MILITARY TECHNICAL COLLEGE
CAIRO-EGYPT

ICEENG 98

FIRST INTERNATIONAL CONF.
ON ELECTRICAL ENGINEERING

PARALLEL IMPLEMENTATION OF IMAGE PROCESSING ALGORITHM FOR DETECTION A MOVING TARGET USING A FULL LINKING TRANSPUTER

S. F. Bahgat **, S. Ghoniemy *, R. N. Mohammed Agam ***

Abstract

A real-time image processing algorithm for detection of a moving targets requires faster, more accurate and more flexible computer systems. For TV image tracking systems, scanning in the camera fielc of view (CFOV), capturing the intended target inside a window, detection of the target center, and tracking process are four modes of operations that need fast execution time. To implement these algorithms in real-time, delay time caused in these modes must be minimized. Parallel processing techniques are exploited as a solution for implementation of such algorithm. This paper describes the parallel implementation of real-time image processing algorithm for detection of a moving target using full linking transputer arrays. The partitioning roughly equal tasks which are written in parallel-C allocating the tasks to processors, interconnection topology of the processors, and the capacity of each communication link over the transputer network are considered. Finally, simulation results show that parallel techniques are very powerful and essential for real - time applications.

Introduction

The conventional TV image tracking system consistes of the following units : video camera, TV monitor, video blaster card, hardware interface, camera driving system, group of software programes for TV image tracking, microprocessor, joy stick manual control, and video tape recorder. The function of this system is as

* Professor , Faculty of Computer and Information Science - Ain Shams University, Cairo, Egypt
** Associate Professor, Dep. of Radar & Guidance, Military Technical College , Cairo, Egypt.
*** Graduite Stuident, Military Technical College, Cairo, Egypt.

follows: the video camera provides a real-time analog images to the system which works in two modes of operation: searching mode and tracking mode. The objective of the group software programes are to recognize the image format provided by the video blaster and changed it to data information which is suiotable for microprocressor operation and to produce the required control signals to control the mode of operation of video blaster to update the image file to repeat the process and produce the error voltage associatd with the new position of the moving target to the camera driving system to follow the moving target. The software modules are classified as follows:

1- Programe responsable to control the mode of operation of video blaster.
2- Programe concerning reading the header file developed by video blaster.
3- Programe concerning to change video blaster format to digital data array .
4- Programe responsable to generate tracking window.
5- Programe responsable to generate the frame difference file.
6- Programe responsable to calculate the coordinates of the field of view of the image frame and the center of the tracking window.
7- Programe responsable for calculating the difference of the measured coordinates and changed it to an error voltage send through the parallel port to control the camera driving system.

In order to implement this system to be closer to real-time application, we propose the following steps:

- * Optimizing the sequential program.
- * Conversion the software witten by C-program into parallel C.
- * Design the graphics library.
- * Splitting the application into parallel tasks.
- * Using INMOS transputer system.
- * hardware configuration, through INMOS links configured by jumpers, booting from the host via the C012 link adapter.
- * Overcoming the restriction of using fileMux (multiplexer file) due to limitation of the transputer links number.

In this paper, the proposed of parallel real-time image processing system of moving target is introduced. Software implementation of the proposed system using IMS-T805 transputer is introduced.

PROPOSED PARALLEL PROCESSING IMAGE DETECTING SYSTEM

Fig. (1) shows the block diagram of the proposed parallel processing image detecting system [2]. The proposed parallel processing image detecting system consists of the following unites: Video camera, TV monitor, video blaster card, camera driving motor, hardware of fine control of the camera driving motor, parallel software programes for image processing, microcomputer, and parallel processors (transputer module). The proposed system is working in one of three modes of operation; searching mode, capturing mode, and tracking mode. In searching mode the camera provides the life image frame to the video blaster through the TV monitor.

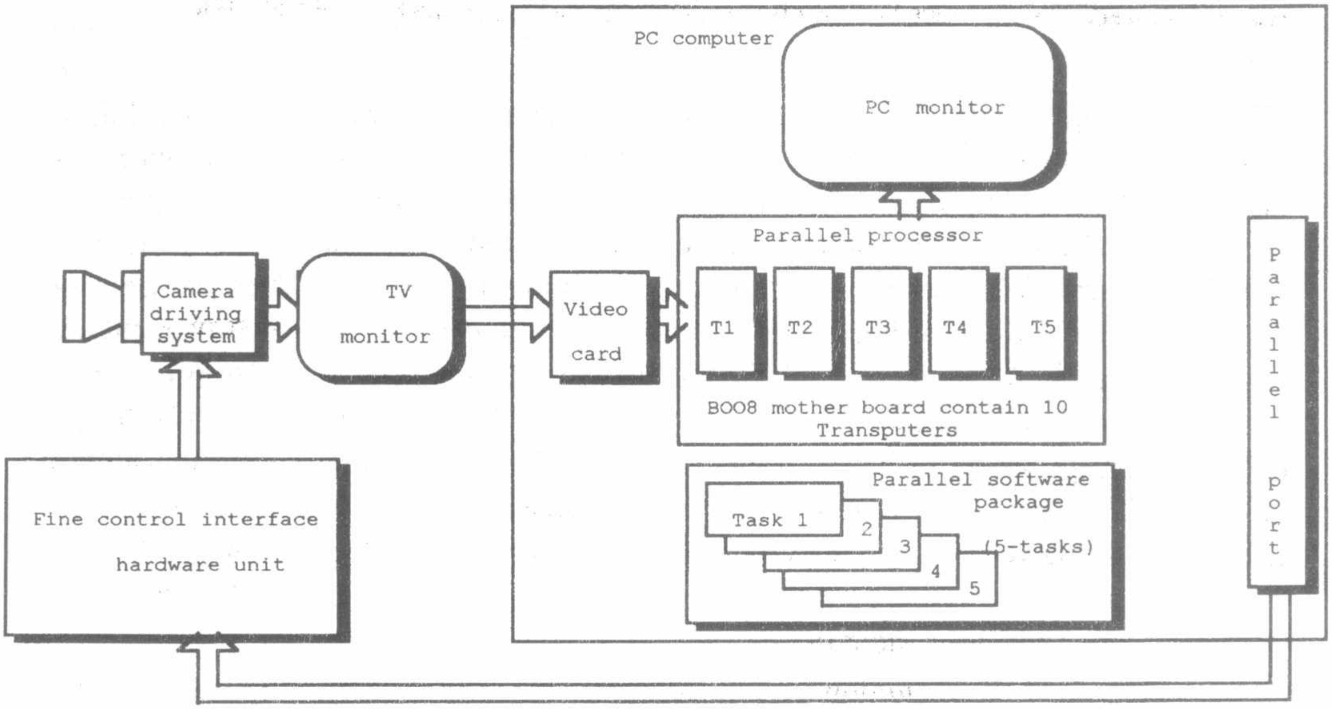

Fig. (1) Block diagram of proposed parallel processing image detecting system

The video blaster receives the life image and changes it into a digital format suitable for displaying on the PC monitor. The first objective of the proposed parallel software is to recognize the image format provided by the video blaster and convert it into a format suitable for the microcomputer operation. The proposed parallel programs work through several stages. The first stage is to initialize and prepare the graphics mode of the computer system, the second is scanning processes in all the camera field of view (CFOV) searching for the targets, and the third stage is the capturing processes for the existing targets. The last stage is the tracking process for the intended target. In capturing mode, the system starts to capture the target image. The microcomputer determine the center of the image frame and the center of the tracking window including the target of interest. The difference between the two centers is calculated and transmitted through the proposed hardware "fine control interface unit" to the camera driving system to align the center of the image frame with the center of the tracking window. In tracking mode; the parallel software algorithm produces the required control signals to control the mode of video blaster to update the image file and repeats the process for producing error signals associated with the new position of the moving target for the camera driving system to follow the intended target.

HARDWARE IMPLEMENTATION OF THE PROPOSED SYSTEM

The parallel processors used in the proposed system are the INMOS transputer system, that consists of a network of five IMS-T805 transputers. The application includes five tasks; the main task and other four parallel tasks. The main task is executed in the root transputer and the other four parallel tasks are executed in the other four transputers, each task is running on a separate transputer.The transputer T1 is the root transputer that connected directly with the host processor (PC processor) through a filter task that works as a matching program between the host and the transputer system. The other four transputers (T2-T5) are communicating with the root transputer and with each other through one-way channels. In Fig.(2), the arrow in sold line indicate the data connection paths in between the network, but the dotted lines indicate the communication paths between the transputers and the multiplexer task that capable each transputer to share the afserver task in host processor.

The INMOS serial communications link is high speed system interconnected between members of the transputer families. INMOS CO11 is a link adapter or a communication device which enables the INMOS serial communication link to the parallel data ports and microprocessors buses. The IMS COO4 is a programmable link switch, that provides a full crossbar switch between 32 link inputs and 32 link outputs[4].

SOFTWARE IMPLEMENTATION OF THE PROPOSED SYSTEM

The software of the proposed system is implemented in parallel-C language[5]. The algorithm is designed in five split tasks which they executed in parallel on five transputers. The main task is the task that executed in the root transputer and control the other four task that executed on the other four transputers such that it distributes the jobs and determines the limits of the scaning zones, then it sends them through the sender channels. The tasks receive the orders from the main task and they work at the same time to scan the all CFOV, four transputers for four quarters. The main goal of scaning the four quarters at the same time is to save about 75 % of the time consumed in scaning operation. The interconnection between the main task and the four parallel tasks are shown in Fig (2). The main task is saved as a parallel-C source file called (searcher.c), the other four parallel tasks are saved also as parallel-C source files called (zone-1.c, zone-2.c, zone-3.c, and zone-4.c). Each task of the five tasks need to share the standard run-time library in afserver task that run in the host processor, that is because they need the DOS commands and standard file I\O, and graphics functions. So the proposed application tasks should be run via the (filemux) which allows several client tasks to share a single file server task by merging (multiplexing) the clients' request streams into a single stream of requests.

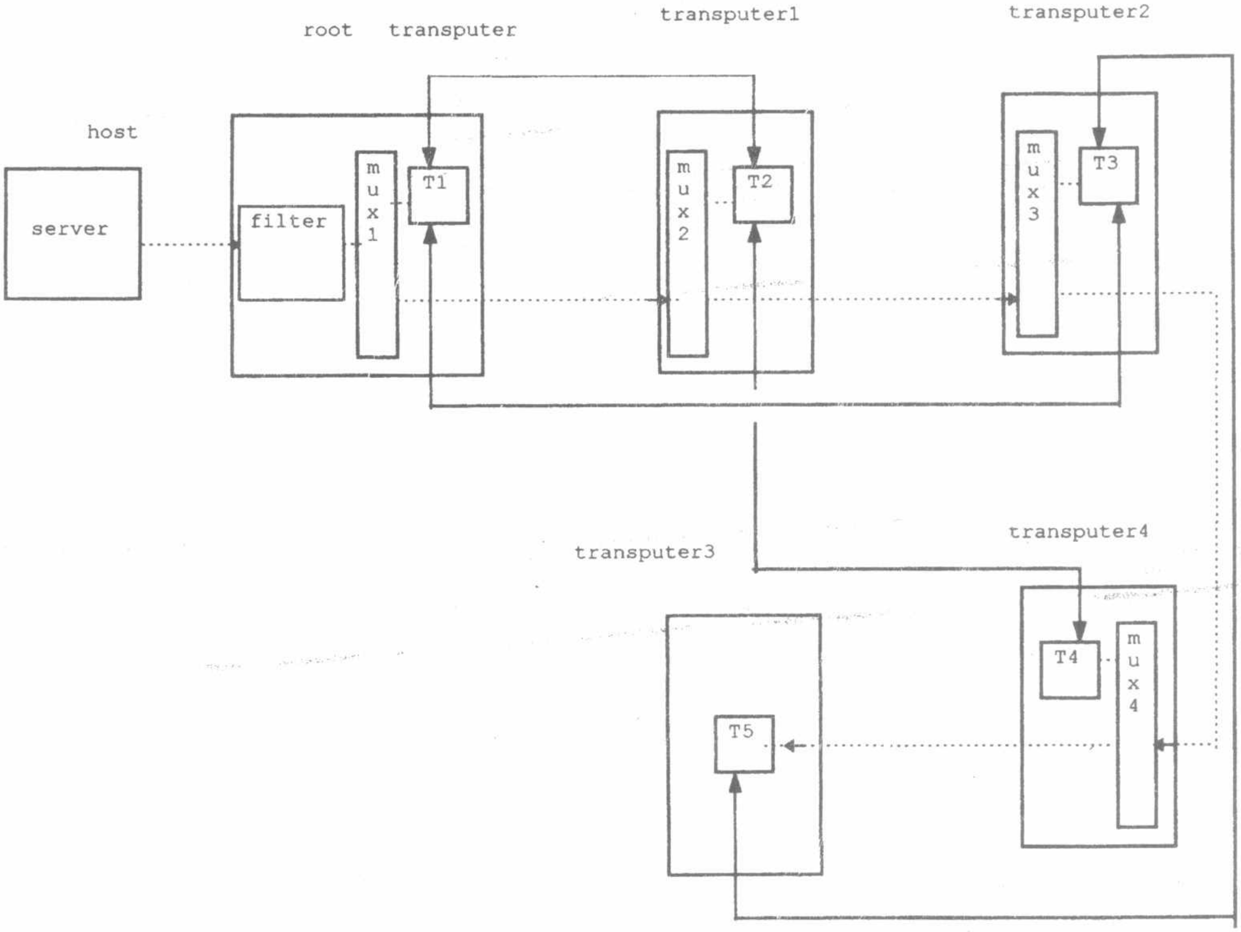

Fig. (2) Interconnection between parallel tasks using 4 multiplexers

SEARCHER TASK ALGORITHM (MAIN TASK)

The searcher task runs in the root transputer and manages the all system tasks. The flowchart of the proposed searcher task is illustrated in Fig. (3). The algorithm will start to initialize the computer graphics mode, initialize video blaster mode, and then calibrate the camera initial position. The searcher task

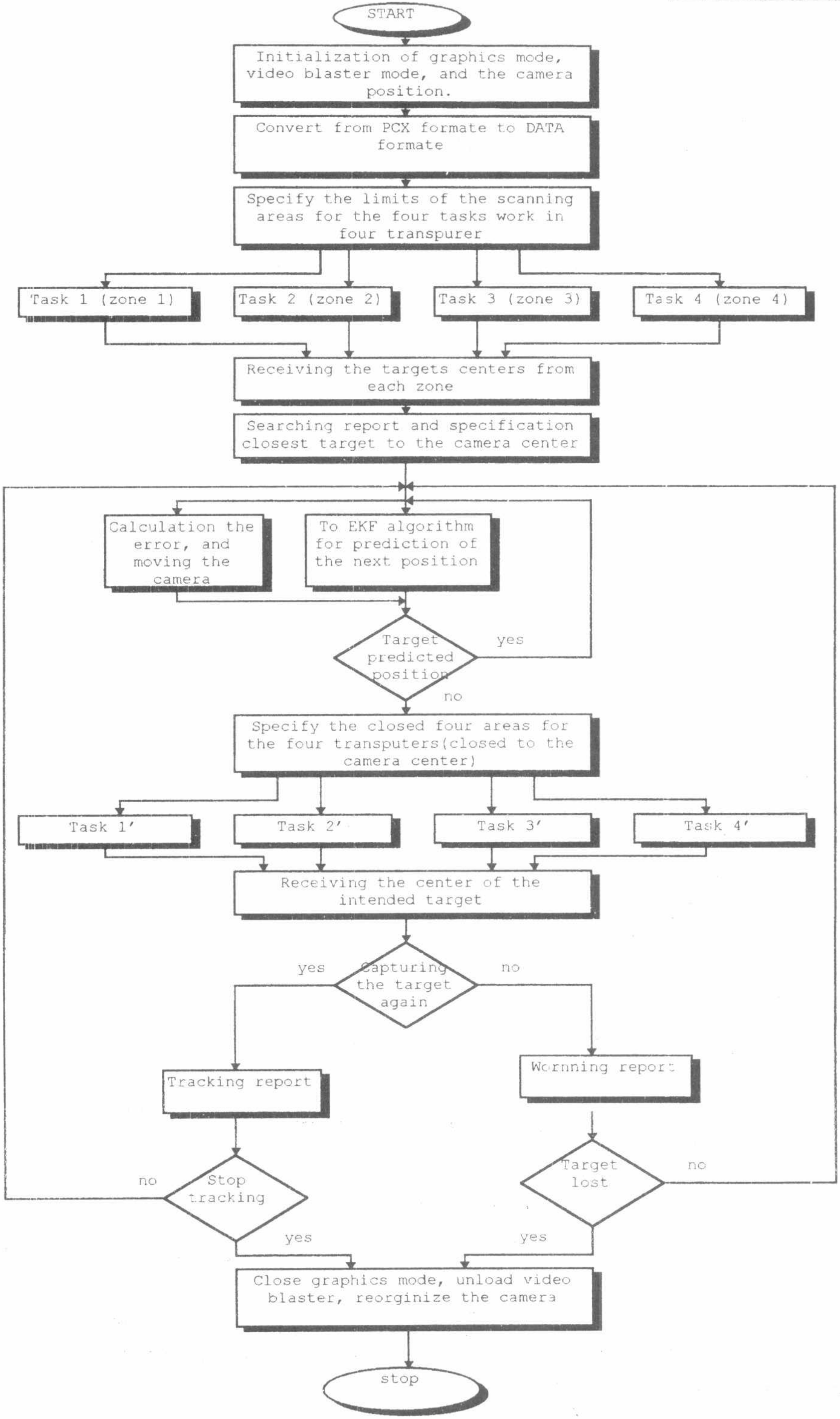
START
Initialization of graphics mode, video blaster mode, and the camera position.
Convert from PCX formate to DATA formate
Specify the limits of the scanning areas for the four tasks work in four transpurer
Task 1 (zone 1)
Task 2 (zone 2)
Task 3 (zone 3)
Task 4 (zone 4)
Receiving the targets centers from each zone
Searching report and specification closest target to the camera center
Calculation the error, and moving the camera
To EKF algorithm for prediction of the next position
Target predicted position
yes
no
Specify the closed four areas for the four transputers(closed to the camera center)
Task 1'
Task 2'
Task 3'
Task 4'
Receiving the center of the intended target
Capturing the target again
yes
no
Tracking report
Wornning report
Stop tracking
no
yes
Target lost
no
yes
Close graphics mode, unload video blaster, reorginize the camera
stop

Fig (3) Flowchart of the searcher task

algorithm allows the video blaster card (VBSE) to capture the CFOV real-time image frame via the camera in PCX format .It calls the pcxy2-program which converts the image from PCX format to DATA format to be suitable for the computer to process and deal with[2]. The image is converted into a DATA form. Then the algorithm converts image into a binary image that makes the separation between the background and the target image (black level for the background and white level for the target). The scanning field is classified into four scane areas. The limits of each scan area is specified by the upper left pixel and the lower right pixel coordinates for each zone. Searching operation in each task will be terminated.The searcher task will receive searching results through four receiving channels. These results are the centers of the existing targets. The searching report will be generated as a conclusion of the searching processes in all the four zone-tasks. The tracking process will start next to searching report. The default tracking process will occur for the closed target to the camera center. Also it can be choosen the priority of the tracking process according to the operator's interest for specific zone area.The difference between the camera center and the target center will be calculated, and the error signals will be generated to control the camera driving system.The tracking process will start by moving the camera towards the target according to the error signals generated and at the same time the predicted target's position can be calculated by actuating the predicted position algorithm. Extended Kalman Filter (EKF) algorithm can used as a predicted algorithm[6]. It receives the target position from the searcher task, then it predicts the moving target next position (Xd,Yd) and returns it back to the searcher task. The searcher task receives the predicted target position (Xd,Yd) through the receiving channels. Then the camera will move directly to align its center to the predicted position. The program will check the existence of the target or not in the predicted position. If the target is not there in the predicted position the searcher algorithm will specify an area closed to the camera center, which is aligned to the predicted position now, and it sends again to all four zone tasks to search for the target; also the closed area will be divided into four zones that will scanned in parallel. So the searching task sends through the four sender channels the limits of the scanning area for each zone task (25 pixels around the camera center). The four receiving channels receive the results from the four zone tasks aconclude the results as follows: As it captures the target again, it will generate the tracking report that informs the EKF by the difference between the center of the predicted position and the center of the corrected one. In other hand, if the searching process fails to find the target again, the warning report will be generated to inform that the target has been lost. The tracking process will be continue between the EKF algorithm which predicts the target next position, and the searcher task with aid of its zones tasks which corrects the predicted position and so on. To stop the working algorithm, the graphics mode must be closed, and the video blaster must be unloaded, and the camera position must be set to the initial position.

ZONE TASK ALGORITHM (ZONE1 TO ZONE 4)

The zone-1 to zone-4 tasks are executed on the transputers T2 to T5 respectifly. they are responsible for scanning the all CFOV. They start to work just they receive the limits of the scan areas from the searcher task which they represents the upper left and the lower right corners pixels of each zone area. Fig. (4) shows the general flowchart of the zone task operation.

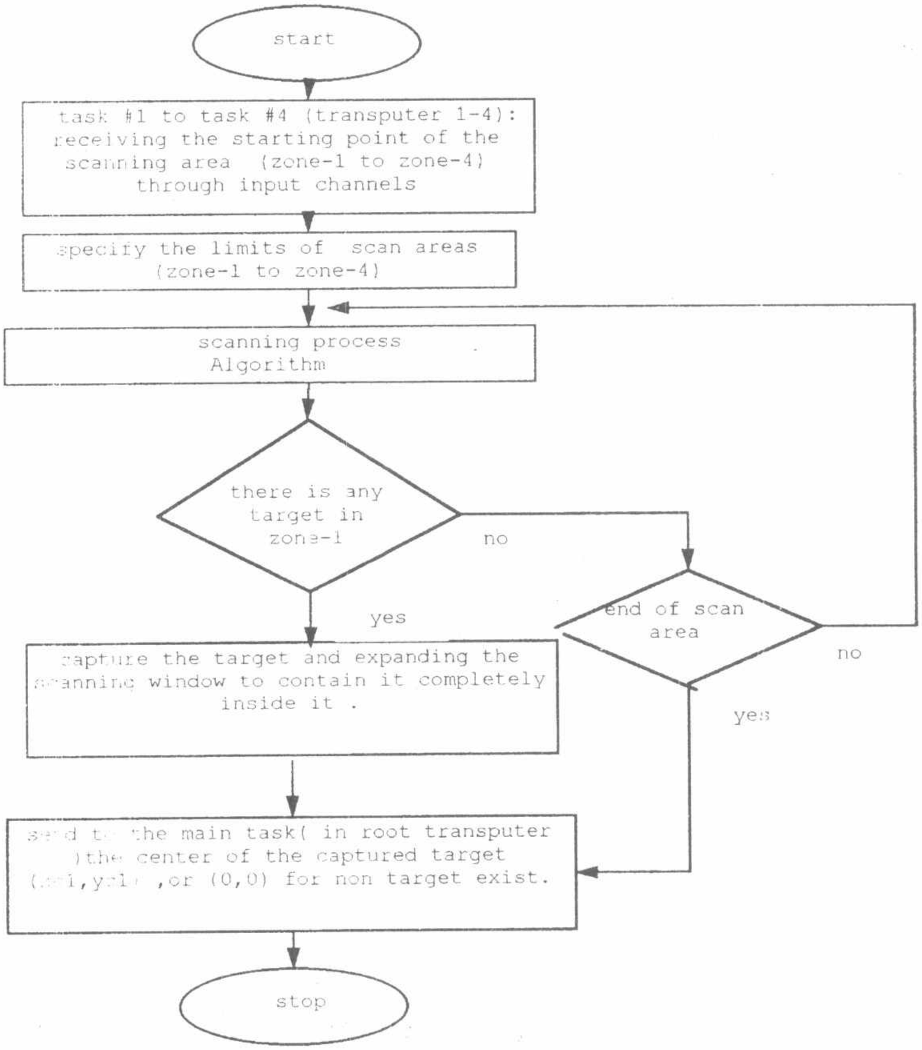
start
task #1 to task #4 (transputer 1-4): receiving the starting point of the scanning area (zone-1 to zone-4) through input channels
specify the limits of scan areas (zone-1 to zone-4)
scanning process Algorithm
there is any target in zone-1
no
yes
end of scan area
no
yes
capture the target and expanding the scanning window to contain it completely inside it .
send to the main task(in root transputer)the center of the captured target (x1,y1) ,or (0,0) for non target exist.
stop

Fig. (4) Flowchart of zone task algorithm.

The limits of the scanning areas are transmitted from the searcher task for zone-1 to zone-4 tasks through one-way receiving channels. Scanning process will started by creating a scanning pixel which initiated in the first pixel in each zone.

The scanning pixel moves horizontally until the end of the line then it moves vertically one pixel down, and restarts to move horizontally again and so on until the end of zone area[7]. The algorithm is designed to remove the current scanning pixel each time before generation of the next new one. The scanning pixel check itself continuously to test the existence of targets. For the capturing the target, the scanning process stop while the capturing process will start. the scanning pixel expands to contain all the target inside tracking window. The expansion process starts in right direction to down direction to up direction then to left direction, and the center of this window is calculated to determine the center of the target inside (Xc1,Yc1) to (Xc4,Yc4). These results will send to the searcher task through the sending channels. Fig.'s 5, 6, 7 show the flowcharts of scanning process algorithm, capturing process algorithm, and error calculation and moving camera algorithm.

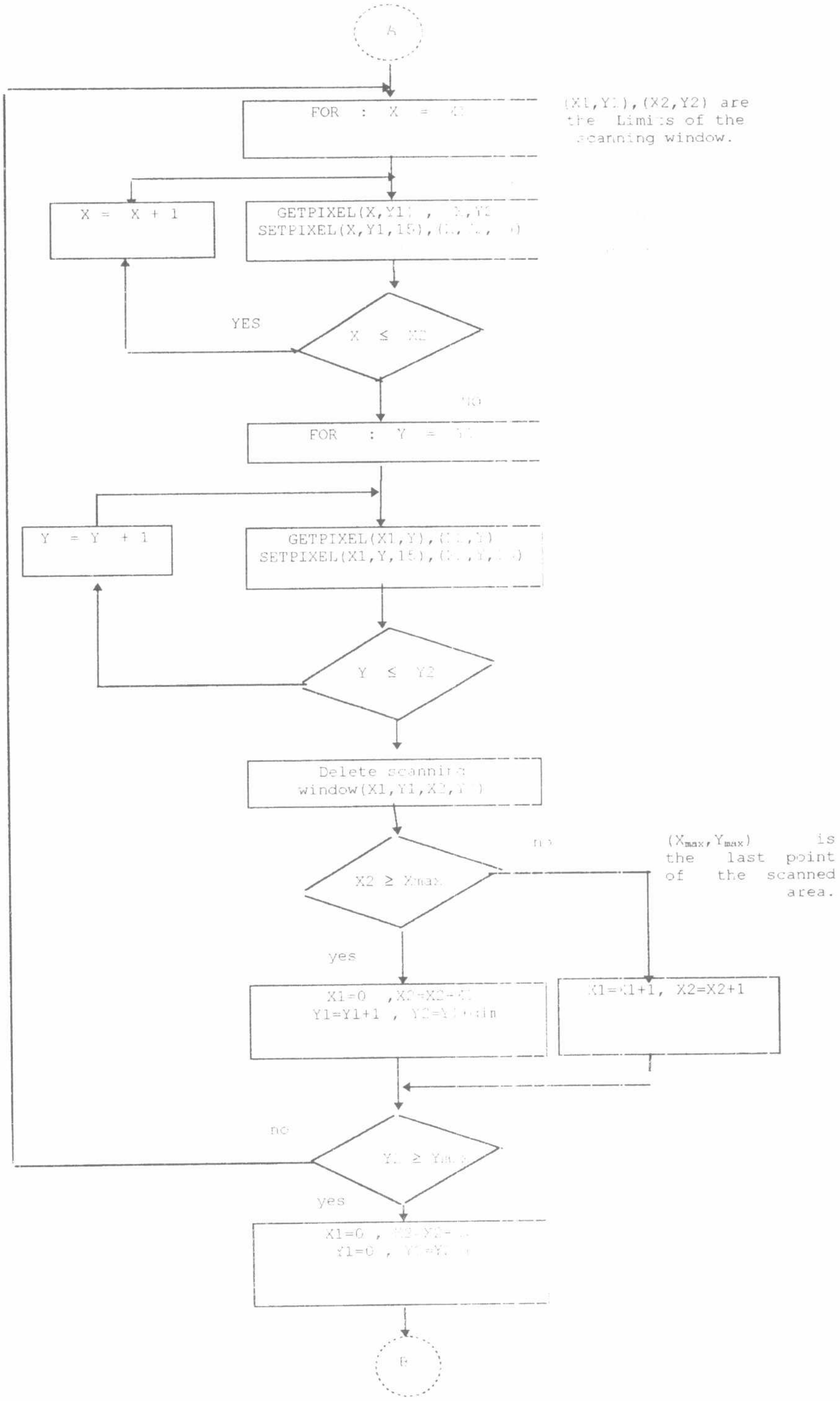

Fig.(5) Flowchart of scanning process algorithm.

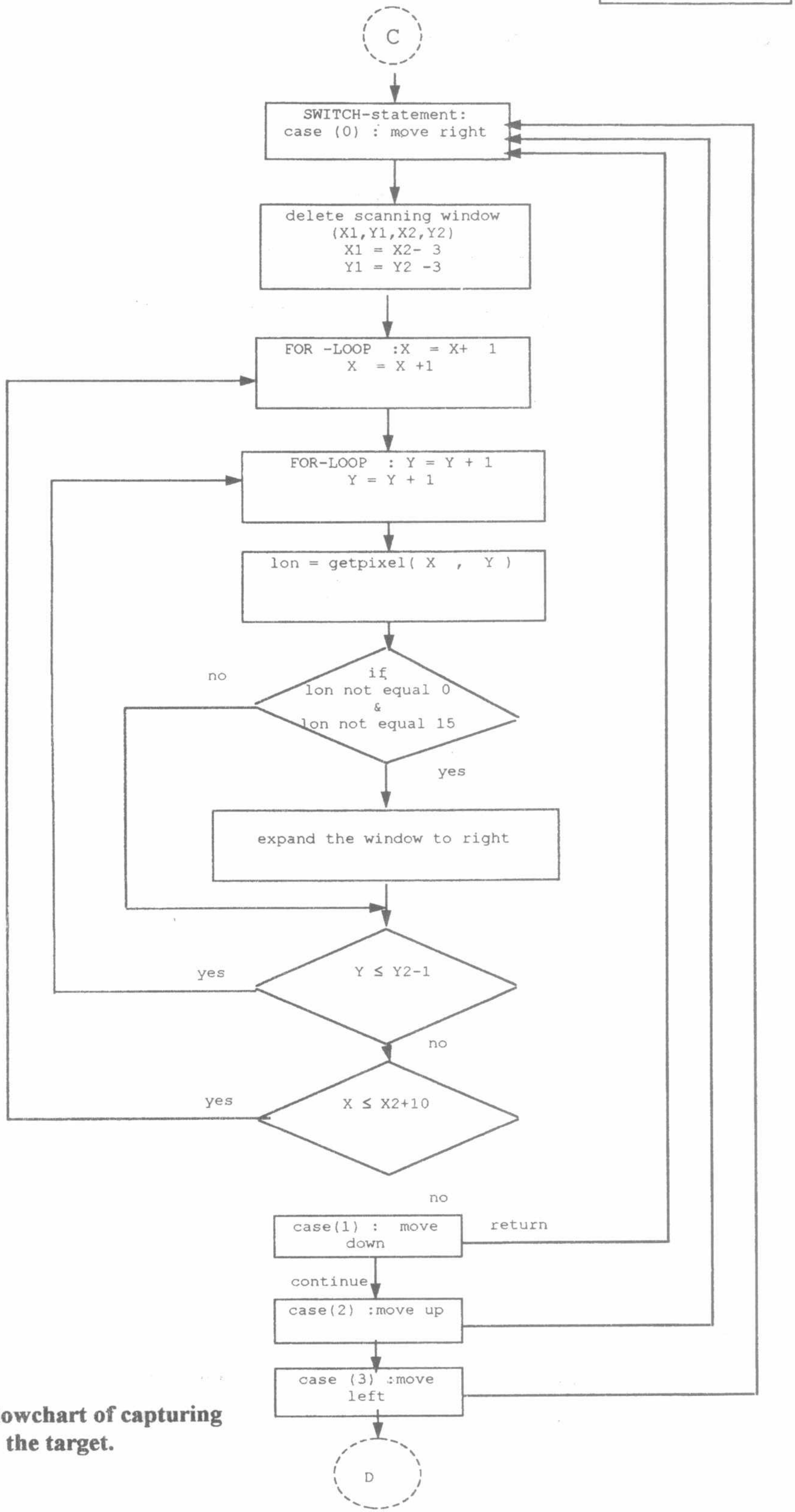

Fig. (6) Flowchart of capturing the target.

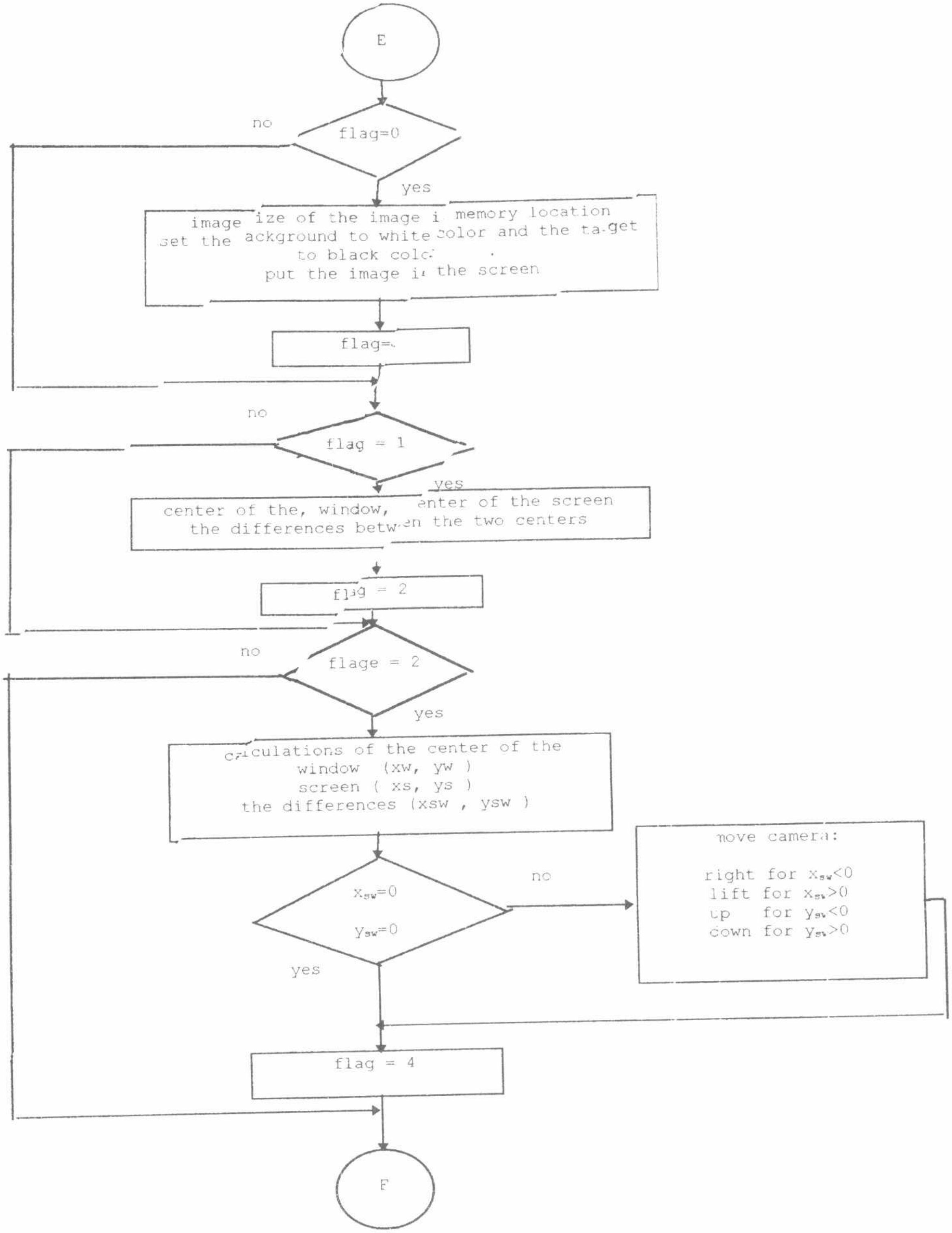
E
flag=0
no
yes
image ize of the image i memory location
set the ackground to white color and the ta.get
to black colc
put the image i the screen
flag=.
no
flag = 1
yes
center of the, window, enter of the screen
the differences betw en the two centers
fl3g = 2
no
flage = 2
yes
calculations of the center of the
window (xw, yw)
screen (xs, ys)
the differences (xsw , ysw)
Xsw=0
Ysw=0
no
yes
move camera:
right for Xsw<0
lift for Xsw>0
up for Ysw<0
down for Ysw>0
flag = 4
F

Fig. (7) Flowchart of error calculation and moving the camera.

Conclusion:

The proposed parallel algorithm had been implemented into the full transputer network [1] . For the purpose of studying the system performance , the implementation had been degraded; starting from implementation into two transputers network, three transputers network, four transputers network and finally, the implementation into five transputers network, (the maximum number of available transputers in the laboratory being 5). It was found that the speed-up factor and parallelisation efficiency are improved with increasing the number of transputers within the implemented transputer network. At first, the application had been compiled and run sequentially using one transputer, and the obtained results are considered as a reference for proposed parallel implementations.The proposed algorithm had succeeded to achieve the parallelism technique that splits it into several tasks working in parallel to save time. Also, it had realized a semi real time tracking process for a slow targets. The thread had been cached and the first step had been walked in the right way of parallelism. Modifications and developments can be done to improve the performance of the parallel system by replacing the available hardware and software capabilities with a high technology and up to date hardware and software components. Highly sensitive TV camera with high response automatic control driving system, high technology video card capable of obtaining full motion digital video, as well as capturing and manipulating video images and save them in main memory instead of the hard disk for fast processing, the recent version of transputers (e.g., IMS T9000 50MHZ) or up, and the recent parallel C package, the real time parallel image processing algorithm for detecting of fast moving target can be realized.

References:

[1] M.K. El-Mahey, An overview of parallel processing techniques (classifications, structures, architectures, software, supercomputers, and performance Metrics)", survey paper, 1996.

[2] G.Kassem , 'TV image processing systems for laser guidance', M.Sc thesis, 1997, MTC.

[3] G.J. Awcock and R. Thomas, "Applied Image Processing", 1996.

[4] SGS-THOMSON, "Inmos-transputerdata book", thired edition, 1992.

[5] "Parallel-C user guide", 3L Ltd, Scotland, 1989.

[6] Kai Hwang and Faye A.Briggs, "Computer architecture and parallel processing", 1994.

[7] S.Ghoniemy, S.F.Bahgat, G.Kassem, R.N.Mohamed,"Hardware and software implementation of image traking system using video blaster and frame difference technique", 7th. international conferenc ASATA,1997.