Mobile HDR Capture and Exposure Fusion

Jason Chaves

Introduction:

High Dynamic Range (HDR) imaging is the process of producing an image with a larger dynamic range than a single camera image, in an attempt to more closely approximate the much larger dynamic range that the human eye can perceive. This allows for more visually appealing images, as it reduces the number of saturated and under-exposed pixels, and of course it more closely mirrors what the human eye would see.

Software-based approaches to HDR imaging involve capturing multiple shots of a scene at different exposures, to capture the large dynamic range of the scene, and then performing a tone-mapping of some sort so that the image can be represented in the usual 8-bit, low dynamic range (LDR) format. The problem of software-based HDR methods can therefore be divided into two steps: exposure bracketing and exposure fusion.

Exposure bracketing involves the selection of exposures, as well as the capturing of shots at the selected exposures. This requires the ability to control the exposure time (and gain) of the camera. Exposure bracketing schemes are free to choose the number of shots captured, the relation between the exposure times, and the absolute scale of the exposure times. The selection of the absolute scale of exposure times can also be done a priori, or on-the-fly using metering techniques.

Exposure fusion involves the combining of the several LDR images, at varying exposures, into a single HDR image. This HDR image is still represented with 8-bit color channels, but fewer pixels are under- or over-exposed, allowing for both bright and dark parts of the scene to be visible. There are a number of exposure fusion techniques, including ones which calculate the radiance maps of the different LDR shots and then tone-map from the overall radiance map, ones which perform a weighted average of the LDR images, and ones which perform laplacian pyramid blending of the LDR images. These techniques usually also include registration steps to account for inter-shot motion, which can range from homography-based methods to optical-flow-based methods.

Implementing HDR imaging on a mobile device introduces certain limitations and requirements that would otherwise not be an issue for solutions that use stand-alone cameras for capture and PCs for computation. Mobile devices obviously have limited CPU and memory resources, so exposure fusion techniques must be able to function under such limited resources. Camera sensors on mobile devices may have even less dynamic range than regular stand-alone cameras, so it may require more shots to capture the same dynamic range. Furthermore, in mobile applications, the experience of the user is very important. Therefore, HDR solutions implemented on mobile should be easy to use and there should not be too much latency between when the user captures the shots and when the user sees the final HDR result.

Previous Work:

In Bilcu et. al., a very simple mobile HDR solution is presented. Their exposure bracketing scheme consists of always taking three shots: one at auto-exposure, one at a shorter exposure, and one at a longer exposure. A brute-force iterative approach is used to determine the exposure times for the under- and over-exposed shots. Their exposure fusion method consists of performing a weighted average between the 3 LDR shots. In response to mobile limitations, their method pre-calculates possible weight values, uses fixed-point arithmetic, and only stores two images in main memory at any given time. Their method is quite simple to implement and can serve as a minimal, fast baseline mobile implementation. Their HDR result images are certainly an improvement over single LDR images, but there is noticeable room for improvement in HDR image quality.

In Gelfand et. al., a more thorough mobile implementation is presented. They implemented HDR metering and viewfinding in order to continuously calculate the proper exposures, and show the user a preview of the HDR result, while the user is positioning the device before pressing the button to initiate the HDR capture. This allows for the exposures to be calculated before the HDR capture is done, so that only those exposures must be captured, as opposed to the brute-force approach of Bilcu et. al., which could result in more than the required three exposures being captured. This predetermination of exposures allows for fast capture of the shots in succession, with less inter-shot motion in between. Their metering method determines optimal short and long exposures based on pixel histograms; for example, for the short exposure, 1-10% of the pixels must be bright (8-bit values above 239). If the two exposures from metering are less than one stop apart (i.e. less than a factor of 2 different), then at capture, only one shot is captured at the short exposure. If the two exposures are between one and three stops apart (i.e. different by a factor of 2 to 8), then a shot is simply captured at each exposure. If the two exposures are more than 3 stops apart, a shot is captured at each exposure, and a shot is captured at a middle exposure, calculated as the geometric mean of the two settings. As for their exposure fusion method, they use a gaussian weighting to produce weight maps for laplacian pyramid blending. However, instead of performing the full laplacian blending, they perform it only up until the second-to-top level, and then the missing detail from the top level is added. This reduces the complexity of the blending, since processing the full-resolution top level is the most expensive, and it serves to sharpen the HDR result. The methods presented in this paper are not too much more difficult to implement, but should produce a more robust HDR solution with better results.

Gupta et. al. presents a Fibonacci bracketing scheme, as well as a generalized registration method based on dense optical flow. In the Fibonacci bracketing scheme, each exposure time is equal to the sum of the two exposure times before it, as in the Fibonacci number sequence. This allows for registration to be performed on sums of frames with equal exposure times, which minimizes registration error. The results shown in their paper are very impressive. However, in their paper, they briefly mention the use of more complicated de-blurring methods and

tone-mapping, so it remains to be scene if their bracketing and registration scheme is as beneficial when more straight-forward exposure fusion techniques are used. It is also important to note that their dense optical flow based registration may be prohibitively slow on mobile devices.

This Work:

In this work, I propose to experiment between different exposure bracketing and exposure fusion techniques on the Tegra Note 7 tablet. My goal is to implement an Android app that can capture a scene using different exposure bracketing techniques, and then process the LDR images into an HDR result using different exposure fusion techniques. This will allow for a comparison between bracketing-fusion pairs in terms of image quality and running time. The image capture will be implemented using the FCam library, and the image processing will be implemented using the OpenCV library.

At the moment, I intend on implementing the bracketing and fusion techniques presented in Bilcu et. al. and Gelfand et. al., using homography-based registration. This will allow for 4 bracketing-fusion pairs to compare. If time permits, I will explore the Fibonacci bracketing and general registration methods from Gupta et. al., while using the fusion methods from the previous two papers.

Technical Approach:

I intend to build an app similar to that in Project-1. It will have Menu items to select the bracketing and fusion schemes, and the rest of the window will be the camera view. Then user will then be able to initiate the HDR capture by tapping on the screen, and the HDR will be present after computation is completed.

I will use FCam to control the camera parameters and request shots. I intend to have a thread performing auto-focus, to find a good focal distance, and that thread will update a global focus parameter which all other Shot requests will use. I will have a thread performing auto-exposure, to give a starting point for the Bilcu et. al. bracketing scheme. I will have another thread running a metering method, to calculate the short and long exposures required for the Gelfand et. al. bracketing. These three threads will rely on capturing a stream of Shots from the camera module. If FCam cannot handle simultaneously scheduling/interleaving multiple streams, then I will need to find another way to implement these, but I believe FCam indeed has this functionality. The gain of the camera may be set as a constant value, or it may be adjusted within the auto-exposure method. FCam has example auto-focus and auto-exposure procedures, so I will have those to build off of and will have to modify them so that they can communicate their results with other threads when needed. These threads will be running in the background during the entire life of the app, although if feasible, it may be good to halt them while the exposure fusion is being done, so that more processor time can be devoted to that task.

When the user taps the screen to initiate HDR capture, the main thread will perform the selected bracketing scheme, using FCam to request the Shots. The Images produced by FCam will

then be converted into OpenCV Mats, and sent to the selected exposure fusion method. When the exposure fusion method returns, the resultant HDR image will be display to the user, as in Project-1.

All of the above computation will be implemented in native C++ code, and logging will be utilized to measure execution time of the fusion method.

Milestones:

5/23: Modify Project-1 code to capture shots at different exposures using FCam and then perform exposure fusion as in Project-1. Add Menu items to select bracketing scheme and fusion scheme. Ask Kari Pulli questions about FCam.

5/30: Add separate thread to app that will perform metering to predetermine exposure values as in Gelfand et. al. Implement Bilcu et. al. bracketing and fusion techniques. Implement fusion technique from Gelfand et. al.

6/4: Project Presentation

6/6: Finish taking HDR shots of different scenes for comparison of methods

6/8: Project Report Due

References:

A. Adams, E.-V. Talvala, S. H. Park, D. E. Jacobs,
B. Adjin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek,
M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli,
M. Horowitz, and M. Levoy. The Frankencamera: An
experimental platform for computational photography.
Proc. SIGGRAPH, 2010.

R. C. Bilcu, A. Burian, A. Knuutila, and
M. Vehvil¨ainen. High dynamic range imaging on mobile
devices. Proc. ICECS, 2008.

M. Gupta, D. Iso and S.K. Nayar,
IEEE International Conference on Computer Vision (ICCV),
pp.1-8, Dec, 2013.

Multi-exposure Imaging on Mobile Devices
Natasha Gelfand, Andrew Adams, Sung Hee Park, Kari Pulli
ACM Multimedia 2010 (short paper & technical demo)