

Parallel Algorithms for 3D Multi-Level Median Filtering with Motion Compensation

P. Liu, M. N. Chong and W. B. Goh

ping@sentosa.sas.ntu.ac.sg
School of Applied Science
Nanyang Technological University
Singapore 2263

Abstract: This paper aims to develop effective parallel processing techniques for 3-Dimensional (3D) Multi-level Median Filtering (3DMMF) with motion compensation on video sequences. Due to the nature of unbalanced load of the filtering algorithm and the objective for the parallel approach, two methods of dynamic load balancing have been proposed and compared. They are Sender-Initiated-Load-Balancing (SILB) and Receiver-Initiated-Load-Balancing (RILB) algorithms. We propose a SILB algorithm which utilises the spatial-temporal characteristics of the processed sequences for load prediction to achieve dynamic load balancing. Both theoretical analysis and experimental results on the IBM SP2 computing surface have been presented in this paper.

1. INTRODUCTION

The recent effort to restore and preserve classic films such as Disney's Snow White and the Seven Dwarfs [1] has highlighted the enormous challenges and demands of digital picture restoration. Forty SGI workstations were used round the clock, seven days a week, for eighteen weeks for just this effort alone. The computational load of motion picture restoration algorithms is mammoth. Our work attempts to address this problem through the use of parallel processing approaches.

Motion Compensated, 3-Dimensional Multi-level Median Filtering (MC-3DMMF) algorithm has been developed to remove impulsive noise in archived film or video material [2]. It has been proven to have better performance compared with the standard 2D techniques for motion picture restoration [3,4]. Compare with the traditional 2D median filter, the new MC-3DMMF allows the median operation to preserve more details in the image sequence. The MC-3DMMF algorithm is computational expensive. The timing performance of the 3DMMF (without motion compensation) algorithm for a 256x256 pixels/frame sequence is 1.6 sec/frame using a SPARC10 workstation under near quiescent state; with motion

compensation, it would take more than a hundred seconds to process a frame.

Parallel implementations of 2D median filtering algorithms on transputer network have been reported in [5,6]. This paper presents a novel load balancing technique which aims to circumvent the load imbalance problem found in the parallel MC-3DMMF algorithm. The proposed Sender-Initiated-Load-Balancing (SILB) algorithm incorporates an a priori workload estimator which is derived from a motion estimation algorithm and the property of motion continuity. Theoretical compute models are derived to analyse the performance of the proposed algorithm with the commonly adopted Receiver-Initiated-Load-Balancing (RILB) scheme [7]. The parallel algorithms are implemented on an IBM SP2 8-node parallel computer, in which the results are used to verify the theoretical compute models.

Unlike the fixed configuration of transputer network which is heavily dependent on network topology, SP2 utilises a high performance switch [8] to route data to all processors.

2. DYNAMIC LOAD BALANCING ALGORITHMS

One of the main factors which affects the compute efficiency of parallel MC-3DMMF algorithm is the imbalance workload distribution among the processors. The statistical characteristics of the sub-images (blocks which are partitioned from an image) can generally be categorised into: (i) no displacement, (ii) exhibits motion continuity property in the displacement (iii) irregular and vigorous displacement which defies the property of motion continuity. Therefore, the computational load of these sub-images varies with their statistical characteristics. Methods based on simple static load distribution scheme would not be efficient for the parallel implementation of MC-3DMMF.

Dynamic load-balancing schemes try to improve static techniques by providing more flexibility in assigning

workload to processors. A common strategy is RILB. RILB is a demand-driven approach which has the drawback of expensive communication cost. The alternative approach is SILB scheme [7]. In SILB, the work splitting is sender initiated in which the generation of tasks/sub-images is independent of the work requests from the idle processors. These tasks are delivered to the slave processors with or without demand by the sender (master). Although the communication cost in SILB is much lower than RILB, SILB requires to know the workload of the application in order to judiciously distribute the workload.

Sender Initiated Load Balancing (SILB) Algorithm

In MC-3DMMF, the sub-filter masks [4] as shown in figure 1 are adopted.

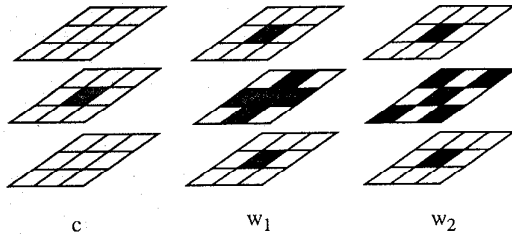


Figure 1 Sub-filter masks used for 3D MMF presented by Alp *et.al* [4]. $z_i = \text{median}[w_i]$, where $1 \leq i \leq 2$. The Filter output = $\text{media}[z_1, z_2, c]$.

A 9x9 block matching motion estimation method is used in the MC-3DMMF algorithm. The search area is 15x15, and the motion estimation algorithm is a threshold block matching search.

For parallel processing, each image frame at time t_k is decomposed into N non-overlapping rectangular sub-image of size n by m , taking into consideration of the number of processors, the communication bandwidth and the setup latency. Three successive frames, which are at time t_{k-1} , t_k and t_{k+1} , are broadcasted to each processor in order for the non-overlapped sub-images to be processed independently on each compute node.

Each node starts with the same number of sub-images. Upon the completion of the restoration process at all the slave processors, the restored sub-images with the information of the motion vectors are sent back to the master. The master processor uses the motion vectors information of frame k to predict the load for frame $k+1$. An a priori workload estimator is used by the master to predict the Load Contention Number (LCN). LCN accounts for the load of the processors, and is used by the master processor

to monitor and distribute the workload for the slave processors [9]. In details, the steps of movement from the pixel I on frame t_k to the corresponding pixel I' on frame t_{k+1} was recorded using threshold block matching method. Therefore, the computational load of pixel I' on frame t_{k+1} which is mainly due to the motion can be predicted accordingly. This predicted movement is used as the LCN to reschedule the workloads for each processors. While master processor is working on the scheduling of workload for each slave processor, each slave processor is concurrently processing frame $k+1$ before it actually receives the load scheduling instructions from the master processor.

Receiver Initiated Load Balancing (RILB) Algorithm

For comparison purpose, the commonly adopted RILB algorithm [9] is also realised and implemented on the IBM SP2 computer. The RILB decomposes the problem into large number of independent tasks (sub-images), which are then assigned to processors one-at-a-time or in small groups. When a processor completes one task, it receives another and the process continues until all the tasks are completed. In our implementation, the image is partitioned up to $p \times 32$ number of tasks/sub-images to minimise potential variations between the workloads, where p is the number of processors. Although the RILB scheme offers more precise load balancing than static schemes, RILB scheme will only be successful when the improvements in processor utilization exceed the overhead which is the communication cost. Therefore the scalability of RILB is limited by the communication cost of a particular system.

3. COMPUTATION MODELS

Two computation models are derived to compare the performance of SILB and RILB algorithms. For SILB algorithm, the processing time for each frame, t_s is:

$$t_s = T/p + t_{err} + t_{Scom} + t_{seq} \quad (1)$$

where T is the sequential processing time for each frame; p is the number of processors; t_{err} is the additional processing time to account for the discrepancy between the predicted load and the actual workload; t_{Scom} is the data transferring time for SILB algorithm, $t_{Scom} = t_{com1} \times (2+p) + t_{com2}$, which includes t_{com1} , the packing and unpacking time for paralleling processing and t_{com2} , the data communication time between master and slaves. t_{seq} accounts for the overheads such as the image tiling process. The extra processing time, t_{err} due to the error in predicting and distributing the workload is critical

to the success of SILB algorithm, the error factor t_{err} can be expressed as:

$$t_{err} = \beta \times (T/p) \quad (2)$$

and
$$\beta = \frac{t_{slowest} - (T/p)}{(T/p)} \quad (3)$$

where $t_{slowest}$ is the processing time measured from the slowest compute node.

Similarly for RILB algorithm, the processing time for each frame, t_r is:

$$t_r = T/p + \Delta t + t_{Rcom} + t_{seq} \quad (4)$$

where Δt is the time difference between the slowest processor and the average compute node; t_{Rcom} is the data transferring time in RILB algorithm and can be defined as:

$$t_{Rcom} = 2 \times (t_{com1} \times (1 + N) + t_{com2}) \quad (5)$$

where N is the number of tasks/sub-images.

It can be seen from equations (1), (4) and (5) that the success of SILB depends on the accurate prediction of workload, while RILB will only be successful when the improvements in processor utilization exceed the communication cost, t_{Rcom} . If the number of processors in a network grows, the number of tasks/sub-images, N must be increased accordingly in order to improve the processor utilization.

In order for SILB to out perform RILB algorithm, the maximum allowable error for β can be derived from equations (1) and (4):

$$\beta_{max} < \frac{(\Delta t + t_{Rcom} - t_{Scom})}{(T/p)} \quad (6)$$

4. IMPLEMENTATIONS AND RESULTS

Two test sequences are used in the experiments:

- (i) Synthetic sequence: the images are generated from a random number generator with a bar moving at the centre as shown in figure 2a. The frame size is 256x256 pixels.
- (ii) "Cheers" Video sequence: the video sequences are corrupted with random noise. The frame size is 256x256 pixels (figure 3a).

PVMe [10] is chosen as the parallel environment and up to 7 nodes are selected as the slave processors for the

studies. The speed-up characteristics of SILB and RILB algorithms on an 8-node IBM SP2 parallel computer, are measured as shown in figures 2b and 3b.

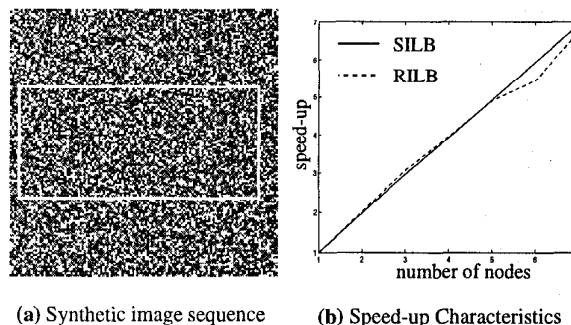


Figure 2 The speed-up characteristics of SILB and RILB algorithms tested on a synthetic image sequence. The rectangle indicates the moving block in the sequence.

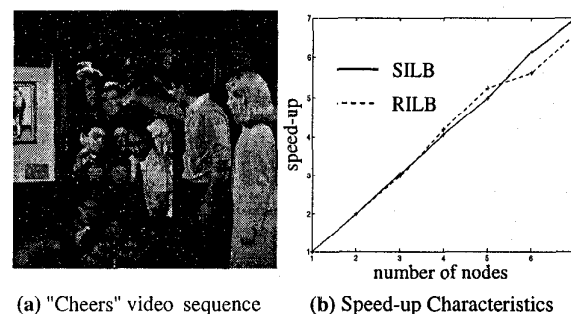


Figure 3 The speed-up characteristics of SILB and RILB algorithms tested on a noise corrupted video sequence.

As shown in figures 2b and 3b, the speed-up characteristic of the proposed SILB is better than RILB algorithm. For the synthetic test sequence, as there is no noise and only translational (small) displacement, the linearity of the speed-up for SILB is quite good. The results shown in figure 2b also indicate that the workload prediction is accurate, thus achieving a higher gain than RILB algorithm. When a TV video sequence is used, it can be seen that the trend of the difference of the performance between SILB and RILB is similar compared with figure 2b, but the scalability for SILB scheme is worsened, this is likely due to the region of the sequence which fails to exhibit motion continuity and the appearance of random noise which affects motion estimation. To examine the details of the timing performance, the processing time of the

individual frames is measured as shown in figure 4. It can be seen that when the a priori estimator more accurately predicts the workload (frames 12, 13, 14 and 17), the difference in performance between RILB and SILB is more significant.

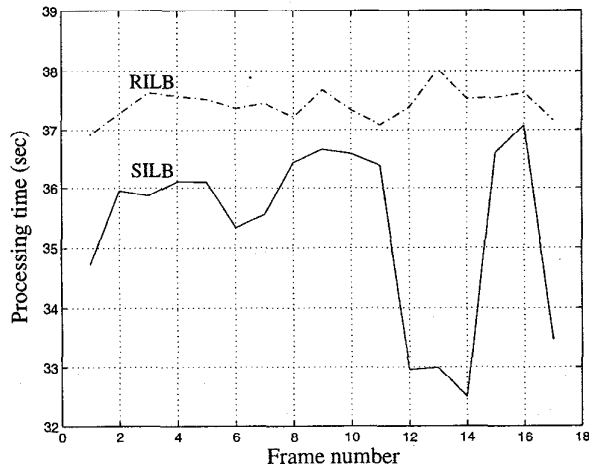


Figure 4 Processing time for restoring the "Cheers" video sequence. 7 nodes on IBM SP2 machine are used as slaves.

5. CONCLUSIONS

A novel SILB algorithm is realised and implemented fully on an 8-node IBM SP2 computer. A successful parallel MC-3DMMF must take into account of the trade-offs between the communication costs and accuracy in predicting the workloads. The prediction of the workloads is in turn depended on the accuracy of the motion estimation algorithm. The condition for the SILB to out perform RILB algorithm is derived as shown in equations (6). The computation models also infer that with the increase in network size, the scalability of the proposed SILB will be more efficient than the RILB algorithm due to the lower communication cost in the proposed SILB algorithm.

ACKNOWLEDGEMENT

This project is funded by the Academic Research Fund (CUED-05) of Singapore. We would like to thank Dr. Peter Rayner and Dr. Anil Kokaram in Department of Engineering, Cambridge University for their assistance and providing the test video sequences.

REFERENCE

[1] B. Fisher "Digital Restoration of SNOW WHITE: 120,000 Famous Frames Are Back" *Advanced Imaging*

pp32-36 Sept. 1993.

[2] Anil C. Kokaram "Motion Picture Restoration", PhD thesis, Department of Engineering University of Cambridge, May 1993.

[3] A C Kokaram and P Rayner "Removal of impulsive noise in image sequences" *Singapore International Conference on Image Processing* pp629-633 Step. 1992.

[4] Bilge Alp, Petri Haavisto, Tiina Jarske, Kai Oitamo and Yrjo Neuvo "Median - based algorithms for image sequence processing" *SPIE Visual Communications and Image Processing* pp122-133 1990.

[5] M. Avvenuti, G. Corsini and C.A. Prete "Parallel Implementation of the Median Filtering of Images by Using a Network of Transputers" *6th Mediterranean Electrotechnical Conference Proceedings* 1.1 pp 439 - 42 1991 USA.

[6] P.J. Narayanan and L.S. Davis "Replicated Image Algorithms and Their Analyses on SIMD Machines" *International Journal of Pattern Recognition and Artificial Intelligence* 6 Iss: 2 - 3 pp335-52 1992.

[7] Vipin Kumar, Ananth Y Grama and Nageshwara Rao Vempaty "Scalable Load Balancing Techniques for Parallel Computers" *Journal of Parallel and Distributed Computing* 22 pp60 - 79 1994.

[8] Craig B. Stunkel, Dennis G. Shea, Biilent Abali, Mark Atkins, Carl A Bender, Don G. Grice, Peter H. Hochschild, Douglas J Joseph, Ben J Nathanson, Richard A swetz, Robert F Stucke, Michael Tsao and Philip R. Varker "The SP2 Communication Subsystem" *IBM POWERparallel Systems: Documentation*. 22 Aug. 1994.

[9] Taieb F Znati and Rami G Melhem "A uniform framework for Dynamic Load Balancing Strategies in Distributed processing Systems" *Journal of Parallel and Distributed Computing* 23 pp246-255 1994.

[10] IBM AIX PVMe User's Guide and Subroutine Reference.