PARALLEL OBJECT CLASSIFICATION ALGORITHMS IN IMAGES

Luong Chi Mai

Viện Công nghệ thông tin

I. Introduction

Image processing and recognition belong to those research areas of artificial intelligence where computers contribute significantly to solving problems which have been only in the competence of the human intelligence. There are several serious demands which lead to the present interest at parallel computers for solving these problems. Huge amount of data to be processed require computers with high operational and memory capacity, as well as reliability. Due to their ability to fulfill these conditions, parallel processors seem to become dominating for future trends in solving image processing, pattern recognition and other related artificial intelligence problems.

Our attention is devoted to the problem of object classification in digital images. Well known statistical approaches were chosen to be modified for parallel implementation. The computational regime under consideration is SIMD (Single Instruction Multiple Data). From this perspective three methods will be examined and parallelized for the SIMD-type computer developed and operated at the Institute of Technical Cybernetics in Bratislava [4]. We restrict ourselves here only to a brief description of its main features.

Computations are performed in parallel modules which are composed of the MDA (Multidimensional Access Memory) block, the arithmetic-logical unit (ALU) and of the permutation network (PN). The MDA memory block consits of 256 x 256 bit subblocks (pages). Data are located in memory fields to which there exits access in both vertical and horizontal directions as well as in their combinations. To each word of the memory one 1-bit processor element of the ALU is assigned. This unit contains also registers of the length 256. One of them serves as a mask register where respectively 1 and 0 at a given position indicates that word in the memory where the result of an operation is or is not written in. The permutation network allows data permutations according to the application under cosideration. Most frequently used are 2^k-type permutations, k=0,1,...,7. The control memory contains the programm for the parallel machine. For the data transfer from the control memory to the MDA memory one 32-bit register is at user's disposal.

The standard functions are realized in a form of macroinstructions and procedures. The application programs from the image processing are described in [2]. This paper brings description of new application algorithms related to the parallel object classification in images. The approaches under parallelization are the Bayes classification, the K-mean and the ensemble average classifier. The Bayes algorithm for the SIMD-type parallel machine is presented in the section 2 of this contribution. It is based on the evaluation of mean vectors and covariance matrices. The automatic K-mean classifier is formulated in the section 3. From the point of view of parallel realization, the most effective seems to be the ensemble average classifier which is described in the concluding section.

We note that the algorithms presented are under programming realization in a parallel assember programming language. Results of computational experiments will be published elsewhere.

2. The Bayes classifier

Let x be a vector (pixel) in the N-dimensional Euclidian space E. For a set of K vector classes $C = \{C_i,\ i = 1,2,...,K\}$, let $P(C_i)$ be a priori class-probability of C_i, $f(x|C_i)$ be class-conditional probability density function and $P(C_i|x)$ be the conditional probability that x belongs to the class C_i.

The Bayes strategy minimizes the misclassification error and it is based on the minimization of the function [5]

$$L_X(C_i) = \sum_{j=1}^{K} P(C_j|x) = 1 - P(C_i|x). \tag{1}$$

Let the set of decision functions is defined by

$$G_i(x) = f(x|C_i)P(C_i), \ i = 1,2,...,K.$$

According to (1) and the relation of propotionality between $P(C_i|x)$ and $f(x|C_i)P(C_i)$, an unknown vector x is classified into the class C_i if and only if

$$iG_i(x) = \max_j \, [G_j(x)] \text{ for } j = 1,2,...,K. \tag{2}$$

Assume that each $f(x|C_j)$, $j = 1,2,...,K$ is N-dimensional Gaussian density function with mean vector μ_j and covariance matrix R_j. Considering $P(C_1) = P(C_2) = ... = P(C_K)$, we obtain [5],

$$G(x) = -\frac{1}{2}\Big(\ln|R_j| + (x-\mu_j)^T R_j^{-1}(x-\mu_j)\Big), \tag{3}$$

where $|R|$ denotes the determinant of R_j and R_j^{-1} is the inverse matrix of R_j.

In practice, mean vector and covariance matrix of each class must be evaluated from training sets of vectors belonging to them. For a given class C_j, $j = 1,2,...,K$ formulas for evaluating respectively the elements μ_j^k, $k = 1,2,...,N$ of m_j and δ_j^{ik}, $i,k = 1,2,...,N$ of R_j, are

$$\mu_j^k = \frac{1}{S_j}\sum_{l=1}^{S_j} y_k^l \tag{4}$$

and

$$\delta_j^{ik} = \frac{1}{S_j}\sum_{l=1}^{S_j}(y_i^l - \mu_j^i)(y_k^l - \mu_j^k). \tag{5}$$

In (4) and (5), y_i^l and y_k^l denote respectively i^{th} and k^{th} component of lth training set vector y belonging to $C_j.S_j$ is the total number of training set elements in C_j.

1st	y_1^1	......	y_1^M	Mth training
training	y_2^1	......	y_2^M	vector of
of class	$\vdots$	......	$\vdots$	class C_1
C_1	y_N^1	......	y_N^M	
	$\vdots$	......	$\vdots$	
1st	y_1^1	......	y_1^M	Mth training
training	y_2^1	......	y_2^M	vector of
of class	$\vdots$	......	$\vdots$	class C_1
C_1	y_N^1	......	y_N^M	
	0	......	0	

Fig.1. The illustration of the training vector location.

In the parallel Bayes algorithm for the parallel associative computer of the SIMD-type, data elements represent N-dimensional vectors. Assume that each component possesses one of 256 gray-level values (8 bits) correspondingly to the gray-level scale of the image. It means that one page of the MDA memory can be divided into M

column fields ($M = 32$) $F_1, F_2, ..., F_M$. Assume that the page of the memory is divided into 4 horizontal blocks, which consist of 64 rows each. These blocks are filled-up in a such manner that all 1st dimension components of the trainig set vectors belonging to the 1st class are located in the first row, the 2nd dimension components of the training set vectors of the 1st class are located in the 2nd row, etc. To store all elements in training sets for all classes, $K * N$ rows of each block are reserved. Free remaining rows of each block should be filled-up by zeros. The location of data in the 1st horizontal memory subblock (64 rows) shows Fig.1. Remaining $S_j - M$ training vectors for each $j = 1, 2, ..., K$ are located in the same way in next horizontal block of the memory page. If the number of elements in training sets for all classes is larger than the memory page size, additional pages of the MDA memory can be used.

Except the fields for storing the training set elements, the algorithm requires to reserve also some other auxiliary fields in the MDA memory. Field H is used to store mean vectors and fields $B_1, B_2, ..., B_N$ are reserved for the elements of the covariance matrices of all classes. These fields are also divided horizontally into 4 blocks of 64 rows each. In ith row, $i = 1, 2, ..., N$ of each subblock in H there is located a mean value corresponding to ith dimension. The field B_1 is used for storing elements $\delta_{11}, \delta_{22}, ..., \delta_{NN}$ of the covariance matrix, the field B_2 for $\delta_{12}, \delta_{23}, ..., \delta_{N-1,N}$ etc., while the field B_n contains δ_{1N}. The illustraton of these fields is shown on Fig.2.

	μ_1^1	σ_1^{12}	σ_1^{12}	$\dots$	σ_1^{1N}
	μ_1^2	σ_1^{22}	σ_1^{23}	$\dots$	0
Class C_1	μ_1^{N-1}	$\sigma_1^{N-1,N-1}$	$\sigma_1^{N-1,N}$	$\dots$	$\dots$
	μ_1^N	σ_1^{NN}	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	μ_1^1	σ_1^{12}	σ_1^{12}	$\dots$	σ_1^{1N}
	μ_1^2	σ_1^{22}	σ_1^{23}	$\dots$	0
Class C_K	μ_1^{N-1}	$\sigma_1^{N-1,N-1}$	$\sigma_1^{N-1,N}$	$\dots$	$\dots$
	μ_1^N	σ_1^{NN}	0	$\dots$	0
	0	0	0	0	0
	H	B_1	B_2		B_N

Fig. 2. The illustration of the contents of the fields $H, B_1, B_2, ..., B_N$ in one memory block of 64 rows.

For calculation of all components of mean vectors by the formula (4), block-permutations to add-up corresponding values in each dimension for all classes are used in the parallel algorithm. The result is stored in the field H.

The algorithm uses two auxiliary fields H_1, H_2 divided also into 4 blocks . In H_1 there is accumulated the sum $\sum_{i=1}^{M}(F_j - H)$.

Both fields are used to apply block-permutations by addition of corresponding partial sums from (5). The result of multiplication of H_1 and H_2 are diagonal elements of the covariance matrices which are stored in the field B_1. Covariance matrix elements in the fields $B_2, B_3, ..., B_N$ are evaluated using the operation of acyclical shift in each subblock and the field multipllication. In last K steps the division of each $B_1, B_2, ..., B_N$ by the number of elements in the corresponding training set is performed.

In the parallel Bayes classification phase, input data represent image pixels stored in the MDA memory in the line interleaved manner,i.e. the first row of the first dimension is located in the first row of the memory page, in the second row of the page there is located the first row of the second dimension,etc., in the kth row there is the first row of the kth dimension, $k = 3, 4, ..., N$. If each pixel possesses one of 256 gray-level values and the number of dimensions equals to N=4 then in one page of the MDA memory could be stored one image of the size 32x64. Besides the pixel values, input data for the classification phase are mean vectors and covariance matrices of all classes. These vectors and matrices are results from the learning phase and they are stored in an external storage device. The field H is divided into blocks of size N, where N is the dimension number. The first block is used to store N-dimensional mean vectors for the first class,etc., the kth block stores the mean vector for kth class, $k = 2, 3, ..., K$. Output of the classification is an image where a pixel represents that class label, to which the corresponding pixel of the original image belongs.

The algorithm consists of two parts, according to the calculation of formula (2) and (3) respectively. Main problem in evaluating the formula (3) is the calculation of the inverse matrix and determinant of each covariance matrix $R_j, j = 1, 2, ..., K$. The algorithm uses a modification of the parallel subroutine JORDAN for solving linear systems of algebraic equations by mean of the Gause-Jordan elimination method with column pivoting [3]. The final operation is to find maximal values according to the formula (2). It is realized in parallel by comparing all columns.

3. Parallel automatic K-mean classifier

This type of clssifier does not require any assumption on probability distribution of pattern vectors. The knowledge available is that there is a sequence of vectors $x_1, x_2, ..., x_N$ from N-dimensional euclidian space. It is assumed that $N << n$. Without making use of training samples, the method classifies a sequence of vectors $x_1, x_2, ..., x_N$ into K classes $C_1, C_2, ..., C_K$,

The automatic K-mean algorithm can be formulated in following steps [1] :

Step 1: Choosing K initial centres $z_1(1), z_2(1), ..., z_K(1)$ of all classes. This selection is done more or less arbitrarily.

Step 2: In k^{th} iteration step, vectors x_l, $l = 1, 2, ..., n$ may be distributed into the classes $C_1, C_2, ..., C_K$ by applying the rule

$$x_l \in C_j(k) \Longleftrightarrow \|x_l - z_i(k)\| < \|x - z(k)\| \tag{6}$$

for all $j = 1, 2..., k; i \neq k$.

In (6), $\|.\|$ denotes the euclidian norm, $C_j(k)$ represents the set of vectors entering the class with the centre $z_j(k)$ and index k denotes the iteration number. In the case of equality in (6), x_l can be classified either to the class C_i or to that class C_j for which the equality holds.

Step 3: Based on the result of step 2, new centres $z_1(k+1), z_2(k+1), ..., z_K(k+1)$ of all classes must be selected minimizing the function

$$J_j = \sum_{x \in C_j(k)} \|x - z_j(k+1)\|, \; j = 1, 2, ..., K. \tag{7}$$

It is well known that the minimum $z_j(k+1)$ of (7) occurs in the average point of all vectors contained in the set $C_j(k)$, $j = 1, 2, ..., K$. It is evaluated by

$$z(k+1) = \frac{1}{N_j} \sum_{x \in C_j(k)} x \tag{8}$$

where N_j is the number of vectors in $C_j(k)$.

Step 4: If

$$\|z(k+1) - z(k)\| < \varepsilon \text{ for all } j = 1, 2, ..., K \tag{9}$$

($\varepsilon > 0$ is chosen), then the algorithm terminates. In the opposite case, repeat from the step 2.

A convergence of the K-mean algorithm depends on a rather large number of factors. In most cases the application of algorithm requires some testing steps concerning the proper value of the parameter K and the initial allocation of the centres of classes., In the parallel K-mean algorithm for the SIMD-type computer, it is considered that the multilevel image is stored in M fields $F_1, F_2, ..., F_M$ of the MDA memory, where the size of each column F_i, $i = 1, 2, ..., M$ depends on the gray-level scale of pixels. It is assummed a multilevel image with the gray-level scale from 0 to 255, i.e. 8 bits must be used to represent one pixel in the memory and in this case $M = 32$. Each N-dimensional pixel is stored in the MDA memory in the interleaved form (Fig.3). For simplicity of illustration let $K = 8$, $N = 4$. Assume that each pixel x and each centre z of class C_j, $j = 1, 2, ..., K$ are represented in a vector form,i.e. $x = (x_{l1}, x_{l2}, ..., x_{lN})$ and $z = (z_{j1}, z_{j2}, ..., z_{jN})$ respectively (see Fig.3), where number $q = 64 * (M - 1)$.

x_{11}	......	$x_{q-63.1}$	z_{11}	Centre
x_{12}	......	$x_{q-63.2}$	z_{12}	for
$\vdots$	......	$\vdots$	$\vdots$	class
x_{1N}	......	$x_{q-63.N}$	Z_{1N}	C_1
$\vdots$	......	$\vdots$	...	$\vdots$
$\vdots$	......	$\vdots$	z_{k1}	...
$x_{64.1}$	......	x_{q1}	z_{k1}	Centre
$x_{64.2}$	......	x_{q2}	...	for
$\vdots$	......	$\vdots$	z_{kN}	class
$x_{64.N}$	......	x_{qN}	0	C_1
F_1	......	F_M	Z	

Fig.3. The illustration of image fields and field for the centres of classes.

A column z with the width of 32 bits is provided for calculating and storing new centres in each iteration. During the evaluation of the algorithm, one boolean matrix M of $K = 8$ columns $M_1, M_2, ..., M_K$ (1 bit per column) and one matrix L of M columns $L_1, L_2, ..., L_M$ (3 bits per column) are needed for intermediate labeling of classes. Field H plays the role of histogram of class labels. The output is a matrix located in the MDA memory with elements possessing values from 1 to 8 correspondigly to that class label which is assigned to a given pixel by the classification. The algorithm is divided into following main parts according to steps formulated above:

(i) Selection of K initial centres of all classes. The first $K * N$ rows in the first column F_1 are selected as K initial centres and stored in the field z.

(ii) Pixels x located in the fields $F_1, F_2, ..., F_M$ are distributed into K classes by the rule (6).

a. Calculation of $x - z_j(k)$ for $j = 1, 2, ..., K$.

b. Calculation of the norm $\|x - z(k)\|$ for $j = 1, 2, ..., K$ using block-permutations with permutation factors 2 and 2 (since $N = 4$). The results of substeps a. and b. are stored in auxiliary memory fields $xz_1, xz_2, ..., xz_K$.

c. Finding maximal values in fields $xz_1, xz_2, ..., xz_K$. Label corresponding to these values will be stored in the M columns $L_1, L_2, ..., L_M$.

(iii) Calculation of new centres of all classes C_j, $j = 1, 2, ..., K$ by (7). In a field H the number of pixels in all classes is stored. During evaluation of the histogram vector H, boolean columns $M_1, M_2, ..., M_K$ are used for marking those positions in $L_1, L_2, ..., L_M$ where the same class label occurs.

(iv) Test of convergence according to (9).

4. Ensemble average classifier

As it was the case in the previous classifier, no assumption is made on the probability distribution or density function of the pattern vectors. The only knowledge provided is a sequence of K sets of training pattern vectors with a known classification. The number of vectors in each class C_j, $j = 1,2,...,K$ must be greater than the dimension of the feature space N. Classification of an unknown pattern vector x into one of the classes is carried out as follows [7]:

Step 1: Estimate the average pattern vector for each class by

$$m_j = \frac{1}{S_j}\sum_{l=1}^{S_j} x_l, \; j = 1,2,...,K, \tag{10}$$

where S_j is the number of training pattern vectors in C_j and x_j, $l = 1,2,...,S_j$ are the pattern vectors belonging to C_j.

Step 2: An unknown pattern vector x is classified into a class C_j iff

$$[(x - m_i)^T(m_i - m_J)]^2 \geq [(x - m_j)(m_i - m_j)]^2 \tag{11}$$

i i j j i j for all $i,j = 1,2,...,K$; $j \neq i$.

After $(k-1)$ comparisons in step 2, the algorithm classifies an unknown vector x into the correct class.

Ensemble average classification algorithm for a parallel SIMD-type computer possesses as input an image stored in the MDA memory by columns in the interleaved form (as in the K-mean algorithm described previously). Let K mean vectors of the training sets be stored in the fields $H_1, H_2, ..., H_K$ of the MDA memory. H is a field containing K 1-bit columns. In each row of this field, there is only one value 1 corresponding to the label of that class, which is assigned to a given pixel by the classification. For K=8, an example of the field H is shown in the following table:

H_j/H	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
2	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0
4	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1

Allocation of the image field and mean vectors are shown in Fig 4a and Fig.4b respectively, where number $q = 64 * (M-1)$ in Fig.4a. For simplicity of representation let us consider $N = 4$, $K = 8$ and 256 gray-levels of image pixels.

The parallel ensemble average classification algorithm is very simple. It is divided into 3 parts. The first part is the calculation of the formula (11) by using only vector subtraction, multiplication and comparision operations. The second one consists in forming the vectors $H_1, H_2, ..., H_K$ by using masked vector operations. As a result we have the field H where in each word there is only one bit possessing the value 1. The third part of the algorithm consists in the histogram calculation from $H_1, H_2, ..., H_K$. It is realized by the technique for parallel evaluation of histogram on the parallel associative computer given in [6].

x_{11}	......	$x_{q-63.1}$
x_{12}	......	$x_{q-63.2}$
$\vdots$	......	$\vdots$
x_{1N}	......	$x_{q-63.N}$
$\vdots$	......	$\vdots$
$x_{64.1}$	......	x_{q1}
$x_{64.2}$	......	x_{q2}
$\vdots$	......	$\vdots$
$x_{64.N}$	......	x_{qN}
F_1	......	F_M

Fig.4a. Image allocation

m_{11}	......	m_{k1}
m_{12}	......	m_{k2}
$\vdots$	......	$\vdots$
m_{1N}	......	m_{kN}
$\vdots$	......	$\vdots$
m_{11}	......	m_{k1}
m_{12}	......	m_{k2}
$\vdots$	......	$\vdots$
m_{1N}	......	m_{kN}
M_1	......	M_K

Fig.4b. Mean vectoc allocation

References

1. Gonsales, Dv. Tu, *Principy raspoznawaniq obrazow.* Moskwa, Mir 1978, str. 50-56.
2. Lucka M., Muchova M. and Vajtersic M., *Image processing algorithms for parallel system of the PPS-SIMD type.* Sis- temy obrabotki znanij i izobravenij. T.1. Smolenice 1986, 157-162.
3. Majlingova O. and Vajtersic M., *Jordanova metoda riesenia systemov linearnych rovnic na paralelnom asociativnom pocitaci.* Vyskumna sprava UTK SAV, Bratis - lava, 1988,.
4. Richter K.et al., *The architecture of PPS-SIMD. Sistemy obrabotki znanij i izobravenij.* T.2, Smolenice 1986 , 51-59.
5. Swain P.H., *Image data alysis in remote sensing.* Digital Image Processing, Vol.5, No.4, 1979, 200-235.
6. Vajtersic M. and Giang Vu Thang : Threshold and histogram algorithms for a parallel associative computer. Computers and Artificial Intelligence, Vol.5, No.2, 1986, 143-161.

7. Zaki F.W., Abd el-Fattah A.I., Enab Y.M. and Konyaly S.H., *An ensemble average classifier for pattern recognition machines.* Pattern Recognition, Vol.21, No.4, 1988, 327-332.

Abstract

The contribution concerns a parallelization of object classification algorithms for a SIMD-type parallel machine. It is assummed that gray-level values of image pixels are located in the orthogonal memory block on which a vector of one-bit processor operates in the bit-serial and word-parallel mode. For this computer, the Bayes classification algorithm, the K-mean algorithm and the ensemble average classifier are described.