Computation Aware Scheme for Visual Signal Processing

Anand Paul[1], Yung Chuan Jiang[2] and Jhing Fa Wang[3]
Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
Email: [1]anand@icwang.ee.ncku.edu.tw,[2]ycjiang@mail.chna.edu.tw,[3]wangjf@csie.ncku.edu.tw

***Abstract*— Computation aware scheme for video signal processing is proposed in this paper. Computation power and time is reduced by dynamically scheduling usage of processor resources environment for video sequence depending up complexity of the video. And different video coding algorithm is selected depending upon the nature of the video. Simulation results show the effectiveness of the proposed method.**

***Index Terms*— Visual Signal Processing, Computation aware system, Dynamic Power Management**

I. Introduction

For the past twenty years several video coding standards such as MPEG-1, MPEG-2/4, H.263, H.264 [1], have been playing a significant role in digital media revolution. The recent advancement in H.264 significantly increases coding efficiency and processor computation power with inclusion of variable block motion estimation and many other coding features [2], and that is very true for HDVT sequences and there is a need to efficiently process the video. Recently, the computation-aware (CA) concept is getting attention of video processing researchers. In software implementations, processors may have to support video coding of different frame rates, frame sizes, and search ranges. In hardware implementations, even if the frame rate, frame size, and search range have been clearly determined, the computation resource (e.g. operating frequency) may still be adjusted according to the battery power for portable devices.

The ability to enable and disable components, as well as tuning their performance to the workload (e.g., user's requests), is important in achieving energy efficient utilization. In this work, we will present new approaches for lowering energy consumption in both system design and utilization. The Policy presented in this thesis has been experimented using an event driven simulator, which demonstrated the effectiveness in power savings with less impact on performance or reliability [5].

The fundamental premise for the applicability of power management schemes is that systems, or system components, experience non-uniform workloads during normal operation time. Non-uniform workloads are common in communication networks and in almost any interactive system.

Dynamic power management (DPM) techniques achieve energy efficient utilization of systems by

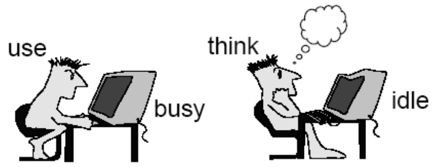


Fig.1 An Interactive Systems is busy or idle depending on the user requests

selectively placing system components into low-power states when they are idle [6]. As illustrated in figure 1.

The goal of CA BMA[Block Matching Algorithm]s is to find the best block matching results in a computation-limited and computation-variant environment. The authors of [3] are pioneers of CA BMAs. They contributed a novel scheme. But it was intented for software and not feasible in hardware.

Different search patterns have different merits and thus should be combined into one CA BMA[4]. Figure 2 compares FSBMA, TSS, and PDS. Among all frames, FSBMA gives the best quality (motion compensated PSNR). On average, PDS is better than TSS. However, when the camera pans very fast, TSS is better than PDS. The results are quite reasonable. When the motion field is small and regular, MV predictor works well, and the diamond pattern can quickly find a good match. As for TSS, the first step search points are dispersed, making final results tend to be trapped in local minima. On the contrary, when the motion field is large and complex, MV predictors do not work well, and the diamond pattern moves slowly toward the best MVs with a high probability of being trapped in local minima. In this case, TSS first glances the entire search area and has better chances to focus on the vicinity of global minimum.

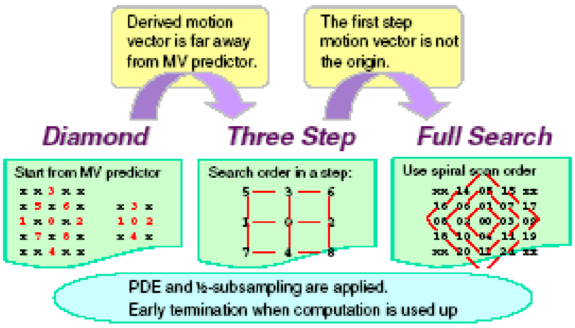


Fig. 2. Adaptive search strategy of [4]

doi:10.4304/jsw.5.6.573-578

II. Video Content Analysis And Requester Model

In order to evaluate the performance of video compression coding, it is necessary to define a measure to compare the original video and the video after compressed. Most video compression systems are designed to minimize the *mean square error (MSE)* between two video sequences *Ψ1* and *Ψ2*, which is defined as

$$MSE = \sigma_e^2 = \frac{1}{N}\sum_t \sum_{x,y} [\Psi_1(x,y,t) - \Psi_2(x,y,t)]^2 \qquad (1)$$

where N is the total number of frames in either video sequences.

Instead of the MSE, the *peak-signal-to-noise ratio (PSNR)* in decibel (dB) is more often used as a quality measure in video coding, which is defined as

$$PSNR = 20\log_{10}\frac{255}{MSE} \qquad (2)$$

It is worth noting that one should compute the MSE between corresponding frames, average the resulting MSE values over all frames, and finally convert the MSE value to PSNR.

A. Predictive Coding

However, the performance is not efficient to compress the fast changing video, so other methods to remove the spatiotemporal redundancy are proposed for video compression. The most famous method is the block matching algorithm, or motion estimation and motion compensation method. The block matching algorithm divides the current frame and the previous frame into several macroblocks, comparing the blocks in the two frames and trying to search for the best matched pairs for each block.

The dissimilarity *D(s,t)* (sometimes referred to as error, distortion, or distance) between two images Ψ_n and Ψ_{n-1} is defined as follows

$$D(s,t) = \sum_{Vy=1}^{p}\sum_{Vx,=1}^{q} M[\Psi_n(x,y), \Psi_{n-1}(x+V_x, y+V_y)] \qquad (3)$$

where *M(u,v)* is a metric that measure the dissimilarity between the two arguments u and v.

There are several types of matching criteria and two most frequently used is MSE and MAD, which is defined as follows:

- *Mean square error* (*MSE*): $M(u,v) = (u - v)^2$ (4)
- *Mean absolute difference* (*MAD*): $M(u,v) = |u - v|$ (5)

A study based on experimental works reported that the matching criterion does not significantly affect the search. Hence, the MAD is preferred due to its simplicity in implementation. These two measures (MSE/MAD) gives us a clear idea about the video frame. And MSE/MAD of the previous frame gives more information about the current frame and its complexity. If the current frame of a video sequence is fast moving(complex) then more processor core are allotted to process else only one processor core is assigned

B. Requester Model

A special entity called “requester” that generates workloads including IO requests and computation needs. Request modeling is one essential part of power management because policies predict future workloads based on their requester models. We consider two requester models for designing policies: single requester, multiple requesters. These models are increasingly complex and close to the programs running on realistic interactive systems like a laptop computer.

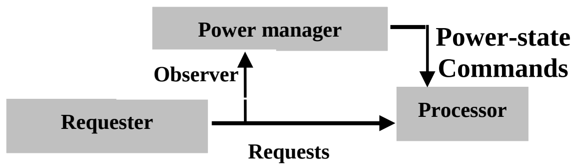


Fig. Single requester model

Figure 3 depicts the concept of the single-request model. The requester generates requests for the device; mean while, the power manager observes the requests. Based on this observation, the power manager issues commands to change the power states of the device. Some policies explicitly use this model in determining their rules to change power states [7]; some other policies implicitly assume a single requester [6].

C. Multiple-requester Model

In complex system, there may be more than one entities that generate requests. For example, in a multiprogramming system, several process may generate requests to the same device. Different processes consume different energy. In particular, a server consumes large energy on both a network card and hard disk.

The request inter arrival times in the active state (the state where one or more requests are in the queue) for all three devices are Poisson distribution in nature. Thus, we can model the user in active state with rate λ and the mean request inter arrival time 1/λ where the probability of the hard disk or the Smart Badge receiving a user request within time interval *t* follows the Poisson probability distribution shown below. The exponential distribution [13] does not model well arrivals in the idle state. The model we use needs to accurately describe the behavior of long idle times as the largest power savings are possible over the long low-power periods. We first filter out short user request inter arrival times in the idle state in order to focus on the longer idle times.

II. Proposed Power Aware System

A general model for dynamic power management learning agent is shown in figure 5, the accumulation of experience that guides the behavior (action policy) is represented by a cost estimator whose parameters are learned as new experiences are presented to the agent.

The agent is also equipped with sensors that define how observations about the external process are made. These observations may be if necessary combined with past observations or input to a state estimator, defining an information vector or internal state which represents the agent's belief about the real state of the process The cost estimator then maps these internal states and presented reinforcements to associated costs, which are basically expectations about how good or bad these states are, given the experience obtained so far. Finally, these costs guide the action policy. The built-in knowledge may affect the behavior of the agent either directly, altering the action policy or indirectly, influencing the cost estimator or sensors.

The experience accumulation and action taking process is represented by the following sequence. At a certain instant of time, the agent:

1. Makes an observation and perceives any reinforcement signal provided by the process.
2. Takes an action based on the former experience associated with the current observation and reinforcement.
3. Makes a new observation and updates its cumulated experience.

A. The Reinforcement Condition of IDPM

The basic assumption of Markov Decision Processes is the Markov condition: any observation made by the agent must be a function only of its last observation from the state transition and action on select the best policy and change the control to the best (plus some random disturbance)

$$o_{t+1} = f\left(o_t, a_t, w_t\right) \tag{6}$$

Where o_t is the observation at time t, a_t is the action taken to

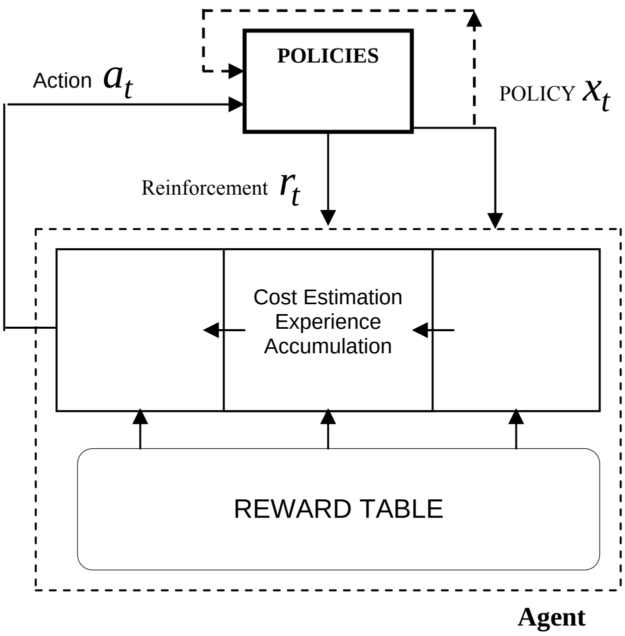


Fig.4. An IDPM model of the learning agent

predict best policy and w_t is the reward weight. o_t provides complete information about x_t . This is equivalent to perfect observability of best policy,

Of particular interest is the discounted infinite horizon formulation of the Markov Decision Process problem. Given

- A finite set of possible actions $a \in A$,
- A finite set of polices $x \in X$,
- A finite set of bounded reinforcements (payoffs $r(x,a) \in \Re$;

The agent gives the reward to which policy minimizes the power consumption. The condition of policies for getting the reward is power saving p_{save} in sleep time should be more than the sum of power consumption at wake up time p_{wake} and power consumption of idle time p_{idle} of embedded system.

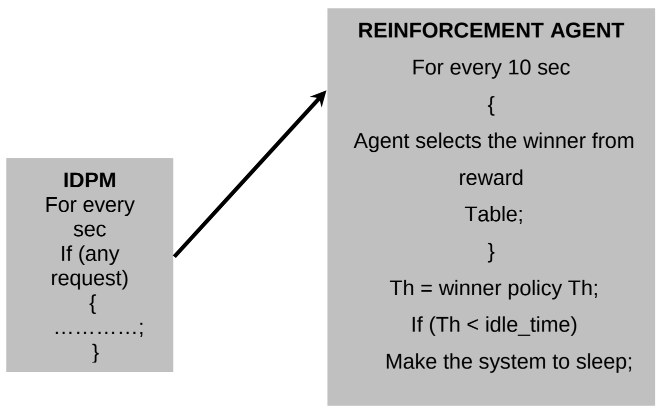
$$T_{th-sleep} \times p_{save} \geq T_{wake} \times p_{wake} + T_{idle} \times p_{idle}$$

To get reward, the policy should make the embedded device to sleep state until the above condition is satisfied. So the threshold time T_{th_sleep} is

$$T_{th-sleep} \geq \frac{T_{wake} \times p_{wake} + T_{idle} \times p_{idle}}{p_{save}}$$

To get reward, the policy should make the system idle state, above or equal to the threshold time T_{th_sleep}.

The pseudo code for the IDPM given below



III. MINIMUM DEPTH ALGORITHM

A PRC parallel partitioning algorithm is presented in this subsection. Determining every MFC in a graph is first introduced. Then the minimum depth obtained recursively by obtaining k-MMGs is explained. Assume that the area constraint is 23, i.e. A_{DRPPU} = 23. For example, 3-MFCs including C_x, C_y and C_z in Figure 4 can be selected for the optimal solution, where 3-MFCs is equal to 3-MMGs.

The minimizing depth algorithm is given in Fig. 5. The algorithm needs to find all feasible floor cones to determine k-MMGs. The major work of parallel partitioning is to find every MFC in G. In every floor cone graph, there are three steps, namely calculating the area, sorting the area, and the labeling nodes.

Application of the traversal technique applies a depth first search (DFS) for calculating the floor cone area. A floor cone is constructed with the node as a root if the floor cone is feasible. In the second step the sorting technique uses MergeSort. Moreover, we label the chosen root v and its predecessors, Pre (v) to get the S_c set, where S_c is a set of MFCs. Let TP (v) = $\bigcup_{u \in Pre(v)} C_u$. Based on the S_c set, our parallel algorithm can find k-MMGs such that the k-MMGs, S_g is obtained by using FFD method in the MFC-processing. When a new DAG $G = G - S_g$ is obtained, we return to the generate-MMG's step in the partitioning procedure to find the new depth of the feasible cone. New MMGs are generated to increase the number of depths so that the partitioning depth in graph G increases until all nodes is covered to the MFC.

Algorithm Determining Depth:

```
while (L ≠ ∅) do
   while (SortList ≠ ∅) do
     Sort area of SortList by MergeSort
          for every floor cones Cv in SortList do
        if Cv is the MFC then
           Sc = Sc ∪ Cv;
           L = L – {v ∪ Pre(v)};
           SortList = SortList – {Cv ∪ TP(v)};
        end of the if loop
     end of the while loop
     for every MFC in Sc
        Finding Sg by FFD;
     end of the for loop
     G = G – Sg;
     Sg = ∅;
     Sc = ∅;
     depth = depth + 1;
end of the while loop
```

$c_1 = 3$	$c_2 = 6$	$c_3 = 2$	$c_4 = 1$	$c_5 = 5$	$c_6 = 7$	$c_7 = 2$

(a)

$c_1 = 7$	$c_2 = 6$	$c_3 = 5$	$c_4 = 3$	$c_5 = 2$	$c_6 = 2$	$c_7 = 1$

Sorting objects

(b)

		$c_7 = 1$
$c_5 = 2$	$c_4 = 3$	$c_6 = 2$
$c_1 = 7$	$c_2 = 6$	$c_3 = 5$

(c)

Fig. 5. (a) A set of MFCs; (b) Sorting these MFCs; (c) Reducing the number of MFCs according to the bin packing FDD.

A. The Complexity of the Algorithm

The sorting method has a low time complexity. Before finding the MFC, all the nodes are processed by sorting. Hence during portioning, the nodes are visited in topological order.

The time complexity of visiting the nodes is $\mathbf{O}(|V|\bullet log|V|)$ where V is the number of nodes in the given graph. The task of finding each MFC includes calculating the area, sorting the area and labeling nodes. Each floor cone area is calculated by the depth first search (DFS) methodology. Hence, the time complexity of calculating the area is $\mathbf{O}(|V|\bullet log|V|)$.

In sorting the area, the number of floor cones is no more than the number of nodes V. The time complexity is $\mathbf{O}(|V|)$. When labeling nodes, a node is visited once. The complexity of labeling nodes is $\mathbf{O}(|V|)$. In the MFC-processing, to apply FFD the number of MFC is no more than the number of nodes V. The time complexity is $\mathbf{O}(|V|\bullet log|V|)$. The minimum-depth partitioning solution selects k-MMG's in each greedy method iteration. Therefore, depth determining takes $\mathbf{O}(|V|^2\bullet log|V|)$ time. In conclusion, the total time complexity is bounded by $\mathbf{O}(|V|^2\bullet log|V|)$.

V. EXPERIMENTAL RESULTS

All the policies suggested so far [14] have either under prediction or over prediction by which they pay performance or power penalty. Our policy makes sure that server is ON, when there is an event in the Service Requester and Service Queue. Which means that under prediction or over prediction will never occur. Performance penalty will never occur by the proposed scheme. Refer figure 6 for comparison of speed performance of ME implementation in different architectures are presented in Tables I. Columns 2 to 4 respectively show the T_{exe} for the same search range as implemented by general purpose processor (GPP), parallel processing elements without the local memory.

Table I. Comparison of different designs for execution time

PRC(Parallel Reconfigurable Computing) SR (Search Range)
GPP (General PurposeProcessor)

Time(ns) SR	GPP	Parallel (1536clbs)	PRC (1536clbs)	Improvement	
				GPP	Parallel
8×8	10000	2200	480	12.10	2.91
16×16	16000	2240	430	14.12	2.11
32×32	24000	10000	1060	14.06	12.12
64×64	33000	14600	1200	17.12	18.76
Average				18.02	11.03

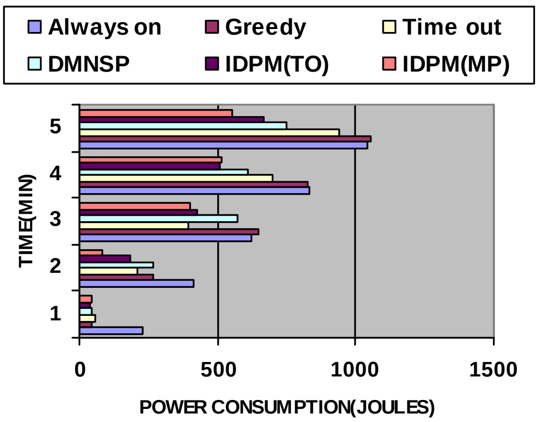


Fig. 6 Power consumption of IBM Hard Disk Drive under different DPM polices

ACKNOWLEDGMENT

The authors wish to thank National Science Council of Taiwan for proving grant for our research, under the NSC 98-2218-E-006-003 project.

REFERENCES

[1] Joint Video Team of ITU-T and ISODEC JTC I, "Draft ITU-T recommendation and final draft international standard of joint video specification." (ITU-T Kec. H.264 ISO/IEC 14496.10 AVC) JVT of ISO/IEC MPEG and ITU-T VCEG , JVT – GO05 march 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003,

[3] P. L. Tsai, S. Y. Huang, C. T. Liu, and J. S. Wang, "Computationaware scheme for software-based block motion estimation," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no. 9, pp. 901–913, Sept. 2003.

[4] Yu-Wen Huang and et.al, "One-Pass Computation-Aware Motion Estimation with Adaptive Search Strategy. ISCAS 2005 pp. 5469-5472

[5] Y. Lu, E. Chung, T. Simuni´c, L. Benini and G. De Micheli, "Quantitative Comparison of PM Algorithms", Design, Automation and Test in Europe, pp. 20–26, 2000.

[6] L. Benini, G. Paleologo, A. Bogliolo and G. De Micheli, "Policy Optimization for Dynamic Power Management", in IEEE Transactions on Computer-Aided Design, vol. 18, no. 6, pp. 813–833, June 1999.

[7] E. Chung, L. Benini and G. De Micheli, "Dynamic Power Management for nonstationary service requests", Design, Automation and Test in Europe, pp. 77–81, 1999.

[8] W. Stallings, "Computer organization and architecture: designing for performance," Pearson Education, 2003.

[9] L. F. Chen, Y. K. Lai, "VLSI architecture of the reconfigurable computing engine for digital signal processing applications," IEEE Circuits and Systems Conference., ISCAS '04. pp. 937-40, May 2004.

[10] Vissers, K.A, "Parallel processing architectures for reconfigurable systems," Design, Automation and Test in Europe Conference and Exhibition, 2003 pp. 396 - 397

[11] H. Schmit et al, "PipeRench: A virtualized programmable datapath in 0.18 micron technology," IEEE Custom Integrated Circuits Conference., pp. 63-66, May 2002.

[12] H. Singh, G. Lu, M. Lee, F. J. Kurdahi, N. Bagherzadeh, E. Filho, R. Maestre, "MorphoSys: Case Study of a Reconfigurable Computing System Targeting Multimedia Applications," Proceedings Design Automation Conference (DAC'00), pp. 573-578, Los Angeles, California, May 2000.

[13] R. Maestre, F. J. Kurdahi, M. Fernández, R. Hermida, N. Bagherzadeh, H. Singh, "Kernel Scheduling Techniques for Efficient Solution Space Exploration in Reconfigurable Computing," Special Issue on Modern Methods and Tools in Digital System Design, in the Journal on System Architecture, 47, pp. 277-292, 2001.

[14] Q. Qiu and M. Pedram, "Dynamic power management based on continuous-time Markov decision processes", Design Automation Conference, pp. 1999.

[15] Comparing System-Level Power Management Policies . Yung-Hsiang Lu, Giovanni De Micheli, Stanford University,

Anand Paul was born in Coimbatore, India in 1978. He is currently pursuing the Ph.D. degree in the electrical engineering at National Cheng Kung University, Taiwan, R.O.C. His research interests include Algorithm and Architecture for motion estimation in video, and Digital Video SoC design for H.264/AVC. His work also include Multi-Core reconfigurable systems and dynamic power management for advance video coding He is reviewer for IEEE transaction on Circuits and Systems for Video Technology, IET image processing and The Computer Journal published by British Computer Society and several international conferences. He has published 5 International Journal papers, 10 international conferences, Two workshop and one patent (pending). He has won many scholarships during 2004-2009 for out standing international student. In 2001 he won national technical level Quiz contest in India and Best Student paper award in National Computer Symposium 2009 in Taipei.

Yung Chuan Jiang received the B.S. degree in electrical engineering from the National Cheng-Kung University, Tainan, Taiwan, R.O.C., in 2000, and the M.S. degree in electrical engineering from the National Chung-Cheng University, Taiwan, R.O.C., in 2001. He is currently pursuing the Ph.D. degree in electrical engineering from the National Cheng-Kung University. His research interests include VLSI/CAD for circuit partitioning and reconfigurable computing.

.

Jhing-Fa Wang is now a Chair Professor in National Cheng Kung University, Tainan, Taiwan. He received his Master and Bachelor degrees in the Department of Electrical Engineering from National Cheng Kung University, Taiwan in 1979 and 1973, respectively and Ph.D. degree in the Department of Computer Science and Electrical Engineering from Stevens Institute of Technology, U.S.A. in 1983. He was elected as an IEEE Fellow in 1999 and now the Chairman of IEEE Tainan Section. He got outstanding awards from Institute of Information Industry in 1991 and National Science Council of Taiwan in 1990, 1995, and 1997, respectively. He has been invited to give keynote speech in PACLIC 12 (Pacific Asia Conference on Language, Information and Computation), Singapore and served as the general chairman of ISCOM 2001. (International Symposium on Communication), Taiwan. He has developed a Mandarin speech recognition system called Venus-Dictate known as a pioneering system in Taiwan. He was an associate editor for IEEE Transaction on Neural Networks and VLSI System. He is currently leading a research group of different disciplines for the development of "Advanced Ubiquitous Media for Created Cyberspace". He has published about 100 journal papers and 230 conference papers and obtained 6 patents since 1983. His research areas include wireless content-based media processing, image processing, speech recognition and natural language understanding.