Real-Time HDR Panorama Video

Lorenz Kellerer[1], Vamsidhar Reddy Gaddam[2], Ragnar Langseth[2],
Haakon Stensland[2], Carsten Griwodz[2], Dag Johansen[3], Paal Halvorsen[2]

[1]Technical University Graz, Austria
[2]Simula Research Laboratory & University of Oslo, Norway
[3]University of Tromsø, Norway

[1]lkellerer-pirklbauer@student.tugraz.at [3]dag@cs.uit.no
[2]{vamsidhg, ragnarla, haakonks, griff, paalh}@ifi.uio.no

ABSTRACT

The interest for wide field of view panorama video is increasing. In this respect, we have an application that uses an array of cameras that overlook a soccer stadium. The input of these cameras are stitched together to provide a panoramic view of the stadium. One of the challenges we face is that large parts of the field are obscured by shadows on sunny days. Such circumstances cause unsatisfying video quality. We have therefore implemented and evaluated multiple algorithms related to high dynamic range (HDR) video. The evaluation shows that a combination of several approaches gives the most useful results in our scenario.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Experimentation; Measurement; Performance

Keywords

High dynamic range (HDR); Panorama video; Real-time

1. INTRODUCTION

Systems providing panorama videos have gained an increased interest as they can deliver both a higher resolution and a wider field-of-view compared to a single camera. Some of the application scenarios include surveillance, navigation, scenic views, educational exhibits and sports. In our research, we use such panoramas in an arena sport scenario [6], and we are currently able to generate a panorama video of a whole soccer stadium in real time.[12]

However, a major challenge is managing variations in lighting conditions. Our current setup works very well in most weather conditions. However, situations with low and/or bright sun give, as seen in figure 1(a), large challenges over the field. There is a large difference between the amount of light available from areas in the sun and in the shadows, i.e., in a clear day, we only have the choice of having a good dynamic range in one region. When there are intensely bright and dark areas in the image (figure 1(a)), most cameras also have problems creating a representative image. Particularly, in our stadium case study, the sun is significantly lower in the sky than most of the habitable world giving challenges as shown in the figure. In such a case, aiming for a good quality in the sunny areas leads to loss of details in shadows. It should also be noted that professional broadcasters also experience these problems. However, to address this problem, they have people manning cameras (and thus also the exposure settings) as well as a someone controlling the live broadcast performing manual adjustments (figures 1(c) and 1(b)), i.e., they manually select the area of interest probably depending on where the ball is located.

(a) Automatic camera (b) Broadcaster 1 (c) Broadcaster 2

Figure 1: Lighting challenges at our case study stadium. Comparison with two professional broadcasters. (NOTE: images from the same game but different situations)

Due to our aim of a fully automated system, our system needs to handle this without human interaction and in real-time. The problem is related to sub-optimal auto exposure and insufficient dynamic range on the camera sensors. Improvements can be achieved in several ways. In this respect, one could solve common auto exposure problems as proposed in [8] , and use real-time assembling of high-dynamic-range (HDR) video by using low-dynamic-range images [2, 5, 4]. Another important detail in our work is that a panorama texture usually contains a lot of pixels, i.e., 2040×5400 in our case, unlike the related works where the maximum resolution handled is Full HD (1920×1080). Even though the algorithms do not get affected, this introduces several practical challenges under a real-time constraint. In this paper, we present a panorama pipeline with an HDR module that can handle difficult lighting conditions. This module is a two-step process, first step creates the high dynamic

MM'14, November 3–7, 2014, Orlando, Florida, USA.

ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2655049.

range image and second performs visually pleasing reduction of the dynamic range to normal screens. We have tested 3 different approaches for each step thus giving 9 possible combinations. Our experimental results show that the visual quality is generally improved over the LDR in all the combinations. Moreover some combinations also process 50 fps input to 25 fps HDR panorama video in real-time in our GPU accelerated implementations.

2. HDR ALTERNATIVES

An HDR image can be created by a process called radiance mapping, where multiple LDR images of the same scene are fused together. The fused image as such cannot be displayed using the conventional devices, so a compression of the dynamic range is performed where we preserve details in all regions and maintain a local contrast, i.e., giving a visually pleasing result. This process is called tone mapping. For the rest of the paper, we refer to the final output of tone mapping as the HDR image. We explored three approaches for radiance mapping, and three for tone mapping. We chose radiance mappers based on the criterion that they must merge multiple exposures. For the tone mappers, we picked a representative subset: one very simple to serve as easy comparison to others, one global and one local tone mapper. After extensive material research, we picked algorithms that subjectively provided the most pleasing results.

2.1 Radiance mappers

Debevec. One of the most cited approaches is presented by Debevec *et al.* [3], where the authors try to recover HDR radiance maps from photographs. It performs an estimation of camera response function and then a weighted selection process where the information is extracted from mid-tone regions of different exposures. We made a GPU implementation for the second step alone, since the first step needs to be performed once in a lifetime for a camera.

Robertson. Similar to *Debevec*, Robertson *et al.* [11] proposed a two step mapping with similar modules. But, they introduced a more extensive approach for recovering the camera response function. Thus, the radical changes are only to the offline step of the algorithm.

Tocci. As opposed to the previous approaches, Tocci *et al.* [13] propose a solution that consists of a single online step. The main assumption is that unless close to saturation in intensity, the preferred output pixels are those from the high exposures. Therefore, they introduce an approach that takes saturated pixels from the neighbourhood into consideration. Since this approach requires fetching the same pixels by different threads, we use the GPU's texture memory to exploit the spatial caching feature.

2.2 Tone mappers

Ward. Ward *et al.* [14] proposed an approach where a global scale factor is applied to each pixel, which is dependent on, among other parameters, the average brightness of the input image. A parallel reduction approach [7] for calculating the average over an entire image is implemented as part of our GPU implementation.

Larson. This algorithm proposed by Larson *et al.* [9] performs tonal compression by creating a look-up table per frame to represent a desired histogram. Unlike a simple histogram equalization, the target histogram is computed taking human contrast sensitivity into account.

Reinhard. Reinhard *et al.* [10] try to emulate a technique called "dodging & burning" [1]. This approach relies on the information from local neighbourhood for tonal compression. Adaptive Gaussian kernels are employed along different dimensions to average the exposure value, the adaptive nature is from the fact that the size of these kernels depends on the local contrast changes.

3. EXPERIMENTAL RESULTS

The computer used for the experiments uses a six-core Intel Core i7-3930K at 3.2 GHz with 32 GB quad-channel DDR3 memory and an Nvidia GeForce GTX 680 graphics processing unit with 3GB memory based on the Kepler architecture.

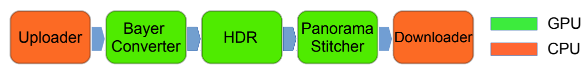


Figure 2: Pipeline modules

We capture the panorama video using five cameras, each with a resolution 2040×1080. Since our challenge is mostly due to the large difference in the amount of light available from two different regions (sun-light and shadow), we make two different exposures where one captures the details in shadow region and the other in hightlights. Hence, to generate one high resolution panorama frame, we have 10 images as input to the HDR module.

As seen in Figure 2, our pipeline consists of multiple consecutive steps. First, the captured frames are transferred to the GPU by the *Uploader* module running on the host. Then, the frames are converted to RGB from the recorded bayer pattern by the *Bayer converter* module which are then fed into the *HDR* module. The processed frames are then transferred back to host memory by the *Downloader* module. It must be noted that this is pipelined architecture which implies that when one frame is being processed by the *HDR* module, the next frame could be under processing in the *Bayer* module, and so on.

3.1 Visual Quality

In the following section, we will provide a subjective assessment of the different configurations of our implemented radiance and tone mappers. The input images displayed in figure 3 show a small part of our panorama frames to highlight details. Then, the output of all possible combinations can be seen in figure 5. The visual quality of the resulting image is based on multiple factors [15].

We performed a limited user study to assess the quality of the different configurations of algorithms. Here we asked 12 participants to rate videos obtained by combining the different algorithms. Several emerging patterns can be observed from the results shown in Figure 4. The first concerns the choice of tone-mappers. Here the average score

(a) Low exposure (b) High exposure

Figure 3: Input images for the HDR processing

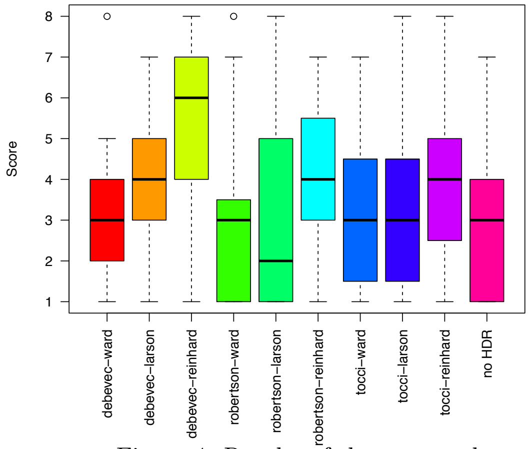


Figure 4: Results of the user study

achieved by *Robertson's* is higher than for the other tone-mappers in every combination with radiance-mappers. A second, less clear trend can be observed for the radiance-mappers. Here *Debevec's* always scores similar or higher than the other proposed radiance-mappers when combined with different tone-mappers

3.2 Execution Time

Real time performance is a major concern for our project. We accomplished this goal on all algorithms except with *Reinhard's* tone mapper. The multiple FFTs for applying the gaussian kernels in this algorithm are performed per frame. Although we use the highly optimized CUFFT library provided by Nvidia, the multiple FFTs still turned out to be a bottleneck. Furthermore, *Reinhard* is consuming a lot of memory. For each stored gauss-kernel, approximately 126 MB have to be allocated.

Furthermore, the combination *Tocci - Larson* is slightly above the real-time threshold, but we believe that with the optimizations mentioned in section 4, we can achieve real time here too.

A detailed listing of execution times can be found in Figure 6. We also executed only the HDR modules without interference from other modules. Those execution times can be seen in Figure 7.

Each of the modules contains several kernels. The scheduling of these kernels on the GPU is managed by CUDA. So, it must be noted that the execution times include the overheads created by scheduling. It can be seen that the execution times of other modules fluctuate when different HDR algorithms are employed. This is from the fact that CUDA schedules the kernels from these modules in different ways depending on the requirements of different modules. The GPU used in our experiments are capable of executing up to 16 kernels concurrently.

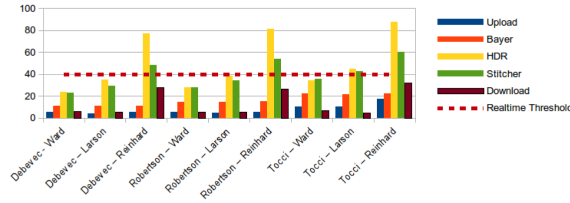


Figure 6: Execution times (ms) of the various modules in different configurations

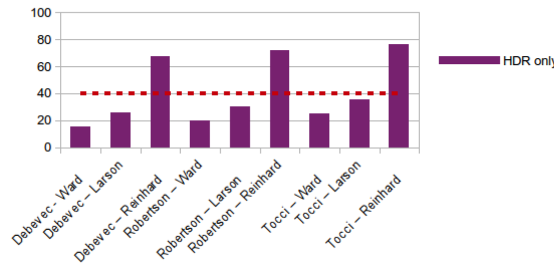


Figure 7: Execution times (ms) of the HDR module in different configurations

3.3 Summary

After taking both visual quality and the execution times into account, we came to a conclusion about the most optimal combination. We believe that using *Debevec's* radiance mapper paired with *Larson's* tone mapper is the best approach. Although *Reinhard's* tone mapper shows promising results, we could not make it pass the real-time requirements, and it also consumed too much memory. Ward's tone mapper produced useless results with our input. *Tocci's* radiance mapper proved to be too complex to efficiently be implemented in CUDA. Also, in our case, it produced undesirable artifacts. Lastly, *Robertson's* radiance mapper also proved to be very good, but the higher dynamic range it produced caused worse results when tone mapped.

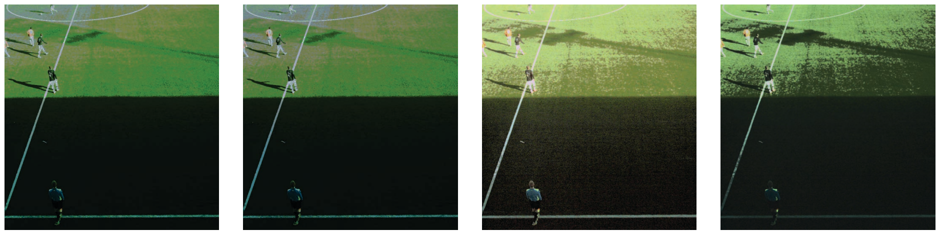
4. DISCUSSION

We experimented with the placement of the HDR module within the pipeline. This step can either be performed before or after stitching the individual images together into a panorama. In the end, we decided to do it before because of three reasons:

1. The input to the HDR module is smaller since the stitcher removes overlap. But, this reduction was not enough to have a large impact on the performance.
2. The stitcher struggled with the high exposure input. Since large areas in this input are pure white, the performance of the stitcher suffered.
3. The HDR module also modifies the frame rate from 50fps to 25 fps. Thus, reducing the number of frames to be stitched by half.

We want to further improve the performance of the implementations we deemed worth pursuing. Therefore, we are looking into ways to parallelize Larson's histogram adjustment. Furthermore, we are exploring possibilities to distribute the HDR module on multiple machines. Since the capturing of the raw video is performed on two computers, it would seem advisable to perform HDR in a distributed

(a) Debevec - Ward (b) Debevec - Larson (c) Debevec - Reinhard (d) Robertson - Ward (e) Robertson - Larson



(f) Robertson - Reinhard (g) Tocci - Ward (h) Tocci - Larson (i) Tocci - Reinhard

Figure 5: Visual quality after HDR for the different algorithms using the input images in figure 3

fashion. Lastly, the output of the tone mappers is scaled to a scene brigthness key value. This determines which value is mapped to middle brightness. In the current setup, this value is not changed even if the lighting conditions change. We want to explore ways to dynamically and intelligently change this key value.

5. CONCLUSION

By integrating HDR into our image processing pipeline, we managed to solve the problem of real-time HDR at large resolutions. We implemented various HDR algorithms and evaluated their performance ending up with a favourable combination. Our pipeline can now handle diverse lighting conditions. We can now display the entire soccer field at once despite the high variation in luminance due parts of it being in dark shadows and other in bright sun light. As opposed to the previous approaches [5], we can perform real-time HDR at a far greater resolutions up to 2040 × 5400.

6. REFERENCES

[1] A. Adams and R. Baker. *The print*. Little, Brown, 1983.

[2] M. A. Ali and S. Mann. Comparametric image compositing: Computationally efficient high dynamic range imaging. In *Proc. of IEEE ICASSP*, pages 913–916, 2012.

[3] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008.

[4] B. Guthier, S. Kopf, and W. Effelsberg. Optimal shutter speed sequences for real-time hdr video. In *Proc. of IEEE IST*, pages 303–308, 2012.

[5] B. Guthier, S. Kopf, and W. Effelsberg. A real-time system for capturing hdr videos. In *Proce. of ACM MM*, pages 1473–1476, 2012.

[6] P. Halvorsen, S. Sægrov, A. Mortensen, D. K. Kristensen, A. Eichhorn, M. Stenhaug, S. Dahl, H. K. Stensland, V. R. Gaddam, C. Griwodz, and D. Johansen. Bagadus: An integrated system for arena sports analytics – a soccer case study. In *Proc. of ACM MMSys*, pages 48–59, Mar. 2013.

[7] M. Harris et al. Optimizing parallel reduction in cuda. *NVIDIA Developer Technology*, 2:45, 2007.

[8] W. C. Kao, L. W. Cheng, C. Y. Chien, and W. K. Lin. Robust brightness measurement and exposure control in real-ime video recording. *IEEE Trans. on Instrumentation and Measurement*, 60(4):1206–1216, 2011.

[9] G. W. Larson, H. Rushmeier, and C. Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Trans. on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[10] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. on Graphics*, 21(3):267–276, 2002.

[11] M. A. Robertson, S. Borman, and R. L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2):219–228, 2003.

[12] H. K. Stensland, V. R. Gaddam, M. Tennøe, E. Helgedagsrud, M. Næss, H. K. Alstad, A. Mortensen, R. Langseth, S. Ljødal, O. Landsverk, C. Griwodz, P. Halvorsen, M. Stenhaug, and D. Johansen. Bagadus: An integrated real-time system for soccer analytics. *ACM Trans. Multimedia Comput. Commun. Appl. (TOMCCAP)*, 10(1s):14:1–14:21, Jan. 2014.

[13] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen. A versatile hdr video production system. 30(4):41, 2011.

[14] G. Ward. A contrast-based scalefactor for luminance display. *Graphics gems IV*, pages 415–421, 1994.

[15] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Image attributes and quality for evaluation of tone mapping operators. In *National Taiwan University*, pages 35–44. Press, 2006.