tanulmányok

206/1988

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest

Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete
Computer and Automation Institute Hungarian Academy of Sciences

THE SECOND HUNGARIAN

WORKSHOP ON IMAGE ANALYSIS

Budapest, June 7-9, 1988

Tanulmányok 206/1988
Studies 206/1988

A kiadásért felelős:

DR. KEVICZKY LÁSZLÓ

Szerkesztette:
Edited by:

ÁLLÓ GÉZA
CSETVERIKOV DMITRIJ

Főosztályvezető:

HELM LÁSZLÓ

ISBN 963 311 248 6
ISSN 0324-2951

RÉSZTVEVŐK LISTÁJA

ALMÁDINÉ TUBÁKOS Éva	Országos Meteorológiai Intézet Budapest, Pf. 32. 1675	H	585-511/40
ALMÁSI István	Országos Meteorológiai Intézet Budapest Tatabánya tér 15-18. 1181	H	585-511/59
AMBRÓZY György	KFKI Budapest, Pf. 49. 1515	H	699-499
ÁLLÓ Géza	SZKI Budapest, Donáti u. 35-45. 1015	H	154-401
BALOGH Zoltán	HTSZ Budapest, Pf. 268. 1519	H	
BANGÓ György	KFKI Budapest, Pf. 49. 1525	H	699-499
BARNA Péter	KFKI Budapest, Pf. 49. 1525	H	699-499 /1813 m.
BERTA Sándorné	Posta Kisérleti Int. Budapest, Pf. 2. 1456	H	471-560
BOGSCH Imre	HTSZ Budapest, Pf. 268. 1519	H	
BOZÓ Pál	Országos Meteorológiai Intézet Budapest, Pf. 32. 1675	H	588-081
BREUER Pál	MICROSYSTEM Budapest, Lenin krt. 77.	H	121-479

BÜTTNER György	Földmérési Intézet Budapest, Pf. 546. 1373	H	636-670
CSEKE István	KFKI Budapest, Pf. 49. 1525	H	699-499
CSETVERIKOV Dmitrij	MTA SZTAKI Budapest, Pf. 63. 1502	H	811-760
CSIRIK János	JATE Szeged Somogyi B. u. 7. 6720	H	62-11-444
CSORNAI Gábor	Földmérési Intézet Budapest, Pf. 546. 1373	H	636-670
CZINEGE IMRE	Bánki Donát Főiskola Budapest Népszinház u. 8. 1081	H	141-438
DOMBAH Ferenc	Országos Meteorológiai Intézet Budapest, Pf. 32. 1675	H	588-081
Jan-Olof EKLUNDH	Royal Inst. of Technology S-100 44 Stockholm 70, Sweden	S	(8) 787-7000
FARKASNÉ NAGY Olga	HTSZ Budapest, Pf. 268. 1519	H	
FAZEKAS ZOLTÁN	KFKI Budapest, Pf. 49. 1525	H	699-499
FEKETE Attila	HTSZ Budapest, Pf. 268. 1519	H	
FERÓ László	SZKI Budapest Donáti u. 35-45. 1015	H	350-180 /191 m.

FISER József	BME EET Budapest, Goldmann Gy. tér 1-3. 1111	H	
GÁRDOS Miklós	KFKI Budapest, Pf. 49. 1525	H	699-499
Erik GRANUM	Aalborg University Badehusvej 23, DK-9000 Aalborg, Denmark	DK	(45 8) 138761
HEGEDÜS GY. Csaba	SZKI Budapest, Iskola u. 16. 1011	H	154-401
HORNJÁK Gábor	SZÁMALK Budapest, Városmajor u. 47/b. 1122	H	668-011 /205 m.
HORVÁTH TIBOR	ÁSH KFT Budapest, Rácz Aladár u. 7. 1121	H	
JURICSKAY István	POTE I.sz. Belgyógy. Klinika Pécs, Pf. 99. 7643	H	72-24-122 /310 m.
KAPOVITS Albert	Országos Meteorológiai Intézet Budapest, Pf. 32. 1676	H	481-130
KELEMEN Dezső	SZKI Budapest, Iskola u. 16. 1011	H	154-401
KELLER Zoltán	HTSZ Budapest, Pf. 268. 1519	H	
KENDE Péter	HTSZ Budapest, Pf. 268. 1519		
KETSKEMÉTY László	Országos Meteorológiai Intézet Budapest, Pf. 32. 1675	H	813-500 /2509 m.
KOCSIS Éva	SZOTE Szeged, Dóm tér 9. 6701	H	62-12-731

KONRÁD Klára	SZKI Budapest, Iskola u. 16. 1011	H	154-401
KOPHÁZI József	SZKI Budapest, Donáti u. 35-45. 1015	H	350-180 /110 m.
KREM Alajos	SZKI Budapest, Donáti u. 35-45. 1015	H	350-180 /191 m.
KUBA Attila	JATE Szeged Árpád tér 2. 6720	H	62-21-022
KÁRPÁTI László	ATE Keszthely Deák F. u. 57. 8361	H	11-299
KÖVÉR Béla	HTSZ Budapest, Pf. 268. 1519	H	
KROPATSCH Walter	Forschungsgesellschaft Johanneum Ges.m.b.H. Wastiangasse 6 A-8010 Graz, Austria	A	
LERCH Attila	MTA SZTAKI Budapest, Pf. 63. 1502	H	811-760
LESZTÁK Sándor	Országos Meteorológiai Intézet Budapest, Tatabánya tér 15-18. 1182	H	588-081
MAGYAR Csaba	MTA SZTAKI Budapest, Pf. 63. 1502	H	811-760

MARÓTI Tamás	KFKI Budapest, Pf. 49. 1525	H	699-499 /1813 m.
MISKOLCZI János	KFKI Budapest, Pf. 49. 1525	H	699-499
OLÁH Ferenc	HTSZ Budapest, Pf. 268. 1519	H	
PAPP Éva	HTSZ Budapest, Pf. 268. 1519	H	
PETŐCZ István	HTSZ Budapest, Pf. 268. 1519	H	
PINTÉR Ferenc	Országos Meteorológiai Intézet Budapest, Tatabánya tér 15-18. 1182	H	585-511
PINZ Axel	Inst. f. Vermessungswessen u. Ferulkunderung Peter Jordan Str. 82. A-1190 Wien Austria	A	
PYATKIN Valery	Computing Center, Sibirian Division of the USSR Acad. Sci., Novosibirsk, USSR	USSR	
RÉNYI István	KFKI Budapest, Pf. 49. 1525	H	699-499
RÉTI Tamás	ITI Budapest, Pf. 29. 1581	H	
RÉVÉSZ Gábor	VIDEOTON Budapest, Krammer S. u. 3. 1202	H	763-733 /218 m.
SOLT Péter	MTA SZTAKI Budapest, Pf. 63. 1502	H	811-760

STASZNY Gábor	SZKI Budapest, Iskola u. 16. 1011	H	154-401
SZABÓ Imre	KFKI Budapest, Pf. 49. 1525	H	699-499
SZABÓ József	SZKI Budapest, Iskola u. 16. 1011	H	154-401
SZIRÁNYI Tamás	VIDEOTON Budapest Vöröshadsereg u. 54. 1021	H	763-733 /344 m.
SZÉKELYNÉ N. Éva	HTSZ Budapest, Pf. 268. 1519	H	
UTASSY Sándor	Kandó Kálmán Villamosip. Főiskola Budapest, Tavaszmező u. 17. 1084	H	335-530
VADÁSZ Vilmos	Országos Meteorológiai Intézet Budapest, Tatabánya tér 15-18. 1182	H	588-081
VERESS Gábor	BME lt. Kémiai tanszék Budapest, Szt. Gellért tér 4. 1111	H	450-300 /1594
ZÖLD Sándor	KFKI Budapest, Pf. 49. 1525	H	699-499

C O N T E N T

SOLT, P.: Industrial picture processing at the INDUSTRIAL HANDLING 88 International Fair 13

MAGYAR, Cs.: A fuzzy nearest neighbor clustering algorithm 15

SZIRÁNYI, T.: Statistical pattern recognition of low resolution pictures 23

HEGEDÜS, Gy.Cs.: The PRIMA (PRoper IMage Analysis) general purpose image processing system 25

STASZNY, G.: A medium-power image processing equipment 29

KONRÁD, K.: Some image processing algorithms realized on parallel processors 33

MISKOLCZI, J. - I. RÉNYI - I. BOGSCH - A. FEKETE - B. KÖVÉR - É. NIKODÉMUSZ: Hardware architecture of the ATLAS-90 interactive image processing display system 39

FARKAS, O. - F. OLÁH - É. NIKODÉMUSZ - I. BOGSCH: Precision image digitizer equipment 47

CSEKE, I. - I. SZABÓ - F. VAJDA: Microprogrammed FFT coprocessor for the ATLAS-90 49

AMBRÓZY, Gy. - I. CSEKE, Z. FAZEKAS - J. PONGRÁCZ - S. ZÖLD - Z. BALOGH - Z. KELLER: Overview of the ATLAS-90 image processing software system 53

ALMÁDI, I.: Kvázipoláris müholdak képeinek földrajzi azonositása 57

DOMBAI, F. - A. KAPOVITS - L. KETSKEMÉTY - T. SZENTIMRE T. SZENTIMREY: Tracking meteorological objects on digital radar images 61

KOCSIS, É. - L. DUX - A. KUBA - F. GUBA: Computer analysis of the structure of the Ca^{2+}-ATPase ENZYME 63

EKLUNDH, J.O.: On shape representation in two dimensions 67

CHETVERIKOV, D.: Detecting defects in texture 75

BANGÓ, Gy. - J. GÁRDOS - J. MISKOLCZI - I. RÉNYI: IPW: an Image Processing Workstation 81

GRANUM, E. - H.I. CHRISTENSEN: Methods for real time motion analysis 85

PINZ, A.: A remote sensing application for a computer vision system 87

BÜTTNER, G. - T. HAJÓS - M. KORÁNDI: Interactive program package for satellite image processing 89

GRANUM, E. - H.I. CHRISTENSEN: Methods for real time motion analysis 95

FISER, J. - A. LERCH: Experiments with the recursively implemented Canny edge detector 109

KROPATSCH, W.G.: Pyramid research for image analysis ... 115

CSEKE, I.: An error-correcting prewitt edge detector in noisy images 119

CZINEGE, I. - T. RÉTI: Quantitative characterization of orientation in binary digitized images 123

CSORNAI, G.: Image analysis methods in the agricultural applications of remote sensing 129

PYATKIN, V.P.: Lineaments detection in aero-space imagery 135

Industrial picture processing at the INDUSTRIAL HANDLING 88 International Fair

Peter Solt

Computer and Automation Institute,
Hungarian Academy of Sciences,
Budapest, POB.63, H-15o2 Hungary

Industrial picture processing equipment is in the center of interest since this equipment is commercially available. The presentation briefly summarizes the typical hardware and software components emphasizing the possibility of standardization at certain points. The picture processing techniques are compared with the discrete sensor technology from a flexible assembly system designer point of view. Picture processing equipment from Switzerland, Sweden, Germany, USA, etc is presented. The main capabilities and the applications are summarized. There were a lot of applications demonstrated at the exhibition. From these one can estimatestate of the industrial vision technology at present and the future trends. A Hungarian firm exhibited a part assembly table for small axis equiped with a vision system. The task and the role of the vision in that assembly cell is described in details as well.

A Fuzzy Nearest Neighbor Clustering Algorithm

Csaba Magyar

Computer and Automation Institute,
Hungarian Academy of Sciences,
Budapest, POB.63, H-1502 Hungary

1. Introduction.

Classification of objects is an important area of research and it has special importance in pattern recognition and data analysis. There have been numerous techniques investigated for classification. Many techniques rely on some notion of similarity or distance in feature space. Nearest neighbor (NN) rules are widely used to perform classification [5]. The real advantages of these methods are their computational simplicity and the good results obtained by their use in many problems of small sample size [6].

One of the drawbacks of using a k-NN classifier is that normally each of the k sample vectors is considered equally important in the assignment of the class label to the input vector.

A similar problem arises when utilizing hard or crisp clustering techniques. In many cases of clustering partitioning the input vectors into mutually exclusive sets decreases the amount of information about the data structure.

To cope with these problems one can utilize fuzzy set theory. One of the best-known algorithms is the fuzzy ISODATA or fuzzy c-means algorithm, which was initially developed by Dunn [7] and generalized by Bezdek [2,3,4]. It has numerous applications in pattern recognition and in data analysis.

In this paper a fuzzy 1-NN classifying technique and an iterative fuzzy clustering algorithm are presented. The latter can also be considered as a fuzzy generalization of the hard c- means process of Ball and Hall [1]. It assignes memberships to data vectors in each cluster according to their distances from the cluster centroids. Then the centroids are recalculated and the process is being continued unless some terminating criterion is satisfied. The clustering algorithm is compared with that of Bezdek via cluster validity measures. It is shown that with certain parameter value the execution time of the algorithm is less than that of the fuzzy c-means process although it keeps all the advantageous properties of the ISODATA algorithm.

2. Fuzzy Nearest Neighbor Clustering Algorithm (FNNC)

The nearest neighbor classification rule assigns an input vector x, which is of unknown classification, to the class of its nearest neighbor.

The hard c-means algorithm [1,3] is based on finding an appropriate hard c-partition [3] of the data set through iterative recalculation of memberships, using nearest neighbor decision.

The drawback of the algorithm is that it partitions the data set into c mutually exclusive, collectively exhausted subsets. In many cases this decreases the amount of information about the data structure. To cope with these problem one can utilize fuzzy set theory. Fuzzy sets are derived by generalizing the concept of a characteristic function to a membership function u: U -> [0,1]. Most crisp set operations (such as union and intersection) and set properties have analogs in fuzzy set theory. The advantage provided by fuzzy sets is that the degree of membership in a set can be specified, rather than just the binary "is or

isn't member". This can be especially advantageous in pattern recognition, where objects frequently not clearly members of one class or another.
In the following the definition of fuzzy c-partition is given.

Definition 1. **Fuzzy c-partition**

X is any finite set; V_{cn} is the set of real c*n matrices; c is an integer 2<=c<=n. Fuzzy c-partition space for X is the set

$$M_{fc} = \{ U \in V_{cn} \mid u_{ik} \in [0,1] \; \forall i,k; \; \sum_{i=1}^{c} u_{ik} = 1 \; \forall k; \; 0 < \sum_{k=1}^{n} u_{ik} < n \; \forall i \}$$

Row i of matrix $U \in M_{fc}$ exhibits (values of) the ith membership function (or ith fuzzy subset) u_i in the fuzzy c-partition U of X. Because each coloumn sum is 1, the total membership of each x_k in X is still 1, but since $0 <= u_{ik} <= 1 \; \forall \; i,k$, it is possible for each x_k to have otherwise arbitrary distribution of membership among the c fuzzy subsets $\{u_i\}$ partitioning X.

In order to give a fuzzy generalization of the nearest neighbor decision rule one can take $U \in M_{fc}$ (a fuzzy c-partition) instead of $U \in M_c$ (a hard c-partition). Let $x_j, x_k \in X$, x is an arbitrary vector to be classified, d is a metric. x_k is the sample vector for which:

$$d(x, xk) = \min d(x, x_j)$$

x_k is required to dominate in determining the memberships of x. To obtain this one can choose a weight function f for which

(i) $f(x, x_j) \to 1$ if $d(x, x_j) \to 0$

and

(ii) $f(x, x_j) \to 0$ if $d(x, x_j) \to$

For example such an f may be

$$f(x, x_j) = 1 / (1 + d^2(x, x_j))^{\alpha}, \quad \alpha >= 1$$

This f has some computational advantage, that is, it is not necessary to compute time consuming square roots. Now to satisfy the condition $\sum_{i=1}^{c} u_{ij} = 1$, one can get

$$(*) \quad u_i = \frac{\sum_{j=1}^{n} u_{ij} / (1 + d^2(x, x_j))^{\alpha}}{\sum_{j=1}^{n} 1 / (1 + d^2(x, x_j))^{\alpha}} \qquad \alpha >= 1, \; i=1,\dots,c$$

In the following the fuzzy nearest neighbor clustering algorithm (FNNC) is given, based on the previous idea. This method can also be considered as a fuzzy generalization of the hard c-means process.

<u>**Algorithm 1:**</u> **FNNC**

1. Fix the number of clusters c, $2 <= c <= n$, where n = number of data items. Fix $\alpha >= 1$. Choose any inner product induced norm metric d, e.g. the Euclidean distance.

2. Initialize the fuzzy c-partition $U^{(0)}$.

At step l, l=0,1,2,...

3. Calculate the c cluster centers $\{v_i^{(l)}\}$ with $U^{(l)}$ and the formula for the ith cluster center is:

$$v_{ij} = \sum_{k=1}^{n} u_{ik} x_{kj} \Big/ \sum_{k=1}^{n} u_{ik} \qquad j = 1,2,\dots,p \quad (x_k \text{ has p coordinates})$$

4. Calculate the memberships of the cluster centers and the formula for the membership of v_j in cluster i is:

$$(**)\quad \mu_{ij} = \mu_i(v_j) = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{otherwise} \end{cases}$$

5. Update $U^{(l)}$; calculate the memberships in $U^{(l+1)}$ with ($\mu_{ij}{}^{(l)}$) , { $v_i{}^{(l)}$ } :

$$(*)\quad u_{ij}{}^{(l+1)} = \frac{\sum_{k=1}^{c} \mu_{ik} \,/\, (1 + d^2(x_j, v_k))^{\alpha}}{\sum_{k=1}^{c} 1 \,/\, (1 + d^2(x_j, v_k))^{\alpha}} \qquad \begin{array}{l} \alpha >= 1 \\ j = 1,\ldots,n \\ i = 1,\ldots,n \end{array}$$

6. Compare $U^{(l)}$ and $U^{(l+1)}$ in a convenient matrix norm; if $\| U^{(l)} - U^{(l+1)} \| < \varepsilon$, stop; otherwise set $l=l+1$ and go to step 3.

The idea of the above algorithm is the following. If an initial fuzzy c-partition is given then the c cluster centers can be computed. Then memmberships are assigned to these centroids. At the next step the memberships of the sample vectors are updated using (*). This means that the cluster center $x_k \in X$ is to the ith cluster center the larger the membership of x_k is in the ith cluster.

3. Results

In this section some results are given comparing the FNNC algorithm with the widely used fuzzy ISODATA process [3]. To illustrate the performance of the methods two cluster validity measures were examined: the partition coefficient and the partition entropy [3].

The first is given by the formula:

$$F(U,c) = \sum_{i=1}^{c} \sum_{k=1}^{n} u^2{}_{ik} \,/\, n$$

while the second is defined as

$$H(U,c) = -\sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} \log_a u_{ik} \,/\, n$$

where U is a fuzzy c-partition. In the example a was chosen as the natural logarithmic base. F and H measures the "fuzziness" of the partitioning. With increasing F the partition becomes "harder", and also this is the case with

decreasing values of H. Interesting properties of F and H can be found in [3].
In Table 1. and Table 2. a two cluster data structure was used as a data set. From the tables it can be seen that with increasing values of α the FNNC algorithm is getting more similar to the hard c-means algorithm. It can be also noticed that with certain α and m parameter values both of the processes terminate with very similar F and H values. The choice of α depends on how "hard" clusters are required. To exploit the advantages of fuzziness it is reasonable to choose a relatively low α, but with increasing α the number of iterations and hence the execution time decreases. Considering the previously mentioned and from examinining numerous test data structures, $\alpha=2$ seems to be the most appropriate choice. This parameter value has other computational advantages too. With this α the FNNC process needs less execution time than the ISODATA algorithm with the most commonly used m=2 parameter value, but in this case the FNNC algorithm results a somewhat harder clustering than the other method. However, it preserves the fuzziness of the data structure, that is, the data vectors which belong to more than one cluster get relatively high memberships in each of these clusters.
From randomly generated initial partitioning it can be observed that both of the FNNC and the ISODATA algorithms are quite stable to the initialization.
The choice of the termination parameter ε was also under investigation. From Table 3. it can be seen that with decreasing ε only the execution time increases while the values of the cluster validity measures remain nearly the same. Therefore it is not necessary to choose a very strict termination criterion.
Finally it can be concluded that the FNNC algorithm possesses all the advantageous properties of the fuzzy c-means process, besides with $\alpha = 2$ parameter value it requires less computational time than the other algorithm with the most

commonly used m = 2 value.

```
*                 *
*  *           *  *
*  *  *  *  *  *  *
*  *           *  *
*                 *
```

Figure 1.
Sample structure with two clusters

α	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
PC	.829	.879	.910	.930	.944	.953	.959	.964	.967	.969
PE	.304	.228	.174	.137	.110	.090	.076	.065	.058	.052

Table 1. FNNC algorithm: Partition coefficient and partition entropy, ε = 0.01

m	1.2	1.4	1.5	1.6	1.8	2	2.2	2.4	2.6	3
PC	.974	.968	.959	.944	.905	.860	.816	.776	.741	.685
PE	.037	.053	.076	.107	.179	.251	.317	.372	.419	.489

Table 2. Fuzzy ISODATA algorithm: Partition coefficient and partition entropy, ε = 0.01

ε	.001	.005	.01	.05	.1
IT	7	7	6	6	6
PC	.944	.944	.944	.944	.944
PE	.109	.109	.110	.110	.110

Table 3. Number of iterations, partition coefficient, partition entropy with different termination parameters. (FNNC, α = 2)

REFERENCES

[1] G.Ball and D.Hall, "A clustering technique for summarizing multivariate data", Behav. Sci., Vol. 12, pp. 153-165, 1967

[2] J.C.Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithhms", IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-2, pp. 1-8, 1980

[3] J.C.Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York: Plenum Press 1981

[4] R.L.Cannon, J.V.Dave, J.C.Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms", IEEE Trans. Anal. Machine Intell., vol. PAMI-8, pp. 248-255, 1986

[5] T.M.Cover and P.E.Hart, "nearest Neighbor pattern classification", IEEE Trans. Inform. Theory, vol. IT-13, pp. 21-27, 1967

[6] B.V.Dasarathy, "Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environment", IEEE Trans. Anal. Machine Intell., vol. PAMI-8, pp. 248-255, 1986

[7] J.C.Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", J. Cybernet. 31 (3), pp. 32-57, 1974

[8] E.Ruspini, "Numerical methods for fuzzy clustering", Inf. Sci., vol. 2, pp. 319-350, 1970

MTA SZTAKI Tanulmányok 206/1988 p.23.

STATISTICAL PATTERN RECOGNITION OF LOW RESOLUTION PICTURES

Tamás SZIRÁNYI

VIDEOTON Development Institute
H-1021 Budapest, Vörös Hadsereg utja 54, HUNGARY

This pattern recognition program has been developed for a sensing array table of low resolution.

This easy-to-make, cheap and simple pyroelectric sensing station has 16x16 sensing elements.

A pattern recognition computer program was development to recognize an object that is on this sensing array table. The sensing element's picture of an object may be broken into several parts and may have additive noises, too.

The object recognition procedure

- connects the picture to get a shape that is similar to the original object silhouette,
- filters noises from the picture,

- obtains measurement values that can be used to characterize the picture and to recognize the object on the sensorized table,
- calculates the most probable object from the feature values.

The first step is the picture reconstruction. It involves the connectivity analysis together with the noise reduction of the original picture. From the reconstructed picture, an edge map is obtained by the bug-following. From the reconstructed picture and the edge map, 21 feature parameters are acquired.

In the teaching period mean values and variances of feature parameters are calculated for every sample object.

During the recognition procedure, from the feature parameters of an object in question Euclidian error-distances are calculated between every sample's feature vector and the object's feature vector in the 21 dimensional space.

Assuming that the probability density of these error-distances has a normal distribution, a nearly exact degree of probability of each object and the reliability of the recognition decision can be calculated.

Effects of different types of picture noises are calculated and compared.

The PRIMA (PRoper IMage Analysis) general purpose image processing system

Gy. Csaba Hegedűs
Computer Research and Innovation Center

H-1251 Budapest, Pf.: 19.

The Mathematical Laboratory of SzKI (Computer Research and Innovation Center) has been engaged in developing general and special purpose image processing systems for about ten years.

Nowadays new hardware technology allows image processing to become widely available in many scientific and industrial applications. According to our experiences,demand for image processing systems based on professional personal computers (PPCs) has greatly increased in the last period. Several facts give reasons for this, e.g.:

- computers of this category have extended processing power and they are available at a decreasing price in the domestic market, no matter whether they are imported or produced in the country;
- small start-up capital is needed to buy them and they can be used for other purposes as well;
- they are easy to install and to remove if necessary.

Taking the requirements into consideration we started the development of PRIMA image processing system based on IBM IBM-PC/XT/AT or compatible PPCs in 1986. Our basic aims were:

- to develop a range of products covering the possible widest requirements of digital image processing;
- to facilitate generation of particular systems for special purposes from the subsystems of PRIMA;
- to apply enhanced algorithms;
- to develop user-friendly but efficient communication methods;
- to integrate into PRIMA the most frequently used general purpose and image processing peripherals;
- to do research work on practical application procedures (basic image processing functions to be integrated into complex algorithms) and to realize the results as macros in PRIMA.

PRIMA Version 1.4 (consisting of the kernel and the most frequently used functions) is now complete, the enhancement work has already started. The main features of the system are as follows:

- efficiency
  - = the set of algorithms is composed as a result of many year experiences in digital image processing;
  - = the system structure allows chaining of either pixels or parts of a frame or frames, resulting in better exploitation of single processor architecture (there are fewer I/O transfers needed); there is a facility to expand the system with additional processors;
  - = adaptive (e.g. context sensitive) algorithms were used whenever possible;
  - = overflows during arithmetic and image manipulating operations are automatically corrected by the system, retaining the maximum possible visual information at the same time;
- user-friendlyness
  - = functions can be activated either by commands like high level instructions or by using mouse controlled menu;

= powerful command-editor is implemented;
= illegal commands are rejected by the command interpreter;
= in addition to the USER'S GUIDE, various menues and help lists are provided to assist the user;
= parameters of the most commands can be given via visual control on the display;
= effects of commands can be modified by means of switches;

-both interactive and multi-level batch mode of operation are possible;

-modular structure: the function set is easily extendable or adaptable to any special task;

-large scale of applicable peripherals (e.g. CDP, PCVISION, PCVISION plus, PROVISION - see later -, DT 2851/53, NRAT5810 frame grabbers; FX-105, HP Laserjet, Colorjet hard copy units);

-the system can be configured to the given hardware environment with setup procedure.

Major functions of Version 1.4. are as follows (note that functions in the system can partly overlap or can be chained, so, it is difficult to list them separately):

- capture image from camera; transfer images or selected parts between any two window (a window can be defined in any size on any possible peripheral device);
- remap images by hardware and software look-up tables,
- define, save, display or delete any table (look-up tables, windows, convolution matrices);
- calculate statistics, display the results in numerical or in graphical form, or use them for computing special look-up tables;
- arithmetic and logic operations between windows or with constants;
- automatic and user-defined scaling;
- convolution, background equalization, RANK operations;
- graphical functions (polygon drawing and painting, entitling);
- creating and running stored image processing programs (MACROs).

The system can be enhanced with using subsystems for special purposes. Main areas of possible applications:

- medical-biological analysis of microscopic pictures;
- material structure testing;
- analysis of infrared photographs or X-ray images;
- analysis of air-photographs and multispectral images.

The PRIMA system - as it was mentioned above - can handle the PROVISION frame grabber, that is a single-board unit developed by the SzKI.

The first part of the frame grabber accepts analog video signal from a TV camera or video recorder, and converts it to digital data. Once in digital form, data are stored as a 512x512 matrix of pixels.

Each pixel is represented on 8 bits providing 256 levels of either grey tones or colours, respectively.

Depending on the requirements, multiple frame grabbers can be used to expand the memory up to 24 bits/pixel (true colour configuration). The stored image can be accessed by the microcomputer for analysis or modification. The frame grabber's memory is directly addressable from the PC, in 64 KB blocks. The contained image can be displayed continuously on colour monitor. Both on the input and on the output side of the frame grabber hardware, look-up tables allow coding of data.

MTA SZTAKI Tanulmányok 206/1988 p. 29

A Medium-power Image Processing Equipment

Gábor Staszny
Computer Research and Innovation Center
H-1251 Budapest, Pf.: 19.

Introduction

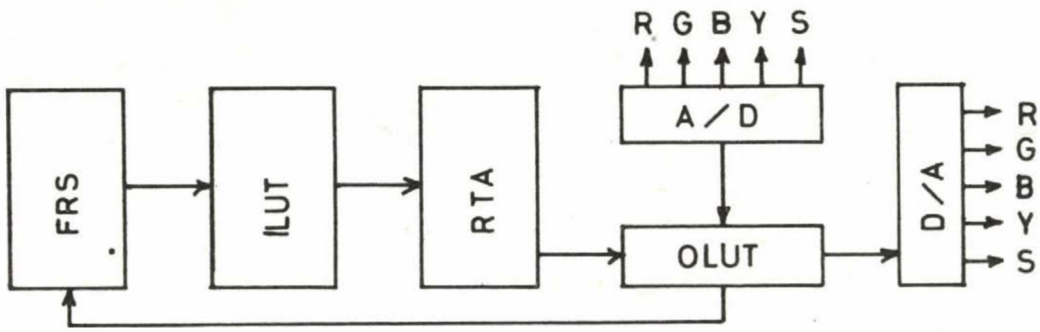
For solving the image processing tasks at the SzKI, we have developed a display processor device (SKM) being able to process true-colour images, as well.
Its basic functions are as follows:

- digitizing of TV-standard videosignals given to its analog Input;
- storing and displaying digital images;
- fast operations between image planes.

Connecting the SKM to a 32 bit IBM/AT or compatible PC (e.g. PROPER-132), we can create a medium-power image porcessing equipment.
It can be used under control of a multitask graphic monitor program.

Architectura

The block-scheme of SKM is shown in the figure below.



The frame-store (FRS) consists of up to 8 pieces of 1 MB/8 bit image-planes, each of them controlled by a HD6348 type graphic processor which itself realises certain hardware functions (scroll, pan, cursorhandling, montage by masks, etc.), as well. A maximum size image consists of 1024^2pixels and 8 or 16 bits/pixel, optionally.
In the image plane, one can arbitrarily define the top left corner and the dimensions of an image.

The real-time arithmetic unit (RTA) is a two-level arithmetic network. Any 4 of the 8 image-planes can be connected to its input simultaneously, through the input look-up-tables(ILUT). By means of it, simple arithmetic (addition, subtraction, multiplying by constant) and logical (AND, OR, NEG) operations can be performed with video-speed.

The results of these operations can be rewritten to any non-input image-plane through the output look-up-tables (OLUT). At the same time, they are sent to the input of the D/A couverters, thus,the results can be seen on the screen of the image monitor.

If an input signal comes from a video-signal source, it will be digitized by the A/D converters.

The digital image itself will be stored in the frame-store through the OLUT which acts as an input-look-up table, in turn.
Accuracy of digitization is 3x4 bits in case of true-colour images and 8 bits in case of black-and-white or narrow-band images, respectively.

Graphic monitor program

The monitor program can be used by several users simultaneously. Its main parts are as follows:

- handlers for realizing multitask environment;
- handlers for partitionning the frame-store;
- image processing subroutines (handling LUT-s and cursors, graphic functions, etc.);
- control programs of RTA

The monitor automatically executes swapping, handles queues and assigns resources to users. It keeps connection with the operating system of the PC, as well.

The image processing equipment can be connected to a host computer, as a whole; in these cases monitor simulates an intelligent terminal.

Real-time arithmetic unit

ILUT and OLUT play an important role in operating of RTA. Filling them properly in, we can compose complex arithmetic operations from the simple ones.

On the first level, it performs operations between two pairs of its 4 inputs, on the second level, in turn, between the two results coming from the first one. On both levels, it works with 12 bits; on the first level, these are converted from the input 8 bits by ILUT.

As an example, convolution filtering with a 3x3 mask consists of 4 LUT-filling in and 4 arithmetic operations which take 320 ms, altogether.
The same operation with a 5x5 mask is performed in 24 steps, wich consumes 960 ms. (In both cases, we suppose that the absolute value of either the mask elements or their sum is not greater than 16.)

Summary

SKM serves for capturing, storing and displaying true-colour and black-and-white images, as well as, for performing elementary operations between them in real time.
By connecting it to a PC, we can build up a medium-power image processing equipment.

Optionally, such a system can be attached any further peripherals (plotter, video tape recorder, etc.) to.
Among others, we find important to mention two devices developed in the SzKI. One of them is a 16 MB capacity double port store, the second one is a very fast multiprogrammable floating-point processor. The auxiliary store can be accessed from the PC by pageing while from the fast processor in continuous addressing mode.

Some Image Processing Algorithms Realized on Parallel Processors

Klára Konrád
Computer Research and Innovation Center (SZKI)

H-1251 Budapest, Pf.:19 .

The colour display processor is necessary for solving image processing problems. At the beginning, this special peripheral device served displaying digital images but nowadays it became a true processing unit with own processors and real-time arithmetics.

SMODUL is a group of image processing algorithms, which has been worked out in the Mathematical Laboratory of SZKI for a parallel processor with the logical structure shown in Fig.1.

The equipment contains lo image-memories. By means of three LUT-s and line-masks, arithmetical and logical operations can be performed among some of them.

The image processing functions of SMODUL are as follows:

- part selection with polygons or rectangles;
- histogram calculation using masks;
- finding inside points of polygons;
- classification according to various parameters (colour-maps, neighbourhood, etc.);

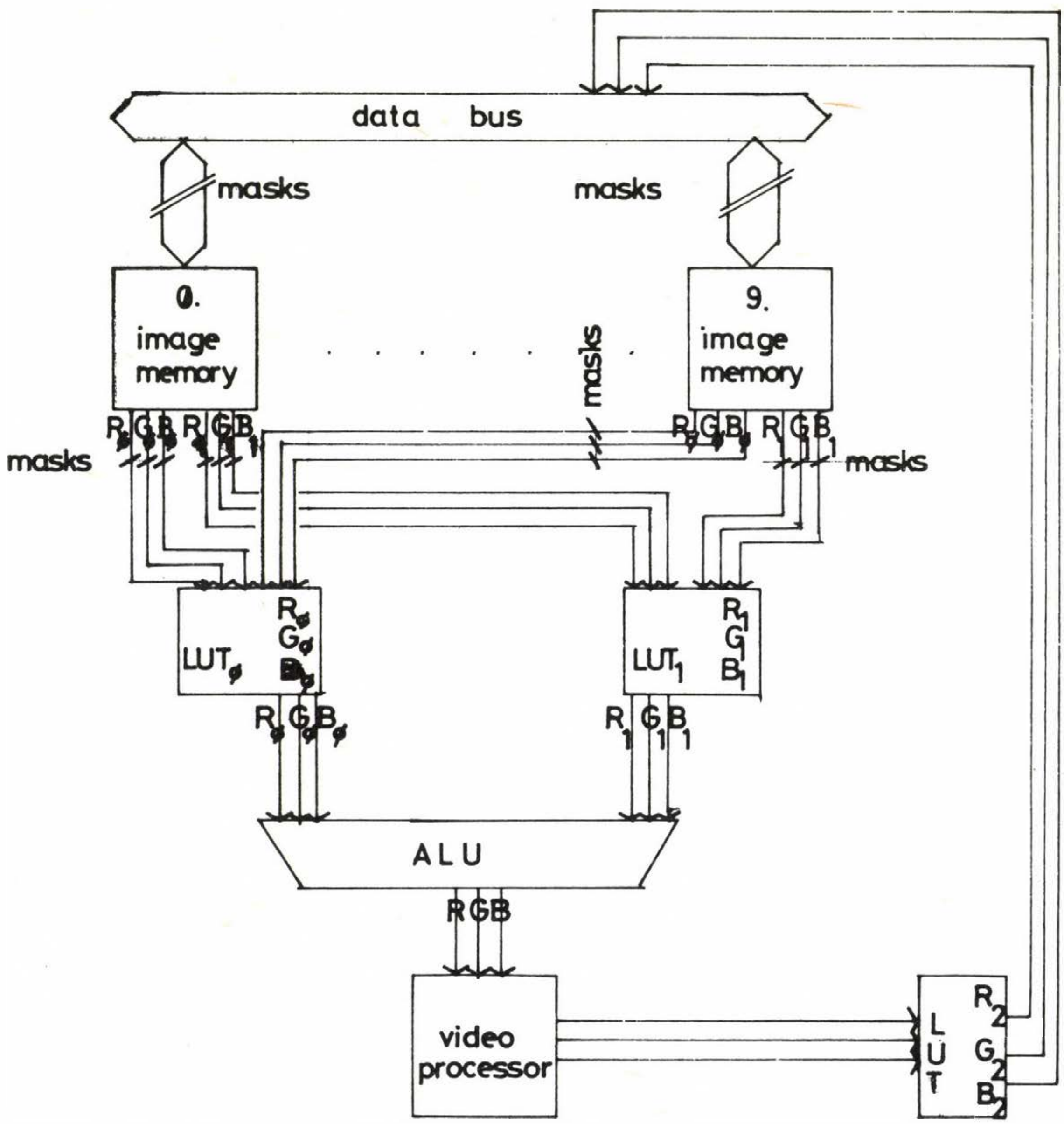

data bus
masks
masks
0.
image
memory
9.
image
memory
masks
masks
masks
LUT$_0$
LUT$_1$
ALU
RGB
video
processor
LUT
R_2
G_2
B_2

Fig. 1.

- calculation of geometrical parameters (diameter, perimeter, area, etc.);
- erosion and dilatation;
- recolouring according to colour-classes or neighbourhood.

Real-time arithmetical and logical operations among images memories were used for realizing all these functions. This possibility is the most important advantage of parallel processors, because in this way the execution time can be reduced drastically.

In the following, we dwell in three algorithms for demonstrating use and advantage of parallel image processors and at the same time we speak about differences between parallel and serial data structures, as well.

1.) Calculation of spot area (or histogram)

Let us suppose that there exists a classification $\{Q_i\}$ of possible colour codes q ($\emptyset \leq q \leq 1023$), so that object in a "natural" image stored in image memory K_1 belongs to class C_i if and only if colour codes $q(k,\ell)$ of its pixels $P(k,\ell)$ belong to Q_i :

$$M:P(k,\ell) \longmapsto C_i \,\big|\, q(k,\ell) \in Q_i \qquad (i=1,2,3,\ldots)$$

We define the element ℓ_j of a look-up-table LUT* according to this equation :

$$\ell_j = \begin{cases} 1, & \text{if} \quad j \in Q_j \\ \emptyset, & \text{otherwise} \end{cases} \qquad (j=\emptyset,1,2,\ldots,1023)$$

At first, we read K_1 into K_2 image memory ($K_1 \neq K_2$) through LUT* ; then we have a two-level image in K_2 with a possible size $2^n x 2^n$.

The number of "1"-s in K_2 can be calculated by using the roll-functios of the videoprocessor which shifts circularly the rows (or columns) of an image with given number of pixels.

Let we roll the image with 2^{n-k} positions repeatedly ($k=\emptyset,1,2,\ldots,n-1$), and add the rolled images to the original one. (During addition LUT-s must be filled in identically.) Then the sum of "1"-s in certain row or column will appear in its first element.

We can leave the procedure at any step. The number of steps is limited by the capacity of memory cells to one pixel.
E.g. in case of n=8, 8 rolls and 8 additions are needed, supposing one memory cell contains 9 bits.
In this case execution time is:

(I) $16 \times T$

where "T" is refreshing time which depends on the processor.

Trying to solve this problem on a serial machine and supposing at least 4 operations and 2 read instructions for every pixel, the required execution time is :

(II) $2^{n+n}(2^2+2) \times t$

where "t" is average operation time of machine.
As a reality, we can calculate with n=8, T=20ms and $t=2.1o^{-6}$s. In this case, we get the following results:

(I/a) $16 \times 2o\text{ms} = 32o \text{ ms} = 0{,}32 \text{ s}$ (on parallel proc.);

(II/a) $2^{8+8} \times (2^2+2) \times 2.1o^{-6}\text{s} = 0{,}79 \text{ s}$ (on serial proc.).

It can be seen that the previous solution is shorter.
In case n 8 the time gaine is more obvious.

2.) <u>Erosion and dilatation</u>

Let us suppose that we have a two-level image. We call objects disjoint connected sets of pixels each of them belonging to the same class.Erosion or dilatation procedures are based border points. Border points are defined as pixels of an object having at least one neighbour not belonging to the object. More exactly, finding border points means finding pixels of an object having a given neighbourhood. Generally, the neighbourhood relations are described by a matrix, e.g. :

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

For performing such a search procedure on serial machine, we have to execute 9 indexing, conditional branching and addition operation in every pixel, respectively. In this case, execution time can be calculated from the equation:

(III) $\qquad 2^{n+n} \times 27 \times t \qquad .$

The numerical value , with the above parameters, is :

(III/a) $\qquad 2^{8+8} \cdot 27 \cdot 2.10^{-6} s = 3{,}54$ s (on serial proc.)

If we realize the erosion or dilatation algorithm on a parallel processor, we have to execute 8 shifting, max 8 additions and one remapping function. The execution time can be calculated from the equation :

(IV) $\qquad (8 + 8 + 1) \times T \qquad .$

In this case the result is :

(IV/a) $\qquad 17 \times 20$ ms = 340 ms = 0,34 s (on parallel p.)

3.) <u>Local operators (Laplace transform, convolution)</u>

Denote $q(k,\ell)$ the gray-level of the pixel with coordinates (k,ℓ) .

Can be modify this value by means of a weighting (or filtering) matrix S, according to the equation

$$q'(k,\ell) \cong \frac{1}{N} \sum_{i=-p}^{p} \sum_{j=-r}^{r} q(k+i, +j) \times s(i,j);$$

where size of matrix is $(2p+1) \times (2r+1)$, $s(i,j)$ is the element of S with coordinates (i,j) related to the centre of S furthermore:

$$N = \begin{cases} 1, & \text{if} \quad R = \emptyset \\ R, & \text{otherwise,} \end{cases} \qquad \text{and}$$

$$R = \sum_{i,j} s(i,j)$$

On serial machine, it is needed p*r multiplications and additions, as far as, one division at all pixels for filtration. Performing this function on a parallel processor it is needed p*r shifting, LUT filling in and one addition, finally , one division operations. Let we now calculate the execution time. In the serial case supposing that $t_m=4t$ (time of multiplication) and $t_d= 8t$ (time of division), the proper equation is

(V) $\quad 2^{n+n} * \left[(p*r) * (t+t_m) + t_d\right]$.

In our case with p=r=3 , it gives :

(V/a) $\quad 2^{16}.(9(1+4)+8).2.10^{-6}s = 6,94s$ (on serial proc).

On the contrary, if we use a parallel processor the execution time is given by :

(VI) $\quad (3 * p*r + 1) * T$,

that gives:

(VI/a) $\quad (27+1).\ 20\ ms = 0,56\ s$, altogether.

The advantages of parallel processors can be exhausted only if we simultaneously use as much as possible image planes, as well as, its other special features.

Summary

In this way we can state that application of parallel processor for image processing purposes can result a considerable great decrease in execution time. Though, they are not equally good in solving every task, but in some cases their advantages over serial machines are obvious. However, their efficient use required rather new art of thinking and programming.

Hardware Architecture of the ATLAS-90 Interactive Image Processing Display System

J. Miskolczi, I. Rényi
Central Research Institute for Physics,
1525 Budapest, P.O.Box 49.

I. Bogsch, A. Fekete, B. Kövér, É. Nikodémusz
Hiradástechnika Szövetkezet, 1519 Budapest, P.O.B. 268.

Advances in the design of image processing display equipment have occurred in two key areas, namely, data flow architectures and pipeline array processor architectures. *Data flow* involves the use of high speed data interfaces for the display processor, a flexible internal bus architecture and efficient access to display processor memories. Image processing power is achieved through the use of *pipeline processors* rather than high speed, large address-space CPUs. The key idea is that the value of the display in processing applications is its fast and convenient access to a large data base. This ease of access has helped the growth in processing applications, and the fact that in many cases the reduced precision used by the display is adequate for analysis by a trained observer. Hardware constraints have an impact on the design of those algorithms which take advantage of the video pixel rates and the computational power of the pipeline processors.

The ATLAS-90 Interactive Image Processing System, developed by a joint team from Hiradástechnika Szövetkezet (HT) and the Central Research Institute for Physics (KFKI), exemplifies the above architectural advances. In the first place the system is intended for processing and displaying *remotely sensed multispectral or multitemporal imagery*, but other application areas requiring fast processing and sophisticated display may also take advantage of the special features of ATLAS-90.

In order to satisfy the potentially broad base of users, each usually with a slightly different set of requirements, the following *objectives* were adopted in the design of the ATLAS-90:

- To provide a flexible data base to accomodate varying image size, precision and operational modes.
- To provide a powerful facility for interchanging data between the image memory and the processing elements and between the image memory and the external data sources.
- To provide a powerful basic processing capability and the possibility of including specialized processors for specific tasks, e.g. image warp, nonlinear filters.
- To provide hardware implementation for utilities so that they can speed up commonly used tasks in image interpretation and analysis such as histogram generation, definition of analysis and display regions by overlay masks.

The two key issues determining the ability of the system to meet these objectives are a) Memory Architectures and b) Processing Power. Following a brief overview of the basic system components, the main features of the image memory architecture will be described along with the means and methods to improve image processing and display operations. To show the present and potential effectiveness of ATLAS-90, a few examples of specific processing tasks will be given.

The basic components of ATLAS-90 are the following:

- *Controlling minicomputer* with coprocessor and conventional peripheral devices (the latter including various disks, tape drives, hardcopy and up to eight display terminals). The controller is a single board, user microprogrammable, PDP11/44 compatible processor (IPC) with an optional 64-bit Floating Point Processor (FPP) extension, both manufactured by KFKI.

- *Image Memory Subsystem* (see later).
- *Pipeline Processor Array* (see later).
- *Display Subsystem.* The processed image can be displayed in the RGB or the IHS (Intensity-Hue-Saturation) system in 512x512 format, 24 bits per pixel. Controlled by the colour assignment memory the processed image can be mixed with the output of the
  - -- graphic memory. A vector generator and four bitplanes are available which can be zoomed and scrolled,
  - -- symbol generator, producing 32x80 coloured characters on the screen,
  - -- two programmable cursor generators.
- *Specialized processors* (see later).
- *Precision image digitizer.* This - optional - equipment is used for inputting image data from various kinds of plain media, e.g. photoprints, films and maps with high precision. For details see the paper "Precision Image Digitizer Equipment" in this Proceedings.

Memory architecture is the foundation on which all of the capabilities of the system are built. Fast and efficient data routing is the goal of the memory design, in support of processing, display and data base operations. To this end, the image memory modules have 5 byte-wide, video speed (75ns/byte) ports: two times four of these are bussed to establish memory input and output lines. The fifth port of each module directly feeds the pipeline processors. Up to 16 modules are incorporated in a system, each holding two blocks of 512x512 bytes, of which one may be accessed during a TV frame time. (Note that the access bandwidth for the entire 8Mbyte image data base is 210 Mbyte/s!) Specialized processors and external equipment access the image memory via the eight I/O buses, the maximum bandwidth here exceeds 100Mbyte/s.

The memory subsystem is controlled by a microprogrammed bipolar microprocessor (MC) which, in addition to orchestrating the array under various operation modes (e.g. row or column sequential byte or word access, random access, refresh) provides "hardware" zoom (2,3,...,16) and scroll/pan facilities and performs rigorous test sequences, including signature analysis on each individual module simultaneously.

The size of a single image in the data base may range from 512x512 (multiband, if you wish) to 2kx4k (single band), of which a 512x512 pixel window is displayed. The screen can be split into four quadrants and each quadrant may show a different image with different zoom and scroll parameters. All these operations are supported by the MC.

Processing power. The key element to the processing capabilities of ATLAS-90 is the *pipeline processing channel*. Using the pipeline processing channels one can generate what will be referred to as a virtual image from a linear combination of all input channels. It is a virtual image because the image seen at the monitor does not exist in refresh memory but is constructed in the three pipelines (red, green, blue) simultaneously. A large number of algorithms can be performed using the input and output function tables, adders, range circuits, etc. of the pipeline processors. A few examples are given at the end of the paper.

The computation of an image histogram is necessary in a wide variety of algorithms. The histogram cannot be obtained by the host since the image to be histogrammed is frequently a virtual image formed in the pipeline processor. In the ATLAS-90 a 20-bit histogram is computed by the so-called

Histometer in two frame times per channel. For the computation, a Region Of Interest (ROI) memory is used to mask the image. The Histometer also measures the area of the ROI and the sum of all pixel values within the ROI.

The introduction of the *Feedback Processor* concept provides a significant advance in the processing capabilities. Previously all video speed processing was limited to a single iteration since the results must be viewed on the screen. With the feedback loop the user may store the image as seen on the monitor back into refresh memory which could then be returned to the host processor or used in subsequent processing steps. The Feedback Processor - currently under development - includes tables, comparators, adders, a multiplier-accumulator and shifter, among others. Application examples will be given later.

The "hardware processing" capabilities of ATLAS-90 can be further extended by adding *specialized processor modules*. There are two architectural features of the system which make the effective use of such processors possible:

- the high memory bandwidth achievable via the multi-operand memory I/O buses
- the flexible addressing capability of the image memory.

Until now no spec. processor module has been implemented. From the first ones considered an image warper could be mentioned. This would produce geometric transformation of up to 1kx1k images using 1.5 degree polynomials with 1/8 pixel accuracy and nearest neighbour or bilinear interpolation. The processing time for a 512x512 image using bilinear interpolation is approx. 1.3s.

The following **list of operations** gives a few examples of the ATLAS-90's capabilities, but does not include all of the power inherent in the system.

1) Addition/Subtraction: The pipeline processors (PL) can be set up to perform a weighted linear combination of images. The process can be applied to monochrome or colour images since separate PLs are available.

2) Interband Ratio: The PL can first perform a logarithmic mapping in the input tables, a subtraction and then an exponential mapping in the output tables.

3) Position Invariant Intensity Mapping: The PL can be used to perform gamma correction, contrast enhancement, level slicing, contouring or any other radiometric operation on monochrome or colour images.

4) Convolution: Using the Feedback Processor (FP) path any NxM kernel convolution can be performed in NxM frame times.

5) Median Filtering: Applying the sorting module of the FP, 2-D median filtering can be performed using any shape of 9-point window in 1.6s.

6) Antivignetting: This operation is carried out in one frame time via the FP path, using the following equation:

$$Out(x,y) = In(x,y) * Weight(x,y)$$

7) Coordinate Transformation in 3-D space: 256K homogeneous coordinate points may be stored using the image warper and 8 image memories. All points are multiplied by a 4x4 matrix in 32 frame times:

$$[x',y',z',w'] = [x,y,z,w] \begin{bmatrix} a1 & b1 & c1 & d1 \\ a2 & b2 & c2 & d2 \\ a3 & b3 & c3 & d3 \\ a4 & b4 & c4 & d4 \end{bmatrix}$$

If only the FP is available for this operation the processing time is increased by a factor of 4.

8) Box Classification: For this simple pixel classification method the FP can be applied if the dimensionality is ≤ 4. Processing time is 2c+1 frame times, where c is the number of classes.

9) Minimum Distance Classification and Clustering: Both the PL and FP are used in these operations. For 4 dimensions and 16 classes the processing time of the MD classification is 3.2s. By iterative application of MD classification, unsupervised or cluster classification can be obtained. The process takes approx. 10 frame times per class per cluster iteration which represents a speed improvement in excess of 100-fold over the host (IPC) implementation.

10) Fast Fourier Transform: The 2-D complex FFT is implemented using special microprogrammed instructions on the concurrently running IPC and FPP processors. This is the subject of another paper in this Proceedings.

MTA SZTAKI Tanulmányok 206/1988 p. 47

PRECISION IMAGE DIGITIZER EQUIPMENT

O.Farkas, F.Olah, E.Nikodemusz and I.Bogsch

HIRADASTECHNIKA SZOVETKEZET

1116 Budapest, Temesvar u. 20
HUNGARY

In order to process some kind of remotely sensed image data - like LANDSAT or SPOT digital images; aerial photo copies; maps etc. - within one common database we developed at HT an image digitizer.

The equipment is able to convert information provided by photocopies or maps into digital form with high photometric precision. Digitized information can be displayed on screen, stored on magnetic tapes or discs, or can be merged with formerly stored digital images. This feature is important when ground references and their correspondent image fragments in remotely sensed digital data have to be identified. Position and scale of the information to be digitized can be adjusted mechanically and optically until corresponding picture elements meet on the merged image.

The illumination system allows picking up both types of photomaterials, films with transillumination and photoprints or maps with illumination. There is a frame for fastening objects to be digitized. This frame can hold maximum 300mm * 300mm, minimum 50mm * 50mm size objects. The object frame can be moved continuously or step-by-step in X and Y directions. Maximum distance of the object frame from its center position is +/- 130mm in both directions. The object frame can be turned round within +/- 10 degrees range continuously or step-by-step.

Optical images are scanned by a high-resolution CCTV camera. Scale of the image can be adjusted by moving the camera vertically made by a stepping motor. Definition of the picked-up image is automatically adjusted during this vertical movement.

Colour images can be digitized by automatic changing of three different colour filters (R;G;B).

Moving and turning round of the object frame, vertical moving of the camera, and changing of the colour filters are done by controlled stepping motors. Electronics for image processing is applied in equipment like 8-bit precision real-time video A/D converter; 8 planes of image memories (512 * 512 * 8 bit each); Video Processor to perform arithmetic and logic operations between image planes during one TV frame time (40 msec). All the image processing electronics is controlled by a PDP 11/44 compatible minicomputer.

Digitized image can be stored on magnetic discs or tapes or can be transferred to another computer or larger image processing system on-line.

The photometrical and geometrical distortion of the system is corrected by a built-in image processing software - supported by a special real-time hardware and microprogram - within a few seconds.

MTA SZTAKI Tanulmányok 206/1988 p. 49

MICROPROGRAMMED FFT COPROCESSOR FOR THE ATLAS-90

Cseke, I., Szabó, I., Vajda, F.
Central Research Institute for Physics
H-1525 Budapest, P.O.B. 49.

Nowadays Digital Signal Processing is used in a variety of fields including Fast Fourier Transform (FFT). There is a direct mutual influence between signal processing architectures and algorithms and the available VLSI technology is the third major factor in the progress of the entire digital signal processing field.

The basic component of an FFT computation is the *"butterfly"*. The computation of a butterfly involves one complex multiplication and two complex additions.

Many variations of FFT algorithms have been developed that exploit both the symmetry and the periodicity of "twiddle" (phase or rotation) factor of the finite duration DFT (Discrete Fourier Transform). The methods of computing an N-point DFT can be classified on the basis of:

- reordering the original N-point (time domain) sequence: DIT (Decimation in Time)

- dividing the output sequences into smaller subsequences: DIF (Decimation in Frequency)
- reordering the input sequences: "*bit reversed*" input order
- using the same locations to store both input and output sequences: "*in place*" algorithms.
- selecting N as a power of 2, 4, 8 etc.: *radix 2*, *radix 4*, *radix 8*, etc. algorithms.

The similarity of IDFT (Inverse Discrete Fourier Transform) to the DFT allows the same algorithms to be used to compute both forward and inverse FFTs (simply by altering the twiddle factors).

The organization of an FFT processing system architecture is usually dictated by performance, cost and available technology. The main issues of multiple processor architecture are:

- Interconnection topology
- Data flow
- Control complexity

Regular combinations of digital signal processing elements can be classified into four types of *organization*:

- Sequential processors
- Cascadeable processors
- Parallel iterative processors
- Parallel array processors

A variety of combinations of these configurations is often found in the literature.

Another problem is the *partitioning* of the logical architecture and allocating these partitions to an appropriate hardware architecture for implementation. The speed of the calculation can be estimated if the basic architecture -

number of data banks, data paths, number of processing elements and their connections - are set.

The basic components used in implementations can be divided into the following groups:

- General purpose MPs (microprocessors) or MCs (microcomputers)
- Bit-sliced elements
- General Purpose DSPs (Digital Signal Processors)
- Building Block Circuits
- Algorithm-specific (Special purpose) digital signal processors

A typical simple algorithm-specific component is an FFT Address Sequencer which generates addresses to perform repetitive butterfly operations. Sine and Cosine generators providing the sine and cosine values can speed up the calculation of the twiddle factor.

In our solution we used microprogrammed coprocessor for FFT calculation. During FFT calculation it is working in a concurrent and *autonomous* manner, i.e. independent of the host processor (opposed to peripheral mode).

In our approach the selected algorithm was mainly defined by the available hardware architecture. It was the central processor unit of the ATLAS-90 picture processing system, built on microprogrammable bit-sliced components and the attached microprogrammable - 64-bit Floating Point - coprocessor. The picture memory storing the data to be processed can be fetched via a special memory window. The two-dimensional FFT has been divided into two one-dimensional FFTs, defined for the rows and columns, respectively. The algorithm is running on two levels: onthe low level a microprogrammed special "user instruction" executes a one-dimensional FFT on a data vector of size 2^N, while on the high level a high level program written in C organizes the two-dimensional transformation, selects the picture memory locations, loads and rearranges the operative store.

The data representation and precision were mainly influenced by speed requirements. The selected floating point format includes a 16-bit 2's complement mantissa and an 8-bit exponent. The latter is always positive and the mantissa is not always normalized. The real and imaginary parts of the result are loaded into the picture memory.

The selected algorithm of the one-dimensional FFT is

- DIF
- Radix 2
- In place
- normal order of input data and bit-reversed order of output data.

The core of the transformation, the "butterfly" requires 6 real additions and 4 real multiplications. In case of 2^N point transformation, we need $N*2^{N-1}$ butterflies.

The multiplication is microprogrammed and requires 17 microcycles. The coefficients are checked for 0 and ± 1 and in these cases only 4 additions are required. The number of multiplications can be significantly decreased by this method.

The two processors (the main processor called IPC: Image Processor Controller and the coprocessor called FPP: Floating Point Processor) share the FFT processing as follows:

IPC: data and coefficients addressing, loading, program flow control

FPP: arithmetic calculation (multiplications, additions, normalizations)

MTA SZTAKI Tanulmányok 206/1988 p.53

OVERVIEW OF THE ATLAS-90 IMAGE PROCESSING SOFTWARE SYSTEM

Ambrózy, Gy.,Cseke, I.,Fazekas, Z.,Pongrácz, J.,Zöld, S.
Central Research Institute for Physics, 1525 Budapest, POB 49.

Balogh, Z., Keller, Z.
Híradástechnika Szövetkezet, 1519 Budapest, POB 268.

The ATLAS-90 is an interactive image processing system, whose main purpose is to support the efficient processing of remotely sensed multispectral images. The software system is built upon the UNIX (V 7) timesharing system (although image processing services cannot be used by several users simultaneously). The main reasons of using UNIX are the environment it provides and the ease of extensibility.
The components of the software system can be grouped as follows:

- the User Interface,
- utility programs,
- the Function Library, consisting of a collection of functions, subroutines, supporting the image memory access, controlling the different hardware components of the system, etc.
- the UNIX kernel extension, activating the microprogrammed Image Memory Controller.

The images are stored on three different storage media (in several forms): on magnetic tapes, on disks, and in the Image Memory. A common characteristic of all these media is that an image always consists of two main components, a descriptor and the image data. The descriptor in each case contains information necessary for using and interpreting the image data . If image files are stored on disk or magnetic tape the descriptor contains information about files related to a given image, such as history, statistics or colouring information etc.

The system has its own User Interface, which is of dual nature, as it has a menu oriented and a command language oriented component.
Both the menu and command interface are partitioned into an application independent program and into a database containing all application dependent informations. Databases of the user interface can be defined in a simple language; the database itself is generated by a translator program. Modification and extension of an existing system can be carried out by modifying the definition and regenerating the appropriate component.
The Menu System enables the user to select actions and parameter values using full screen menus, as well as, to define parameter values interactively. The Menu System checks the received parameter values against ranges and substitutes default values if necessary. There are further services available for the convenience of the user, e.g. history accumulation, inline command execution, etc.
The ATLAS 90 Command Language facility enables the user to work more efficiently and it ensures an interface compatible with that of the UNIX system. The realized Command Language is more user friendly, because it allows using of keyword and positional parameters; option and parameter identification by mnemonics is possible. Similarly to the Menu System parameter checking, default value substitution is also possible.

The ATLAS 90 Module Library is a set of user programs, which have to be designed according to some conventions. The program parameter interface (options, optional parameters, parameters) has to meet the requirements of the Menu System and Command Interpreter. Furthermore, the programs can access the Image Memory or activate any image processing services only with the help of the ATLAS Function Library.

The first functional group of the Module Library is that of the utility programs:

- Image File Handling: the purpose of these modules is transferring the image files between the system and the external world, or within the system: between different storage media, with the necessary administration.
- Screen Display Control: these modules are responsible for screen definition, assignment of image memory areas to subscreens, changing parameters (scroll, zoom), grey level control, split screen etc.
- Test Programs: the Image Memory System has its own diagnostic programs, including one of them which is signature analysis.

The second group consists of the image processing modules. Our intention is to provide a wide variety of them , although the main area of our interest is the processing of multispectral images. The Module Library is now under development, some interesting examples of the available modules are as follows:

- Preprocessing of images (filtering, smoothing, corrections, etc),
- Image statistics,
- Geometric transformations,
- Fourier and inverse Fourier transform (partially implemented in microcode),
- Supervised classification,
- Clustering with ISODATA algorithm,
- Image arithmetics: the program evaluates simple

arithmetic expressions, where the operands are blocks of the Image Memory.

The components of the Module Library are programmed mainly in C language, however, FORTRAN 77 language can also be used.

The ATLAS 90 Function Library is a set of routines programmed in C and UNIX assembler languages. Its main purpose is to provide an efficient interface for the developers of the Module Library.

At present, the Function Library contains the following functional groups:

- Image Memory Access: the functions move data blocks between the Image Memory and the operative store of the control computer, handle the contents of the Image Memory through windows for processing purposes, etc.
- Logical Screen Control: these functions serve for defining subscreens, assigning Image Memory areas, scrolling or zooming displayed data , etc.
- Display Subsystem Control: these functions handle the complex facilities of the Display Subsystem, and of the Pipeline Processor, as well the interactive devices.
- Image File System Functions: this group provides elementary functions for image file handling.

The Image Memory Controller, a microprogrammed special processor is attached to the IPC controlling computer. Its physical handling is realized by a device driver included into the UNIX kernel.

Kvázipoláris müholdak képeinek földrajzi azonosítása

Almádi István

OMSZ Számítóközpont

1675 Budapest, Tatabánya-tér. 15-18. Pf: 32

Előadásomban a KEI Távérzékelői osztálya számára készülő programcsomagot szeretném röviden ismertetni. A programcsomag kvázipoláris müholdképek geometriájával, földrajzi azonosításával kapcsolatos feladatokat végző ill. ezeket kisegitő programokat tartalmaz Az algoritmusok még általánosan a kvázipoláris müholdakra iródtak, de a programok finomításuk során a NOAA holdak képeinek feldolgozására specializálódtak. Az eljárások további sajátosságait a TPA 11/48-as számítógép lehetőségei adják A programok jelenleg FORTRAN-IV. ny elven íródtak, de hamarosan egyes rutinok macro-assembler betétekkel lesznek felváltva.

A programok alapját két, a kétirányú földrajzi-műholdképi azonosítást végrehajtó rutin jelenti. Ezek, az általánosabban elterjedt illesztőpontos eljárásokkal ellentétben, a műhold konstansok vett és aktuális pályaadatai alapján végzik az azonosítást. A számítások gömbháromszögtani összefüggések felhasználásával történnek. Azok az egyenletek, melyekkel egy műholdképpont földrajzi koordinátáit megadják az irodalomból jól ismertek. A visszafelé történő transzformálást egy, talán újszerű nek mondható elképzelés alapján ugyanezekkel az egyenletekkel lehet végrehajtani.

A különböző geometriai feladatú programok futását adatbevivő és adat-előfeldolgozó, valamint a számítási pontosságot javító segédprogramok támogatják.

A csomag segítségével jelenleg elvégezhető feladatok:

- Műholdképpont-földrajzi pont kétirányú azonosítása.
- Tetszőleges polársztereografikus vetületü térkép-pont-földrajzi pont kétirányú azonosítása, az adatokból táblázat készítése.
- Műhold vetületű képre tetszőleges kontúrok felrajzolása töréspontok bevitele alapján.
  (A NOAA felbontásának megfelelő sűrűségben Magyarország, Balaton, Duna, Tisza megyehatárok töréspontjai lemezen tárolva)
- Térképvetületű képre ugyanilyen kontúrfelvitel.
- Műholdkép adott térképvetületre transzformálása.
- Műholdkép területhelyes vetületbe vitele.

A további tervezett feladatok:

- Az egyes valamilyen vetületbe transzformált pontok azonnali corsoros megjelenítése MIP rutinok segítségével.
- Két műholdas közös vetületben történő megjelenítése
- METEOSAT képekre a poláris holdaknál felsorolt feladatok elvégzése (ezek részben már elkészültek)

A felsorolt feladatok kissé részletesebben a térképvetületre transzformáló programot szeretném bemutatni, mely az eddigiek közül a legnehezebb problémát jelentette. A programok operatív célra készülnek, - a közeljövőben az intézetben beindul a NOAA képek rendszeres vétele-, igy minden megoldásnál elsődleges szempont a gyorsaság. (Ez a feltétel eleve elvetette az azonosítópontos korrekciós eljárásokat). Ez a TPA-n több problémát is felvet, főként a 64 Kb-ból történő kilépés nehézkessége miatt. A program jelenlegi formájában nem is lép ki ennél nagyobb memóriaterületre, igy a térkép kis részletekben készül el. Futási ideje 256x256-os kivágatra a NOAA 4 sávjára 5-6 perc. A fő időt a viszonylag sok lemezművelet mellett (a blokkolt adatmozgatási forma ellenére) az igen nagy mennyiségű valós számmal végzett művelet igényli. A számítások nagy része azonban olyan, hogy adott műholdra előre elvégezhető és az adatok táblázatos formában eltárolhatók. Ettől az újabb lemezműveletek ellenére a program jelentős gyorsulása várható. A táblázatosítás először a meteorológiánál használt S6-R radartérképhez fog megtörténni, mely vetületen az aktuális radaradatokkal való összehasonlítás válik lehetővé.

A jelenlegi feladatok tehát a táblázatosítás, valamint a további gyorsítást szolgáló macro rutinok elkészítése, és a METEOSAT tal kapcsolatos programok szerves beépítése a csomagba mely így az intézetben folyó munkákat komplex módon fogja támogatni.

MTA SZTAKI Tanulmányok 206/1988 p. 61

TRACKING METEOROLOGICAL OBJECTS ON DIGITAL RADAR IMAGES

F.Dombai,* A.Kapovits,* L. Ketskeméty,** T.Szentimrey***

Central Institute for Weather Forecasting, POB 32, H-1675 Budapest *
Polytechnical University of Budapest, H-1111 Budapest, Egri J.u.20. **
Central Meteorological Institute, POB 38. H-1525 Budapest ***

Nowcasting, having appeared in the last decade, is a new, rapidly developing area in the field of meteorological forecast. It has a meaning of description of current weather with high resolution both in space and time, and extrapolation of the weather systems 0-2 hours ahead.

Most meteorological satellites and weather radars provide the necessary detailed information on the cloud-precipitation systems, on their developments and movements. Different techniques are available for determining and extrapolating the movements of these objects seen on digital images. There are methods for general use and there are others regarding the special features of the objects and sources of data. The algorithms giving exact information on weather systems require great computing power and sometimes special hardware tools. Cross-correlation is the most widely used method for extrapolation.

In this investigation our aim was to find method(s) which can be realized on mini-computer an used operationally with short running time. A variety of such methods has been tried (movements of weighted and non-weighted gravity centers, the adaptive window cross-correlation method and its simplified version, the Boolean cross-correlation technique). In addition on these, substitution of simple geometrical shapes for modelling of objects has been also tried. Modelling was made by applying curves of second and fourth order with ellipses of dispersion and their modified versions. For testing, in certain cases digital radar images with different spatial resolution were used in order to see the impact of different resolution on the results and to save running time.

There has been already researched in Hungary to describe by objective methods the movement of cloud-precipitation systems. It was based however on digitized analogue satellite pictures. For the present investigation we decided to use digital radar images because of their excellent high resolution (better than that of meteorological satellite images available in Hungary) and because the objects on them are of more compact and more easly identifiable than those on satellite pictures.

Preprocessing the raw digital radar pictures (calibration, filling up the gaps of scan-conversion, filtering, eliminating the image noise etc.), determining the basic statistical parameters of the objects to be tracked (sizes, mean values, gravity center coordinates, windows etc.) and their identification on consecutive pictures have been the important stages of achieving a proper data base for nowcasting the precipitation field.

Though our work has not been finished yet, some conclusions can be made. The most important one of them is that there is no single method applicable to every case. Different method should be applied to the objects, depending upon their sizes, shapes, intensity distribution and their changes in space and time. For this particular reason we intend to develop a special software package that has facilitates the optimum choice of methods for automatic tracking based on the statistical parameters of the objects.

COMPUTER ANALYSIS OF THE STRUCTURE OF THE Ca^{2+}-ATPase ENZYME

KOCSIS É.[1] DUX, L.[1] KUBA, A.[2] GUBA, F.[1]

1:ALBERT SZENT-GYÖRGYI UNIVERSITY OF MEDICAL SCIENCES
INSTITUTE OF BIOCHEMISTRY, SZEGED, DÓM TÉR 9., HUNGARY
2:ATTILA JÓZSEF UNIVERSITY OF ARTS AND SCIENCES
LÁSZLÓ KALMÁR LABORATORY OF CYBERNETICS, SZEGED, ÁRPÁD TÉR 2., HUNGARY

Introduction

The techniques for the study of protein crystals with electron microscope developed rapidly in the past few years. A substantial progress has been achieved in the application of computer image analysis methods to electron micrographs of single molecules and planar protein arrays lacking long-range order.

Averaging processes enhance the common structural features of the molecules by suppressing the background noise on the micrograph.

Our aim was to improve the signal to noise ratio (SNR) of the pictures and to measure some characteristic data of the Ca^{2+}-ATPase molecule. We applied computer averaging of the electronmicrographs for this purpose.

Biochemical background

The Ca^{2+}-ATPase of sarcoplasmic reticulum forms two dimensional crystals on the surface of membrane vesicules in the presence of vanadate. The structure of the Ca^{2+}-ATPase dimers has been described at 15 Å resolution previously. In a hyposmotic medium these crystals disrupt into recognizable free dimer chains. The molecules released from the intact structure may loose some intramolecular forces which accomodate them to the curved surface. Therefore the isolated Ca^{2+}-ATPase dimer chains may provide some new insight into the detailed molecular organisation of the enzyme oligomers.

Methods

Sarcoplasmic reticulum vesicles were prepared and the Ca^{2+}-ATPase enzyme was crystallized with vanadate as described by Dux and Martonosi 1983. /J. Biol.Chem. 258. 2599-2603/. The intact crystalline vesicles were transferred into hyposmotic medium and electronmicrographs were taken at 80, 100-120000x instrumental magnification at 80 kV accelerating voltage, after negative staining with 1% uranyl acetate. The positive prints of the electronmicroscopic negatives were digitalized and recorded with a MO8X computer-TV camera HT 680 Colour display system. The images were enlarged two-fold using linear interpolation technique. Averaging procedure was performed on these pictures by a window method. Molecules in the isolated dimer chains lay in identical position. This allowed us to average the molecules in several paralell chains. The slight desorientation within the dimer chains can be corrected by changing the position of the selected window area and calculating the minimal value of the square difference function. Representing the values of the window as a matrix (A_i) /i=1.. k, averaging k-times/, the final result of the averaging process can be given: $A=\sum A_i/k$. The dimers obtained by this averaging process were magnified two-fold again, and the remaining point like noises were filtered out by a two-step enhancement procedure using the

$\begin{pmatrix}111\\111\\111\end{pmatrix}$ convolution function.

Characteristics of the systems

The system consists of a M08X computer with 64 kbyte memory and a HT 680-Colour Display.

The display can be connected to 625-row OIRT or CCIR TV systems. The AD part of the apparatus digitizes black-white videosignals, the DA part produces black-white and RGB signals from the stored numerical values.
The geometric resolution of the system is 384x288 picture points, the intensity resolution is 64 levels per pixel.

Results and discussion

The width and length of the two molecules forming the dimer are measured 44+2.65 Å and 62+1.95 Å respectively in the ruptured crystals, the same values are 46 Å and 62 Å in the intact crystal /Taylor, Dux, Martonosi 1986. J.Mol.Biol. 187. 417-427/. Despite the unchanged dimensions of each enzyme molecule after the hyposmotic lysis, the distance within the dimers /between the two Ca^{2+}-ATPase monomers/ was found to be significantly smaller /67+2.76 Å versus 72 Å/ in the isolated chains when compared to the intact membrane crystals.

This difference can be related to the reorientation of the enzyme from a tilted position in the membrane crystal when the protein-protein interactions on both sides of the dimers are released.

Further high resolution image analysis of the isolated Ca^{2+}-ATPase dimer chains may provide more detailed features of this enzyme.

On Shape Representation in Two Dimensions

Jan-Olof Eklundh
Computer Vision and Associative Pattern Processing Laboratory

Numerical Analysis and Computing Science
Royal Institute of Technology
S-100 44 Stockholm
Sweden

Introduction

The understanding of shape and geometry plays an important role in our process of visual perception. It is therefore not surprising, that problems on determining shape descriptions have been given much attention in computational vision. This concerns both the problem of what should be represented and how, and the problem of how to compute these representations.

Formal descriptions of shape can be obtained in terms of notions from various branches of mathematics: geometry, differential geometry, approximation theory etc. These disciplines offer means of abstracting shape properties. However, this is not sufficient to address the issues in computer vision. Here it is essential to consider the problems of what it is one wants to capture by these descriptions and also in what way and by what primitives such descriptions can be computed from real life data, like noisy image contours.

In this paper we will consider some of these problems in the case of two-dimensional shapes with specific reference two image contours. Even though we limit ourselves to this case we would like to make of few remarks on the more general situation. There are two ways in which the shape representation problem can be generalized into three dimensions. First, there are three-dimensional shape representations of similar types. Being far from simple, this generalization is in a sense nevertheless straightforward. Secondly, there is the generalized case of when the two dimensional contours are projected silhouettes of the boundaries of three dimensional objects. The isues raised here are substantially more difficult. In particular, the important properties to be represented may now be different. We will briefly relate our two-dimensional discussion to these problems at the end of our talk.

Dynamic shape

Traditional mathematical and computational approaches to shape representation (see for instance the overview by Pavlidis (1980) or Faux and Pratt (1981) takes of view of shape as a static entitity that can be given a single representation in terms of some primitives, e.g. some family of functions or parameters. However, such a view does not capture two paramount aspects of real life shapes: One is that structures in the world occur at many different levels of scales. A forest seen from an airplane at high altitude looks a smooth surface. A closeup look reveals the treetops as a sort of almost periodic modulation of this "envelop". At even finer scales the treetops start to look different and to have branches, leaves and so on. The important thing is that above the actual physical resolution (below that we can observe nothing) **all of these** shape descriptions are valid. Moreover, each of them exists at one appropriate (interval of) scale(s), but not at the other scales. Hence, general shape representations useful for the purpose of computer vision should capture all these scales*). The second aspect has to do with perception of shapes, including classification and recognition. For example, many proposed shape recognition schemes are based on the derivation of and matching of features or invariants. Sometimes such schemes also take into account that the shapes can undergo various transformation or deformations and still relain the classification, see e.g. Pavlidis (1977) and the classical paper by Fischler and Elschlager (1974). Nevertheless, it is not until recently that one started to model a more general notion of similarity or association of shape. Traditional feature-based methods, based on e.g. corners, curvature properties and symmetries do not account for the fact that slight perturbations or changes of the shape may change these feature descriptions drastically.

*) In a specific application one might, of course, know at what scale the important events occur.

These variations may be due to other causes than just noise. For instance, one shape can have developed from another by some manufacturing process or by natural growth or wearing. A flexibel representation scheme should capture also this kind of dynamics.

Over the past few years some important contributions. Witkin (1983) introduced the notion of scale-space to describe the structure of shape at several levels of scale, the simple but crucial observation being that scale should be seen as a continuous parameter. Witkin proposed that the scale-space representation should be obtained by gaussian blurring. It has later been shown, first by Babaud et al (1983) that this is the only sensible way, if one does not want to obtain new structure in a certain sense. Koenderink and van Doorn (1986) have furthered the theory substantially and have indicated a number of unsolved problems. It should be observed that existing theories on pyramid representation really does not address the same type of problems. Pyramids have been used for various efficient algorithms based on coarse-to-fine strategies. However, work on pyramids has not explored the fact that important structures only may exist at a certain scale. Neither is there any corresponding analysis of the causality between structures at different scales.

In principle, scale-space allows representation of deformation and growth processes of shapes as well. Another type of representations that capture such information at a higher level and, possibly, in a more general sense has been proposed by Leyton (1984, 1988), who introduced shape grammars based on the dynamic change of properties. For simple shapes he gives a compact and complete description of the set of possible deformations, a description that could be used as a basis of categorization for examples, the number of steps needed to transform one shape into another could be used as a measure of similarity.

Leyton´s representation is related both to the so called codon theory by Richards and his co-workers, see e.g. Richards and Hoffman (1985) and to the symmetry-based representation proposed by Brady and his co-workers (see e.g. Brady (1983)). The latter are in fact, the only ones who have tried to turn their representational models into computations, see also O'Connell and Brady (1987).

Computational aspects

There are several criteria that could be put on a good representation of shape (see e.g. Eklundh and Kjelldahl (1985) for a general discussion). One decisive aspect is that the representation should be possible to compute. In the case of two-dimensional shape representations of data obtained from images, the problem consists of robust determination of shape representations from noisy planar contours, in general obtained by edge detection and contour tracing. Hence, one can expect positional noise, often at a fine resolution, as well as shape events occurring at different levels of scale. To obtain a dynamic shape representation in one of the senses mentioned above, one could proceed in a number of ways. One way would be to do gaussian blurring of the shape and in that way pick up the different properties or events at different scales. This was the original proposal by Witkin, who applied it to waveforms (functions of a single variable). Mokhtarian and Mackworth (1986) applied to idea to two-dimensional contours.

There is one problem with this approach, when it is turned into a discrete algorithm. The blurring operation moves the events, like corners, curvature extrema or inflection points. This happens in a predictable way, but still the location and sometimes (like for the corners) the existence of these events is not correctly or explicitly represented at any specific level, except (for location) at the lowest level. This is unfortunate, since certain events exist at a wide range of scales and since efficient algorithms should be able to take advantage of this fact. As a simple example we could consider a noisy polygon with sides of roughly the same length. Such a figure should remain stable over a large range of scales, with its polygonal shape explicit. The information is nevertheless available in the full scale-space representation, so possibly algorithms can be designed to recover it. We will, however, propose another computational approach and demonstrate its usefulness for computing dynamic shape representations.

A proposed algorithm

The algorithm we propose works as follows. First we compute a family of polygonal approximations in a densely sampled scale-space*). This family is nested, that is the coarser representations are subsets of the finer ones. We argue that since we aim at image contours the spatial noise is small, that is, outliers do not occur (they would be thrown away at edge detection). Therefore, a pure interpolation scheme is used. It is based on a split-and-merge procedure which tests both on maximal errors and on offsetting errors (using signed relative area). This **family of polygons** is then used as input to an approximation with higher order curves. We have in particular considered conic arcs, but at this stage we have the information about the significant shape features **at all scales** and can use any approximation scheme, like spline approximation. We will in our talk show how we from these algorithms can perform shape preserving approximation with subjectively reasonable results, and that we can determine important shape features basic to e.g. Leyton's dynamic shape grammars. A detailed description by (a now) early version of the algorithm can be found in Bengtson et al (1986). Experiments will be presented at the talk.

Conclusions

We have discussed the goals of shape representations for computer vision. We have focused on the need for dynamic representations capturing the fact that structures occur at many levels of scale in real scenes and that similarities often reflect certain deformation or growth processes. We then consider the two-dimensional case and propose a method for computing these multiresolution representations. Experimental results confirm that the method achieves these goals.

---

*) This means that the incremental step between resolutions equals the original resolution or a small multiple of it.

References

Pavlidis, T., *Algorithms for Shape Analysis of Contours and Waveforms,* IEEE PAMI, 4:2, 1980, 301-312

Faux, I.D., Pratt, M.J., *Computational Geometry for Design and Manufacturing,* Ellis Horwood, 1981

Pavlidis, T., *Structural Pattern Recognition,* Springer, New York, 1977

Fischler, M. A., Elschlager, R.A., *The Representation and Matching of Pictorial Structures,* IEEE TC, 22, 1973, 67-92

Witkin, A.P., *Scale-Space Filtering,* Proc. IJCAI -79, Karlsruhe, 1983, 1019-1022

Koenderink, J.J., van Doorn, A.J., *Dynamic Shape,* Biological Cybernetics, 53, 1986, 383-396

Babaud, J., Witkin, A.P., Baudin, M., Duda, R.O., *Uniqueness of the Gaussian Kernel for Scale-Space Filtering,* FLAIR Technical Report 20, Palo Alto, Ca., 1983

Leyton, M., *Perceptual Organization as Nested Control,* Biological Cybernetics, 51, 1984, 141-153

Leyton, M., *A Process Grammar for Shape* ,Artificial Intelligence, 34, 1988, 213-247

Richards, W., Hoffman, D.D., *Codon Constraints on Closed 2-D Shapes,* CVGIP, 31, 1985, 265-281

Brady, M., *Crieria for Representations of Sahpe,* in Rosenfeld, A., Beck, J., (eds.), *Human and Machine Vision,* Erlbaum, Hillsdale, N.J., 1983

O'Connel, J.H., Brady, M., *Generating and Generalizing Models of Visual Models*, Artificial Intelligence, 31, 1987, 159-184

Eklundh, J-O., Kjelldahl, L., *Computer Graphics and Computer Vision - Some Unifying and Discriminating Features*, Computers & Graphics, 9, 1985, 339-349

Moktharian, F., Mackworth, A., *Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes*, IEEE PAMI, 8, 1986, 34-43

Bengtson, A., Eklundh, J-O., Howako, J., *Shape Representation by Multiscale Contour Approximation*, TRITA-NA-8607, Stockholm, 1986, to appear in IEEE PAMI

DETECTING DEFECTS IN TEXTURE

Dmitry Chetverikov

Computer and Automation Institute,
Hungarian Academy of Sciences,
Budapest, POB.63, H-1502 Hungary

Introduction.

In automatic surface inspection, one often faces tasks that combine texture analysis with image segmentation. But it is not necessarily segmentation by texture. In some cases, local imperfections (defects) in uniform texture fields are to be automatically detected. The size of the defects is usually so small that they cannot be considered as distinct texture patches. Texture imperfections are non-textured patches that locally break the homogeneity of a texture pattern. Detection of such regions differs from the traditional segmentation by texture.

Application problems related to defect detection in texture are currently solved by diverse ad hoc methods, without any general methodology. Furthermore, the majority of the existing application algorithms for defect detection in texture cannot cope with a slow spatial variation of the distorted texture pattern that sometimes appeares in the industrial or other imagery. In recent studies [1,2] we have made an attempt to generalize the problem of defect detection in texture. In the present paper we discuss possible frameworks for detection of imperfections, and describe a new algorithm for location of texture defects in the presence of texture variation.

1. Detecting texture imperfections.

Basically, there are two categories of approaches to the detection of imperfections. Local approaches compute a set of texture features in a sliding window and search for significant local deviations in the feature values. Global methods apply global image transformations. They require more detailed information about the shape and intensity of defects.

In this study we confine ourselves to the local approaches. The majority of the existing imperfection detection algorithms developed for different inspection tasks belong to the local type. The conventional local approach involves computing texture feature values in a sliding window and comparing them to the reference values representing a perfect pattern. A threshold is set which separates acceptable and defective patches.

In practice, there is sometimes a slow global variation of texture throughout the image. The conventional approach based on the simple non-adaptive comparison to the reference pattern may fail depending on the magnitude of the global texture variation. To cope with the problem, instead of using the reference pattern one can compare the feature values computed in neighboring windows in order to extract drastic local changes. The suggested adaptive procedure uses such a difference operator.

2. A new adaptive procedure.

For the sake of simplicity, consider the case of single texture feature F. Let O be an arbitrary pixel and let (i, j) denote the coordinates of the pixel. Consider the eight-window configuration centered on O which is shown in Fig.1. Introduce the local difference operator

$$\Delta_F^{(8)} = \max_{k=0,1,2,3} \left| F^{(k)}(i,j) - F^{(k+4)}(i,j) \right| \qquad (1)$$

where $F^{(k)}(i,j)$ is the value of F in the k-th window, k=0, 1, .., 7.

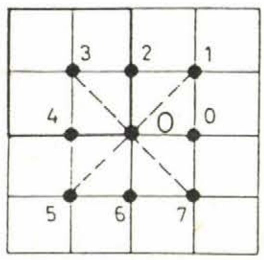


Fig.1 The eight-window configuration.

This operator responds to the borders of imperfections. It is less sensitive to slow global variation of texture pattern. The eight-window configuration is rotationally flexible and produces more border points than a simple four-window operator would produce.

The difference operator (1) is applied to the analysed pattern to obtain the <u>difference image</u>. In order to speed up the computation this operation is executed in two steps. First, the <u>feature image</u> is computed which contains the feature values for each window, then the difference image is obtained frum the feature image using (1).

The difference image is thresholded to detect those boundary points which yield great difference values. These points usually appear in clusters which do not form connected contours. Superfluous points may also appear. Hence a digital filter must be applied to the difference image prior to (or after) the thresholding in order to extract distinct clusters of the imperfection points and filter out the superfluous points. In this study, the median filter was used for this purpose.

In the conventional approach, the thresholding is applied to the feature image. An experiment has been carried out in order to compare the suggested procedure to the conventional one and to investigate the effects introduced by the median filtering. For this purpose the procedure has been executed step by step and the intermediate results have been stored and displayed. The experimental results are discussed in [3].

Concluding remarks.

In this paper, a new procedure was suddested for adaptive detection of texture imperfections in the presense of a slow spatial variation of the texture pattern. The results of beginnig experiments with the new detector are promising. The new procedure seems to be efficient when the conventional one fails because of texture variation. The experiments also show that the shape and size of the detector window must be adjusted to the shape of imperfections. Further work is needed to develop a method for automatic setting of the optimum window size of the detector.

References.

[1]**Chetverikov, D.** (1987). Texture imperfections. Pattern Recognition Letters 6, pp.45-50.

[2]**Chetverikov, D.** (1987). On some basic concepts of texture analysis. Proc. 2nd Int.Conf. Computer Analysis of Images and Patterns, Wismar, GDR, pp.196-201.

[3]**Chetverikov, D.** (1987). Detecting defects in texture. Proc. 5th Scandinavian Conf. Image Analysis, Stockholm, pp.427-433.

MTA SZTAKI Tanulmányok 206/1988 p. 81

IPW: AN IMAGE PROCESSING WORKSTATION

Gy. Bango, M. Gardos, J. Miskolczi, I. Renyi
Central Research Institute for Physics,
H-1525 Budapest, P.O. Box 49

The appearance of relatively powerful and inexpensive professional personal computers rapidly reshaped our view of (among other things) image processing equipment. Dozens of smaller firms started to produce *add-on boards* for the most successful PCs, which turned them into simple and versatile "image computers" for a fraction of the price of previous systems. The processing on these "PC + frame grabber" systems is performed solely by software. For a number of applications where the processing time is not really important (e.g. laboratory or scientific applications) this is the most straightforward solution. In most real applications, however, there is a more or less strict constraint on response time which, for most cases, cannot be met by software means.

The solution to the speed problem is the use of special purpose hardware or processor architecture. During the last 10 or 15 years, special architectures for image processing have become separate scientific and practical fields. A great number of ideas, solutions and systems have evolved with a common goal: the architecture should make efficient use of the special structure and inherent parallelism implied in most image processing operations.

In addition to the above mentioned goal, the concepts behind the design of the *Image Processing Workstation* were the following:

- Cost effectiveness and small, desktop size.
- At least an order of magnitude increase in throughput for most operations compared to frame grabber + sw solutions.
- Easy adaptability to specific application requirements by open ended architecture.
- The use of the widespread IBM PC/AT as the controlling or host computer.
- The use of the microprogramming, local parallelism, and pipeline concepts in the processor modules.
- Application of the most up-to-date VLSI components available for signal processing.
- Convenient menu driven and command language user interfaces.

To illustrate the above basic concepts we give a short oveview of the IPW architecture and its software design principles. The modules of the IPW are connected by two *bus systems*:

- Both programmed and DMA transfers may take place via the extended *PC Bus* (Master Bus) for controlling the IPW modules (e.g. downloading microprograms, look-up tables, reading/writing the image memories by the PC, etc.).
- For high speed data exchange among image memories and processors a multioperand, time-shared *Synchronous Bus* (Slave Bus) is used. A special characteristic of this bus is that it does not carry address information.

To achieve effective utilization of the resources, hardware supported image processors usually require a large amount of memory. The use of up to 64 image planes (1/4 MByte each) is not uncommon. Mainly for economic reasons the IPW offers *image* *memory* modules of two different kinds:

- The *Video Memory* (VM) is built up of dual port video RAM components and stores three 512x512 pixel (8-bit/pixel) images for video rate acquisition and display. The VM makes it possible to undersample the digitized scene and to zoom and roam the displayed image.
- The *Bulk Memory* (BM) module uses high capacity DRAM components to store four 16-bit (or eight 8-bit) image planes. It has a very flexible addressing scheme: up to four independent quasi concurrent operand addresses can be generated using a high power, microprogrammed data address generator device. By combining half of two image planes for a single image access, full video speed can be achieved. Up to eight BMs can be used in the system.

The power and effectivity of the IPW is primarily determined by the *special purpose image processors*. In general, they provide rapid execution of algorithms that require frequent sequential image memory access (or where an access algorithm can be given, e.g. FFT), counting, decision making, complex sequences of mathematical or logical functions, etc. The processor modules have microprogram control. Resources include a single-chip Digital Signal Processor (DSP), multiplier/accumulators, large data buffers and tables, special counters, sorters, logic filters and so on.

To illustrate the use of the special processors in low-level vision, a few examples will be listed below:

- shift invariant point operations, histogram conversion
- arithmetic and logic operations between two images or one image and a constant (8- or 16-bit accuracy)
- local linear and nonlinear neighbourhood operations (16-bit internal accuracy)
- complex 2-D Fast Fourier Transform (16-bit block floating point operations)
- matrix (vector) operations (e.g. addition, vector inner product, matrix multiplication, transposition, etc.)

- polynomial evaluation, geometric transformation (correction)
- binary logical operations (using 3-by-3 windows)
- area and perimeter measurements
- component labelling
- generation of local and global statistics, co-occurrence matrix.

All operations are performed on n-by-m sized image windows or matrices ($1 \leq n,m \leq 1024$) and - unlike most other systems - the processing time is proportional to nxm. *Processing speed* is roughly 1/2 video rate for most operations. With the DSP a 16-bit multiplication and 40-bit accumulation can be carried out every 160 nsec. The estimated time for a 512x512 pixel complex FFT is 30 sec.

The *software system* of the IPW is composed of three layers:

- *User Interface Layer*. The user has two choices for communicating with the system:
  - -- For inexperienced users a flexible menu system is available, using the fashionable window technique.
  - -- The so-called command language makes it possible to write sequences of commands, branchings, etc. in a simple language-like manner, with appropriate parameter and diagnostic information passing between commands.
- *Program Library Layer*. The components of this layer make up the continuously growing set of low-level and high-level vision programs and the utilities (including image I/O, file handling, overlays).
- *Basic System Layer*. This is accessed in form of a basic routine library and can be regarded as an extension of the operating system. It includes a special collection of mathematical routines, routines handling the image memory, the spec. processors, the interactive peripherals, and several image processing primitives.

MTA SZTAKI Tanulmányok 206/1988 p. 85

METHODS FOR REAL TIME MOTION ANALYSIS

Erik GRANUM - Henrik I. Christensen

Laboratory of Image Analysis
Institute of Electronic Systems
Aalborg University
Denmark

Methods for image sequence and motion analysis can largely be grouped under the following 3 headings: Fourier methods, Optical flow and Token matching. These approaches are reviewed, compared and evaluated for real time applications.

Token matching seems most promising in this respect. Matching methods and consequences of using low and high level tokens, respectively, are evaluated.

Still, - motion analysis in real time by 1988 depends on restricted scenaria and special algorithms and/or special hardware. Various possibilities are considered.

In the end, processing speed is by far the only problem. Development of reasonable methods (with regard to complexity) which can actually take advantage of observed motion patterns, handle occlusions and predict motion for tracking and collision avoidance, etc. are possibly more interesting challenges.

An experimental system which addresses these problems and has the potential of operation at TV-frame rate is presented and various examples of applications are given.

MTA SZTAKI Tanulmányok 206/1988 p. 87

A REMOTE SENSING APPLICATION FOR A COMPUTER VISION SYSTEM

Axel PINZ

Institut für Vermessungswesen und Fernerkundung
Universität für Bodenkultur
A-1190 Wien,
Peter Jordan-Strasse 82
Austria

A b s t r a c t

After a short survey of the architecture of the Vision Expert System VES the problem of forest damage classification using color-infrared airborne photographs is discussed.

To achieve completely automatic interpretation of digital images of the above specified kind, it is necessary to:

a) find trees in the image,
b) recognize the species of the tree found in step a),
c) evaluate the color of the tree top - this gives a hint about the vitality of the tree.

Finally an example shows, how the VES works on real images, examining closer the knowledge representation, inferences and image processing tasks of the system.

INTERACTIVE PROGRAM PACKAGE FOR SATELLITE IMAGE PROCESSING

Gy. Büttner - T. Hajós - M. Korándi

Földmérési és Távérzékelési Intézet
(Remote Sensing Centre)

1373 Budapest Pf. 546.

INTRODUCTION

Digital image processing software can be divided into three general categories. The first applies specific high technology hardware systems, the second offers a broad collection of algorithms taken from the published literature and the third addresses specific applications, generally in the form of turnkey systems.

The Remote Sensing Package (RSP) described in this paper has been developed to process high resolution images taken by Earth resources satellites, such as Landsat and SPOT. This software belongs to the last of the above categories. Main aim of it is to provide an effective tool for multichannel image processing and computer assisted image interpretation (Figure 1) in a highly interactive environment. A detailed description of RSP is provided by Hajós et al. (1987).

To illustrate the capabilities of RSP some results of the Hungarian PEPS (Programme d'Évaluation Préliminaire SPOT) study are presented (Büttner et al. 1987). RSP was used for solving the following tasks:

- data quality and information content analysis of SPOT data;
- supervised classification of vegetation (crop types) and soils;
- enhancement of areas with low vegetation index according to their soil brightness using bi-temporal imagery.

SYSTEM DESCRIPTION

Experience has shown that long response times have disruptive effect on the continuity of the vision - mind process of the user, degrading overall performance and possibly resulting in loss of information. Therefore, run times in RSP are near real-time, no more than 1-2 minutes. The result of a processing is immediately visible on the color monitor.

RSP is running on PERICOLOR 2000E hardware, manufactured by Numelec, France. The hardware consists of an INTEL 8086+8087 management processor for system-user communication and a programmable serial bit-sliced processor for fast image operation. For program and data storage a Winchester and a floppy disk units are used. The system can have up to ten 512*512*8 bits image memories and four 512*512*1 bit graphic overlays. The management processor can be programmed in PLM and Fortran, while an interactive micro-assembler is available for the fast processor.

FUNCTION SET OF RSP

Usage of RSP is accomplished by a menu. It combines the benefits of the command line interpreter and the common menu (Figure 2). The menu containing both alphanumeric and graphics characters appears on the right side of the color monitor. It executes logical control over the called image processing task and hides all the unimportant parameters, so the user can focus on the high level image processing tasks.

Each of the application functions has a separate menu. Zoom and roam are available inside each image processing task.

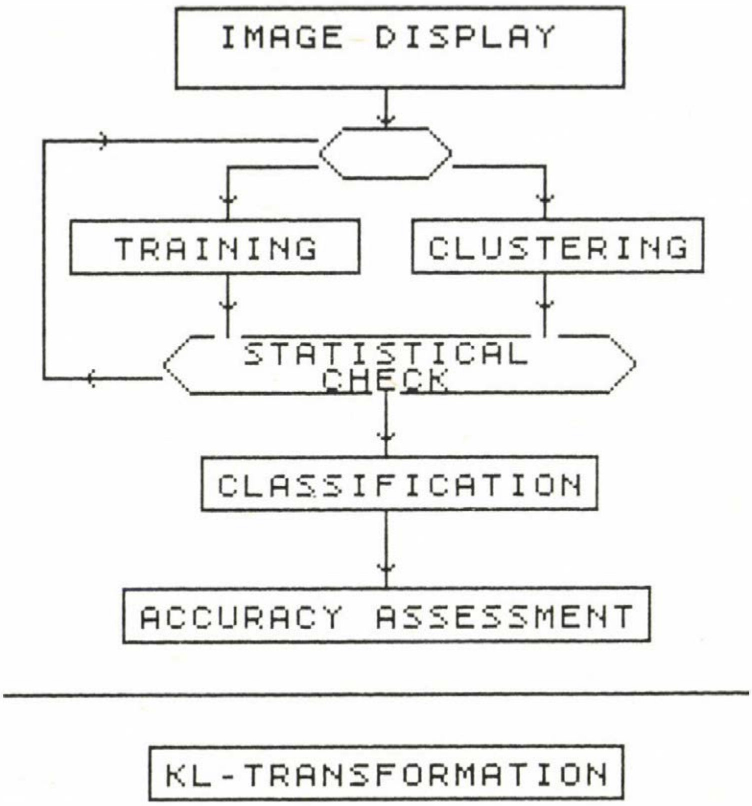


Fig.1.
Flow chart of the RSP

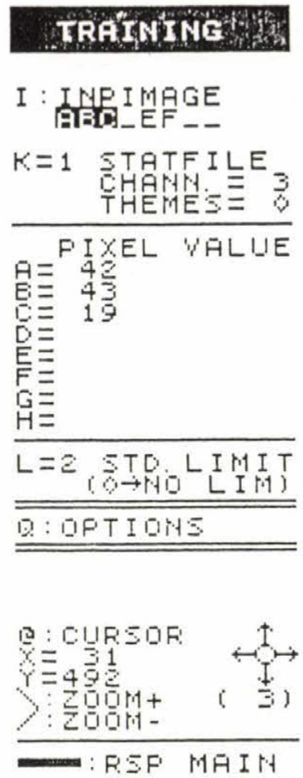
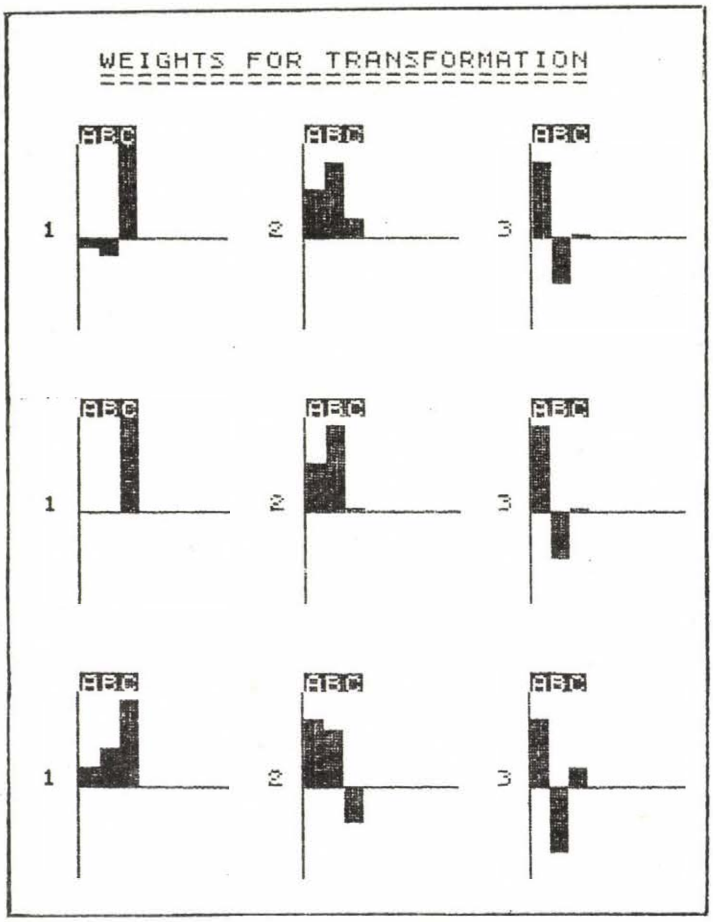


Fig.2.
Example of a menu

DISPLAY composes images for optimal visual evaluation and can be used for modifying colors of a thematic (classified) image.

Principal component (PC) transformation serves for dimensionality evaluation or data COMPRESSION. The conclusions of the PC analysis carried out for three SPOT satellite images of the same area were as follows (Figure 3):

- 86 - 92 percent of the total variance is contained in the PC channel on each date, which is dominated by the near infrared band.
- The third PC channels are determined by the difference of the highly correlated visible bands, therefore they are practically noise.
- Two principal components cover 98.5 - 99.7 percent of the total variance on each date, ie. SPOT data have two intrinsic dimensions.



	Variance PC1	PC2	(%) PC3
04-May-86	89.3	10.2	0.5
31-July-86	92.5	7.2	0.3
13-Nov-86	90.8	7.7	1.5

Fig. 3. PC analysis for three SPOT images of the same area (A=green band, B=red band, C=near-infrared band)

TRAINING serves for supervised training on multichannel images. An iterative procedure is available for cleaning of training data, which eliminates some of the subjectivity of human decisions. Class population, mean and covariance matrix for each class are stored in a statistics file.

STATISTICS manipulates with the training statistics: computation of spectral separability matrix, construction of separability tree, merging and/or deleting statistics. Several interclass distances are available.

Separability tree is a binary graph representation of between class distances. It is excellently suited to the philosophy of interactive image processing, because it is the shape of the tree which has primary information for the user, instead of numerical values. The tree is constructed in such a way that in consecutive steps the most similar class pairs (represented by the smallest between class distance) are merged. After m-1 steps all the classes are joined and the tree is obtained. Every node in the tree is characterized by a distance. A proper training category set is represented by such a tree, where the branches are formed by classes which are really similar, while dissimilar classes or group of classes are joined only at large distances. Figure 4 displays the separability tree belonging to the soil - vegetation classification of spring SPOT data. Training areas were selected for wheat, spring and winter barley, alfalfa, rapeseed, bare soil, water and forest. Categories of bare soil, vegetation and water seem to be properly separable. Alfalfa and blooming rapeseed are well separable from the other crops, but mixing of barley and wheat, and forest and wheat can be expected.

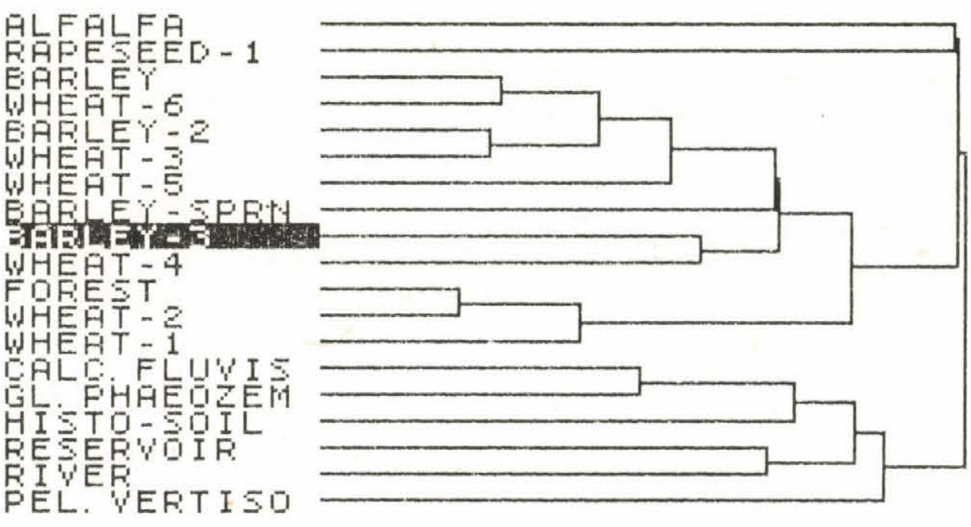


Fig. 4. Separability tree for soil and vegetation categories. (SPOT data, 04-May-1986)

To produce a thematic image based on training statistics a minimum distance CLASSIFICATION is available. A threshold can be used to set the amount of unclassified pixels.

The last logical step of the image analysis is the EVALUATION of the results. The comparison of two different thematic images can serve as accuracy assessment or as change detection.

Some other functions are under development in cooperation with the Technical University of Budapest. CLUSTER is an unsupervised means of training statistics generation, by finding data groups in the feature space. An ISODATA-type iterative algorithm is used with manual as well as with automatic selection of initial cluster centres. After each iteration the user can visually examine the resulted cluster map and can check the corresponding set of statistics. It is possible to delete or merge clustèrs in each iteration step.

Two types of SMOOTHING are going to be used. For image files a median filter is recommended, which preserves edges, but filters out random noise. For thematic files a minimum-area filter is available. The user can define the minimum size of the contiguous area for each class. In the case of a small area it is reclassified to the class, that dominates the first order neighborhood of the area. It is possible to inhibit any of the possible class transitions.

ACKNOWLEDGEMENTS

This work has been supported by the Ministry of Agriculture and Food and the State Office of Technical Development, Hungary. The application results are part of the Programme d'Evaluation Preliminaire SPOT (PEPS) project number 64.

LITERATURE

BüTTNER et al.: Comparative Study of Crop and Soil Mapping Using Multitemporal and Multispectral SPOT and Landsat Data, in: SPOT-1 Image Utilization, Assessment, Results, Cepadues Editions pp.99-106, 1987.

HAJóS, T. - BÜTTNER GY. - KORÁNDI M.: Remote Sensing Software Package for Satellite Image Processing, 38th Congress of the International Astronautical Federation, IAF-87-143, Brighton, 1987.

Methods for Real Time Motion Analysis

Erik Granum & Henrik I. Christensen

Laboratory of Image Analysis
Institute of Electronic Systems
Aalborg University
Denmark

Abstract

Methods for image sequence and motion analysis can largely be grouped under the following 3 headings: Fourier methods, Optical flow and Token matching. These approaches are reviewed, compared and evaluated for real time applications.

Token matching seems most promising in this respect. Matching methods and consequences of using low and high level tokens, respectively, are evaluated.

Still, - motion analysis in real time by 1988 depends on restricted scenaria and special algorithms and/or special hardware. Various possibilities are considered.

In the end, processing speed is by far the only problem. Development of reasonable methods (with regard to complexity) which can actually take advantage of observed motion patterns, handle occlusions and predict motion for tracking and collision avoidance, etc. are possibly more interesting challenges.

An experimental system which addresses these problems and has the potential of operation at TV-frame rate is presented.

1. Introduction

Motion may be considered as an essential perceptual primitive in a dynamic world, serving a variety of perceptual tasks, e.g. shape from motion etc. In this context, however, we will only consider motion for "its own sake".

Motion implies changes in time and our means of analysing these changes by computer is typically through time varying information in the image plane of a camera. The image plane is a projection of a scene and motion "signals" may be due to time variations in the scene and/or due to variations of camera position, orientation and/or adjustments (e.g. zoom). In any case, the motion information we can register in the image plane will always be relative to the camera.

In the general case it is useful to classify the possible scene phenomena we can observe into at least 3 groups:

- Static "objects"
- Moving "objects"
- (Sudden) events.

Static "objects" introduces no new challenges, while "sudden events" will present so much of a challenge to any perceptual system that it will be left in a state of surprise for a while after such an event has occured.

We concentrate on the moving objects, and to monitor these in a digital system we need samples as a function of time, i.e. repeated capture of complete images. The result is an image time sequence, which can be analysed by looking for changes from image to image in the sequence.

Performing this analysis in real time means firstly to process data as fast as they are acquired. Secondly, it means that images are sampled fast enough to allow for "unambiguous" analysis of the changes of interest. These considerations make criteria for real time motion analysis application dependent. More importantly though, they allow for the use of concepts such as "smoothness of motion" and "path coherence", which restricts the variety of possible motion patterns to be considered. From accumulated descriptions of observed motion patterns, - motion histories -, reasonable estimates of what to expect can be established.

By following this philosophy we may change the very time consuming problem of relating detailed information of pairs of successive images into an appeearently simpler problem of checking whether expected changes actually happened. I.e. some basic image analysis is exchanged for some higher level perceptual functionality.

Most of these ideas have been around for some time, but to come back to what has actually been done so far, let us consider the following perspective.

- Single frame methods/systems corresponds to traditional image analysis. Images are usually analysed seperately and possibly to the smallest detail.
- Dual/triple frame methods/systems are the type which this paper is mainly about. Pairs of images are analysed in quite some detail.
- Multiple frame or continuously operating methods and systems are what the future is likely to be about.

In the following section 2 the "traditional" principles of motion analysis are reviewed and compared, - in particular in respect of real time application. Some of the problems of dual frame approaches are addressed by "model driven motion analysis" in section 3, and section 4 describes a specific implementation and some results. Speculations and attempts regarding future research are presented in section 5, before the paper concludes with a few remarks.

2. Motion Analysis Principles

A wide spread interest in motion analysis can be seen from the large variety of methods which have been reported in the literature. For surveys see Aggerwal (1987) and Nagel (1981, 1983). These methods are usually catagorized under one of three major headings (Huang, 1981):

- The Fourier principle
- Differential methods
- Token tracking.

The methods are reviewed and their current potential for real time implementation is evaluated.

Fourier Principle

If we consider a two dimensional signal $f(x,y)$ then a simple translation $v = (x_o,y_o)$, which corresponds to the signal $g(x,y) = f(x + x_o, y + y_o)$, may be expressed through the Fourier transform. Let F(u,v) be the Fourier transform of f(x,y) and G(u,v) the transform of g(x,y). It is then easy to show that the transforms of F and G are related by

$$G(u,v) = F(u,v)\ e^{+j2\pi(ux_o+vy_o)}$$

This means a translation in the time domain corresponds to a phase shift in the frequency domain. Rotations map into a rotation of the transform in the frequency domain, while scaling, i.e. motion in the z-direction, is much more complex.

One of the primary problems of the Fourier methods is that they can only be applied to image segments where all pixels have the same motion. If we have several objects which move independently, they must be segmented and analysed separately. The image is segmented somehow and the background is padded with zeros (Huang 1981). This problem usually complicates the analysis and the segmentation itself implies one of the other principles. Another problem with the Fourier methods is speed, i.e. computing an FFT for a 512 x 512 image requires a substantial amount of computational resources. Using available technology one may achieve processings speeds up to the order of 1 image/second.

For examples on systems employing Fourier based motion analysis see Nagel (1981) and Kabuka & McVey (1986).

Differential Methods

The simplest methods of this category is image differencing methods where pairs of time sequential images are subtracted. In this situation the time invariant components will vanish and the resulting images will only contain noticeable values in regions associated with moving objects. The principle is illustrated for a 1-D example in Figure 1.

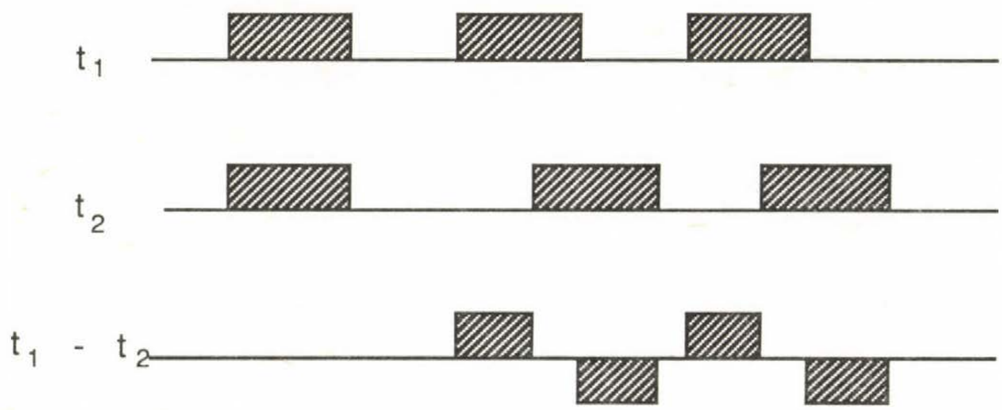


Figure 1 Example illustrating the image differencing principle

This principle can be very fast as one may use special hardware to substract time-sequential images. The primary problem is usually to establish the relation between the original image and the difference-image. In Figure 1 the movement of each object results in two non-connected regions, and the result must be postprocessed before the final result can be obtained. Several have reported algorithms for this purpose, for examples see Wiklund & Granlund (1987) and Gonzales & Wintz (1987).

The group of methods which have received the most attention are those based on first order intensity differentials in space and time. These methods are usually referred to as optical flow or image flow methods.

Let us consider an image I(x,y;t) and another image I(x,y;t+dt) acquired dt later; then we may describe a small movement between elements of the images by a taylor expansion of the first image, i.e.

$$I(x+dx, y+dy; t+dt) = I(x,y;t)$$

$$I(x+ dx, y+ dy;t+ dt) - I(x,y;t) = 0$$

$$I(x,y;t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt - I(x,y;t) = 0$$

$$\frac{\partial I}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t}$$

If we substitute u for dx/dt and v for dy/dt we obtain

$$I_x u + I_y u = -I_t$$

$$(I_x,I_y) \, . \, (u,v)^T = -I_t \quad (*)$$

This equation is often referred to as Horns equations (Horn & Schunk 1981). In this approach intensity changes are assumed to be based on motion in the scene observed, hence the illumination is assumed to be stationary. One may relax this condition by applying 2nd order flow field descriptors, as described by Vaxman & Wohn (1985).

The equation (*) expresses the perpendicular velocity, i.e. the component (u,v) are in the direction of (Ix,Iy). Hence the flow v_1 = (u,v) is called the perpendicular flow. The perpendicular flow is rarely the true optical flow as may be seen from Figure 2.

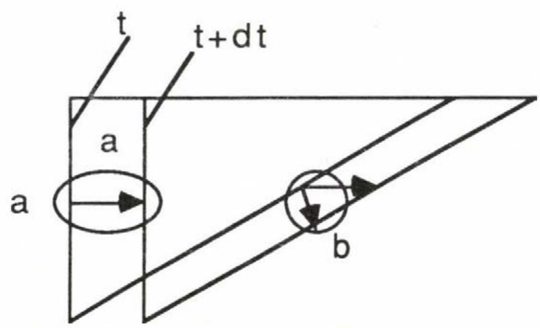


Figure 2 The aperture problem.

In situation "a" in Figure 2 the perpendicular flow equals the true flow as the edge (gradient) is perpendicular to the movement, but in situation "b" the flow differs from the true flow. This problem is usually referred to as the aperture problem as one cannot locally determine the true flow.

To obtain the true optical flow one must postprocess the perpendicular flow field. To extract the flow one usually assume the 3-D motion or the objects are constrained in one way or the other. Hildreth (1982) assume in her smoothest optical flowfield method that all motion is in a plane parallel to the image plane, i.e. no motion in the z-direction or rotations about the x and y axes. Buxton et. al. (1985) assume all scene objects are composed on rigid planar surfaces that may have arbitrary motion. Given this constraint they have developed a matrix equation expressing surface motion and orientation. This equation may be solved if the perpendicular flow is known for 8 image points belonging to that surface.

Heeger (1987) bypasses a number of these problems by extracting the true image flow directly from the image sequence. This is achieved through spatio-temporal filtering using Gabor filters.

The flow methods provides a very detailed analytical description of scene dynamics, which is a valuable scientific property, but it is complicated to extract general and over all descriptors. The optical flow is expensive to compute, but the perpendicular flow may be calculated using simple hardware (Buxton et. al., 1984). The computation of true optical flow requires an interpretation that is difficult to implement in hardware. In the approach by Heeger (1987) the complexity of the spatio-temporal filtering rules out real-time usage.

Token Tracking

In this group of methods single pixels or unique primitives (segments, tokens, edges, lines, objects etc.) are tracked between images. These methods may be divided into three groups, that is:

a) Pixel based matching (correlation)
b) Low level matching
c) High level matching.

For all these methods the motion parameters are calculated on the basis of difference in location which is detected by identifying correponding tokens in two subsequent images.

a) Pixel based matching (correlation)

The matching is based on correspondence of segments using a correlation measure, i.e. a segment from one of the images are crosscorrelated with the other image and the location with maximum correlation is taken as the position of the segment in the latter image. A major disadvantage with this type of matching is sensitivity to orientation, i.e. if the segments can have any orientation, one must repeat the correlation corresponding to each of the possible orientations. This might be very time consuming. For examples, see Gonzales & Wintz (1987) and Aggerwal et. al. (1981).

b) Low level token tracking

Low level tokens are here simple primitives like edge elements, line segments and corners, i.e. descriptors that may be obtained using simple preprocessing algorithms. These low level tokens are then matched using some measure of similarity including minimum distance. A primary problem in token matching is ambiguity, i.e. it is not necessarily obvious which token in one image should match which one in the following image. A simple example is shown in Figure 3. The approach is also rather sensitive to noise. These small tokens may not always be detected, and their uncertain presence introduce more ambiguity. The ambiguity may also arise when two or more objects temporarily overlap, i.e. they may appear as a single object with an entirely new contour. This phenomena is often referred to as occlusion.

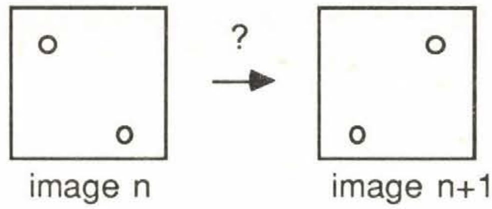


Figure 3 Illustration of matching ambiguity.

The ambiguity calls for the use of costly and sophisticated matching functions, such as probabilistic relaxation. For a detailed analysis of the matching ambiguity, see Ullman (1979). A large number of researchers have addressed these methods. Examples are Roach & Aggerwal (1978), Wiklund & Granlund (1987), Crowley et. et. (1988) and Nagel (1981, 1983).

c) High level token tracking

In the high level approaches the tokens represents larger identified image segments, i.e. entire objects or the like described by feature vectors. The high level tokens are usually obtained by some fast preprocessing methods, e.g. the fast interval processor (Shippey et.al., 1978). Some such special (and possible constraining) approach to the preprocessing is necessary for the potential of these methods in a real time context.

For high level token tracking the problem of ambiguity still applies. However, fewer tokens are usually considered per frame and more sophistication are employed in matching. Also some uses the gathered information on the motion of the tokens to predict the next frame, and some ambiguities are resolved by matching the predicted with the measured frame.

In this category Andersson (1985) uses 1st-4th order moments to describe tokens, while Sethi & jain (1987) use object centroids combined with a special optimization method. Chow & Agggerwal (1979) describe entire objects by shape descriptors and match these using a simple distance measure. Granum & Munk (1986) use image segments obtained through thresholding together with a rule based matching while Christensen & Granum (1988a) use probabilistic relaxation.

Often the token based methods are based on a fast preprocessing of the images that may speed up processing. Again the real-time performance is only obtainable using special hardware and by constraining the scenario. (Crowley et. al., 1988) has reported a system based on linesegment which is capable of analysing 12 images/sec. Here the major problem is occlusions, i.e. the system is unable to handle occluding scene objects. Christensen & Granum (1988a) on the other hand has reported a system based on image segments obtained through thresholding which is capable of handling occlusions. This system is capable of processing 5 images/second using a conventional UNIX computer.

Summary

In general one may conclude that real time (25 images/sec) performance only is obtainable using special hardware, a constrained scenario and/or special (engineered) algorithms. In 1988 the token tracking methods seem to offer the most promising approach to real-time motion analysis.

3. Model driven motion analysis

The major problems in token tracking methods are ambiguity and occlusions, as pointed out in previous section. For methods based on low level tokens some of the primitives may disappear once in a while due to noise, i.e. the preprocessing algorithms used for line extraction or the like are normally sensitive to image noise. These occasional losses of tokens lead to ambiguity in the matching. Tokens may also vanish temporarily due to occlusion, i.e. objects which overlap. The sensitivity may be reduced by using higher level tokens such as image regions, entire contours, objects etc.

To employ as few computational resources as possible one wants to sample as infrequent as possible. This requirement contradicts the ambiguity requirement, i.e. objects will move even further between sampled images if the sampling frequency is low. This kind of problem will always exist in systems where samples acquired at different times are compared. In conventional control theory this problem is often solved by modelling the system and employing prediction with this model to foresee future changes, i.e. the Schmidt regulator (Aaström & Wittenmark, 1984). By employing prediction one now compares samples which coincide in time, and samples will now be less ambiguous if the dynamic system is well behaved.

A similar approach may be employed in token tracking systems. I.e. in the analysis of an image sequence we can register the motion history of individual tokens, and this information may form the basis for a dynamic model of the world. We may use this model for prediction of future frames, if we assume that we sample sufficiently frequent. This latter requirement can be expressed by the "smoothness of motion" and "path coherence" constraints defined by Sethi & Jain (1987). From the discussion above it appears that we should be able to solve a major part of the ambiguity problem by employing this model driven approach together with some kind of high level tokens. A typical block diagramme for such a system is shown in Figure 4.

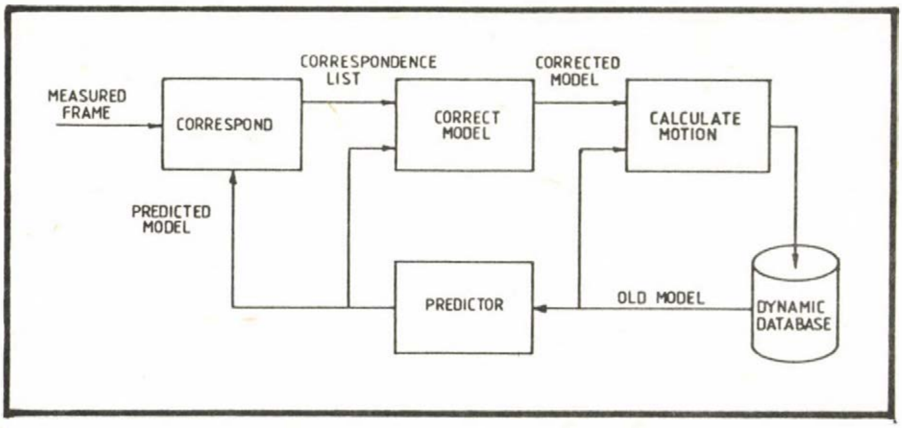


Figure 4 Block diagram for a model driven motion analysis system.

A number of reports on the use of this approach with success exists, e.g. Chow & Aggerwal (1979), Granum & Munk (1986), Wiklund & Granlund (1987), Christensen & Granum (1988a, 1988b) and Crowley et. al. (1988).

4. A sample system (DANTE)[1]

[1] DANTE = Dynamic ANalysis based on Token Extraction.

This section briefly outlines the system of Christensen & Granum (1988a) and present some initial results.

In this system the tokens are image regions (objects) that have been segmented using thresholding, region growing or the like. The image regions are preprocessed using the fast interval processor (Shippey et. al., 1981), and the input to the token tracker is a feature vector for each of the image segments. This vector contains position of centroid, size and shape description. Based on this input and the design criterias listed below a structure as shown in Figure 4 was deviced.

a) Objects (= tokens) of the observed frame are to be matched with an estimated version of the same frame.

b) An object in the observed frame may correspond to two or more objects currently occluding.

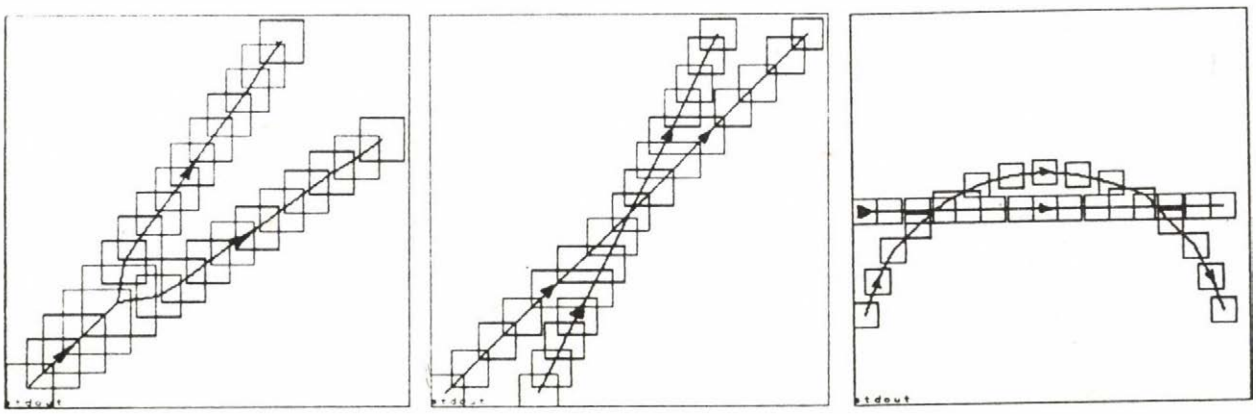
c) An object in the estimated frame is a <u>single</u> object, if and only if, it has previously been observed as such.

d) The maximum number of objects per frame is limited to, - say 10.

e) Standard hardware and software should suffice for implementation, and time of execution should be less than one frame interval.

In this system the dynamic model is built from the feature vectors received from the preprocessing combined with the dynamic information obtained by the token tracker (velocity and acceleration). Based on this model the next frame is predicted using a 2nd order taylor expansion. The actual matching is performed using probabilistic relaxation. The matching result is used in an update of the predicted model and in the calculation of dynamic information that can be used for updating of the dynamic model.

This approach has proven to be able to handle occlusions, and similar ambiguous situations. To illustrate this a number of testsequences are reproduced below in Figure 5.



(a) (b) (c)

Figure 5 *Results Obtained with DANTE.*
The rectangels are a super-position of the measured tokens per frame while the solid lines connects the centroids of the tokens in the dynamic model. i.e. the lines indicate the interpretation obtained by the system.

In figure 5.A two objects occludes initially, but as the sequence progresses the objects move apart. From the system interpretation we may see how the system replaces the token representing the occlucing objects with the separate ones as soon as they are detected. In figure 5.B two objects that are separate initially occlude in the middle, but become separate again in the latter part of the sequence. From the traces of the model centroids we may see how the system maintain the separate models during the occlusion, i.e. due to the prediciton it has foreseen the situation. In Figure 5.C a similar situation is illustrated for an object performing straight line motion while the other moves in a semi circle, here two occlusions occur, but the traces clearly show that the system copes with this. These examples demonstrate that the system is capable of handling ambiguous situations such as occlusions. For a more detailed description, see Christensen (1987) and Christensen & Granum (1988a)

5. Future research

The systems/approaches which have been described so far bave all been based on analysis at a single level of spatial resolution. This may be sufficient for uncomplicated motion patterns such as circular and straight line motion, but if we want to analyse a secnario as shown in Figure 6, however this approach may be inadequate.

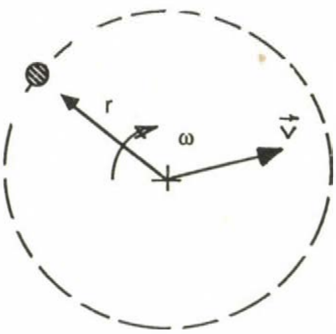


Figure 6 A scenario with a compound motion pattern.

In Figure 6 a small object is attached to a circular plate which has the same reflextance properties as the background. The circular plate has the radius r and rotates with the angular velocity ω in addition the entire plate moves with the translatoric velocity v. If we trace the object through a number of images we may obtain a path as shown in Figure 7.

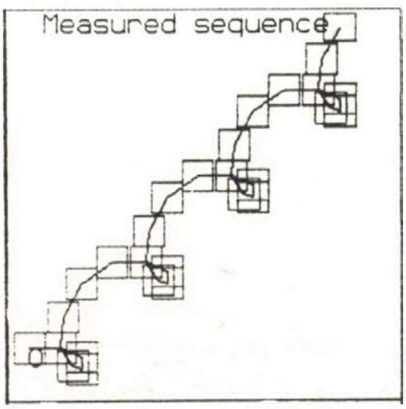


Figure 7 A composite motion path

If we analyse the sequence at a single spatial resolution it may be difficult to relate the obtained descriptors to the real world phenomena. On the other hand we may choose initially to extract the translatoric motion of the plate at a low resolution and then use this information to track the plate. In doing this we may extract the circular motion at a higher spatial resolution. This approach introduces a hierarchical motion description which is potentially easier to relate to phenomena in the world percieved. In the general situation one may employ a pyramid of images, where the top represents an image at a very low resolution while the bottom represents the same image at the highest spatial resolution, e.g. a scale space approach, see also (Rosenfeld, 1984). A model driven motion analysis structure for this kind of processing is shown in Figure 8.

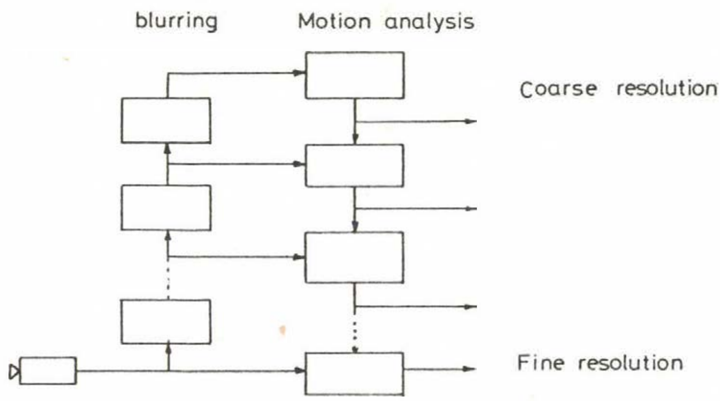


Figure 8 Structure for a multi resolution motion analysis structure.

Some initial experiment using this kind of procesing has been reported by Christensen & Granum (1988b). Here only two resolutions are employed as shown in Figure 9.

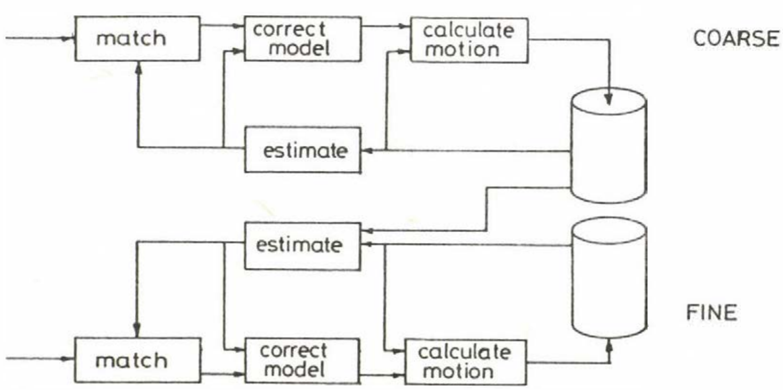


Figure 9 A system used for initial experiments in multiscale motion analysis.

This system has been tested on the sequence in figure 9. The result obtained at the finer resolution is shown in Figure 10, i.e. from the tracking of the centroid of the plate we obtain the circular motions.

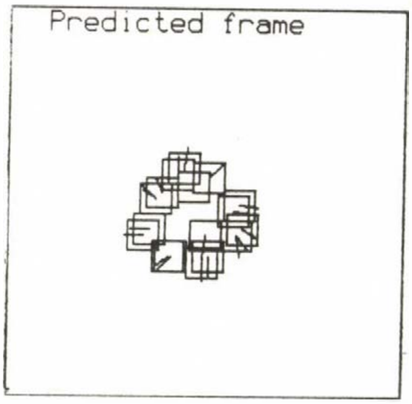


Figure 10 Result obtained with a bilevel motion analysis system

The result indicates that this kind of approach may produce motion descriptors that are easier to interpret, but a lot of work still remains in this area before the true potential and possible generality can be established.

6. Summary

We have discussed various alternative approaches to motion analysis and found that the token tracking methods currently seem most promising for real time application. Particularly important though, it seems to be that temporal context can be used to reduce ambiguity. Under the label of a "Model Driven Approach" a sample system was presented. It accumulates motion history etc. in a dynamic "world model" and uses this information to predict next frame for matching with the measured frame. It was shown that using this model, also allowed for copying with occlusions.

Eventually some speculations on motion pattern descriptors in scale space was illustrated with an implemented example.

References

Aaström, K.J. & Wittenmark, B., *Computer Controlled Systems*, Prentice-Hall Inc., Englewood Cliffs, N.3, 1984.

Aggerwal et. al., *Correspondance Processes in Dynamic Scene Analysis*. IEEE Proceedings, Vol. 69, No. 5, pp. 562-571, May 1981.

Aggerwal, J., *On the Computation of Motion from a Sequence on Monocular or Stereo Images, - An Overview*. In: Machine Intelligence and Knowledge Engineering for Robotics

Applications, A.K.C. Wong & A. Pugh (Eds), NATO ASI series F. No. 33, Springer Verlag, 1987.

Andersson, R.L., *Real Time Gray Scale Video Processing using a moment Generating Chip,* IEEE Jour. Robotics and Automation, Vol. RA-1, No. 2, June, 1986.

Buxton et. al., *Machine Perception of Visual Motion,* GEC Journal on Research, vol. 3, no. 3, 1985.

Buxton, B., *Computer Vision-Image Formation*, Lectures at Queen Mary College, Jan. 1984.

Buxton, et. al., *Parallel computations of optic flow in early image processing*, IEEE Proceedings, Vol. 131, No. 6, PP. 593, Oct. 1984.

Chow, C.K. & Aggerwal, J.K., *Computer Analysis of Planar Curvelinear Moving Images,* EEE Trans on Comput., Vol. C-26, pp. 179-185, Feb. 1977.

Christensen, H.I. & Granum, E., *On Multi Scale Motion Analysis*, proceedings of NOBIM Konferencen, Noršk Regnesentralen, Oslo, Norway, pp. 83-88, June 1988b.

Christensen, H.I. & Granum, E., *On Token Matching in Real-time Motion Analysis,* BPRA 4th ICPR, and Lecture Notes in Computer Science, Vol. 301, J. Kittler (Ed.), Springer Verlag, New York, pp. 448-457, 1988a.

Christensen, H.I., *Monitoring Moving Objects in Real-time,* M.Sc. Thesis, Institute of Electronic Systems, Aalborg University, June 1987.

Crowley, J.L. et. al., *Measuring Image Flow by Tracking Edge-Lines*, submitted to 2nd ICCV, Florida, Dec. 1988.

Gonzales, R.C. & Wintz, P., *Digital Image Processing*, 2nd edition, Addison Wesley, Massachusetts, 1987.

Granum, E. & Munk, K.H., *Monitoring Moving Objects in Real-time*, BAG-memo 8607, Insititute of Electronic Systems, Aalborg University, March 1986.

Heeger, D.J., *Optical Flow from Spatio-emporal Filters,* Procs. 1st ICCV, IEEE, pp. 181-190, June 1987.

Hildreth , E.C., *The Measurement of Visual Motion,* 1983 ACM Distinguised Dissertation, MIT Press, Cambridge Mass., 1982

Horn, D.J., *Determining Optical Flow*, Artificial Intelligence, Vol. 17, Aug. 1981.

Huang, T.S., *Image Sequence Analysis*, Springer Verlag, New York, 1981.

Kabuka, M.R. & McVey, E.S., *Input-output characteristics for Image Transducers,* IEEE Journal on Robotics and Automation, RA-2, No. 2, pp. 106-110, June 1986.

Nagel, H.H., *Image Sequence Analysis: what can we learn from applications?*: In: Image Sequence Analysis, T.S. Huang (Ed.), Springer Verlag, New York, pp. 19-213, 1981.

Nagel, H.H., *Overview on Image Sequence Analysis*. In: Image Sequence Processing & Dynamic Scene Analysis, T.S. Huang (Ed.), NATO ASI serie F. No. 2, pp. 2-39, 1983.

Roach, J.W. & Aggerwal, J.K., *Determining the movement of objects from a series of images*, IEEE Trans on PAMI, PAMI 2, 1980.

Rosenfeld, A. (Ed.), *Multiresolution Image Processing and Analysis,* Springer Verlag, New York, 1984.

Sethi, I.K. & Jain, R., *Finding Trajectories of feature points in a monocular image sequence,* IEEE Trans on PAMI, PAMI 9, No. 1, pp. 56-73, jan. 1987.

Shippey, et. al., *A fast Interval Processor*, Pattern Recognition, Vol. 14, Nos. 1-6, pp. 345-365, 1981.

Ullman, S., *The Interpretation of Visual Motion*, MIT Press, Massachusetts, 1979.

Waxman, A.M. & Wohn, K., *Contour Evolution, neighbourhood deformation and global image flow: planar surfaces in motion*, Int. Journal Robotics Research 4., No. 3, pp. 95-108, 1985.

Wiklund, J. & Granlund, G., I*mage sequence analysis for object tracking*, Procs. 5th SCIA, Stockholm, June 1987.

Experiments with the Recursively Implemented Canny Edge Detector

J. Fiser(*) and A. Lerch(**)
(*)Technical Univ. of Budapest,
Dept. of Electron Devices
1111 Budapest, Goldmann Gy. tér 1-3.
(**)Computer and Automation Institute
1111 Budapest, Kende u. 13-17.

Introduction: Edge detection is a relevant tool of image processing. Its purpose is to transform a grey level image into an "edge image". A lot of algorithms have been developed to achieve better results. Edge detection suffers from the very same problem as most image processing tasks do, namely that different edge detectors can be found optimal for different classes of images. The goal of this pilot experimental study is to select a fast and reliable edge detector which can cope with rather bad quality pictures without loosing important features. A recent paper of Deriche [1] presents a new robust algorithm which is claimed to perform very well in noisy images. In order to evaluate the performance of this algorithm we chose the well-known Prewitt edge detector [2], which is one of the simplest proposed operators. We compared the edge images obtained by the two operators.

I. The Canny edge detector and its modification

One expects an edge operator to fulfill the following criteria:

1. It should respond to every valid edge point, i.e. there should be a low probability of missing real edge

points.

2. It should not respond to false points which are introduced by noise even though they are local gradient maxima, i.e. there should be a low possibility of marking points falsely.

3. Points marked as edge points should coincide with real edge points, i.e. the average (or maximum) distance between the ideal and detected edge points should be small.

These criteria are treated by Canny [3] for a specific edge and noise model. He used a more or less acceptable assumption that the input image is degraded by additive Gaussian noise. The first two criteria result in investigation of the signal-to-noise ratio (SNR). The maximum achievable SNR corresponds to the best detection. The third criterion is the "localization" one, which involves the reciprocal of the standard deviation of the spatial distribution of detected edge points centered at the ideal edge. The examination of these three criteria yields an intuitively clear uncertainty principle between localization and detection, i.e. trying to better localize edges by using smaller operators results in weaker responses. Canny also introduced an additional constraint: the detector should not give multiple responses to one edge.

He used numerical optimization for determining the function class which maximizes the product of the detection and localization criteria under a constraint derived from the multiple response criterion. He approximated the optimal function with the first derivative of a Gaussian whose variance is a parameter of the filter. The 2D filter performs smoothing in edge direction by means of a Gaussian with the same σ. The broader this tunable filter is the better detection and the worse localization we obtain and vice versa.

Recently, Deriche [1] proposed an efficient realization and a small modification of Canny's filter. The Deriche filter is recursive, which means considerable savings in computational effort when applying larger filters. Only a small constant number of operations is necessary in each image point independently of the filter size. Using infinite support filter Deriche concludes that function

$$f(x) = c\, e^{-\alpha|x|} \sin \Omega x \qquad (1)$$

(in the gradient direction) is optimal. Here α and Ω are positive values and c is a constant.

We chose Deriche's algorithm because it has strong mathematical support and is optimal in the case of additive Gaussian noise. Also, its computational requirements are constant at different scales.

II. Investigation

When studying the features of the Deriche edge detector the following questions were investigated:

1. Tuning the filter is done by varying two parameters α and Ω. The behaviour of the filter was studied in the 2D parameter plane. We tried to separate the regions where the filter function oscillates and where one parameter dominates.

2. Several scale filters were tested in synthesized images and the results were compared to the results of the Prewitt operator. The issues considered were detection and localization in the presence of synthesized noise.

3. The performances of the Prewitt and the Deriche filters were compared for bad quality low resolution real-world images. Typical indoor pictures and biological images were digitized into 128*128*6 bit matrices.

4. To avoid too subjective evaluation of the resulting gradient pictures a post-processing phase was added (simple thresholding or thresholding with hysteresis, and nonmaximum

suppression). Automatic threshold selection seemed impossible thus thresholding was performed manually.

III. Results

1. According to Deriche [1], the best trade-off between localization and detection is achieved when the α/Ω ratio is large enough, although it is the worst case in the sense of the multiple response criterion. Our experiments seem to support Deriche's conclusion that sacrifying the multiple-response criterion does not mean any perceivable loss in the operator performance for bad quality pictures. In this case the single parameter α is sufficient for determining the filter behaviour, therefore tuning the operator becomes easier. The exponent in the filter function (1) suppresses the oscillation. We determined the region where the one-parameter-tunability is acceptable. Within this region we calculated the operator widths corresponding to different α values.

When α is comparable to Ω and the filter function oscillates, one can also achieve nearly as good results as in the previous case. However, we concluded that using this region is not advisable because small changes in the parameters can entail drastic changes in the resulting images.

2. Comparing the Prewitt and the Deriche filters <u>of similar size</u> for real-world imagery we experienced that

- the Deriche operator does not indicate more significant edges than the Prewitt operator
- the Deriche operator indicates fewer false edges due to noise; this effect was stronger in better quality images with higher contrast.

Increasing the operator size made the Deriche filter less noise sensitive as expected, but this suppressed a number of high frequency edges. The edge images became clearer, only the strong edges remained. However, effects due to strong blurring, such as rounding-off and bad localization, became significant.

We conclude that in the case of relatively low resolution, low contrast pictures using the Deriche edge detector does not offer considerable advantage compared with the very simple Prewitt operator . The reason presumably is that low dynamics of pictures has stronger effect than noise. In certain applications the illumination of the scene can be similarly weak, therefore using simple hence computationally less demanding edge detectors should be considered.

In our investigations we did not exploit the advantages of multi-resolution edge detection. Selecting the "best-tuned" operator is quite difficult. Some approaches examine the behavior of the edge image in scale-space, i.e. continuously changing the operator size. Since Deriche's filtering is of constant complexity in scale-space, his filter is very favourable for such purposes.

References

[1] R.Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", International Journal of Computer Vision, pp. 167-187, 1987.

[2] J.M.S.Prewitt, "Object Enhancement and Extraction", in Picture Processing and Psychopictorics, B.S.Lipkin and A.Rosenfeld (Eds.), N.Y., Academic Press, 1970.

[3] J.Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, Nov. 1986.

Pyramid Research for Image Analysis

W. G. Kropatsch
Institute for Image Processing and Computer Graphics
Wastiangasse 6
A–8010 GRAZ
AUSTRIA

1 Introduction

The analysis of digital images can be seen as a translation process. At the input there are large numerical arrays of (grey/colour/...) values that are successively transformed into symbolic/semantic representations that form the basis for description and interpretation at the other end of this process. One of the first steps in this process is the segmentation of the image into homogeneous regions and continuous contours.

To handle the large amounts of data, hierarchical computational structures are necessary to achieve real time processing. Multiresolution or "pyramid" approaches to image processing and description have become of considerable interest over the past few years [19]. Pyramid structures represent information about an image at a set of exponentially decreasing scales. The represented information will take the form of a series of abstractions that describe the content of the image. At the same time, the information will allow reconstruction of approximations of various degrees to the image. [1], [6], [17] give good overviews on the pyramid research of the last few years.

2 Numerical Computations

Grey level pyramids are commonly constructed by recursive subsampling of low-pass filtered images. The values stored in the cells of the pyramid can be interpreted as parameters of a model that averages the light intensity reflected by the corresponding region in the image plane [7]. In a general case any set of numerical (statistical) values can be used if necessary.

Statistical pyramid computations use basic operations such as summing, counting, averaging, and computing the variance, which can be done in parallel on a pyramid architecture [3]. These operations can be extended to compute least square fitted polynomials, statistical moments, and to detect bimodality and anomaly. One step further is the detection of "coarse" features. Several methods are described in the literature that extract primitive regions such as bright spots or blobs in grey level pyramids [2].

3 Edge Detection and Contour Coding

Edges can be detected at all levels of a grey level pyramid in parallel and stored in the original structure (e.g. [20]). There are several problems with this approach:

(A) edges are normally represented by two measurements (magnitude and orientation) rather than by symbolic codes;

(B) continuity checking is very difficult and needs sequential (horizontal) processing;

(C) relations between edges at different levels of the edge pyramid are not obvious;

(D) edges represent relative changes in grey value, information about brightness or darkness or colour is lost.

Problems (A), (B), and (C) are solved by the concept of binary coded curve relations [15], [10], which is defined on a square grid. One curve crosses the sides of a square twice. The presence of such a curve in any square cell is stored by marking the two sides connected by the curve.

Using curve relations as the contents of the pyramid cells, a parallel symbolic reduction procedure is able to generalize sets of connected curve relations of a 2×2 reduction window into one larger square cell without destroying their connectivity. Furthermore, it can be shown that the resulting curve pyramid sorts the sets of curve segments that are connected in the base of the pyramid by their spatial extend, e.g. long curves are propagated to high levels of the pyramid whereas short curves remain in the lower levels [11], [13].

The geometrical structure of this curve pyramid differs from the conventional $2 \times 2/4$ (quadtree) structure in that the number of cells is reduced by a factor of two instead of four which allows adjacent 2×2 reduction windows to overlap by one corner cell [16]. The axes of the reduced grid are rotated by 45 degrees with respect to the finer grid. Crowley was the first to use this structure [4], [5].

The hierarchical structure code (HSC) of [8], [9] is a coding scheme on a hexagonal pyramid similar to the curve pyramid.

4 Dual Pyramids

The approach in [14], [12] overcomes also problem (D) by a concept involving two geometrically dual pyramids: the nodes of the $3 \times 3/2$ grey level pyramid contain areal measurements like grey values, the other $2 \times 2/2$ curve pyramid stores symbolic curve codes in the nodes. At every level the two square grids are *dual* to each other.

Elementary operations move the information within a local neighborhood horizontally from one pyramid to the other and vertically up and down in the pyramids. Some examples have shown how to use effectively the complementary information provided by the two pyramids to perform simple image processing tasks. Some recent experiments are reported in [18]. But this is only a small fraction of the possibilities to manipulate semantic/symbolic pictorial entities.

5 Conclusion

The research in this field opens a wide field for further investigations. There are reasonable chances that it could bridge the large gap between the low level numerical image processing and the high level part of image understanding which involves to a great part techniques from Artificial Intelligence and Expert Systems.

References

[1] E. Adelson, C. Anderson, J. Bergen, P. Burt, and J. Ogden. Pyramid methods in image processing. *RCA Engineer*, Vol. 29-6, Nov./Dec. 1984.

[2] R. P. Blanford and S. L. Tanimoto. Bright-spot detection in pyramids. In *Proc. Eighth International Conference on Pattern Recognition*, pages 1280–1282, IEEE Comp.Soc., Paris, France, October 1986.

[3] P. Burt, T. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11(No.12):pp.802–809, December 1981.

[4] J. Crowley and A. Parker. A representation of shape based on peaks and ridges in the difference of low–pass transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI–6:pp.156–170, 1984.

[5] J. L. Crowley and R. M. Stern. Fast computation of the difference of low–pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–6:pp.212–222, 1984.

[6] W. I. Grosky and R. Jain. A pyramid-based approach to segmentation applied to region matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(No.5):pp.639–650, September 1986.

[7] R. L. Hartley. *Multi–Scale Models in Image Analysis*. PhD thesis, University of Maryland, Computer Science Center, 1984.

[8] G. Hartmann. Principles and strategies of hierarchical contour coding. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1087–1089, Montreal, Canada, 1984.

[9] G. Hartmann. Unified description and recognition of continuous contours and regions. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pages 1201–1203, Paris, France, October 1986.

[10] W. Kropatsch. Kurvenrepräsentation in Pyramiden. In W. Kropatsch and P. Mandl, editors, *Mustererkennung'86*, pages 16–51, Oldenbourg, 1986.

[11] W. G. Kropatsch. Curve representations in multiple resolutions. *Pattern Recognition Letters*, Vol. 6(No. 3):pp.179–184, August 1987.

[12] W. G. Kropatsch. Ein Konzept mit zwei sich ergänzenden Pyramiden. In W. G. Kropatsch and P. Mandl, editors, *Mustererkennung'86*, pages 158–171, R. Oldenburg, 1986.

[13] W. G. Kropatsch. Elimination von "kleinen" Kurvenstücken in der 2x2/2 Kurvenpyramide. In E. Paulus, editor, *Mustererkennung 1987*, pages 156–160, Springer Verlag, 1987.

[14] W. G. Kropatsch. Grauwert und Kurvenpyramide, das ideale Paar. In G. Hartmann, editor, *Mustererkennung 1986*, pages 79–83, Springer Verlag, 1986.

[15] W. G. Kropatsch. *Hierarchical Curve Representation in a New Pyramid Scheme.* Technical Report TR-1522, University of Maryland, Computer Science Center, June 1985.

[16] W. G. Kropatsch. A pyramid that grows by powers of 2. *Pattern Recognition Letters*, Vol. 3:pp.315–322, 1985.

[17] L. O'Gorman and A. C. Sanderson. A comparison of methods and computation for multi–resolution low– and band pass tranforms for image processing. *Computer Vision, Graphics, and Image Processing*, Vol. 37(No. 3):pp.386–401, March 1987.

[18] G. Paar. *Beibehaltung von Objekträndern in Bildpyramiden.* Master's thesis, Technische Universität Graz, 1987. Diplomarbeit.

[19] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis.* Springer, Berlin, 1984.

[20] M. Shneier. Two hierarchical linear feature representations: edge pyramids and edge quadtrees. *Computer Graphics and Image Processing*, Vol. 17:pp.221–224, 1981.

MTA SZTAKI Tanulmányok 206/1988 p. 119

An Error-correcting Prewitt Edge Detector in Noisy Images

I. Cseke
Central Research Institute for Physics
H-1525 Budapest, P.O.B. 49.

1. Abstract

Image segmentation is the first step in image analysis. The purpose of segmentation is to divide an image into homogeneous regions (i.e. segments). The most important segmentation technique is the edge detection, which aims to detect the boundaries of homogeneous regions or the abrupt changes of some pixel properties (e.g. gray level). The problem of edge detection is rather complicated because there is no unique edge/image model. Some of the difficulties are caused by noise and the fact that many variations of neighbouring pixels have no edge-significance while some continuous regions may still contain some edge-information. In the past two decades a lot of edge detection techniques were proposed [1]. These algorithms differ significantly in speed, universality, isotropy and noise immunity. Because of speed and universality requirements the commonly used edge detectors are the enhancement/thresholding ones. The results obtained by these

methods are seriously affected by noise. As far as noise immunity is concerned, the optimal enhancement/ thresholding edge detector is perhaps the Prewitt operator [2]. On the other hand the edge magnitude and orientation detected by the enhancement/thresholding edge detectors depend strongly on the actual edge direction (anisotropic feature). This paper investigates the edge magnitude produced by the Prewitt operator and proposes a fast correcting algorithm for edge magnitude. A simple figure of merit is defined for measuring the effect of the magnitude correction in noisy images. Finally some experimental results are presented.

2. Some details

In the case of the Prewitt edge detection method the discrete image array $F(j,k)$ is spatially convolved with two masks $H_i(j,k)$

$$G_i(j,k) = F(j,k) \otimes H_i(j,k)$$

and these values are combined by a point operation

$$G(j,k) = | G_1(j,k) | + | G_2(j,k) |$$

to produce a gradient image (edge magnitude). The next step is thresholding of gradient image

$$E(j,k) = \begin{cases} 1 & \text{if } G(j,k) \geq T \\ 0 & \text{if } G(j,k) < T \end{cases}$$

in order to create the edge map. Thus one pixel is an edge element if its edge magnitude is greater than an appropriate threshold. The edge orientation is calculated by

$$\phi(j,k) = \tan^{-1}(G_2/G_1) .$$

Considering a simple image and edge model we can easily determine the functions

$$A_D = f_1(b-a,\ \theta) \qquad \theta_D = f_2(\theta)$$

where A_D and θ_D denote the detected edge magnitude and orientation produced by the Prewitt edge detector, b-a and θ are the actual edge magnitude and orientation, respectively [3]. From the functions f_1 and f_2 we can yield the actual edge magnitude

$$b-a = A_D \cdot f(\tan\theta_D) \ .$$

Since the gray levels are quantized, we can deduce that $\tan\theta_D$ is also quantized (i.e. having only a finite value-set). So the function f can be implemented with the help of a look-up table and the edge magnitude correction will be rather fast. The minimal size of the look-up table can be determined from the following condition

$$|\ A_D \cdot f(\tan\theta_D) - A_D \cdot f(\tan\theta_{Dkvan})\ | < 1 \ .$$

For studying the error-correcting Prewitt operator we have used different simple test images, which contain central-vertical and diagonal line and in the other case a ring. An independent zero-mean Gaussian noise was added to each test image using different signal to noise ratios. We have defined a simple figure of merit in order to measure the goodness of edge detectors. This figure of merit is based on two required properties of a well-detected edge map, namely on the continuousness and thinness of the edges [4]. The figure of merit is defined as

$$R = w_t + w_l + w_r$$

where w_t is the measure of thinness, w_l and w_r are the measures of continuousness at the left and at the right, respectively. We have to compute these measures at every edge point considering their 8-neighbourhoods and then we have to average them on the complete image. The measure of thinness is given from the number of edge neighbours. The measure of continuousness is computed as follows. The function

$$a(d,d_i) = \begin{cases} \left|\, 4 - |d - d_i| \,\right| & \text{if neighbour } i \text{ is edge pixel} \\ 0 & \text{otherwise} \end{cases}$$

represents the difference between d and d_i edge orientations, where $i = 0,\ldots,7$ and $d, d_i = \{0,\ldots,7\}$ according to the eight main edge directions. The measure of left continuousness is given by

$$w_l = \max \left\{ a(d,d_{i-1}), a(d,d_i), a(d,d_{i+1}), a(d-1,d_{i-1}), a(d+1,d_{i+1}) \right\}$$

where $i = (d + 2) \bmod 8$. We get the value of the measure of right continuousness similarly. The proposed figure of merit does not require the a priori knowledge of edges and it has low computional cost. It ranges from 1 for the ideally detected edges, to zero for the hopeless edge maps. The experimental results show that the derived figure of merit is suitable for qualifying the edge maps. Comparing the Prewitt and the error-correcting Prewitt edge detectors we found essential improvement in quality at not very noisy images, where signal to noise ratio was greater than 5. The experiments were performed on the ATLAS-90 digital image displaying and processing system.

3. References

[1] Levialdi, S.: Edge Extraction Techniques, in Fundamentals in Computer Vision (O.D. Faugeras ed.), Cambridge 1983, 117-144.

[2] Abdou, I.E., W.K. Pratt: Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors, Proc. of the IEEE 67, 1979, 753-763.

[3] Kittler, J.: On the Accuracy of the Sobel Edge Detector, Image and Vision Computing 1, 1983, 37-42.

[4] Kitchen, L., A. Rosenfeld: Edge Evaluation Using Local Edge Coherence, IEEE SMC 11, 1981, 597-605.

MTA SZTAKI Tanulmányok 206/1988 p. 123

QUANTITATIVE CHARACTERIZATION OF ORIENTATION IN BINARY DIGITIZED IMAGES

Imre CZINEGE* and Tamás RÉTI**

* Bánki Donát Polytechnic, H-1081 Budapest, Népszinház u. 8.

** Institute of Industrial Technology, H-1148 Budapest, Fogarasi ut 14.

INTRODUCTION

Several methods are known from the literature describing the oriented microstructures [1-4]. The major part of them is based on manual measuring and evaluating technique. This is why the adaptation of them to the modern image analysing instruments causes difficulties. Coming into details, in the computerized analysis of digitized micrographs the determination of the orientation axis and the rose diagram frequently gives different results - depending on the applied geometrical- topological model and the algorithm.

This paper presents a new, generalised method for the determination of the orientation axis and the degree of orientation suitable for characterisation of images including not only one oriented system of features but two or more systems as well. The measuring technique is based on a special geometrical-topological model of binary digitized image. The determination of the parameters of elementary orientations is carried out by a modified hill climbing algorithm.

CHARACTERIZATION OF THE ORIENTATION

The main problem consists in interpreting the "intersection with a digital test line" on the basis of numerical criteria and subsequently determining the number of intersections. An "analogous image" which can be considered as the plane geometrical model of the digital image is assigned to the "digital shape" represented by "1" elements. We used for our tests the M4 type geometrical model proposed by Réti [5]. The characteristics of the model can be studied in Figure 1. Fig. 1/a shows some 2x2 digitized windows and their geometrical representation while in Fig. 1/b and 1/c we can see an example of a digitized image and its plane geometrical model, indicating also the intersections with a digital test line.

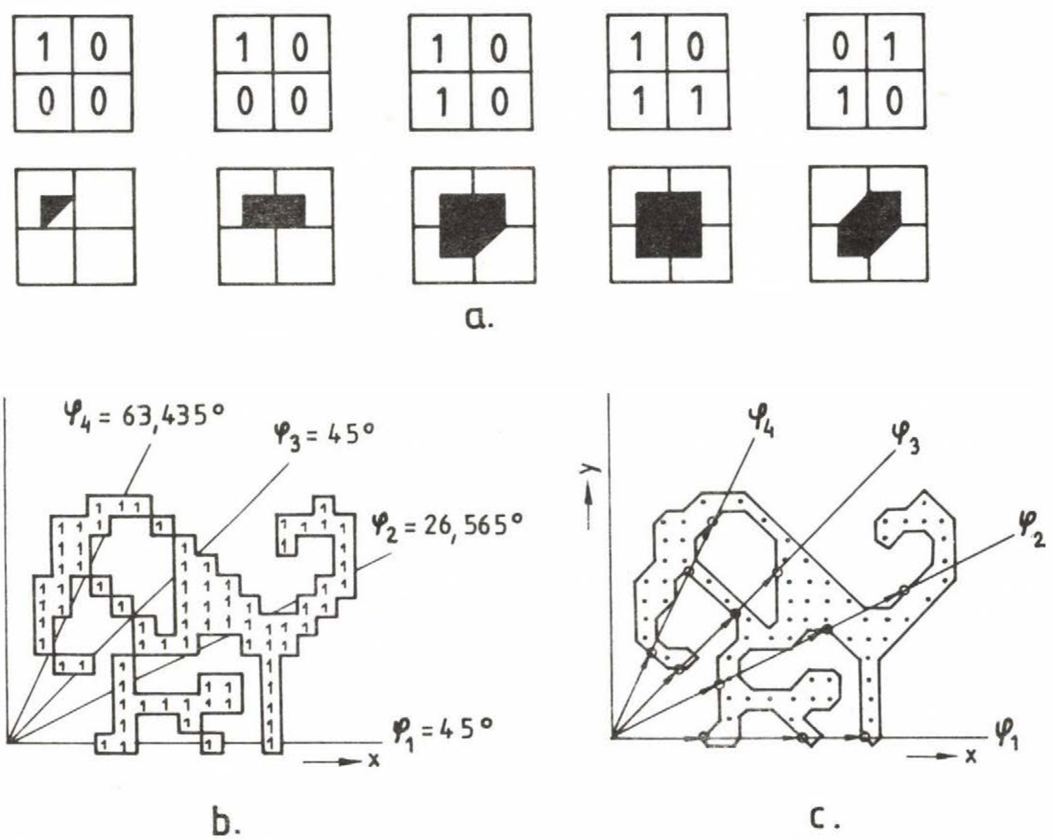


Fig. 1. Intersections with a digital test line
a./ Plain geometrical model. b./ Digitized binary image
c./ Analogue image generated by M4 model

On the binary image the number of intersections were determined by test windows of the sizes 2x2, 2x3 and 3x2. The test windows and the obtained main directions of intersections are presented in Figure 2. The test window technique is described in detail by Hajnal et al [6]. The characteristics of the 2x2 test window (see in Fig. 1/a.) discussed in [6] can be transferred suitably to 2x3 and 3x2 windows.

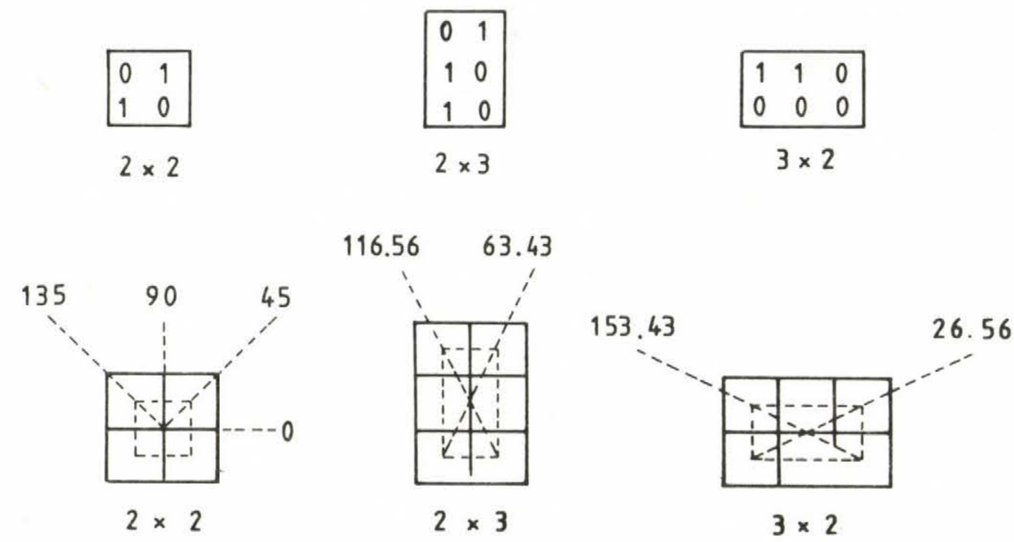


Fig. 2. Test windows and directions of test lines

Scanning the digital image with the windows, the frequency of the various window configurations is obtained and the summation of the adequate windows furnishes the number of intersections in the main directions. Dividing this by the total length of line we obtain the number of intersections per unit length.

DERIVATION OF QUANTITATIVE PARAMETERS

The orientation analysis was carried out by a micro-processor-based image analyzer type HT 680-X (made by HTSZ, Hungary) which produced a digitized image with square raster consisting of 300x400 pixels. After threshold selection the values of the number of intersections per unit length $N_L(\varphi_i)$ were measured in the main directions characterized by the angles $\varphi_1=0°$, $\varphi_2=26,6°$, $\varphi_3=45°$, $\varphi_4=63,4°$, $\varphi_5=90°$, $\varphi_6=116,6°$, $\varphi_7=135°$ and $\varphi_8=153,4°$, then the the mean intercept lengths R_i $(i=1,2,...n)$ defined as

$$R_i = \frac{1}{N_L(\varphi_i)}$$

were computed (n=8).

It is well known from the literature that plotting the number of intersections $N_L(\varphi_i)$ versus φ_i in a polar coordinate system the so called "intercept rose diagram" is obtained while the points of the mean intercept length R_i give the orientation ellipse. It is obvious that if only one characteristical orientation exists in the image the points of R_i function form only one ellipse. From these measured points the parameters of the orientation ellipse can be estimated by the least square method [7].

If the image includes more than one characteristical oriented system of features the function $R_i(\varphi_i)$ can be given as the sum of the ellipses belonging to the different oriented systems. In this case for the evaluation of the orientation we suggest the polar function $R_K(\varphi)$ generated as a sum of K ellipses

$$R_K(\varphi) = \sum_{k=1}^{K} \frac{a_k \cdot b_k}{\sqrt{a_k^2 \cdot \cos^2(\varphi - \varphi_k) + b_k^2 \cdot \sin^2(\varphi - \varphi_k)}}$$

where the variable φ is the polar angle, a_k and b_k are the major and minor axes of kth ellipse belonging to the kth oriented system and φ_k means the angle between the x-coordinate and the major axis a_k. Using the parameters of the ellipses, the direction of the orientation can be characterized by the angle φ_k and the degree of orientation g_k (k=1,2,...K) which is defined as the function of a_k and b_k:

$$g_k = \sqrt{1 - \left[\frac{b_k}{a_k} \right]^2}$$

Minimizing the function H_k the unknown parameters a_k, b_k and φ_k can be estimated by numerical method [8], where

$$H_K = \sum_{i=1}^{n} \left[R_i - R_K(\varphi_i) \right]^2$$

Figure 3/a. shows some examples of high-speed steels with various carbid dispersion and their orientation characteristics [7] supposing that only one oriented system exists. The image signed by K4 is different from K1...K3 as it shows a bidirectional orientation. This image was analysed by the above described method presuming that two characteristical orientations can be found in the image (K=2). Figure 3/b. shows the $R_K(\varphi)$ diagram where the two ellipses characterize the different orientations separately while the outer curve represents the measured points (the sum of the ellipses). For this experimental analysis 21 points of the $R_K(\varphi)$ diagram were measured and the 6 orientation characteristics $(a_1,\ b_1,\ \varphi_1,\ a_2,\ b_2,\ \varphi_2)$ were estimated with the aid of the hill climbing algorithm [8].

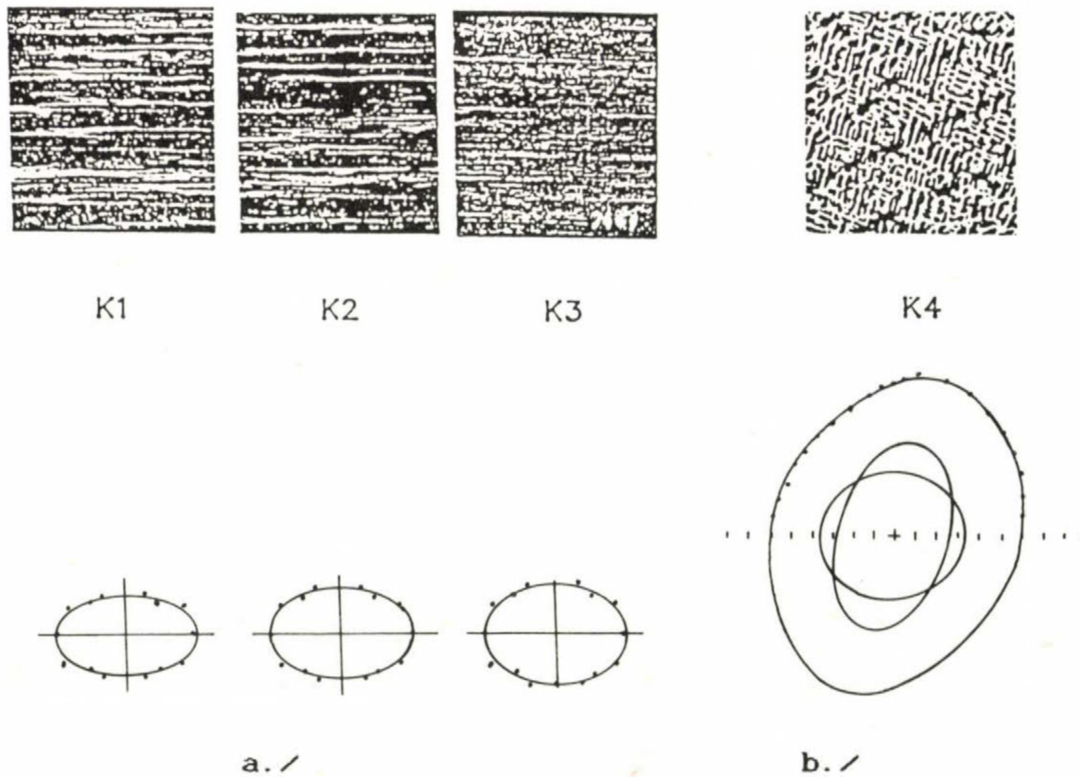


Fig. 3. Experimental results
a./ Orientation of high-speed steels
b./ Orientation ellipses of image K4

SUMMARY

From our theoretical considerations and experimental results the following conclusions can be drawn:
If there exists only one orientation axis in the binary image (case K=1), then the parameters of the orientation ellipse a_1, b_1 and φ_1 can be easily estimated by the well-known least square analysis [7]. If the image morphology is characterized by two main direction of orientation (case K=2) then the parameters a_k, b_k, and φ_k (k=1,2) can be computed with the new method proposed in this paper. Generalising the problem for K>2 that means that if more than two orientation systems are present in the image the characteristics of the elementary oriented systems can be calculated in similar way.

References

1. Stoyan, D., Peisker, D., Ohser, J.: Neue Hütte 26 (1981) H.2. 65/69.
2. Saltikov, Sz.A.:Sztereometricseszkaja metallografia. Metallurgizdat, Moscow, 1958.
3. Czinege,I., Réti, T.: Determination of Local Deformations in Cold Formed Products... Proc. of the 18th Int. Machine Tool Design and Research Conf., London (1977) 159/164.
4. Hilliard, J.E.: Trans. AIME 224 (1962) 1201/1214.
5. Réti, T.: BKL-Kohászat 119 (1968) Nr.1. 14/23.
6. Hajnal, M., Könyves-Tóth, M., Réti, T.: ACTA IMECO (1982) Proc. of the 9th IMECO Congr. W-Berlin, 283/292.
7. Réti, T., Czinege, I.: Simple computerized methods for evaluation of orientation in digitized microscopic images. Acta Stereologica (1987) Vol. 6. 579/584.
8. Nelder, J.A., Mead, R.: The Computer Journal (1965) Vol. 7. 308/313.

IMAGE ANALYSIS METHODS IN THE AGRICULTURAL APPLICATIONS OF REMOTE SENSING

G. Csornai

FÖMI, Remote Sensing Centre
H-1149 Budapest
Bosnyák tér 5.
HUNGARY

The availability of multispectral satellite data has increased our capability to acquire most important information on the stage and development of the major crops in Hungary. The adequacy of these data lies in their parameters: low cost, timeliness, area coverage - typically 35000 km^2 - well suited spectral channels and spatial resolution (0.4-0.1 ha/pixel) plus the ease of acquisition through data distributors in Europe and worldwide. The area survey, the crop stage and development assessment and the reliable yield forecast are those basic problems that can be addressed by the analysis of remotely sensed data.

Different tasks and preliminary results in crop monitoring

The area survey that provides framework to the other tasks should be solved first. For ecologically homogeneous 100000-300000 hectare areas (land systems) our classification methods gave good results when using monotemporal Landsat

MSS data (Csornai et al, 1983a). Recently, the methods have been tested on a complex county, Hajdú-Bihar with approx. 600000 hectare area to compare remote sensing method to the traditional field sample based information system. The results (Csornai et al, 1987b) are encouraging being close for wheat, corn area estimation (2-4%) to that of the existing state information collection system. The crop stage and development monitoring have resulted in only relative classification maps and figures as opposed to correlated biometrical measurements (phenology, leaf area, biomass etc) (Csornai et al, 1983b).

A fairly complex set of pattern recognition techniques based methods has been developed to facilitate complex regional (600-2000 thousand Ha) area surveys. User class and subclass definition is helped by the effective ISODATA type clustering method that can control the size, population and the shape of data clusters to some extent. Different cluster probability density function based measures are used to control the cluster merge process and help the class configuration assessment procedure. Cautiously using the mixture assumption per user class 90-95% classification accuracies could be achieved with the common maximum likelihood and Bayes classifiers using even monotemporal Landsat MSS data (Csornai et al, 1983a, 1984).

Data processing system

The data processing system serves as a basis during the development of remote sensing based crop information system (Fig.1). Later, the controller unit should become a set of expert systems each dedicated to one of the basic crop monitoring tasks.

The subsystems are at different development stages and the integration level of the whole system is low at present.

The image analysis subsystem (IAS) has been the most important part so far being the basic tool for the agricultural land use mapping studies. It provides for the major geometrical, registration, radiometrical, supervised and unsupervised classification plus error assessment procedures. First it was implemented on a TPA 1148 (PDP 11 series equivalent) computer configuration having recently switched to microVAX compatible processor. All the necessary human interaction in the image analysis is done via a Pericolor raster terminal.

The reference data handling subsystem (RDS) facilitates the input and data management of all the necessary ancillary data that are originally in map form, including ground truth, soils maps, meteorologic and topographic maps. The data base system is of vector type with the attributes coupled. RDS provides a small, but quite flexible set of features (Kiss, 1988) that are customary in a vector based geographic information system (GIS).

Partly because of the vector principle, most of the geographic information systems don't support multidimensional, multisource analysis and modelling enough. Therefore a separate subsystem for geographic data modelling (GDMS) was devised. This division of the common GIS functions into two groups supports image analysis and modelling better when different data sources are involved creating a many dimensional problem of the parameters that are spatially distributed.

The term geographic information and modelling system (GIMS = RDS + GDMS) seems to be more appropriate.

The IAS and GIMS are planned to be in a fairly symmetric relation. One direction when IAS gives result maps etc. to the GIMS is common. Inversely, the integration of a geographic information system (GIS) into the image analysis (Csornai et al, 1987a) is hoped to result in not only stabilized, high classification accuracies using context dependent methods but also a relative independence of the factors (soils, terrain, climate, cultural practices etc.) that can influence estimates and mapping. This latter method makes use of the unique feature of the Hungarian agriculture (Csornai et al, 1987b).

Suggested further research

The Landsat data time series were certainly superior to monodate data, but the potential of the former is still to be exploited using a model, that is suited to vegetation development better than the one has been used. The sampling problems and the complementarity of multiple new generation data sources (Landsat TM, SPOT etc.) are going to be investigated in the near future.

The whole image analysis and GIS subsystems should be integrated on the microVAX compatible minicomputer, making use of the raster image display more effectively.

References

1. Csornai G., Dalia O., Gothár Á., Vámosi J.(1983a): Classification Method and Automated Result Testing Techniques for Differentiating Crop Types Proc. Machine Processing of Remotely Sensed Data, West Lafayette, USA
2. Csornai G., Vámosi J., Dalia O., Gothár Á. (1983b): Vegetation Status Assessment and Monitoring in Agricultural Areas by Remote Sensing XXXIVth IAF Congress, Budapest
3. Csornai G., Dalia O. (1984): Efficient methods for the classification of

multispectral satellite images, (in Hungarian), 5th Earth Photo Seminar, Budapest

4. Csornai G., Dalia O. (1987a):
   Remote sensing based crop survey and information system, conference on "Remote sensing applications", (in Hungarian), Békéscsaba
5. Csornai G., Dalia O., Farkasfalvy J., Nádor G., Vámosi J., Zabó P. (1987b):
   Research and development results on the regional crop survey methods, report (in Hungarian), Budapest
6. Kiss Z. (1988):
   REBEKA - a reference data handling system, Users' Manual (in Hungarian), Budapest

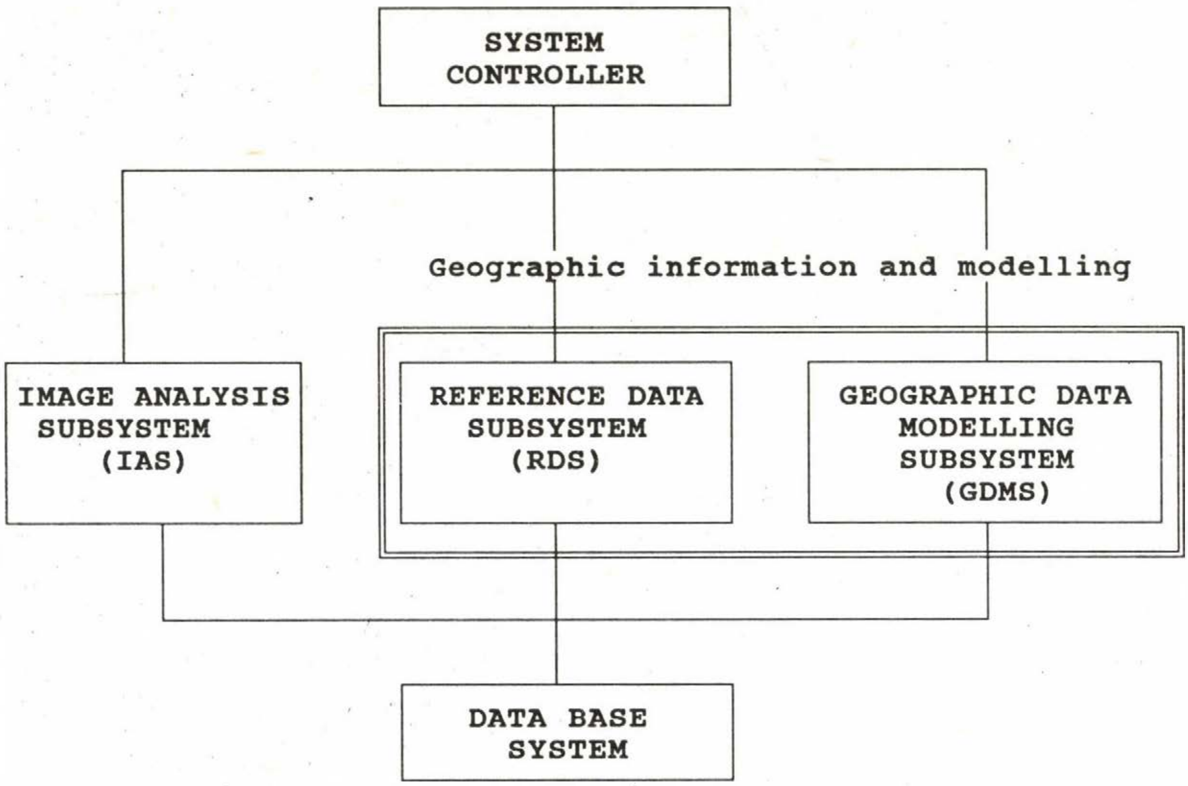


Figure 1. The scheme of the satellite data processing system

MTA SZTAKI Tanulmányok 206/1988 p. 135

LINEAMENTS DETECTION IN AERO-SPACE IMAGERY

V.P. PYATKIN

Computing Center, Siberian Division of the USSR Academy of Sciences, Novosibirsk, USSR

A b s t r a c t

Nowadays the nature of lineaments is not absolutely clear, neither exists their comprehensive classification. As a rule, their geological interpretation meets the tasks of a particular investigation and does not exhaust all the essence of the object considered. In most of the papers on lineament analysis automation, the results are obtained due to visual decoding of quasi-linear elements of surface structure. This approach is rather subjective and not accurate enough. Therefore, the task to automate lineament detection in aero-space imagery seems rather important and has a great practical interest. A conventional approach to this task presented in the paper is connected with application of gradient type operators. This approach to lineament detection in aero-space images possesses a number of drawbacks. Thus, the need to search for new approaches to solving the task of automated lineament detection in aero-space imagery arises. Two such approaches, tomographic and statistic, are considered in the paper. The basis of the first approach is the criterion of projection optimality in computing tomography. In statistic approach, the conclusion of the lineament presence (absence) in aero-space imagery is made on the basis of the result obtained by testing the statistic hypothesis. These two approaches have been used practically in processing of real aero-space images. The preliminary results testify to the prospects of the statistic and tomographic approaches to detecting circular structures in aero-space images.

A TANULMÁNYOK SOROZATBAN 1987-BEN MEGJELENTEK:

195/1987 Telegdi László: Bináris változók strukturájának vizsgálata

196/1987 Rónyai Lajos: Algebrai algoritmusok

197/1987 Hernádi Ágnes - Bodó Zoltán - Knuth Előd: A tudásábrázolás technikái és gépi eszközei

198/1987 Miguel Fonfria Atan: A data base management system developed for the Cuban minicomputer CID 300/10

199/1987 Bach Iván - Farkas Ernő - Naszódi Mátyás: A magyar nyelv elemzése számitógéppel

200/1987 Publikációk'86 /Szerkesztette: Petróczy Judit/

201/1987 Eszenszki József - Hévizi László - dr. Kas Iván - dr. Laufer Judit - Palotási András - Szőnyi Tamás - Dr. Vörös Károly: Tanulmányok a számitástechnika nyomdaipari alkalmazásához

202/1987 Problems of Computer Science
Proceedings of the joint workshop of Computer and Automation Institute of HAS and Computing Centre of Armenian Academy of Sciences held in Budapest, September 1987. /Edited by: G.B. Marandzjan,
B. Uhrin/

1988-BAN EDDIG MEGJELENTEK:

203/1988 KNVVT EG-25. Problems and tools of the integration of information systems. Proceedings. 1987.
Edited by: Rumjana Kirkova - Tibor Remzső
Ferenc Urbánszki

204/1988 Csetverikov Dmitrij: Digitális texturavizsgálat néhány uj módszere

205/1988 Hernádi Ágnes: Uj eszközök a fogalmi modellezésben