HDR Photographic Pipeline for Camera Modules in Mobile Devices

R.Mantiuk[1] & J. Cybularczyk[2] & M. Cichowicz[2] & M. Smyk[2] & B. Bazyluk[1]

[1]West Pomeranian University of Technology in Szczecin, Poland
[2]BlStream Ltd., Poland

Abstract

We replace the standard image capture pipeline in mobile phones with an HDR acquisition pipeline based on the multi-exposure method. We report timings for basic HDR algorithms implemented in a smartphone with ARMv6 processor and discuss programming techniques that speed-up execution and reduce RAM memory usage. The results compare favourably to proprietary iPhone 4 HDR implementation and show that the HDR pipeline can be efficiently implemented on existing camera phones using high-level APIs and no dedicated hardware.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Image Processing and Computer Vision]: Applications

1. Introduction

Together with a CMOS/CCD image sensor, an Image Signal Processor (ISP) defines the image quality and the processing speed of the camera subsystem in mobile devices like smartphones or PDAs. At present, these subsystems are optimised to deliver LDR (Low Dynamic Range) photos.

HDR photography pipelines offer features unavailable in standard cameras like post-capture exposure adjustment or extended dynamic range. Moreover, many advanced techniques of classical photography can be significantly enhanced (e.g. contrast boost, denoising, etc).

In our project we replace the standard image capture pipeline of mobile phone with an HDR acquisition pipeline based on the multi-exposure method. This concept, although known previously from Apple's iOS 4.1, is difficult to implement in devices where soft- and hardware components of the camera subsystem are not specifically tuned to the goal of HDR photo capture. We propose a pipeline architecture that follows all the features of the standard HDR pipeline (see Sec. 2). The limitations of mobile devices are overcome by reducing the accuracy of algorithms and exploiting both hardware-specific and general optimisation techniques, rather than removing crucial operations from the pipeline. We present limitations of mobile platforms (see Sec. 3) and show how to work around them (see Sec. 4).

2. HDR photographic pipeline

In our HDR photographic pipeline the multi-exposure acquisition technique is used to extend the dynamic range of a final photograph. In contrast to standard photographic pipeline where only one input image is used, a sequence of images can also be utilised to reduce noise, merge flash/no-flash photographs, etc.

The HDR pipeline features direct access to the raw sensor data to execute image processing operations directly on Bayer channel data (before demosaicing, denoising, or sharpening). If sensor data is stored in a disk file, lossless compression is used. The only processing performed on the raw sensor data is noise suppression.The access to unmodified raw data enables the use of a range of algorithms for further processing, and the adjustment of the accuracy of these algorithms for specific device's capabilities. Moreover, no information is lost nor degraded due to undesired processing by the camera's ISP or upstream software. To reduce performance overhead due to large pixel numbers, further data processing is task-oriented. For example, it is not necessary to process the whole data set from a 12Mpx sensor if a final photograph is going to be displayed on a 640×360 pixels LCD. The data size is thus limited by the precision and format of an output stream. The pipeline ends with lossy data compression suitable for storing the data in possibly compact form.

3. Mobile platform limitations

The depicted photographic pipeline is difficult to implement in contemporary mobile devices due to their hardware and software limitations. Processing ability of smartphones is constrained by both RAM size and CPU power, with floating point operations being especially slow. There are more and more mobile devices on the market with programmable GPU units, but architectural decisions guided by focus on improving gaming performance usually make it impossible to use shaders for processing large data sets typical of HDR.

Camera modules in mobile devices produce results of lower quality than dedicated standalone cameras. Interestingly, the parameters of specific individual parts of the camera are often pushed up for marketing targets and do not fit well into the whole hardware-software system. A typical example is a very high resolution sensor combined with mediocre optics and low RAM capacity.

However, we find not the hardware but the software limitations of camera systems to be the most problematic issue. The existing mobile APIs for the ISP modules do not allow the direct access to the sensor raw data and often provide only high-level interfaces that deliver JPEG-compressed and sharpened images (Frankencamera API [ATP*10] may change this in the future). Processing (e.g denoising) of images after the mentioned hardware processing is often ineffective even using advanced state-of-the-art algorithms. Finally, in imaging pipelines unoptimized for rapid shooting there is a noticeable gap between subsequent exposures (often above 1s), which combined with hand-held nature of mobile photography necessitates aligning the exposures.

4. Example implementation

We implemented the basic HDR operations on a mobile phone with ARMv6 architecture processor (700 MHz, no NEON SIMD instructions), based on merging of two 2EV-separated exposures. To counter image misalignment between exposures, we perform image registration using modified Adams et al. algorithm [AGP08]. Our changes include increasing low-pass filter size to 7×7 pixels to improve robustness of corner detection, addition of histogram equalisation that improves pairing of corresponding points in differently exposed images, and avoiding using corners found in these areas of an image that are likely to be under- or overexposed on the other one. Total time for this stage is 1200 ms/Mpx, including bilinearly filtered in-place image transformation (to save memory). It is worth noting that if low-level access to the sensor was available and the two exposures could be captured immediately one after another, this stage would effectively be made unnecessary. In the next step, a HDR merge combines two images at 200 ms/Mpx using response curve recovered from the device. The global photographic tone mapping [RWPD05, Sec.7.3.6] operating at 600 ms/Mpx is used to produce final LDR version of the photograph. Our peak memory consumption is $250KB + px \times 8\ bytes$, where px is the total number of pixels in a single exposure. Given our device's limit of approx. 80MBs of memory available for application use, we are able to process 8Mpx images.

These results compare favourably to Apple's proprietary HDR implementation in iPhone 4, which operates at about 1000 ms/Mpx on a considerably faster CPU (Apple A4) and without image registration step.

Compared to naive implementation, processing time was significantly reduced by using fixed-point math, replacing time-consuming computations like exponential or gamma-correction functions by look-up tables, and using low-level optimised implementation of logarithmic function. Additionally, the shared exponent of 32-bit Radiance RGBE pixel format [RWPD05, Sec.3.3] was exploited as a memory-saving intermediate HDR representation. Special attention was paid to cache-friendly data layout and instruction ordering that avoids CPU pipeline stalls, a measure that is especially important on an in-order ARMv6 architecture.

5. Future work

To further increase processing speed and reduce peak memory usage in our imaging software, we plan to implement multi-threading and tiling techniques [MP08], which should enable us to build interactive applications featuring more advanced algorithms, including pyramid-based and gradient-domain techniques.

6. Acknowledgements

This work was partially supported by grant no. N N516 508539 founded from MNiSW (Poland).

References

[AGP08] ADAMS A., GELFAND N., PULLI K.: Viewfinder alignment. *Comput. Graph. Forum 27*, 2 (2008), 597–606.

[ATP*10] ADAMS A., TALVALA E.-V., PARK S. H., JACOBS D. E., AJDIN B., GELFAND N., DOLSON J., VAQUERO D., BAEK J., TICO M., LENSCH H. P. A., MATUSIK W., PULLI K., HOROWITZ M., LEVOY M.: The Frankencamera: an experimental platform for computational photography. *ACM Trans. Graph. 29* (July 2010), 29:1–29:12.

[MP08] MANTIUK R., PAJAK D.: Acceleration of high dynamic range imaging pipeline based on multi-threading and SIMD technologies. In *LNCS (Proc. of ICCS'08)* (2008), vol. 5101/1, pp. 780–789.

[RWPD05] REINHARD E., WARD G., PATTANAIK S., DEBEVEC P.: *High Dynamic Range Imaging. Data Acquisition, Manipulation, and Display*. Morgan Kaufmann, 2005.