Virtual HDR/LDR Image Synthesizer in Multi-Core Platform

Tsun Hsien Wang, Jen Wen Wang , Ching Te Chiu

Department of Electrical Engineering, National Tsing Hua University
Department of Computer Science, National Tsing Hua University
(email : ctchiu @ cs.nthu.edu.tw)

Abstract

In this paper, we analyze the performance of implementing inverse tone mapping operation in various multi-core environments. We parallelize the procedure of producing five HDR images and weighting value, the ratio of the two functions decreases and the total simulation time decreases around 50% under six-core environment. The reduction is because these functions are parallel implemented with multi-thread on multi-core platform. The hardware performance analysis shows that our proposed method is easy to be realized in a multi-core system to speed up the performance. In our future work, we will study the effects and performance of applying our method to video.

1. Introduction

The concept of multi-core system has been becoming increasingly prevalent, since the increasing number of personal computer users and more applications are designed for them. To cope with huge computation complexity and reach real time processing, the need for faster computation speed on one single computer is urgent. Though the fast developing of processor technology, chip speed and performance with one CPU has been improved exponentially over the years, the current computer industry trend is shifts toward creating processor chips that contain multiple computing cores, such as Intel's four cores processor Core™ i7, Cavium Networks's Octeon 16 cores processor, and even the terminal user of Microsoft Xbox 360 has three cores processor. Each core of their processor is high-performance computation unit.

Using multiple cores on a single chip takes advantage of powerful processing ability. The multi-core system can effectively divide the total amount of computation to several pieces to each core and parallelize them to reduce elapse time and raise performance.

However, speedup by increasing the number of cores on a single chip challenges the communication and coherence across the cores, such as whether all cores be homogenous or highly specialized more efficient, and the distribution of the work load for each core. Proper parallelization is the key to speedup.

Since the multi-core system is powerful especially on graphics and 3D computations with high-resolution image for the easy distributed property, we proposed novel multi-core schemes on HDR-based image synthesizer and performed experiments separately on one core, six cores, and sixteen cores system with different schemes by simulation to access the simulation statics and performance comparison among these results. We adopt the Open Virtual Platform (OVP™) as the simulation environment, which includes model architecture for a single core and multiple cores, free open software models and public APIs, and a fast simulation and performance evaluation platform.

The rest of the paper is organized as follows. Section II describes the multi-core platform used in the simulation. Section III describes the HDR/LDR synthesizer and the realization in the multi-core platform. Section IV shows the simulation results. Section V gives the conclusion.

2. Model architecture

As in Figure 1, the single core model is composed of one OR1K processor with an internal memory, two memories individual storing programming space and stack, and one peripheral (UART) for interrupt management which are connected to DecodeBusLT (decodes the address and choose the corresponding target) by TLM (transaction level modeling). Also the model adopt direct memory interface to access memory. The computation frequency of OR1K processor is 100 MIPS (million instructions per second).

The multi-core model, see the right part of Figure 2, is a 4x4 mesh architecture consist of sixteen OR1K processors, as same as the one used in above single core system, connected by 32-bits wide TLM buses. As the left part of Figure 2, each core has one router to transmit data between cores, which includes receiver core number, the starting address and the total size of the data to transmit. The router sequentially redirects the data to the target cores. The blue frame indicates the central core (master core) which controls other slave cores (the

978-1-4799-7591-4/15/$31.00

yellow ones). Outside of the mesh there are two memories for program and stack with one peripheral shared between sixteen cores, adopt direct memory interface as well.

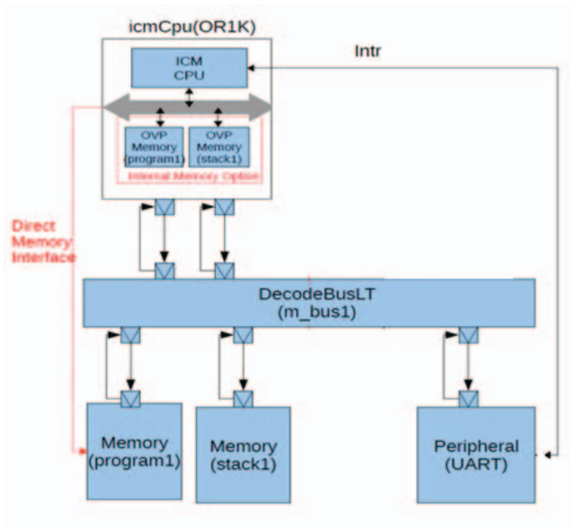


Figure 1. A single core model architecture

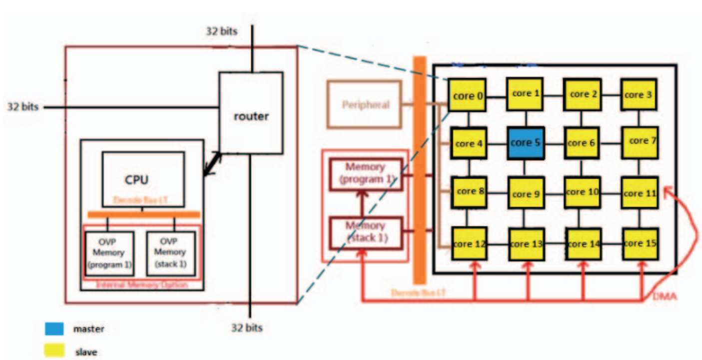


Figure 2. The architecture of a multi-core model

With the availability of the high quality display devices, the acquisitions of HDR images or videos become more and more urgent. Conventional image acquisition produces LDR images because of the limitations of sensors or bit resolution. As a result, images captured by a conventional imaging system may end up being too dark in some areas or possibly saturated in others.

T.H. Wang, etc, [9] propose an emulated multiple exposure inverse tone mapping that utilizes multiple non-linear curves to convert one optimized LDR image to multiple images with different brightness. The primary concept of using multiple non-linear curves is that the luminance of each pixel is expanded. The discrepancy between region-based algorithms and our proposed method is that the former deals with surrounded multiple pixels and the latter deals with multiple luminance for each pixel. In our proposed method, we utilize weighting average to combine multiple luminance to avoid artifacts and unnatural appearance from non-linear curve inversion. The objective of this paper is to evaluate the implementation of the HDR synthesizer framework for legacy LDR images in multi-core environment.

3. Design of HDR/ LDR synthesizer

In Wang's proposed HDR/LDR synthesizer, input LDR images generated from conventional CCD or CMOS sensors are assumed to be optimized for distinct light under variety of real scenes. An exposure-dependent S curve is used to convert one optimized LDR image to multiple images with different brightness, which are then fused into a HDR images with wide dynamic range. The S curve can avoid saturation in bright regions and boost the intensity of pixels in dark regions. While our algorithm has a few parameters, these depend only on brightness of an input image and the required dynamic range of an output image. If the input image is optimized in its brightness, the parameters can be fixed for a variety of images. According to our implementation results, the dynamic range can reach from 10^{-4}~10^{2}. The emulated multiple exposure HDR images have luminance values in the order of 10−4~102 that increases the computation complexity compared to LDR images in which the resolution for each pixel is eight bit. To reduce the implementation cost, we could generate emulated multiple exposure images and perform the weighted average operation in the LDR domain. The inverse mapping function is added after to generate the final reconstructed HDR image. This is the LDR-based emulated multiple exposure synthesizer. The main discrepancy between emulated multiple exposures HDR and LDR method is that the light parameter adjustment and weighted average for the former works on the HDR domain while the latter in the LDR domain.

The data flow of the HDR/ LDR synthesizer of the single core system is in Figure 3 [9]. The green part indicates that only when running LDR-based flow it would go through this step.

First we read an LDR image into an array, set several variables for further used, then start inverse gamma correction for adjusting image brightness for displays and perform RGB to YUV for transferring the 3-channel image to gray scale.

Next, we perform local region segmentation from the normal LDR image based on the probability partition method to separate the original LDR image into four different luminance regions. Afterward, in HDR-based, we apply inverse tone mapping to generate five different exposure HDR images from the original LDR image. Different from the above method, in LDR-based, we apply inverse tone mapping to the original LDR image for generating one HDR image, and go through tone mapping to generate four different exposure LDR images.

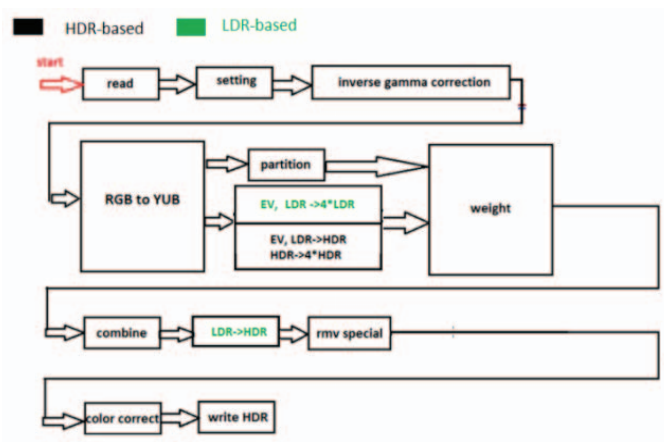


Figure 3. Data flow of a single core system

To fuse these five emulated HDR-based/ LDR-based images into the final reconstructed HDR/ LDR images, the next step is to weight-average the luminance from them and finds the weighting value for each image. After combination, the LDR-based method will apply to another inverse tone mapping for generating the final HDR image from the one LDR image after combined. Last, we remove all pixels which are NaN and infinite after estimation, resume the gray scale image back to 3-channel image, and write it to a .hdr file.

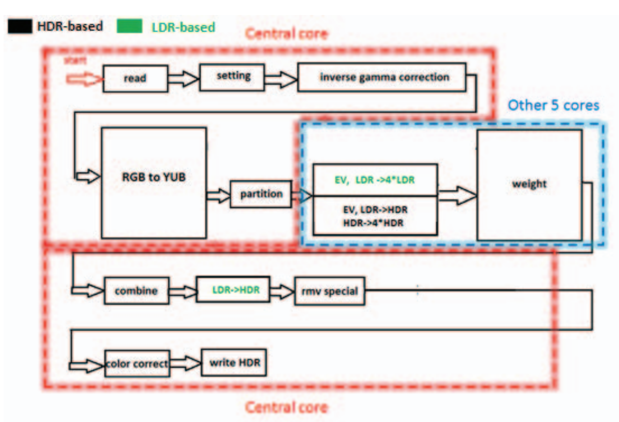


Figure 4. Data flow of a 6-core core system

With multi-core system, we proposed two designs on six cores and sixteen cores with parallelization to reduce the total simulation time compared to the procedure of the single core. First, see Figure 4, we made a little change of the previous data flow and apply it on six cores system. The key is that we parallelize the production of the five different exposure images and finding the corresponding weighting value. The sections framed in red are the part central core responses for, while the blue one operates by other five cores.

The central core processes images up to segmentation and sends the original LDR image and segmentation results to other five cores, separately produce five HDR images and weighting values at the same time. For further combination, the five cores send the emulated HDR images and weighting values back to the central core to combine all HDR images. The central core then post-processes the combined image.

Figure 5. Data flow of a16-core system

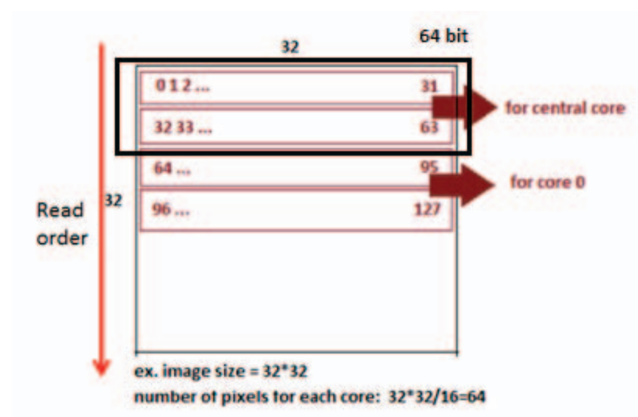
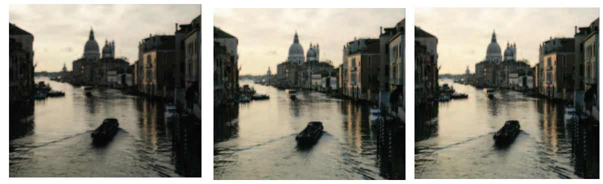


Figure 6. The reading order of an image

For the sixteen cores system, the HDR-based data flow is in Figure 5. First, the central core read in the original LDR image, sees Figure 6, and distributes the pixels equally to other fifteen cores and also the central core itself. All cores process RGB to YUV and reverse gamma correction. When it comes to normalizing the image after the above procedure, the local maximums within each core are needed for finding the global maximum of all pixels. That is to say, the central core will receive fifteen local maximums from other cores and send back the maximum among them to other cores to complete normalization individually. Next, all cores estimate luminance of the image after processing reverse gamma for further 3-channel mapping, and normalize it as well. All cores then find local histogram and combine them in central core. Continually, central core perform probability partition and come out the segment point, sending them to all other cores for generating segmentation images. All cores produce five HDR local images with different exposures and find the weighting value for each pixel. To come out the weighting value, central core first receive the local maximum, minimum of each core belongs to the 3rd HDR emulated image, pixel numbers and sum of each region from other cores, then send back the five coefficients (Lmed) after processing. Last, all cores combine the local five HDR images into one HDR image, perform color correction, and centralize the all sixteen local HDR images into the final HDR image in the central core.

4. Experimental results

In Figure 7, there are results of the HDR-based and the LDR-based method.



(a) (b) (c)

Figure 7. (a) Original image (b) HDR-based image (c) LDR-based image

According to the experimental results of the simulation statics of OVP™, as in Table I, it reveals that the" produce weighting value" function is the bottle neck both with LDR-based and HDR-based method in the single core system.

Table I. Function instruction load comparison between HDR-based and LDR-based method in the single core system.

Function name	1 core	
	HDR-based	LDR-based
Read image + setting	0.09%	0.10%
Reverse gamma correction	8.26%	4.43%
RGB to YUB	6.76%	7.22%
Image Segmentation	8.30%	8.86%
Produce 5*HDR /Produce 4*LDR	13.81%	13.25%
Produce Weighting value	40.44%	39.34%
Combination	16.22%	17.31%
LDR->HDR	-	2.95%
Color correction	6.13%	6.54%

In another comparison as Table II below, we can see that after parallelizing the procedure of producing five HDR images and weighting value, the ratio of the two functions decreases and the total instruction count of six cores system became almost a half of the single core system.

Furthermore, as Table III, with the input 256x256 image, the simulation time of sixteen cores system is almost 10x faster than the single core one and reach real time processing.

5. Conclusion

We proposed accelerating schemes in a multi-core system such as six cores and sixteen cores. In the six cores system, we parallelize the production of the five emulated HDR images and the corresponding weighting values in five different cores. The improvement ratio over the single core system is 0.5. In the sixteen cores system, we distribute the total pixel numbers of the original LDR image equally to all cores and perform proper parallelization. The time improvement ratio over the single core reaches 10. The multi-core schemes both retained good results while raising the processing speed.

Table II. Function instruction load comparison between the single core and six-core system with HDR-based method

Function Name	Single core		Six cores		Instruction ratio (single core/Six cores)
	Instruction count (million)	Instruction count ratio (%)	Instruction count (million)	instruction count ration (%)	
Read image + Setting	0.2	0.002	0.2	0.004	1
Reverse gamma correction	611	8.28	611	14.52	1
RGB to YUB	500	6.77	500	11.89	1
Image Segmentation	614	8.32	6.14	14.6	1
Send /Receive luminance and segmentation image	-	-	0.4	0.01	-
Produce 5 HDR	1021	13.83	204	4.85	5.005
Produce weighting value	2991	40.53	623	14.81	4.801
Send /Receive weighted value	-	-	1.2	0.03	-
Combination	1199	16.25	1199	28.51	1
Color correction	453	6.14	453	10.77	1
Total	7378.2	100	4205.8	100	1.754

Table III. Simulation time comparison of the single core, six core, and sixteen cores system with HDR-based method

Simulation statics	1 core	6 core	16 core
	HDR-based		
Simulated time	**7.36(s)**	**2.24(s)**	**0.67(s)**

References

[1] H. Seetzen, L. Whitehead, and G. Ward, "A high dynamic range display using low and high resolution modulators," In Society for Information Display Internatiational Symposium Digest, Tech. Rep., 2003.
[2] H. Seetzen, H. Li, L. Ye., L. W. G. Ward, and W. Heidric, "Guidelines for contrast, brightness, and amplitude resolution of displays," In Society for Information Display (SID) Digest, pp. 1229–1233, 2006.
[3] S. Mann, R.W. Picard, S. Mann, and R. P. Being, "Undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures," in Proc. IST's 48th Annual Conference, 5 1995, pp. 442–448.
[4] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in Proc. SIGGRAPH, 1997, pp. 369–378.
[5] S. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000.
[6] K. Saito, "Electronic image pickup device," Japanese Patent 08-340486, 12 1996.
[7] E. Ikeda, "Image data processing apparatus for processing combined image signals in order to extend dynamic range," U.S. Patent 5801773, 9 1998.
[8] R. A. Street, "High dynamic range segmented pixel sensor array," U.S. Patent 5789737, 8 1998.
[9] Tsun-Hsien Wang, Cheng-Wen Chiu, Wei-Chen Wu, Jen-Wen Wang, Chun-Yi Lin, Ching-Te Chiu, Jing-Jia Liou:
Pseudo-Multiple-Exposure-Based Tone Fusion With Local Region Adjustment. IEEE Transactions on Multimedia 17(4): 470-484 (2015)