

# Memory Reduction in Advanced Driver Assistance Systems for Multi Camera Based System

Amritha.S<sup>1</sup> Student PG<sup>1</sup>

Department of Computer Science & Technology,  
Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu,

**Abstract**—Advanced Driver Assistance Systems (ADASs) have become a vital research area in intelligent transportation systems. The major challenge of ADASs is to meet the real-time image processing requirements, especially when ADAS acquire multiple cameras. The image processing algorithms are highly depends on data, which shows that it is very difficult to reduce memory access latency for improving performance of image processing. However, there is no efficient memory reduction techniques have been implemented for ADASs. In this paper the concept of Image Processing Acceleration Unit (IPAU) is introduced, for optimizing the memory bandwidth of real-time video processing systems. The first feature of IPAU is block mode data layout in DRAM devices. The second feature is multiscale block mode hardware support through reorder buffer. The last feature is hardware rotation accelerator. Based on evaluation results, memory with IPAU can improve memory bandwidth by 49% compared to a memory hierarchy without IPAU, and reduce the energy consumption by 30% on the average.

**Keywords**—ADAS; Image Processing; Memory; Optimization

## I. INTRODUCTION

With the city growing and more cars running, traffic congestion become a major challenge for building Wisdom City. Road Traffic Accidents Increase Dramatically Worldwide. In order to improve traffic safety the scientific community start to the research of driver assistance systems.

Initially, simple mechanisms like analog camera for rear view are integrated into car, and then more complex devices, such as multi-camera surround view park assist systems, lane detection and pedestrian detection systems, were developed [1]. In the last decade, driver assistance systems have moved forward to more intelligent systems that referred as Advanced Driver Assistance Systems (ADASs). ADASs assist the driver in marking decisions, send out signals in possibly dangerous situations, and execute counteractive measures. With ADASs, road transportation systems can be more efficient, more human-friendly and safer. For examples, the adaptive cruise control system maintains a safe gap between vehicles and the lane departure warning system acts when the car is driven out of a lane inadvertently [2].

The ADASs technology fields include microelectronics, artificial intelligence, robotics, multi-sensor fusion, communication, and control. In particular, real-time image processing is a key technology in ADASs. Recently the Multi-Processor System-on-Chip (MPSoC) been introduced

that can meet the computation requirements that allow real-time video image processing to contribute substantially to safety technology.

Generally ADASs deploy four to five or more cameras for accurate images around the car. Although extremely complex and highly computation demanding, it can deliver valuable information from images [3]. Computer vision is a powerful means for sensing the environment and has been widely employed to deal with most of tasks in the automotive applications.

Computer vision based systems like lane tracking, face recognition and obstacle detection mature an enhanced range of driver assistance systems are becoming practical. The image processing of multi-camera need more resources in memory capacity and access latency.

In this paper, we focus on memory bottleneck of image processing in multi-camera based ADASs, such as Motion Estimation, Object Detection and Classification. Based on memory access characteristics of image processing algorithms and architecture of DRAM devices, we present an Image Processing Acceleration Unit (IPAU) for the optimization of the memory hierarchy of real-time video processing systems.

## II. MULTI-CAMERA BASED DRIVER ASSISTANCE SYSTEMS

More and more sensors and control systems have been integrated intelligent vehicles. So the vehicles could understand the surround environment that the driver can be warned of potential hazards. Vision is the most important sense employed for driving assistance systems, therefore camera sensors are the most used sensor in these systems.

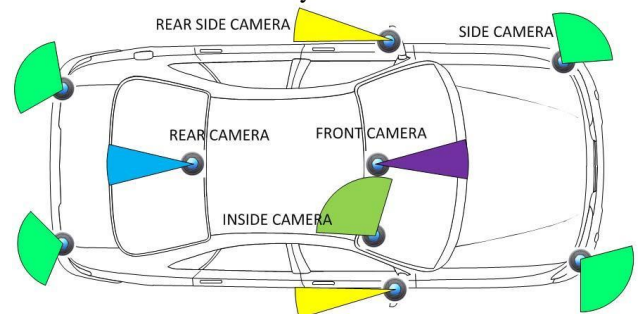


Figure 1. Multi-camera based Driver Assistance Systems

Vision subsystem for intelligent vehicles need to recognize visual state such as traffic sign, road lane and

obstacle shape. ADASs deploy four to nine cameras for accurate images around the car as shown in Figure 1. Sometimes compression can be used to reduce the required video communication bandwidth.

Vision subsystem in ADASs can analyze the video content to get valuable cues for Traffic Sign Recognition (TSR), Lane Departure Warning (LDW), and Auto Parking Assistance System (APAS) [4].

In APAS which is one kind of Multi-camera based Driver Assistance Systems, side cameras capture and display on-screen the environment surrounding the car from a virtual top view for parking assistance. The perspective is dynamically moved depending upon the trajectory of the car, providing a 360° view around it.

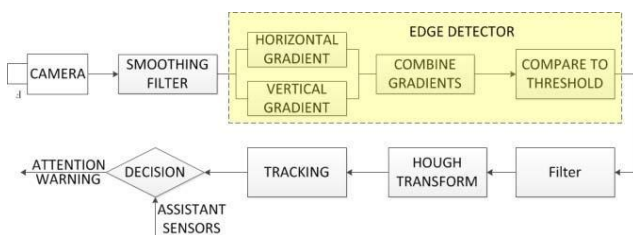


Figure 2. The structure of lane-departure algorithm

Front view camera designed as lane departure warning subsystem to warn a driver when the vehicle begins to move out of its lane on freeways and arterial roads. The lane-departure algorithm is shown as the Figure 2. Rear view camera help drivers identify an obstacle or pedestrian in the back of a car and back up safely and maneuver conveniently into parking spaces. In addition the distance of objects can be measured and braking intervention can be triggered.

### III. COMPUTER VISION TECHNIQUES IN ADASS

The following subjects in multi-camera based ADASs depend on computer vision techniques: Traffic Sign Recognition (TSR), Blind-spot detection (BSD), Pedestrian Protection Systems (PPS), Lane Departure Warning System (LDW), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW), Event Video Recording (M-DVR), Auto Parking Assistance System (APAS), and Driver Fatigue Warning System (DFWS) [5][6].

The Computer vision techniques or real-time image processing is supported by embedded computing such as MPSoC, Digital Signal Processor (DSP), field programmable gate array (FPGA). The real-time image processing is computation demanding. Many-core technology could be proposed as a solution to improve the performance of modern intelligent vehicles systems.

Vision-based driver assistance is on the edge of current developments in computer vision techniques. It aims at real-time understanding of complex environment as they may possibly occur in a traffic context [7].

For any object in an captured image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from the image, can then be used to identify the object when attempting to locate

the object in successive images containing many other objects [8].

Recognizing an image by scale invariant feature transform algorithm is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes. To perform reliable recognition, it is important that the features extracted from the training contrast regions of the image, such as object edges. The most of algorithms processing image in block with different scale.

During image processing, the camera subsystem captures the images and buffers those images into DRAM at a fixed period. Conventionally, the images stored in line, but accessed in block afterwards.

DRAM, as the most cost effective semiconductor storage device, is used in a wide variety of devices. Although DRAM chips could be in variety of packages and different sizes, but their essentially operation is the same. DRAM chips are composite of a few same large rectangular arrays of memory cells, and refresh circuitry to maintain the stored data. Memory arrays are arranged in rows and columns respectively. Each memory cell has a unique address defined by bank, row and column. A memory bank is a logical unit of storage in electronics, which is hardware dependent. Every bank has a row buffer. Because the banks share the data bus, command bus and address bus, so only one bank could be accessed in a single read or write operation.

Efficient memory usage is an important consideration for system designers because external memory accesses usually have high latencies. While memory controller for ADASs need to support larger off-chip memories, it is important to be judicious in transferring only the useful data needed for the application. The data should be reused to an extreme [9].

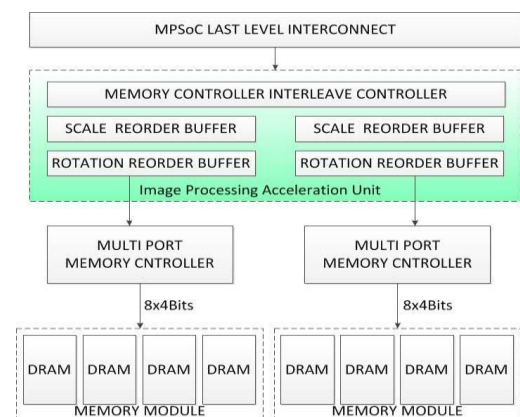


Figure 3. The structure design of DRAM

Because real-time image processing is very computation demanding, frame buffers must be set up in external memory, as shown in Figure 3. In the scenario, while the image processing algorithms operates on one buffer, the other buffers is being filled by the camera sensors. A simple mutex semaphore can be used to maintain synchronization between the frames [10].

In this paper, we present there DRAM bandwidth optimizing techniques based on memory access

characteristics of image processing algorithms and architecture of DRAM devices.

#### IV. IPAU MEMORY OPTIMIZATION IN ADASS

In the design of multi-camera based ADASSs, memory efficiency play a very important role, and often impact significantly the system's performance, power dissipation, as well as overall cost.

Memory access bandwidth optimization can be divided into three levels, including application layer, the memory controller layer and device layer. In application layer, the processor or ASIC issue the memory read or write operations. The optimization could be independent of hardware. In memory controller layer, memory controller can control the DRAM devices by issuing commands such as REFRESH, ACTIVE and PRECHARGE to improve access bandwidth, which is more accurate than application layer. And the improvement of DRAM architecture referenced as the device layer optimization.

We have developed an Image Processing Acceleration Unit(IPAU) for the optimization of the memory bandwidth of real-time video processing in ADASSs. The IPAU module is located between the processor core and memory controller. So it is one kind of application layer optimization. It includes three methods, the memory block method, multiscale block method and rotation hardware accelerator. The last two method of IPAU module are implemented in hardware for accelerating [11].

##### A. The Memory Block Method

Most of image processing algorithms work on the principle of spatial locality. They access the image data in two-dimensional format, for example, 16x16 pixels of block. If the image is stored in external memory in linear format, as shown in Figure 4. To access the block, multiple memory accesses need to be performed. In the DRAM device, each memory access can visit up to 1024 bytes. However, because the image block is not stored contiguously in the memory, only partial pixels are useful in each memory access. Multiple accesses are generated to fetch the rest bytes of the data.

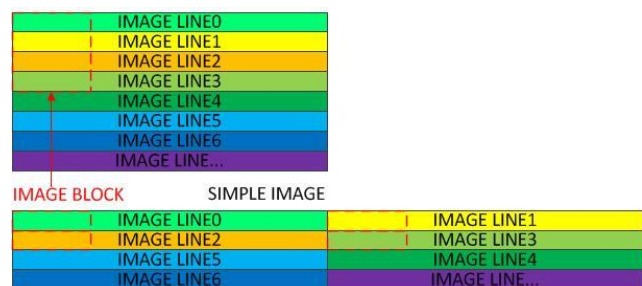


Figure 4. Image stored in the memory linearly

Instead, if the video data is stored in bi-dimensional block format as Figure 5 shown, much fewer accesses are needed to read the entire block of data. This is the strategic motivation for storing the video data in the block format [12].

In the article we propose an optimization scheme according to the DRAM access behavior characteristics of image processing and the parallel characteristics of DRAM device architecture to fit the available memory bandwidth constraint. Block object transfer optimization: block and sub-block. Its primary function is increase efficiency of a 2D block memory access, primary handling efficiently 2 dimensional video data mapped in blocks.

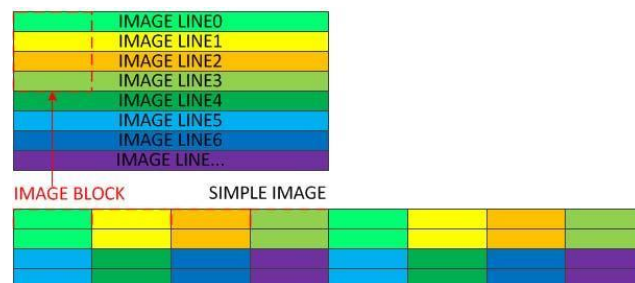


Figure 5. Image stored in the memory in micro block

The block is designed to offer bi-dimensional data locality in a single SDRAM page. It has been sized to 1024 bytes (the size of the smallest SDRAM page). By representing a 2D block, we could ensure that any access that fits within a block is atomic in the SDRAM controller and fits in a single SDRAM memory page, and minimize the number of SDRAM page openings per 2D block transfer.

In order to make full use of row buffer capacity, the buffer should be occupied by one block. For instance, almost all the row size of DDR3 chip is 1024 bytes, so we can partitioning image in 32x32 blocks. each block occupy a row exactly. In image processing, we read four consecutive rows for a 64x64 block in zigzag mode. And we can read in a group of row from DRAM as a bigger imageblock.

To reduce the access delay, we propose that image data block store in adjacent DRAM device via address interleave as Figure 6, then it could be access continuously according to the bank's concurrency features. Continuous datum were distributed in different banks to improve memory access performance by hiding row buffer activation delay.

| SYSTEM ADDRESS | DRAM ADDRESS     | SYSTEM ADDRESS | DRAM ADDRESS       |
|----------------|------------------|----------------|--------------------|
| 0x0000-0x03FF  | MPMC0 BANK0,ROW0 | 0x0000-0x0FFF  | MPMC0 BANK0-3,ROW0 |
| 0x0400-0x07FF  | MPMC1 BANK0,ROW0 | 0x1000-0x1FFF  | MPMC1 BANK0-3,ROW0 |
| 0x0800-0x0BFF  | MPMC0 BANK1,ROW0 | 0x2000-0x2FFF  | MPMC0 BANK4-7,ROW0 |
| 0x0C00-0x0FFF  | MPMC1 BANK1,ROW0 | 0x3000-0x3FFF  | MPMC1 BANK4-7,ROW0 |
| 0x1000-0x13FF  | MPMC0 BANK2,ROW0 | 0x4000-0x4FFF  | MPMC0 BANK0-3,ROW1 |
| 0x1400-0x17FF  | MPMC1 BANK2,ROW0 | 0x5000-0x5FFF  | MPMC1 BANK0-3,ROW1 |
| 0x1800-0x1BFF  | MPMC0 BANK3,ROW0 | 0x6000-0x6FFF  | MPMC1 BANK4-7,ROW1 |
| 0x1C00-0x1FFF  | MPMC1 BANK3,ROW0 | 0x7000-0x7FFF  | MPMC0 BANK4-7,ROW1 |

SINGLE ROW SIZE(1KB) INTERLEAVE

FOUR ROW SIZE(4KB) INTERLEAVE

Figure 6. Multiple Memory Controller Address Interleaving

Furthermore, the IPAU memory interleaving size of accesses is set to 1 KB - a block size - so that any request that fits within a block fits in a single SDRAM memory page. Any request that spans over 2 or 4 blocks, is distributed on all of memory controllers.



Taken Micron's 8 bits DDR3 DRAM chip into consideration, it has 8 banks, 32768 rows in every bank, 1024 column in every row. When video data stored in line, the active row have 1024 bytes in which only 64 bytes are useful in a block reading. The data utilization is only 6.25%. But if every line of block stored in DRAM successively as shown in Figure 5, the row data utilization efficiency is up to nearly 100%.

Furthermore, banks in DRAM are hardware independent although they shared the same interface. Each bank could be activated concurrently. DRAM devices can have up to 8 or more banks which could be read or write successively without active delay. But in order to limit the total power consumption of DRAM device, no more than four rows in different banks can be activated concurrently according to the DDR3 specification.

### B. Multiscale block method

In most of image processing algorithms, the basic video operational components are blocks in different scale, such as 4x4, 8x8, 16x16, 32x32 or 64x64 pixels of block.

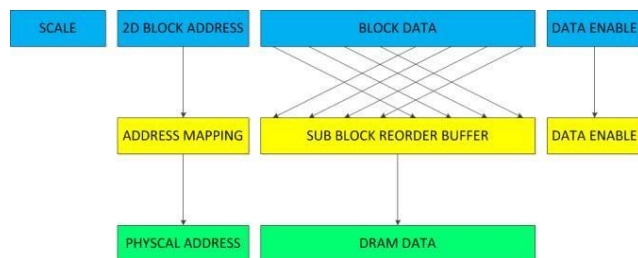


Figure 7. Hardware Multiscale block design

When the image processing algorithm processes the block in a smaller scale than 32x32, the new image block is not stored contiguously in the memory any more, which will penalize the memory bandwidth.

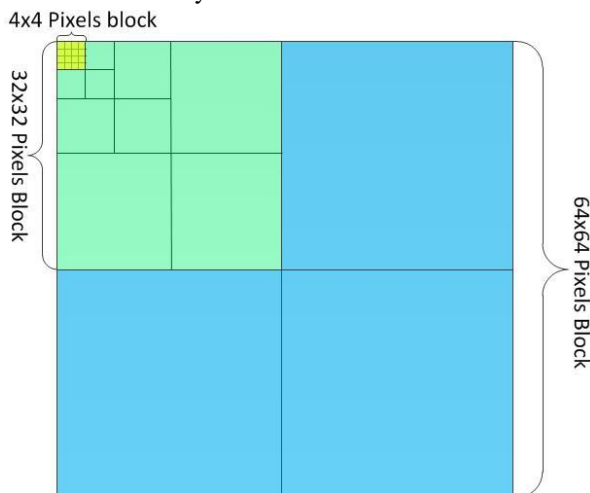


Figure 8. Multiscale tiller organization

To resolve this problem, the second optimization method, multi-scale block method, is proposed. The method introduces a block reorder buffer as hardware accelerator as

Figure 7 shown. The blocks are stored in pyramid organizational structure. The bigger block is comprised of four smaller blocks in quarter size as Figure 8 shown [13].

Under the multi-scale block method, when the image processing algorithm accesses the block in different size, all the smaller blocks which comprised of the accessed block are transferred into reordering buffer as Figure 8 shown. It will reorder the image block according to the value of parameter SCALE in block reorder buffer control register. By the multi-scale block method, we could access blocks in any scale contiguously without any DRAM access interrupt interference.

Furthermore, the block reorder buffer is set to 1 KB, a 32x32 pixels of block size - so that any request that fits within a block, fits in a single DRAM memory page. Any request that spans over 2 or 4 blocks, is distributed on a maximum number of memory controllers.

### C. Hardware rotation accelerator

Rotation operation is another hot action when image processing. The rotation reorder buffer is a submodule within the IPAU aimed at efficient handling of two dimensional data by the use of block layout. IPAU optionally manages the memory fragmentation and zero-copy physical frame buffers swapping through a page-grained translation. The memory optimization submodule making on-the-fly, zero overhead transforms, such as 90°, 180° or 270° rotations, with either a vertical or horizontal reflection.

All these transforms correspond to the composition of a 0°, 90°, 180°, or 270° rotation with an optional reflection. Mathematically the nature of this orientation is based on the three following binary parameters: the first parameters is the direction of the x axis of the image block, the second is the direction of the y axis of the image block, and the last parameters to swap the x and y axis.

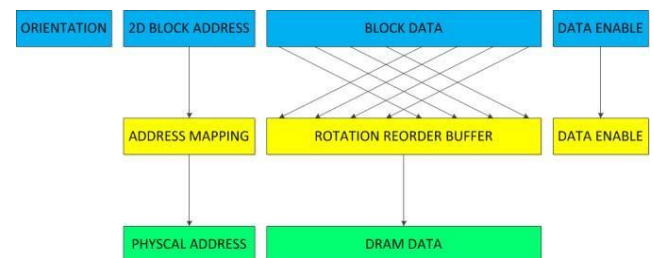


Figure 9. Hardware rotation accelerator

The architecture of hardware rotation accelerator is similar to architecture of multi-scale block method. The core design of hardware rotation accelerator is the Rotation Re-Ordering Buffer (RROB) as Figure 9 shown. The RROB is responsible for managing the re-ordering buffer and performing the data reordering due to the orientation setting after requested the data from DRAM devices. Because of the image block in all of eight orientations directly, the image processing algorithms running on the MPSoCs are more efficient by making fewer accesses to the SDRAM, to fetch blocks of data [14].

## V. MEMORY OPTIMIZATION EVALUATION

In the section we evaluate the optimizations of DRAM bandwidth in detail. We construct the evaluation system based on a x86\_64 PC with intel i7-2600 processor, and 64-bit version of ubuntu 11.10 operating system. The kernel version of ubuntu is 3.0.0-32, and GNU compiler is x86\_64-linux-gnu. The evaluation system that shown as Figure 10 is composite of the Image processing algorithms model and a memory system simulator. We revise DRAMSim2 as to simulate the DRAM memory system. DRAMSim2 is a cycle-accurate simulator of DRAM controller and memory modules [10]. It can simulate memory controller and memory modules in different DDR2 and DDR3 chips from MICRON company. It can be driven by trace file or by simulator such as GEM5, SIMICS and QEMU [15].

The ADAS model only simulates the memory accesses of image processing such as reading sub blocks, buffering video like a real application. The model drives the DRAM memory simulator to evaluate the performance of the system with bandwidth optimization methods proposed in this paper.

### 2) DRAM Row hit rate

In order to evaluate the performance of data layout in block, we add a row buffer hit log modules for DRAMSim2. It logs the row buffer active history with every active row hit counter. As Figure 12 shows, most of row hit rate is not more than 8 times in linearly mode, but when stored in block according to block size, the most of hit rate more than 100 times.

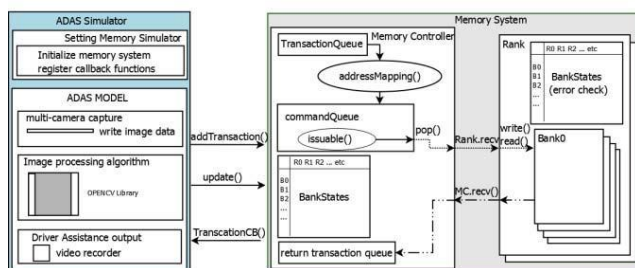


Figure 10. Memory optimizations evaluation platform

We use a DDR3 DRAM timing model from Micron company, DDR3\_micron\_32M\_8B\_x8\_sg15.ini, with 8 banks, 32768 rows, 1024 columns and 8-bit data width. The system parameter row\_BUFFER\_POLICY is set as open page policy, both TRANS\_QUEUE\_DEPTH and CMD\_QUEUE\_DEPTH are set as 4 to 32, and the EPOCH\_LENGTH is set as 10,000 cycles. Both latency of multi-scale block re-order and rotation re-order are set as 4 cycles.

### 1) Memory bandwidth performance

In order to evaluate the performance of IPAU, we compare memory bandwidth of the ADAS model with and without IPAU module. The ADAS model without IPAU module is called linearly design. The ADAS model with block feature of IPAU module reference as IPAU-1, with multi-scale block feature of IPAU module reference as IPAU-2, and with hardware rotation feature are called IPAU-3 design respectively.

We use DRAMSim2 to examine DRAM memory system throughput with fully saturated queues. DRAMSim2 outputs data file consist of the bandwidth, latency, and power for each simulation epoch. Figure 11 shows that the IPAU-1 can improve bandwidth throughput by as much as 45% over that of a general memory without IPAU at 5.33 Gb/s (DDR3). And with hardware rotation support of IPAU-3, the

bandwidth improvement greatly as much as 49% over that of a general memory.

In Figure 11, the real bandwidth increase to 9.23GB/s when changing data layout from line mode to block mode, and further when DRAM command queue depth is set to 32, the bandwidth is up to 11.3GB/s.

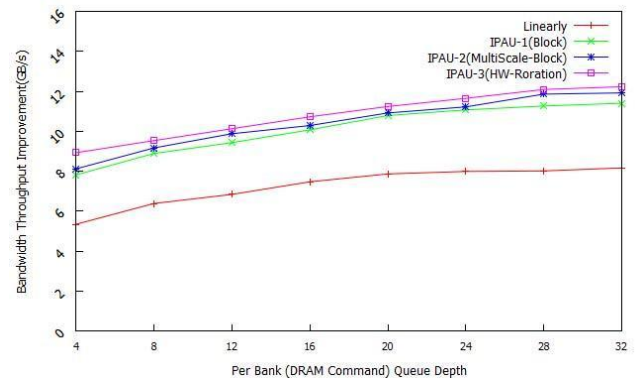


Figure 11. DRAM Bandwidth throughput

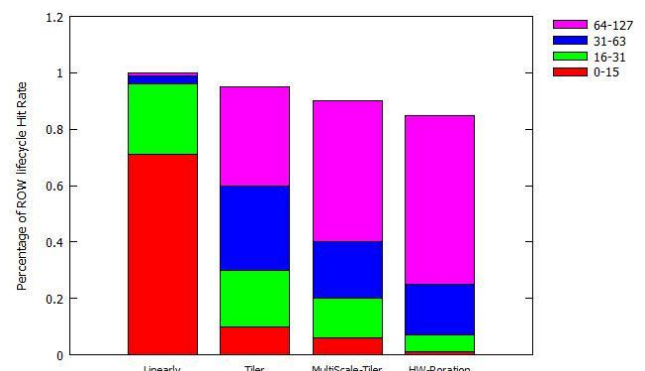


Figure 12. DRAM Row hit rate

The origin data bandwidth log of DRAMSim2 is all of the data burst through memory modules. But in practical application only a few of burst bytes are useful. So we add two data transform counter to record the total read bytes and total write bytes. By that, we could get the efficient bandwidth of the memory system instead of data throughput.

### 3) power consumption

At last, because we improve row buffer hit rate, reduce the row active and precharge operations, and increase the coefficient of data utilization, the memory access number is decrease. Therefore the power consume is reduced either. The power is composed of Background power, ACTIVE and PRECHARGE of row power, burst power, refresh power as

Figure 13 shows. The main parts of power consumption are background power and ACTIVE and PRECHARGE power. The background power is nearly invariant for all applications. But we can save ACTIVE and PRECHARGE power by improving row buffer hit rate. When the most of row buffer hit increase from 5-8 times to 101-128, the ACTIVE and PRECHARGE power consumption decreased from 450mW to 200 mW.

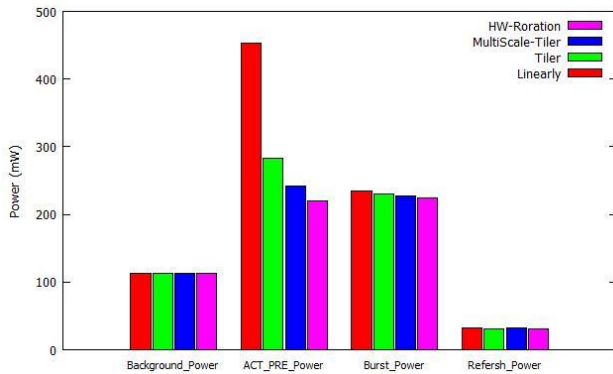


Figure 13. The power consume of DDR3

## VI. CONCLUSIONS

Image processing is the most critical part in video based driver assistance systems. It is largest consumer of computation time and memory bandwidth. And it has greatest impact on performance in response time. In this paper, we propose an external memory storage optimization for image processing in ADASs. We introduce block data storage scheme and have improved the row buffer utilization up to nearly 100% compared to conventional linear data storage scheme. Besides, by cooperating with multi-scale block method and hardware rotation accelerator, the row data utilization rate raised from 6.25% to more than 90%. The proposed architecture reduces about 79% of row-activations and increases the memory bandwidth by 49%. In addition, the proposed architecture reduces the energy consumption by 30% on the average. According to experimental results, the proposed algorithm can effectively improve the performance by 47.7% in average.

## REFERENCES

- [1] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, Vision-based intelligent vehicles: State of the art and perspectives, *Robotics and Autonomous Systems* 32 1 - 16, 2000.
- [2] M. Jones and D. Snow, Pedestrian Detection Using Boosted Features over Many Frames, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- [3] S. Marsi, G. Impoco, A. Ukovich, S. Carrato, and G. Ramponi, Video Enhancement and Dynamic Range Control of HDR Sequences for Automotive Applications, *EURASIP J. Advances in Signal Processing*, vol. 2007, p. 9, 2007.
- [4] A. Shashua, Y. Gdalyahu, and G. Hayun, Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance, *Proc. IEEE Intelligent Vehicles Symp.*, pp. 1-6, 2004.
- [5] Kaempchen, N.; Dietmayer, K.C.J.: Data synchronization strategies for multi-sensor fusion, in *Proceedings of the 10th World Congress on Intelligent Transport Systems and Services*, Madrid, Spain, 2003, No. T2250
- [6] Stiller, C.; Hipp, J.; Rössig, C.; Ewald, A.: Multisensor obstacle detection and tracking, *Image and Vision Computing*, vol. 18, pp. 389396, 2000.
- [7] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea and M. E. Lopez, Real-Time System for Monitoring Driver Vigilance, *IEEE Transaction on Intelligent Transportation Systems*, Vol. 7, Issue 1, pp. 63-77, 2006.
- [8] Nicholas Apostoloff and Alex Zelinsky. Robust vision based lane tracking using multiple cues and particle filtering. In *Proc. IEEE Symposium on Intelligent Vehicles*, 2003.
- [9] Xiangyu Dong, Yuan Xie, Naveen Muralimanohar, Norman P. Jouppi. Simple but Effective Heterogeneous Main Memory with On-Chip Memory Controller Support. *SC '10 Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010
- [10] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems: Cache, DRAM, Disk*. New York: Kaufmann, 2008.
- [11] B. Akesson, K. Goossens, and M. Ringhofer, Predator: A predictable SDRAM memory controller, in *CODES ISSS*, New York, NY, USA, 2007.
- [12] LEE, K.-B., LIN, T.-C., AND JEN, C.-W. An efficient quality-aware memory controller for multimedia platform SoC. *Circuits and Systems for Video Technology*, *IEEE Trans. on* 15, 5 (2005), 620 633.
- [13] BORKAR, S. 3d integration for energy efficient system design. In *Proc. 48th ACM/EDAC/IEEE Design Automation Conf. (DAC)* (2011), pp. 214219
- [14] AHN, J. H., EREZ, M., AND DALLY, W. J. The Design Space of Data-Parallel Memory Systems. In *Proc. ACM/IEEE SC 2006 Conf* (2006).
- [15] D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, and B. Jacob, *DRAMsim: a memory system simulator*, *SIGARCH Comput. Archit. News*, vol. 33, no. 4, pp. 100107, 2005.