDSZER

Az előző pontban vázolt kisérleti rendszer bonyolultságát részben a meglevő eszközök minél gyorsabb felhasználása, illetve egyes gyári szoftverek viszonylagos zártsága indokolja. A végleges rendszer a képdigitalizálót, pipeline képfeldolgozó processzort, Transputer kártyát és a robotvezérlőt egyetlen IBM-PC-AT-ben fogja össze:

- Az 512x512x8 bit felbontású digitalizálóhoz speciális videobuszon pipeline célhardver kapcsolódik, mely a valósidejű előfeldolgozó műveleteket (hisztogram, shading korrekció, NxM konvolúció, stb.) az IBM PC-AT-nél kb. két nagyságrenddel gyorsabban végrehajtja.

- A robotvezérlő program módosításával megoldott a digitalizálás, robotvezérlés és jelfeldolgozás párhuzamosítása, vagyis a robot egy taskjának végrehajtása közben - kilépve a vezérlő programból - az IBM PC-AT egyéb feladatokat hajthat végre.

- A Transputer modul az előfeldogozott digitális képből a tárgy típusát, pozícióját és orientációját leíró 3D információt emeli ki.

A várható számítási igény és az adatfolyam sebességek számbavétele után célszerűnek tünt, hogy az alkalmazott transputer hálózat logikai elrendezését az aktuálisan futó algoritmusok optimalizálása érdekében dinamikusan változtassuk. A Microway Quadputer kártya fizikai kialakítása támogatja ezt a számítási stratégiát, mivel a T800 elemek teljesen összekapcsolt hálózatot alkotnak (minden trasputer minden másikkal össze van kapcsolva a link vonalakon keresztül). A kép előfeldolgozó eljárások során mátrix, illetve pipeline architekturát választottunk, amelyek kiválóan alkalmazhatók pl. a hullámfront és szisztolikus jellegű algoritmusok futási idejének minimalizálására. A tulajdonságtérben végzett felismerő műveletek implementálásához a processzor farm elrendezést rendeltük, amely az un. "dinamic load balancing" elv alapján működik. Ebben az esetben a hálózat master és slave elemekre oszlik; a slave-ek mindegyike az adott probléma egy-egy szegmensét dolgozza fel, a master pedig gondoskodik arról, hogy azok a slave elemek, amelyek végeztek az adott feladatszegmenssel, új adathalmazt kapjanak.

Ilyen módon a hálózat potenciális teljesítőképessége maximálisan kihasználható. A hálózat elemei közti feladatparticionálás - amely a processzor farm elrendezés legkritikusabb problémája - optimalizálására egy saját fejlesztésű teljesítményanalízis programot használtunk, amely egy futási idejű logikai óra alapján részletes kimutatást készít az egyes processzorok kihasználtságáról. A programnak a kifejlesztése során alapelvnek tekintettük, hogy a létrejövő rendszer újabb transputer kártyákkal egyszerűen bővíthető legyen, tekintettel a szóbajöhető konkrét ipari felhasználások sebességigényére.

III. LÁTÓRENDSZER FŐBB FELDOLGOZÁSI LÉPÉSEI

III.1 "Lézer pontok" megkeresése soronként (3. ábra):

A gradált képen egy hármas ablakkal (4. ábra) mozogva kiszámítjuk a középső és szélső ablakok átlagát, majd megkeressük ezen átlagok különbségének maximumát. Ezen algoritmus számos verzióját kipróbáltuk [3],[4] - elsősorban a gyorsítás, zajérzéketlenség és soronkénti több lézervonal egyidejű követésének szempontjait figyelembevéve.

III.2 A szegmentálás során meghatározzuk, hogy mely lézerpontok tartoznak egy egyeneshez, és második lépésként e közelítő egyenest húzzuk meg. Több ismert algoritmust kipróbálva az 5. ábrán látható megodást választottuk.

III.3 Az alakfelismerés fázis feladata minimális számú fénykés alkalmazásával (adaptív megvilágítás vezérlés szükséges) az alakzat típusának, fontosabb méreteinek, pozíciójának és orientációjának meghatározása. Egyszerű (konvex, nem takart) tárgyakra már végeztünk sikeres kiértékelést, de a munka zöme e területen még hátra van.

III.4 Időszakos kalibráció:

A digitális képből (2D vetület) csak akkor következtethetünk a lézer szegmensek csúcspontjainak 3D koordinátáira, ha a kamera és megvilágítás helye ismert. A manipulátorra szerelt lézerdióda helyzetét mi határozzuk meg, míg az egyszerű lyukkamera modellből kiindulva annak helyzetét teszt látvány kiértékelésével gyorsan elvégezhetjük.

A lézer fénypontok megkeresését és a szegmentálást hasáb alakú és hengeres testre a 6. és 7. ábra mutatja be.

IV. ÖSSZEFOGLALÁS

- A röviden bemutatott IBM PC-AT bázisú képfeldolgozó rendszer kedvező ár/teljesítmény viszonya lehetővé teszi az ipari célrendszerekben való alkalmazást.

- A hardver és szoftver moduláris felépítése lehetővé teszi a konkrét feladatokra való egyszerű optimalizálást (kamerák, lézer megvilágítás, transputerek száma, szoftver további particionálása).

- 3D információk zajos ipari környezetben való gyors kiemelésére az aktívfényes módszer bizonyult leginkább alkalmasnak. Továbblépésként bonyolultabb 2 dimenziós minták generálásával komplex alakzatok gyorsabb felismerésével kívánunk foglalkozni.

V. IRODALOM

[1] Dr. Loványi István - Nagy Ákos: Személyi számítógép alapú valósidejű képfeldolgozó rendszer. Mérés és Automatika, 37. évf. 1989. 6. sz. 347. old.

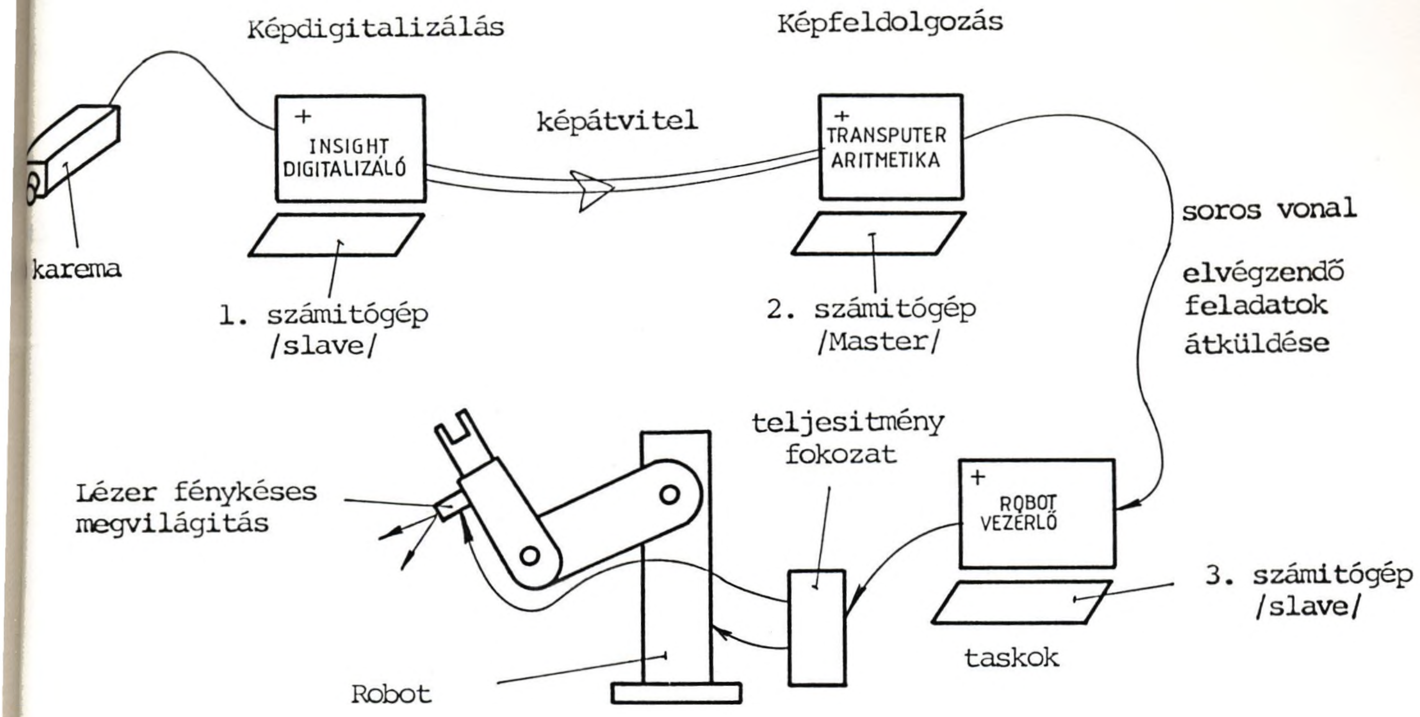
[2] Dr. Loványi István - Nagy Ákos: Számítógépes képfeldolgozás ipari alkalmazásai. Mérés és Automatika, 38. évf. 1990. 3. sz. 146. old.

[3] Pierre-Yves Pasco-Denis Callonnec: Développement d'un systéme de vision 3D pour la commande d'un robot, Budapest, 1990. (Rapport de stage - kézirat)

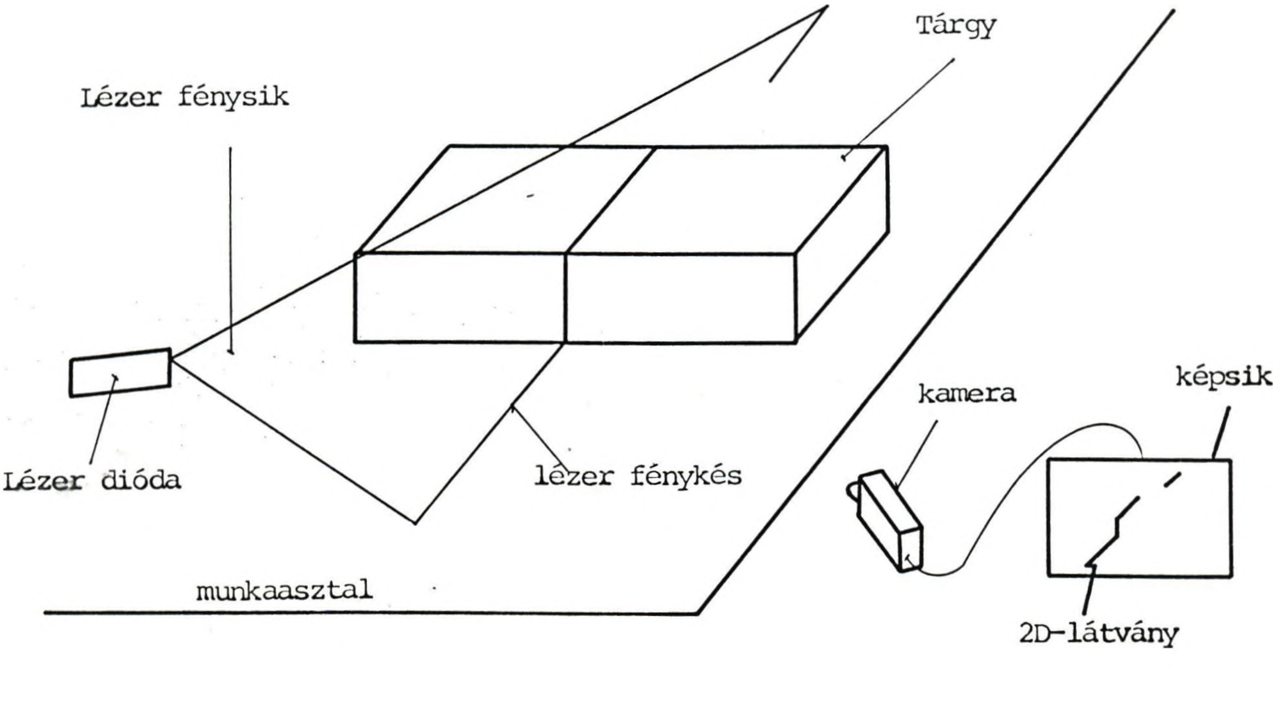
[4] A RÉMY: VISION LASER 3D. Budapest 1991. február (Rapport de stage technique - kézirat)

[5] Beni-Mackwood: Recent advances in robotics. Wiley, 1985.

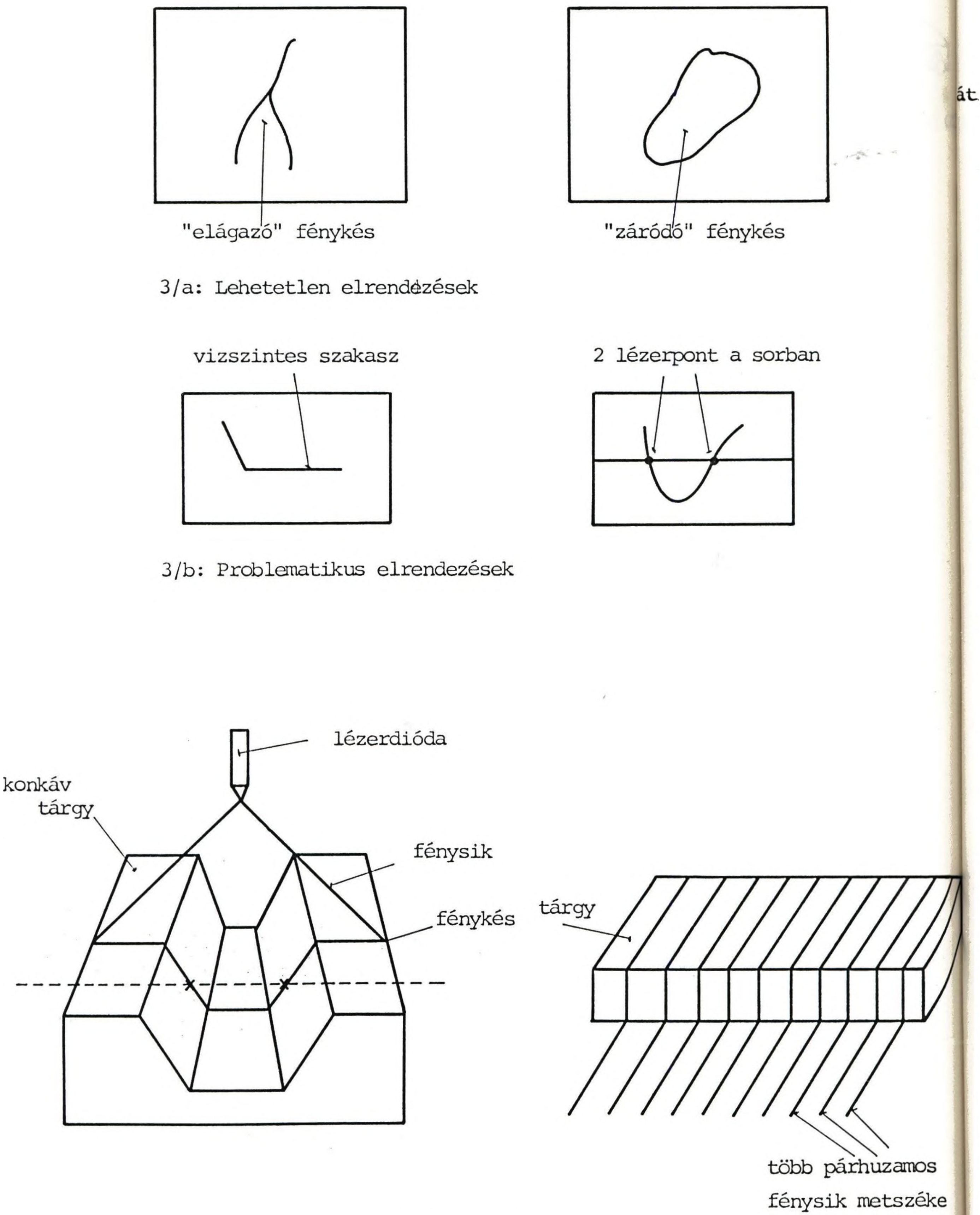
[6] Dr. Alpek Ferenc, Dr. Berkes Ottó, Dr. Loványi István, Nagy Ákos, Nagy Zoltán, Szélig Károly: Látórendszer és robotcsuklóba integrált erőmérő alkalmazása robotintelligencia növelésére, szereléshez". "A fokozott pontosságú anyagalakító technológiák és berendezések K+F feladatai" OMFB-6 tárcaprogram fejlesztési eredményei. Szimpózium, Budapest, 1990. dec. 12-13.



1. ábra: Kisérleti robot látórendszer felépitése



2. ábra: Lézer fénykéses megvilágitás elve

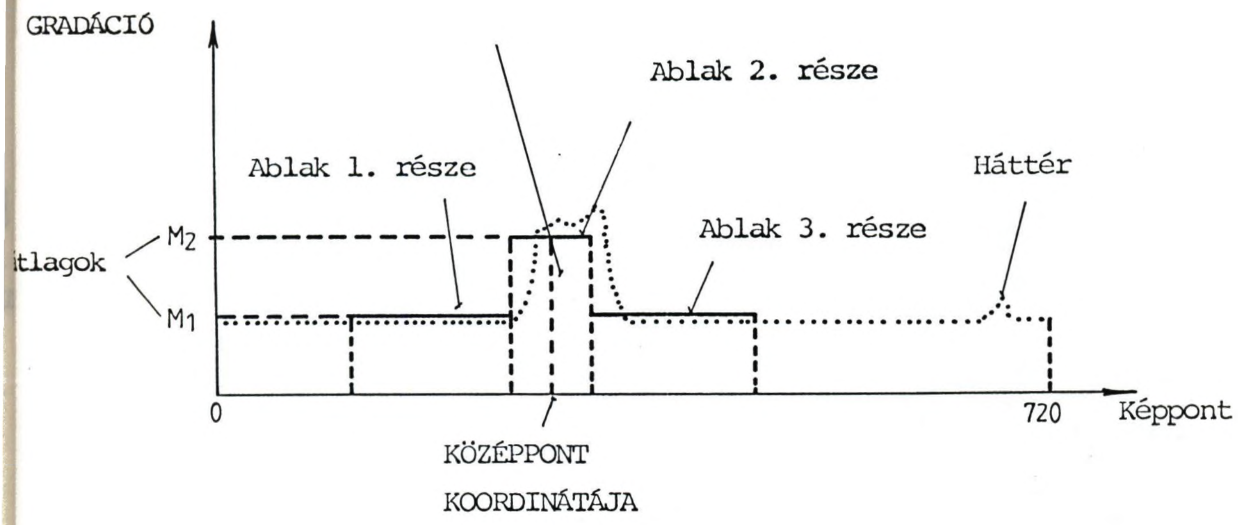


3/a: Lehetetlen elrendézések

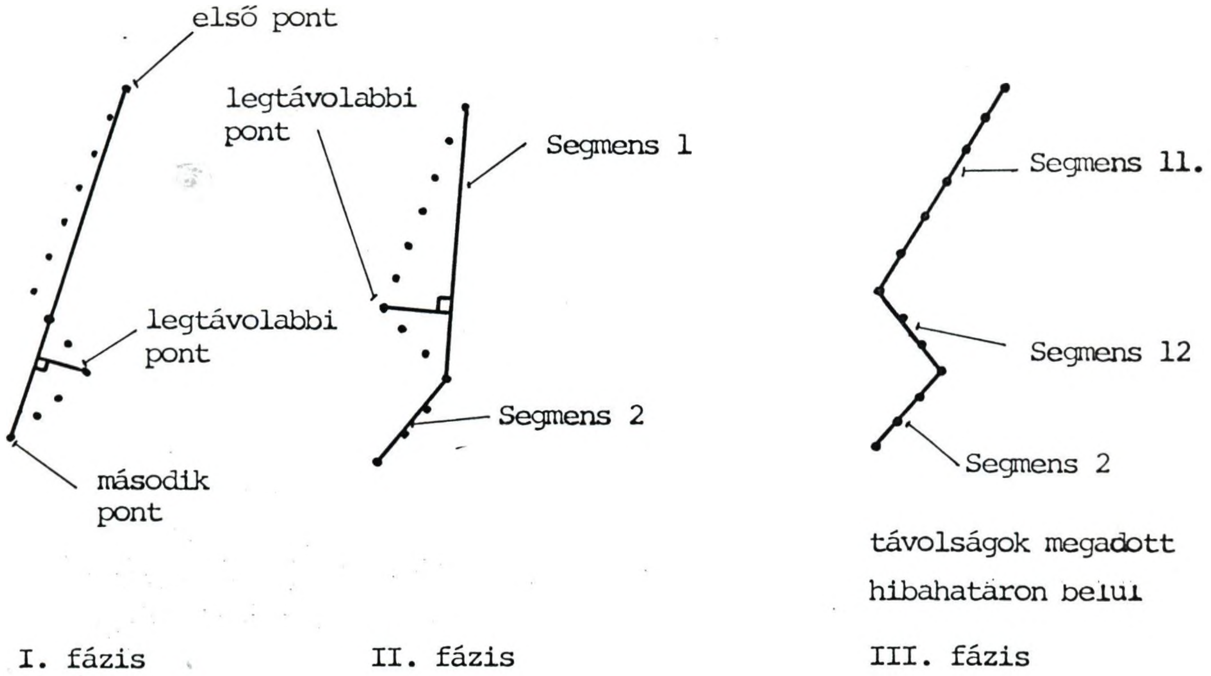
3/b: Problematikus elrendezések

3/c: kettő vagy több pont egy sorban

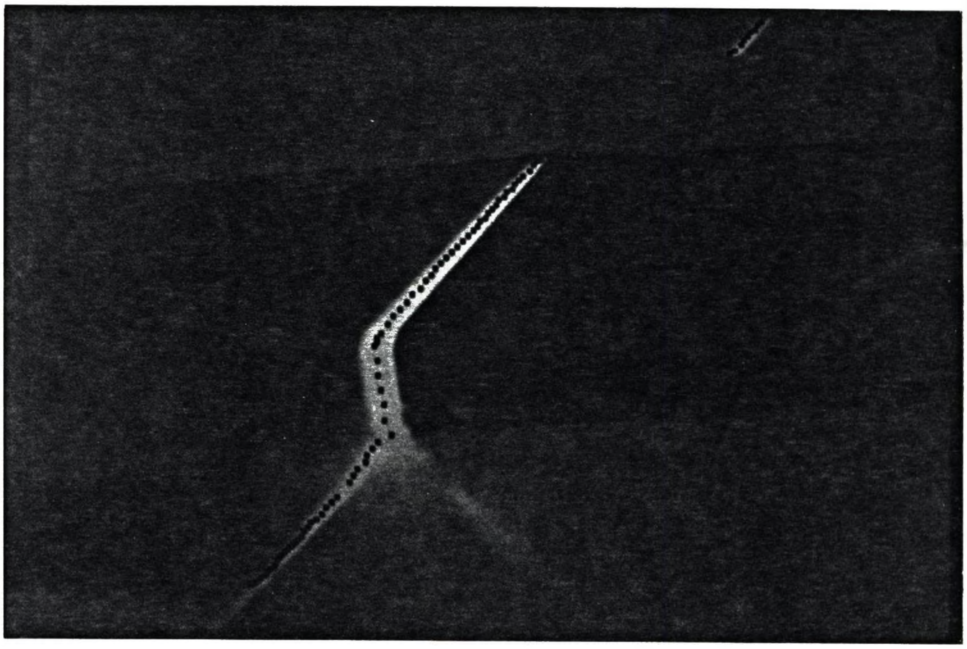
3. ábra: Lézer fénykés metszékek



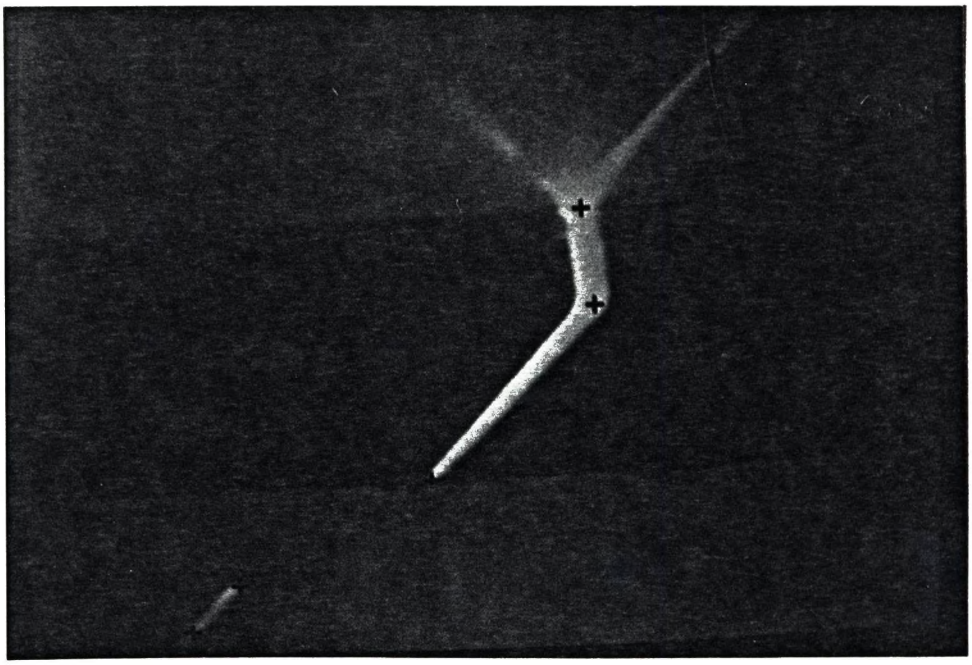
4. ábra: Lézer fénypont keresése egy sorban



5. ábra: Szegmentálás

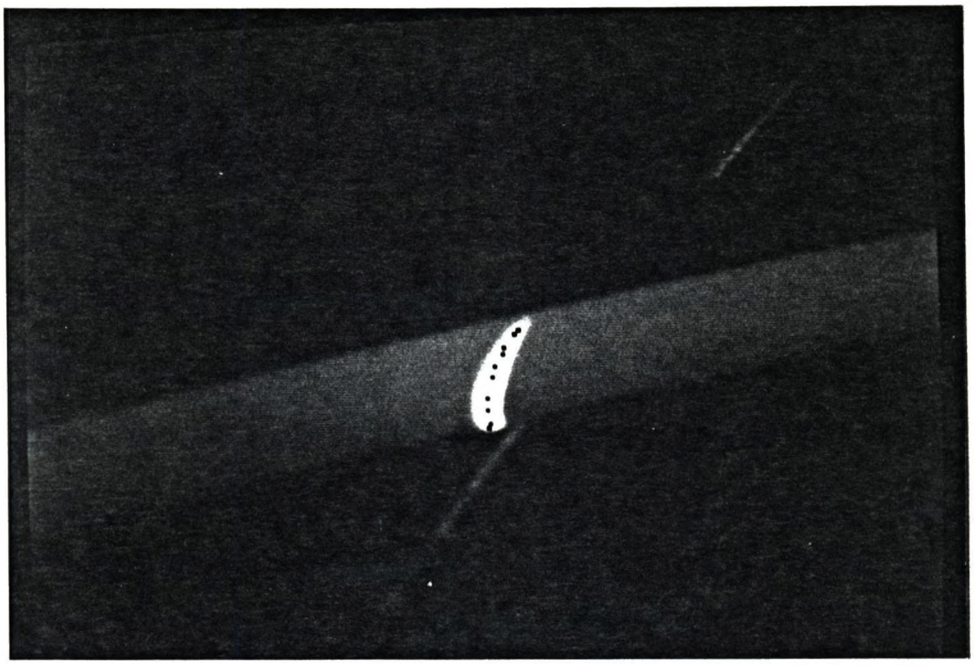


6/a: Szignifikáns lézerpontok megkeresése

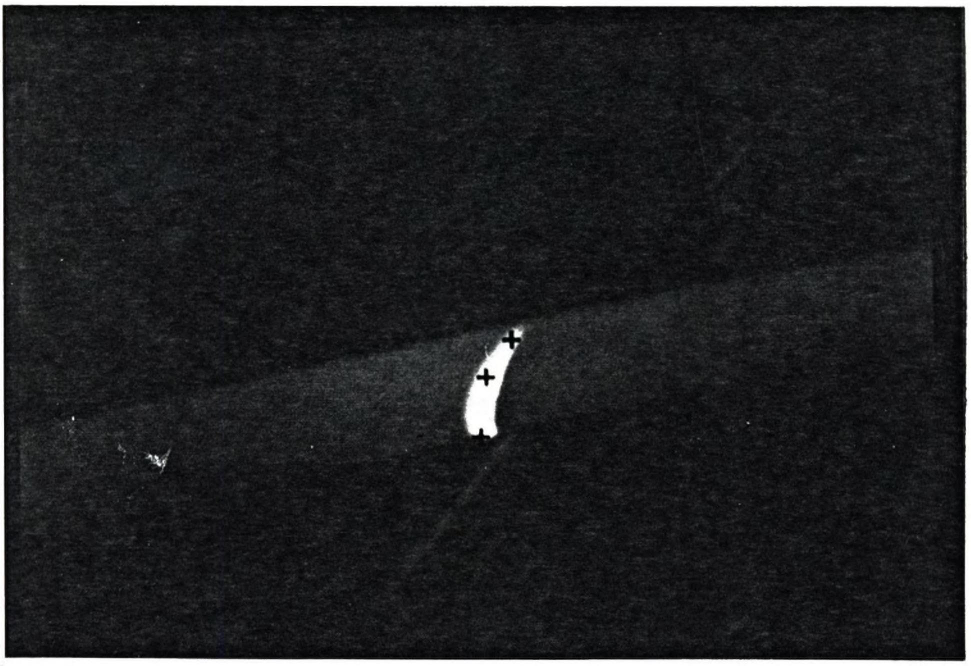


6/b: Szegmentálás

6. ábra: Lézer fénypontok megkeresése és szegmentálása zajos képen HASÁB



7/a: Szignifikáns lézerpontok megkeresése



7/b: Szegmentálás

7. ábra: Lézer fénypontok megkeresése és szegmentálása
zajos képen HENGERES TEST

COMPUTER ASSISTED PARTICLE SIZE MEASUREMENTS

S.Selinger and M.Bocu

Institute for Computer Technique Research
Cluj-Romania

Abstract

We propose a method to detect particles and to determine their statistical parameters with high accuracy.For detection of particles we use a image processing system MIPS.

System overview

The MIPS (Microscopical Particle Survey) system is an dedicated purpose picture processing IBM PC/XT, PC/AT based systems.
Connect tv-camera to a low resolution optical microscope systems will be able to analyze the two dimensional images of particles sizing.The particles image obtained by microscop is taken from video signal output of the system.
MIPS system acquires, digitises this image and stores it in its image memory (256 x 256 x 8 bit).Here local image processing is possib
This image will be displayed.For maximum operator convenience are used two monitors: monochrom and color for the image presented and the other one is the control monitor where all operational information, menus and results appear.Dedicated application software package is available the user with complete access to the functions of the system with a simple MENU format.

Investigation

In mineral processing with physical separation technique the effective concentration of an ore often depends on the degree of liberation achieved during grinding.

Traditional liberation analysis has been an expansive and time consuming process.

The locked and liberated program provides a detailed characterization of the composition distribution of particle population.The results include are fraction of ore as a function of particle sizing and the relative amounts of included, locked and free grains.

Features to be measured are selected by virtue of their grey level contrast.The analysis involves the characterization of a feature in terms of its grey level intensity and its frequency histogram or predefined thresholds.The program is capable to analyse individual features even though the feautures are overlapping and partially hidden by predefined thresholds to produce contours.

This dedicated software package is capable of computing advanced statistical results from feature analysis data bases.All statistical data is plotted cummulatively for an analysed filed of samples.

Other dedicated application software package have been developed for cytophotometric diagnosis and morphometry.The results concerning two nuclear parameters (longest axis and area) in cells.The particles has been detected after boundaries have been reconstructed.Are interested other kind of feature of regions i.e. perimeter, size, compactness, moment, distances from center of gravity to boundary points slope and curvature of boundaries, statistics, other features.

Experimental investigation

The automated analysis of particle size measurments contains the following tasks:

- digital acquisition of images with corresponding spatial resolution and storage of the digital images.
- image processing, that means reduction of the electrooptical noises and elimination of light distorsions.For optical correction entry of two matrix calibration factors, created durring acquisition,

this correction may be applied by software as a grey level linear transformation.This feature corrects ertifacts in an image wich may be caused by systematic effects such as uneven illumination or optical abberation.

- detection of particles and generation of a perticle list wich contains the number and diameter of particles.
- interpretation of the results.

The table (l) reflects the results obtained from image cummulation of particles.

Where:

- median value of particles $$X = \frac{\sum f_m}{n}$$
- standard deviation $$\sigma = \sqrt{\frac{\sum f(m - X)^2}{n}}$$
- asymmetry $$SK = \frac{\sum f(m - X)^3}{n\sigma^3}$$
- sharpness $$K = \frac{\sum f(m - X)^4}{n\sigma^4}$$

m - middle of particle class

f - frequency of the particles in the class

n = loo%

Conclusion:

The advantages of this system are its simplicity, rapid data processing, and high accuracy.

The system described is a powerful, cost-effective, measuring tool that enables the scientist to evaluate quantitativ two dimensional densitometry of the particle sizing in a time saving manner.At the same time to obtain a permanent record of graphics and tabulated data.

The program is intended for use on a hard disk but can if required run on a floppy.

The simple MENU format requires minimal training.

References

1. M.Bocu, S.Selinger: Work-station for mineralogic analysis. CAIP'89 Berlin.

2. M.Bocu, S.Selinger, M.Vaida: Analize granulometrice cu sistemul bazat pe prelucrări de imagini. PROCOMP'89 Bucureşti.

3. C.D.Clinici,M.Bocu, S.Selinger, M.Vaida: Value of morphometric investigations in the diagnosis of pleural effusion. The 8th Balcan Biochemical and Biophisical Days.21-25 May.Cluj.

```
                 STATISTICA GRANULOMETRICA

  FI              2,25    2,5     2,75    3       3,25    3,5     3,75
MICRONI   <<      210,2   176,8   148,7   125     105,1   88,4    74,3
NR.PART.    00000   00000   00000   00000   00000   00000   00000   00000
----------------------------------------------------------------------------
  FI      4       4,25    4,5     4,75    5       5,25    5,5     5,75
MICRONI   62,5    52,6    44,2    37,2    31,3    26,3    22,1    18,6
NR.PART.    00000   00000   00000   00000   00000   00000   00000   00000
----------------------------------------------------------------------------
  FI      6       6,25    6,5     6,75    7       7,25    7,5     7,75
MICRONI   15,62   13,13   11,04   9,29    7,81    6,57    5,53    4,65
NR.PART.    00000   00000   00000   00000   00000   00000   00001   00002
----------------------------------------------------------------------------
  FI      8       8,25    8,5     8,75    9       9,25    9,5     9,75
MICRONI   3,9     3,28    2,76    2,32    1,95    1,64    1,38    1,16    <<
NR.PART.    00000   00003   00000   00003   00003   00002   00000   00000
----------------------------------------------------------------------------
Coef rezolutie = 8.10  Nr. total particule=14  Nr. imagini=2
     DENUMIRE PROBA : GHFK
   STATISTICA PROCENTUALA SIMPLA SI CUMULATA  I.T.C.I. CLUJ

MICRONI   <<      210,2   176,8   148,7   125     105,1   88,4    74,3
            0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
            100.0   100.0   100.0   100.0   100.0   100.0   100.0   100.0
MICRONI   62,5    52,6    44,2    37,2    31,3    26,3    22,1    18,6
            0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
            100.0   100.0   100.0   100.0   100.0   100.0   100.0   100.0
MICRONI   15,62   13,13   11,04   9,29    7,81    6,57    5,53    4,65
            0.0     0.0     0.0     0.0     0.0     0.0     7.1     14.3
            100.0   100.0   100.0   100.0   100.0   100.0   100.0   92.9
MICRONI   3,9     3,28    2,76    2,32    1,95    1,64    1,38    1,16    <<
            0.0     21.4    0.0     21.4    21.4    14.3    0.0     0.0
            78.6    78.6    57.1    57.1    35.7    14.3    0.0     0.0

Media   aritmetica      X= 8.66071E+00 FI ;   2.74304E+00 microni
Deviatia standard      DX= 5.57875E-01 FI ;   1.12782E+00 microni
Coeficient asimetrie   Sk= -4.49431E-01
Coeficient ascutime     K= 1.91678E+00
LAMBDA =  1.54503E-01
continua
```

Table 1.

AUTOMATIC THRESHOLD SELECTION FOR IMAGE SEGMENTATION

Mircea Bocu
INSTITUTE FOR COMPUTER
TECHNIQUE CLUJ,ROMANIA

1. Introduction

Image segmentation in accordance with a threshold is the most utilisated method in digital picture analysis.
Weszka's published work/1/ surveis the methods of the thresholds selection techniques.The most important of them are:

- selection by a maximum on the histogram;
- selection by a minimum finded into two significant maximum on the histogram;
- selection by the point of maxim concavity on the histogram;
- selection by the calculation of the betweenclass separation BCS(Otsu) of histogram;
- selection by the points with significant Laplacian on the histogram;(Weszka,Nagel,Rosenfeld) /2/ ;
- selection by the maximum of the gradient(Watanabe and its variants of Kohler,Wang,Haralick) ;
- selection by image approximation(Jalobeanu) /3/.

2. Preliminaries

Our purpose was that to improve the last method of the image approximation suggested by Jalobeanu /3/, by attributing the best value to that approximate image.So,we were addmited for measure the Euclidian distance between the f function and the approximation function g.

We were supposed - only to simplify the judgement - that we have had a square image field of dimension d.

Let p-1 the highest value of gray level in sistem.

Choising the coordinates $i,j \in [1,2,\ldots\ldots d]$, $0 \leq f_{ij} \leq p-1$,the measure would be:

$$D^2 = \sum_{i=1}^{d} \sum_{j=1}^{d} (f_{ij} - g_{ij})^2$$

There,g would be choise like that:

$$g_{ij} = \begin{cases} a & f_{ij} < t \\ b & f_{ij} \geq t \end{cases}$$

where t was the threshold, and a,b were constants, choising fo giving minim on the proper intervals.

3. Purpose method

Let $H_k = H(k)$ the distribution given by image histogram, with the gray level. The Euclidian distance on $k \in [0,t]$ interval, would be:

$$D^2 = \sum_i \sum_j (f_{ij} - g_{ij})^2 = \sum_0^{t-1} k^2 H_k - 2a \sum_0^{t-1} kH_k + a^2 \sum_0^{t-1} kH_k = \mathbf{F}(a)$$

For one given t, the sums were constants. The minimum of F was realisate for a mean value, that was :

$$a = m_i = \frac{\sum_0^{t-1} kH_k}{\sum_0^{t-1} H_k}$$

Analogous, b would be choised :

$$b = m_s = \frac{\sum_t^{p-1} kH_k}{\sum_t^{p-1} H_k}$$

The approximation function choised by us would be therefore:

$$g_{ij} = \begin{cases} m_i & \text{for } f_{ij} < t \\ m_s & \text{for } f_{ij} \geq t \end{cases}$$

We could now calculate the measure for an approximation of the whole picture:

$$D^2 = \sum_{i=1}^{d}\sum_{j=1}^{d} (f_{ij} - g_{ij})^2 = \sum_{i}\sum_{j} (f_{ij}^2 - 2 f_{ij}g_{ij} + g_{ij}^2) =$$

$$= \sum_{0}^{t-1} k^2 H_k - 2m_i \sum_{0}^{t-1} kH_k + m_i^2 \sum_{0}^{t-1} H_k + \sum_{t}^{p-1} k^2 H_k - 2m_s \sum_{t}^{p-1} kH_k + m_s^2 \sum_{t}^{p-1} H_k =$$

$$= \sum_{0}^{p-1} k^2 H_k - \left[\frac{\left(\sum_{0}^{t-1} kH_k\right)^2}{\sum_{0}^{t-1} H_k} + \frac{\left(\sum_{t}^{p-1} kH_k\right)^2}{\sum_{t}^{p-1} H_k} \right]$$

Now,here was necesary to choise a t for a minimum, therefore t for which the function

$$F = \frac{\left(\sum_{0}^{t-1} kH_k\right)^2}{\sum_{0}^{t-1} H_k} + \frac{\left(\sum_{t}^{p-1} kH_k\right)^2}{\sum_{t}^{p-1} H_k} = m_i^2 A_i + m_s^2 A_s$$

was maximum.

An interpretation which used the means values and the areas was suitable only to prove that the maximum existed.

Practically,the function F was calculated from picture histogram, point by point,choising the k value for maximum,and that value was declarated the selected threshold (t) .

4. Multiple segmentation

The above presented calculation may be extended for a number n of thresholds (t_1 ,t_2 ,.....t_n), which may give a segmentation in the $n+1$ intervals.The measure in this case is:

$$D^2 = \sum_{0}^{p-1} k^2 H_k - \left(\frac{\left(\sum_{0}^{t_1-1} kH_k\right)^2}{\sum_{0}^{t_1-1} H_k} + \frac{\left(\sum_{t_1}^{t_2-1} kH_k\right)^2}{\sum_{t_1}^{t_2-1} H_k} + \cdots\cdots + \frac{\left(\sum_{t_n}^{p-1} kH_k\right)^2}{\sum_{t_n}^{p-1} H_k} \right)$$

This problem is very difficult.We want only to reveal that D is

improved when multiple thresholds are choised.
So,we are suggesting an approximation on the problem solution, choising a treatment by dycotonomia; after algoritm's applicati a threshold will be find $t_{n/2}$, on $[0,t_{n/2}]$ and $[t_{n/2},p-1]$, intervals,than the algoritm will be applied again,finding $t^{i}_{n/4}$ $t^{s}_{n/4}$, and so on...
This method has the advantage of simplicity and it ofer very good and practical results.

5. Experimental results

Our method was applied and was worked on personal image proces sistem.The algoritm was applied to granule's selection on the field of granulometry,the multiple segmentation procedure was applied with great succes in densitometry.

References

1 Weszka J., " A survey of threshold selection techniqes" Comput.Graphics and Image Proc. 7,1978,259-265

2. Rosenfeld A.,De la Torre P., "Histogram concavity analy as an aid threshold selection",IEE Trans.System Cybern.,SMC-13,1983,231-235

3. Jalobeanu M., "Threshold Selection based on Image Appro tion", L'Analyse numerique et la Theorie de L'Ap ximation,2/1986,131-139

ARGUS (IPW)

IMAGE PROCESSING WORKSTATION

IPW is a state-of-the-art Image Processing Workstation, advanced development tool for **image manipulation and analysis.**

IPW BASIC CONFIGURATION

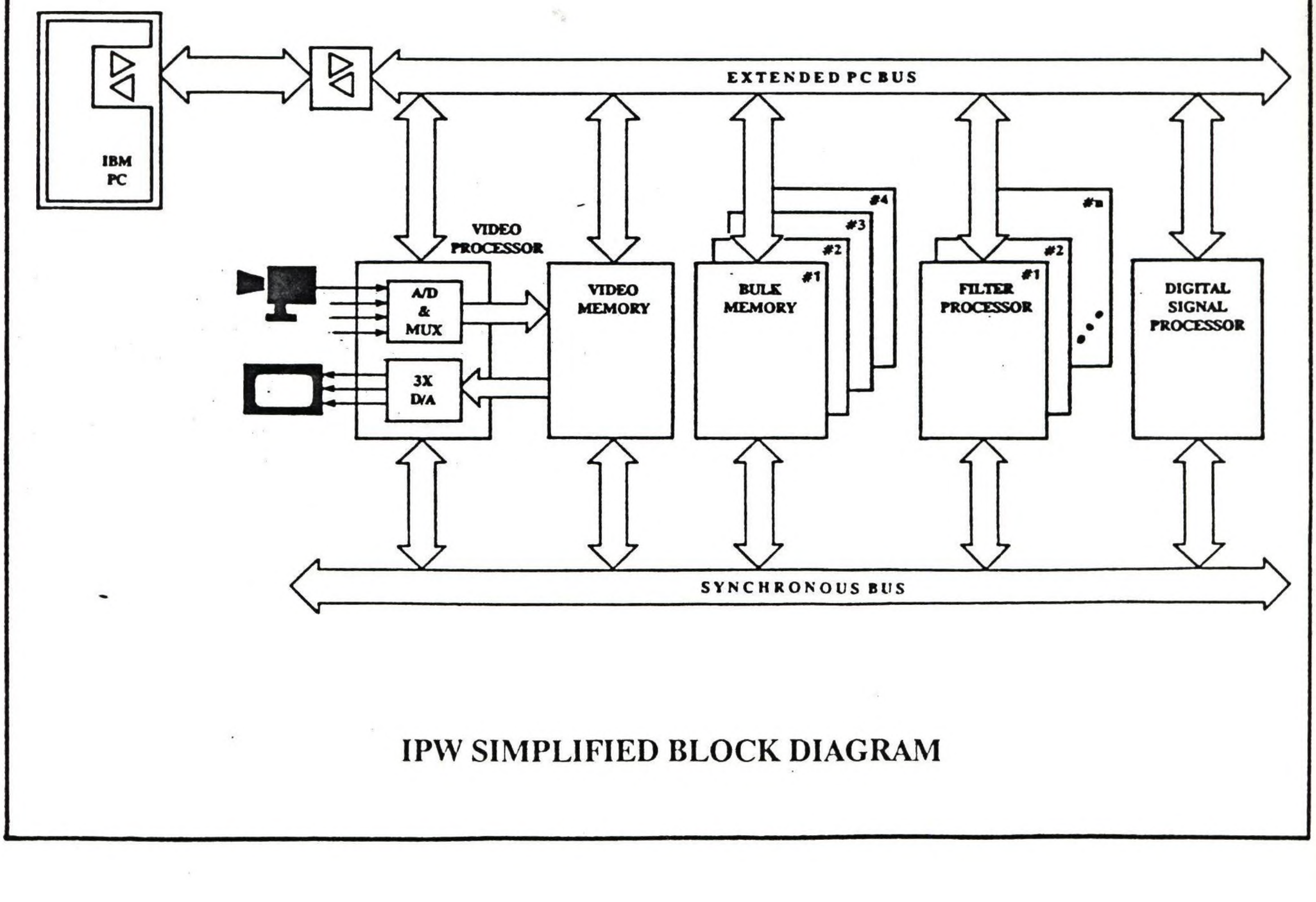
Host computer (IBM AT compatibles)

IPW modules

- processors
- image memories

INPUT camera

OUTPUT colour display



IPW SIMPLIFIED BLOCK DIAGRAM

KFKI RASTER Ltd. • KFKI RASTER Ltd. • KFKI RASTER Ltd. • KFKI RASTER Ltd.

IPW MODULES

PROCESSORS

Video Input-Output Processor

Input
- TV camera (colour or b/w)
- CCD line-scan camera
- video recorder

Lookup tables for colour and gray level transformations
Four displaying modes for colour and gray images
Various cursor patterns
Output RGB colour display, 3 x 8 bit resolution

Digital Signal Processor
16 x 16-bit Multiplier
4o-bit Accumulator, Barrel Shifter
Direct connection to IBM and Real-Time buses
Microprogram Control
32 Kbyte buffer memory
Fast picture processing
Histogram Generator

Filter Processors
Optionally added Filter Processors
- Real-Time Filtration
- Real-Time Arithmetics
- 3 x 3 window for Real-Time Convolution, etc.

MEMORIES

Video Memory
Three independent memory blocks with video-rate acess
3 x 512 x 512 byte capacity
Graphics and text overlay facilities
Pan, Zoom, Shift operations

Bulk Memory
High capacity Picture and Data Store
Capacity: 16 images of 8 bits or
8 images of 16 bits
Up to 4 Bulk Memories (16 Mbytes)
Direct connection to Real-Time bus
Flexible microprogramable addressing

IPW SOFTWARE

IPW SOFTWARE STRUCTURE

USER INTERFACE
PROGRAM LIBRARY
BASIC SYSTEM LIBRARY

USER INTERFACE
MS-DOS based flexible menu system

PROGRAM LIBRARY
Interactive display of pixel-data
Statistical Data calculations
Geometrical transformations
Filter functions, Median filtering
FFT, Filtering in frequency domain
Classifications

BASIC SYSTEM LIBRARY
Image input/output operations
Image Memory Mapper Handling
Camera control functions
Lookup Table Management
Data and Image Transfer functions
General purpose keyboard management
Display management

IPW APPLICATIONS

Industrial Quality Control
Object recognition
Medical sciences
Environment and Resource
Life Sciences
Microscopy
Agriculture and Meteorology

For further information please turn to:
Central Research Institute for Physics • **Raster Ltd.** 1525 Budapest Pf.49 Hungary
Phone: (36) 1 169-9499/1698 telex: 22-4289