

MILITARY TECHNICAL COLLEGE
CAIRO-EGYPT

ICEENG 98



FIRST INTERNATIONAL CONF.
ON ELECTRICAL ENGINEERING

PERFORMANCE ANALYSIS FOR PARALLEL IMAGE PROCESSING ALGORITHM USING TRANSPUTERS

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

ABSTRACT

This paper describes a computable measure that may be used to discriminate between attributes based on configurations of parallel image processing algorithms for achieving optimal system performance. This measure is a compromise between the speed-up factor, the parallelisation efficiency, the number of used transputers, and the complication of the hardware and software algorithm configuration. This measure is described in the context of parallel implementation of digital image processing algorithm for detection of a moving target using a full link transputer

INTRODUCTION

Parallel algorithms are computer programs configured in both hardware and software. The formation of attribute configuration for a specific algorithm determines the performance of the parallel system. Many metrics are used to judge the performance of parallel algorithms running on parallel processing systems. Granularity ratio, execution time, throughput rate, speedup factor, and efficiency are some of these metrics that can be used to judge the proposed system. The proposed parallel algorithm was implemented in parallel C making available the concurrency and communication features of the transputers. The application is declared as a set of tasks which are independently compiled and linked. The set of tasks can then be mapped into a processor network, with tasks communicating via channels which can be realized as transputer links for tasks on adjacent processors.

* Professor , Faculty of Computer and Information Sciences-Ain Shams University, Cairo, Egypt.

** Associate Professor, Dep. Of Radar & Guidance , Military Technical College, Cairo, Egypt.

*** Graduate Student , Military Technical College, Cairo, Egypt.

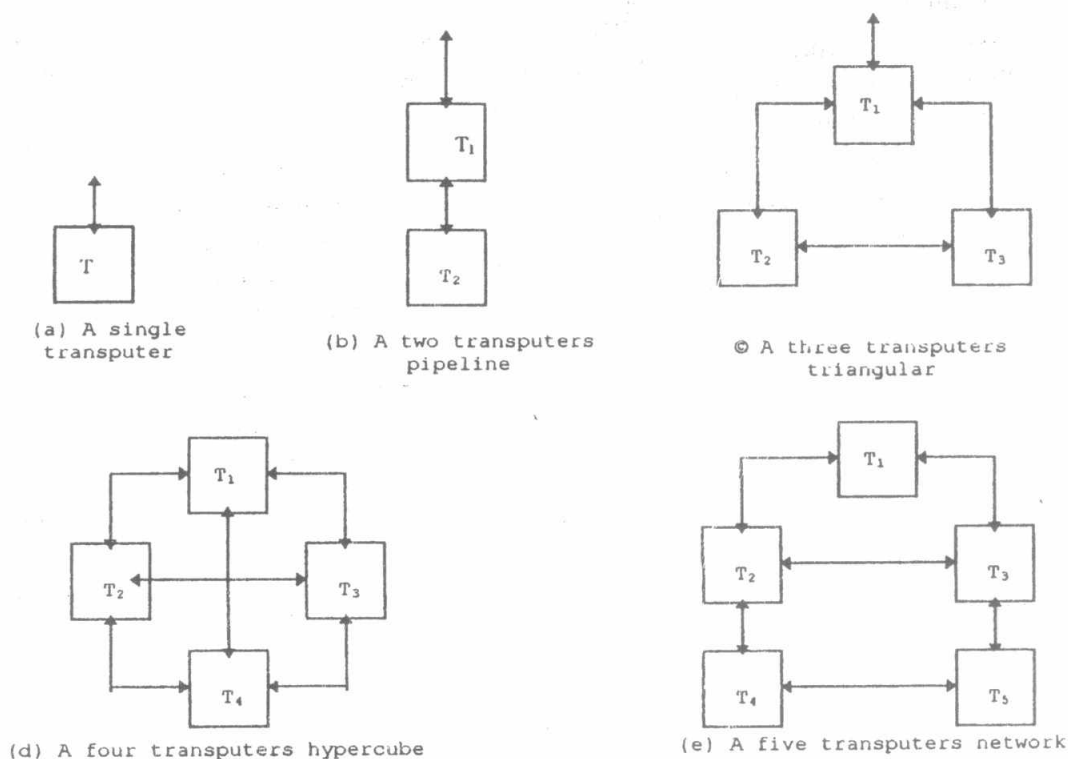


Fig (1) Transputers networks topologies.

The moving target detection algorithm is mapped into full link transputer topologies, these topologies are: single transputer topology, two transputers pipeline topology, three transputers triangle topology, four transputers hypercube topology, and five transputer network topology. Fig (1) shows the different transputers networks topologies. The performance of the proposed parallel system is studied and the results of the different transputer's topologies are analysed.

PROPOSED PARALLEL SYSTEM MAPPING STRATEGIES [1] :

The proposed system had been realized through the available capabilities. The IMS T-805 25MHZ transputer mounted on PC computer 486 66MHZ IBM compatible, 3L Ltd parallel C software package, video blaster card (VBSE), and the TV camera of type,hs6zm2-5, 8-48m, 1:1.42cs, tc9948a. The proposed parallel system is working as a semi real time [2]-[5]. The general goal for parallel image acquisition and processing is to bring pictures into the domain of the computer, where they can be displayed, manipulated and detect its position then track it. Four processes are involved here: input, display, manipulation, and output. The input signal is captured with a TV camera, real life image is displayed on its screen and is passed to the video blaster card via the PC computer to display the image on its screen. With video blaster kit, the system is capable of obtaining full motion digital video, as well as capturing and manipulating video images. The output data of the image from the video blaster is in form of PCX format, which save on hard disk, then through a conversion program it will be converted into a data format file. With respect to the parallel tasks, the main task is responsible for opening the image data file on the PC computer screen, then specifying the boundaries of scanning areas for the other tasks. For the two transputers algorithm it splits the screen into two scanning zones, for three transputers algorithm it split the screen into three scanning zones, and so on. The

scanning windows will appear on the screen; one for each zone, then the process of scanning will continue until it catches a target or in case of no targets it reaches the end of zone and stop scanning [3]. The captured target is contained completely inside a rectangle, and the difference between its center and the camera center will be calculated. This difference will be translated as an error signal that drives the TV camera towards the target. The center of the camera and the target will coincide. Then, automatically another image will be captured, and saved on the hard disk. The main task will open the image data file again and distributes the jobs for the tasks. At this time the image data file is less than the original one because it is for the area which is very close to the screen center. So, the scanning time will also be less than in the first time. The camera will move again towards the target to coincide the two centers again, and so on, the tracking process will continue however the target exist.

The mapping strategies were used to schedule the subtasks for the transputers since the parallelism is efficiently extracted from the algorithm [8]. Although it does not assure that the proposed parallel algorithm is the most effective different options were tried until the lowest execution time was obtained. These options include task scheduling, balancing loads, minimizing the communication among the processors, blocking transferred data, and avoiding duplicating tasks. When allocating the algorithm into a transputer network, all subtasks running on one transputer are blocked into one task to avoid time-slicing and internal communication overheads. The results are summarized in table (1).

Table (1) Performance of various transputer topologies.

# of Tasks in the algorithm	# of transputers	Execution time for all screen (CFV) (μs)
One task	1	9955154
Two tasks	2	6294458
Three tasks	3	5676093
Four tasks	4	5027533
Five tasks	5	4529775

For each parallel version, a Gantt chart is given to illustrate the computational requirements and the balancing of loads and how the inter-task and the inter-processor communications can be minimized [9]. The rectangles which contain the sub-tasks name represent the order of the computations from left to right and the arrows represent the inter-processor communication with the data transferred message appearing on the same arrow. The drawings assume that all the sub-tasks take the same execution time. Fig.(2) and Fig.(3) show an example of the Gantt charts of moving target algorithm mapped into two transputers array and three transputers triangular network respectively.

THE PROPOSED SYSTEM SPEED -UP FACTOR AND EFFICIENCY

One of the useful features of the transputer is that tasks can be allocated to separate processors, or may share a processor, without any recompilation of the task code. This is achieved by the use of channels for task communication, which may be realized as links for tasks on separate processors, or implemented on chip for a shared processor. Hence the task

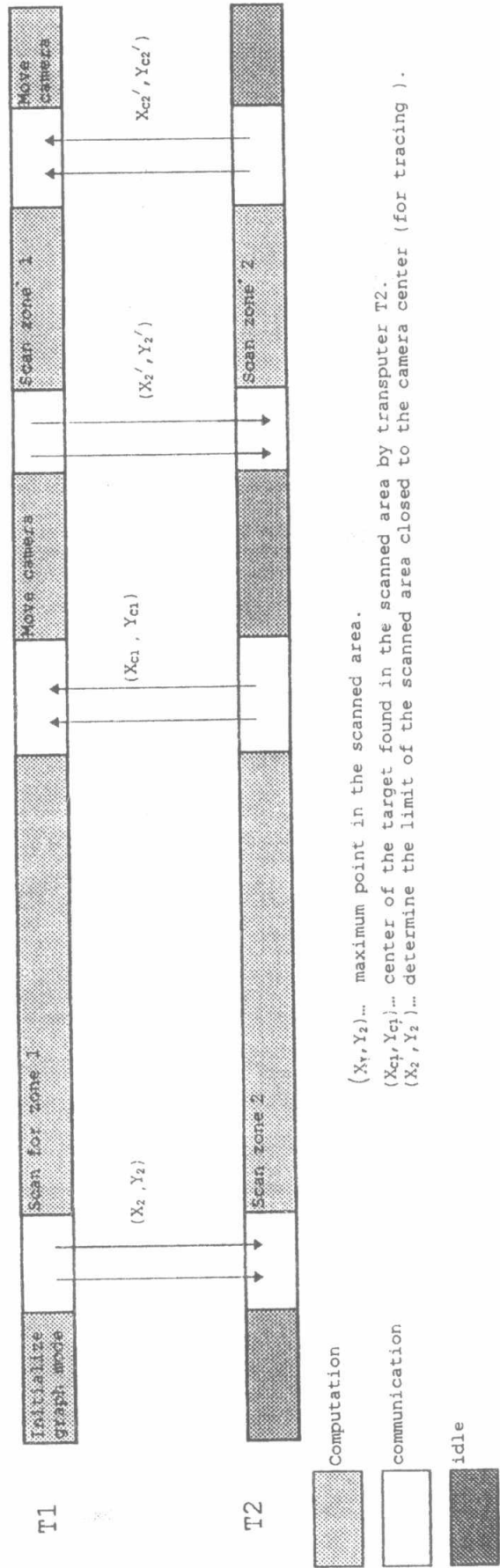


Fig. (2) Gantt chart for moving target detection algorithm mapped into two transputers pipeline.

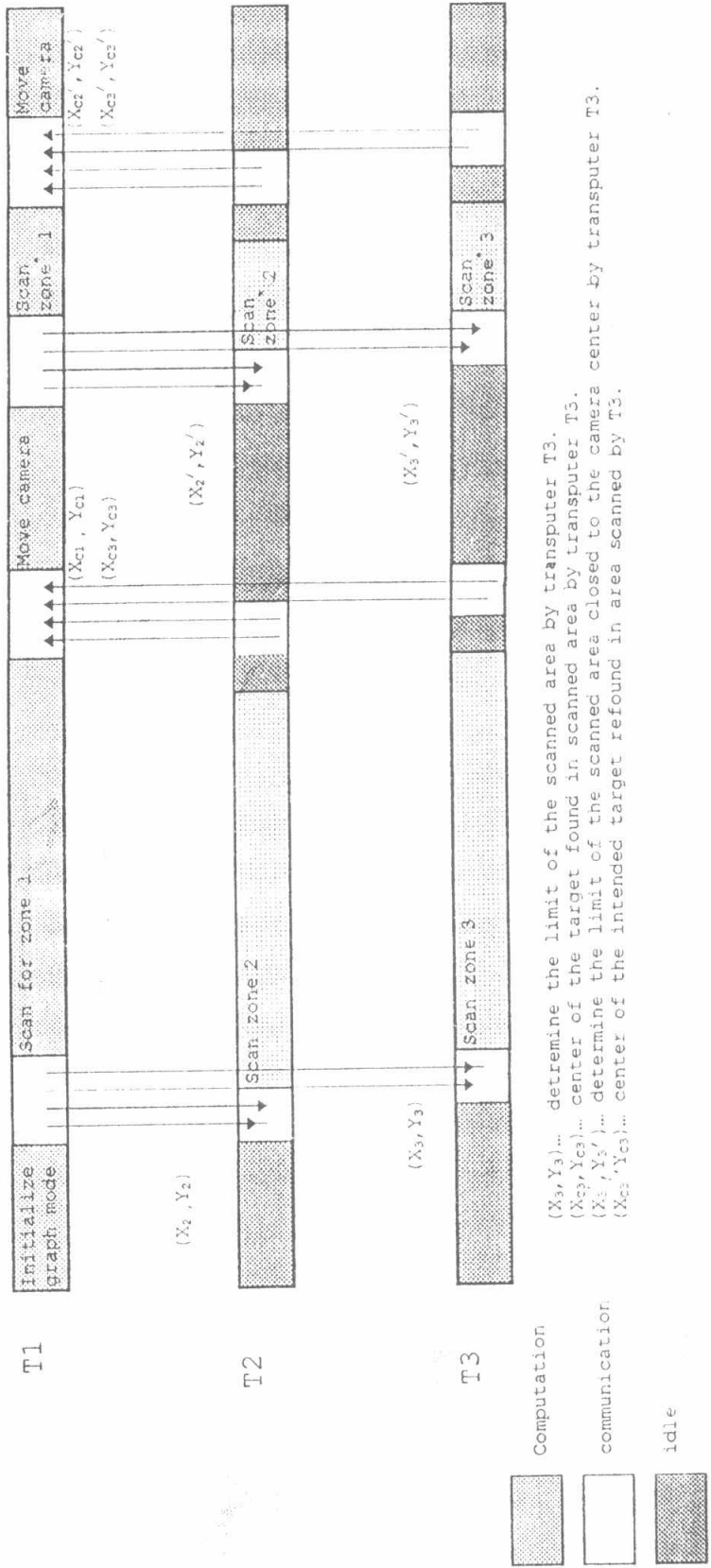


Fig. (3) Gantt chart for the moving target detection algorithm mapped into three transputers triangular.

network can be developed and debugged on a single processor, then distribute to a multi-processor environment once completed. In the same manner, placing all the tasks on one processor puts all the code in the critical path and allows timing to estimate the amount of code involved. This can be compared against the multiprocessor timings, for estimating the speedup factor and parallelisation efficiency.

SPEED-UP FACTOR

The speed-up factor (S_p) for the proposed system is defined as :

$$S_p = T_1 / T_p$$

Where : T_1 is the execution time of the single transputer system.

T_p is the execution time of the p transputers system.

Then the speed - up factor for the proposed system had been calculated as follows:

1. For one transputer system $S_1 = T_1 / T_1 = 9955154 / 9955154 = 1$
2. For two transputers system: $S_2 = T_1 / T_2 = 9955154 / 6294458 = 1.582$
3. For three transputers system: $S_3 = T_1 / T_3 = 9955154 / 5676093 = 1.753$
4. For four transputers system: $S_4 = T_1 / T_4 = 9955154 / 5027533 = 2.198$
5. For five transputers system: $S_5 = T_1 / T_5 = 9955154 / 4529775 = 2.731$

These results can be integrated in Table (2). Fig. (4) illustrates the speed-up factor of the proposed parallel system via the number of transputers in the system network.

EFFICIENCY

The parallelisation efficiency of the system performance, is defined as: the initial efficiency of the one transputer system multiplied by the value of one added to the difference between the execution time on one transputer (T_1) and execution time on i transputers (T_p) divided by the execution time on one transputer (T_1), expressed in percentage.

$$\eta(T_i) = (1 + ((T_1 - T_p) / T_1)) \eta_0$$

where: i is the number of transputers exploited in the system.

η_0 is the initial efficiency of the one transputer system. We assume that the system efficiency for one transputer (η_0) is expressed at $T_1 = 9955154 \mu s$. So the efficiency of proposed system can be calculated as follows:

1. For one transputer system $\eta(T_1) = \eta_0$
2. For two transputers system: $\eta(T_2) = (1 + ((T_1 - T_2) / T_1)) \eta_0$
 $= (1 + ((9955154 - 6294458) / 9955154)) \eta_0 = 1.36 \eta_0$
3. For three transputers system: $\eta(T_3) = (1 + ((T_1 - T_3) / T_1)) \eta_0$
 $= (1 + ((9955154 - 5676093) / 9955154)) \eta_0 = 1.42 \eta_0$
4. For four transputers system: $\eta(T_4) = (1 + ((T_1 - T_4) / T_1)) \eta_0$
 $= (1 + ((9955154 - 5027533) / 9955154)) \eta_0 = 1.49 \eta_0$
5. For five transputers system: $\eta(T_5) = (1 + ((T_1 - T_5) / T_1)) \eta_0$
 $= (1 + ((9955154 - 4529775) / 9955154)) \eta_0 = 1.54 \eta_0$

These results is shown in table (3).

Table (2) speed-up factor for the proposed parallel system.

# of Tasks in the algorithm	# of transputer	speed-up factor (S_p)
One task	1	1
Two tasks	2	1.582
Three tasks	3	1.753
Four tasks	4	2.198
Five tasks	5	2.731

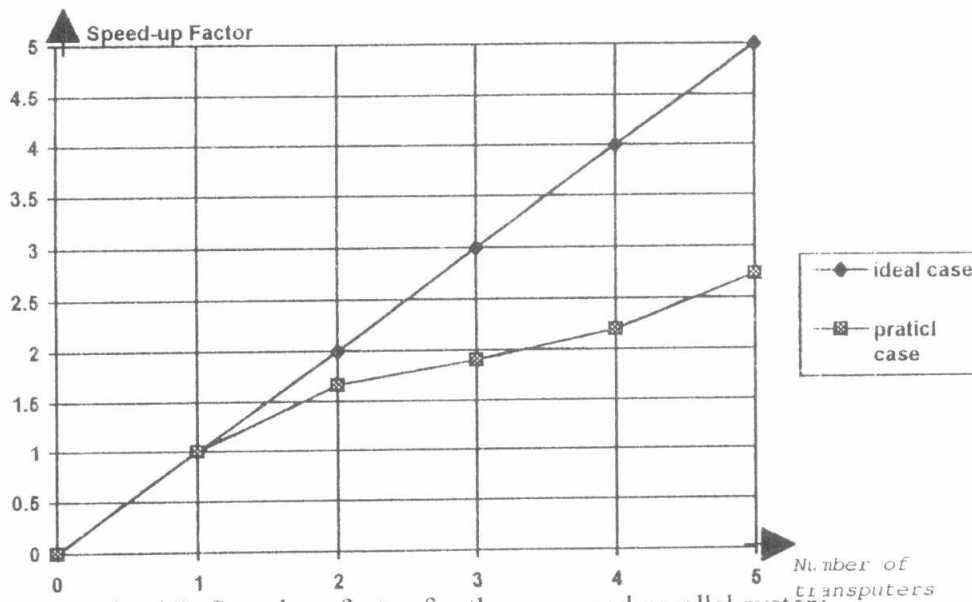


Fig. (4) Speed-up factor for the proposed parallel system.

Table (3) Parallelisation efficiency for the proposed parallel system

# of Tasks in the algorithm	# of transputers	efficiency $\eta(T_i)$
One task	1	η_0
Two task	2	$1.36 \eta_0$
Three tasks	3	$1.43 \eta_0$
Four tasks	4	$1.49 \eta_0$
Five tasks	5	$1.54 \eta_0$

SYSTEM CONFIGURATION RESULTS AND ANALYSIS

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 observations are concluded:

The maximum speedup factor (in excess of 8) was achieved when the proposed algorithm is optimized to the specific order of moving target detection problem. This is far greater than any subsequent speedup achieved due to parallelisation. On one transputer, the advantage of the partitioning appears since it makes full advantage of the decoupling of the tasks specifying the CFOV (camera field of view) zones. Even after optimization, however, the timings quoted in these results do not quite match the results given by the same application written in Occam language. Generally it is accepted that C compilers are slightly less efficient than Occam compilers, and this is generally due to the sequential nature of the C communication functions. The transputer is capable of communicating at full speed across a link while continuing with calculations, but this is not permitted with the current generation of C compilers, which insist that the processor must halt until the communication is complete. This problem is more acute for inter-processor communication, since the time taken to send a message (using C) can be approximated to: $12\mu\text{s} + 0.128\mu\text{s}/\text{byte}$ for on-chip communication, and: $15\mu\text{s} + 0.748\mu\text{s}/\text{byte}$ for processor to processor communication. Hence the reduction in parallelisation efficiency observed for similar partitioning strategies.

The available transputer version is IMS T805 25MHZ, and the available parallel C software package is 3L Ltd, 1989 [11]. One of the shortages in this software package's capabilities was the missing of the graphics library. So, the design of the graphics library including the essential graphics functions needed for the proposed application was the first step done before converting of the sequential algorithm into parallel one. Another shortage of the software package was that the main task which exploited the root transputer is the only task that can deal with the standard library in afserver (which can use DOS commands), but the other tasks in other transputers can not. They use only the stand alone library which cannot see the afserver and cannot share the DOS commands. The nature of the proposed application required to run each parallel task with standard library and share the afserver with the main task in root transputer, so using the multiplexer technique was the only solution for that problem. The multiplexer that available in this software package which called file-MUX was the solution that gave opportunity for each parallel task to share the afserver with the main task in the root transputer. However, the file-MUX caused a delay time in the execution of the task connected with it. This delay time for scanning all the CFOV was found to be $2538225\mu\text{s}$. Another disadvantage of using the file-MUX is that it derives the transputer from two links, used to connect the file-MUX with the previous and

next transputers. These disadvantages may force the designer to use greater number of transputers than the number of the parallel tasks used.

The current parallel C software package is convenient to the applications that need amount of complex computations rather than these image processing applications. Although this serious opinion, the parallelism as an absolute technique is very essential for the applications that need to run in real time which need high speed and high accuracy. This can be achieved through the higher technology INMOS transputers, e.g., IMS T9000, and modified version of the parallel C software package.

Theoretically, the relation between the number of used transputers and the speed-up factor is a linear relation see Fig. (4), it means that for use a two transputers network, the speed-up factor will equal two, and for three transputers network, it will equal three, and so on. Practically, there are some factors that make this relation nonlinear. The delay time due to waiting for communication intervals between the tasks, at these times the calculation processes will stop, waiting for completing the communication processes. Also, the delay time due to inserting the multiplexers (file MUX), in between the tasks will increase seriously due to use layers of these file MUXs. All these factors cause the nonlinearity relation. To compensate these time delays, more transputers will be used. But this compensation will complicate the system hardware and software configuration. Although the increasing of the transputers number and the complexity of the system hardware and software, the speed-up factor was not sufficiently increasing to worth this work done. So it is important to make a compromise between the number of used transputers, the required speed-up factor, and the system hardware and the software configurations. As a result, each application has to implement in a certain (critical) number of transputers which realize this compromise.

One of the important specifications of using parallelisation technique is to realize high speed efficiency using the available low speed processors realize as if you use a high speed and technology processor. In the proposed algorithm, the system performance had been realized by parallel implementation of the system into a five transputers network instead of using a sequential (conventional) algorithm that implemented into Intel-486 66MHZ processor. The high technology transputer version will give the opportunity to speed up for the system more than the ordinary (sequential) up to date processor can do.

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 and parallel C package, the real time parallel image processing algorithm for detecting of fast moving target can be realized.

CONCLUSION

Parallelisation techniques are very powerful and essential for the applications that need to speedup the execution time (e.g. real time applications). The proposed parallel system performance can be measured in some metrics such that granularity ratio, execution time, throughput rate, speedup, and efficiency. Optimization of the conventional algorithm, speeded up the system about eight times. Results and analysis of the sequential algorithms, conventional and optimized algorithms, and the parallel algorithms performance conclude that the relation between the number of transputers used to implement the system and the speeding up factor is a nonlinear relation. The nonlinearity of this relation is due to the communication intervals in-between the parallel tasks and the idle time due to the processors waiting the results from each other. The best performance of the parallel system can be achieved by using specific number of transputers that achieved by compromise between the maximum speedup factor with the noncomplicated hardware or software. Fast and high technology transputers achieve better results, e.g., (IMS T9000 50MHZ). Parallel Pentium processors are the up to date parallel processor, until now, that achieved the best parallel system performance.

REFERENCES

- [1] S.F.Bahgat " Parallel Implementation of Image Processing Algorithm for Detection a Moving Target Using a Full Linking Transputer" First International Conference on Electrical Engineering Cairo, Egypt, March 24 - 26 , 1998.
- [2] G.J. Awcock and R. Thomas, "Applied Image Processing", 1996.
- [3] G.Kassem , 'TV Image Processing Systems for Laser Guidance', M.Sc Thesis, 1997, MTC.
- [4] O.EL-Mowafy, "Image Processing Systems For TV Image Tracking", M.Sc.Thesis, 1994, MTC.
- [5] LSI/DAIA - Division of Artificial Intelligence and Automation (H2.TXT), Internet.
- [6] S.Yalamanchili, W.N.M artin, and J.K.Aggarwal "Extraction of Moving Object Discription via Differencing" .Computer Graphics and Image Processing 18,pp 188-201, 1982.
- [7] Ji-Chien Lee, "Digital Motion Estimation and Image Restoration on VLSI", H7.TXT, Internet, July 1991.
- [8] Kai Hwang and Faye A.Briggs, "Computer Architecture and Parallel Processing", 1994.
- [9] M.K El-Mahey, "Parallel Implementation of Target Tracking Kalman filter onto Full-Linking Transputer Network", Paper in Germany Comferance, 1993.
- [10] SGS-THOMSON, "Inmos-Transputer Data Book", Third Edition, 1992.
- [11] M.K. El-Mahey, "Effecient Parallel Implementation of Target Tracking Kalman filter Using a Network of Transputers", Aerospace Vehicle Dynamics and Control Conference, 1992.
- [12] "Parallel-C User Guide", 3L Ltd, Scotland, 1989
- [13] 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.
- [14] M.K. El-Mahey, An Overview of Parallel Processing Techniques (Classifications, Structures, Architectures, Software, Supercomputers, and Performance Metrics)", Survey Paper, 1996.
- [15] Peter Persson, Database consultant NCR SÆCopenhagen, "Prallel Data Systems", 1996.
- [16] AT&T Globale Information Solutions Comany, AR&T Workmark"5100M Server Brochure", 1995.
- [17] NCR Corporation,Ohio USA, NCR System 3600, Product Discription, 1992.

