Volume 30 (2011), Number 2 pp. 1–2 COMPUTER GRAPHICS *forum*

Towards Mobile HDR Video

T. K. Castro, A. Chapiro, M. Cicconet and L. Velho

Visgraf/IMPA, Brazil

Abstract

We present a novel method for High Dynamic Range video where the critical phases of the pipeline are based on histograms. It is possible to achieve high framerates, since the algorithm generates one HDR frame per captured frame. Also, the method is of low computational cost, making it particularly suited for devices with less powerful processors. An implementation of the capture process for the Nokia N900 smartphone, using the recent FCam API, is detailed.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Miscellaneous—

High Dynamic Range video reconstruction is more challenging than the image-related problem because, from the hardware side, it requires a programmable camera and, from the software side, the data is dynamic. The earlier reference in this case is [KUWS03], where classical vision methods for motion estimation are used to deal with the motion between frames. For a review of methods we refer to [Mys08]. Our approach for HDR video is based on histograms. It is efficient, simple and robust to noise. The creation of HDR video with a programmable camera was briefly introduced in [Vel07]. This work contains further elaboration of this novel method. The HDR reconstruction algorithm is a modified version of [RBS99]. The histogram-based image registration technique is brought from [War03], and the Radiance Map reconstruction with ghost removal is made in a similar way to what is described in [MPC09].

HDR Video Our method has three steps: first, the camera response function is estimated using a histogram-based technique; second, multiresolution alignment of threshold images based on histogram cuts is performed; third, the radiance map is reconstructed observing the variances of radiance values for each pixel.

The input of our algorithm is a sequence of triples of images $\{F^i\}$, where $F^i = \{F^i_1, F^i_2, F^i_3\}$. The exposure of F^i_1 is updated during the video capture, through an auto-exposure algorithm. F^i_2 and F^i_3 have exposures that are, respectively, a half and twice the exposure of F^i_1, for all i. We are assuming, based on [Vel07], that exposure changes preserve monotonicity of pixel values. Intuitively, the n brightest pixels in a frame with exposure e_1 correspond approximately to the n brightest pixels in a subsequent frame with exposure e_2, even though their actual values are not the same. Let $\{p_i\}$ and $\{q_i\}$ be the sets of pixels from two consecutive frames (say, P and Q), of the same size, sorted according to the luminance value of the pixel. The radiance mapping $M_{P,Q}$, between P and Q, is defined simply by $M_{P,Q}(p_i) = q_i, \forall i$. Finally, the actual *pixel value* to *radiance value* mapping can be recovered by applying any of the algorithms available in the literature. For this particular implementation we have used the parametric approach described in [MN99].

Aiming not to reduce the HDR-reconstructed video framerate when compared to the captured video, we generate an HDR frame for each frame in F^i. Therefore, once $j \in \{1,2,3\}$ and i are fixed, the remaining frames of the triple should be aligned with F^i_j to compensate for camera movement. We perform a multiresolution alignment based on image pyramids, which is described in detail in [War03].

One of the main difficulties of our method consists of correctly displaying object movement in the scene. This follows from the fact that our method uses each captured frame for the resulting video. Three methods were used to deal with this problem. These methods are shown in the video provided with the supplementary material. The solutions consist of either using a simple average based on well-exposedness, detection of abrupt variations in a pixel's luminance, or a more complex method, as follows: disregarding the first and last two frames of the video and observing an arbitrary frame F, there are three possible HDR images for F. Considering the consecutive frames $\{D, E, F, G, H\}$, there is an intermediate HDR image for each set $\{D, E, F\}$, $\{E, F, G\}$, $\{F, G, H\}$, all of them aligned to F. The average of these HDR images results into our final HDR frame $\hat{F}$ for the captured frame F. Using $\hat{F}$ we get a smoother and more robust result in comparison to what could be achieved by just selecting one of the intemediate HDR images. $\hat{F}$ also generates a motion blur effect, which may not be desirable in some applications. This method is applied to the entire capture.

Video Capture In this work, we used a Nokia N900 running Maemo 5 and the FCam API [ATP*10], which allows to have full control over the camera parameters, such as exposure time, gain and focus. The major challenge concerning the capture and processing of HDR frames on a mobile device is to maintain a fine balance between framerate, memory usage and processing power. The current implementation uses three different exposure settings, which provides good results without harming this balance. We limit our application to capture only short videos due to memory limitations.

Processing HDR video The processing of the captured frames is done after all the frames were captured. This is a way to guarantee that there is no slowdown during the capture process. Since this stage is done independently of the first, virtually any HDR method could be used here. This stage could be performed on the mobile device, on another device, or even on a cloud. Our current implementation performs this step on a desktop computer for testing purposes; however, we are currently working on an application which will perform the full method on the camera itself. This stage of implementation is straightforward, since the same code can run on both machines and the chosen method does not demand high processing power.

Results In the current implementation stage the algorithm being used on the Nokia N900 returns a sequence of images with varying exposures. We then proceed by transferring the results to a desktop computer in order to process the data. After this step, we have a set of HDR images, which correspond to each frame of the captured video. In order to be able to visualize our results on regular LDR devices, a tone-mapping algorithm is also necessary. This step is done by using the `pfstmo` library, from the Max Planck Institute. After testing several tone-mapping methods, we have decided to use an implementation of [DMAC03] present in the library, due to both its speed and the quality of the results obtained. The final resulting tone-mapped output, along with a sequence of three differently exposed frames generated by our program can be seen on Figure 1.

Future Work An interesting challenge involves finding a better way to deal with the device's small memory, in order to increase the amount of frames that can be captured. A possible solution lies in creating a low priority thread that would save images to the device's hard disk while the program is still running. Another improvement might come from other means of pixel correspondence for object movement in the scene, such as optical flow, to further enhance the quality of the results. Also, other Tone Enhancement techniques could be used to improve the quality of the captured videos.

Figure 1: *(Top) Captured frames. From left to right: optimally-, sub- and super-exposed shots. (Bottom) Corresponding results with tone-mapping. Notice that the background (resp., the building interior) is only well-exposed on the second (resp., third) captured frame. Both areas of the image are well visible in the three tone-mapped results.*

References

[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*, 4 (2010), 1–12. 2

[DMAC03] DRAGO F., MYSZKOWSKI K., ANNEN T., CHIBA N.: Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum 22* (2003), 419–426. 2

[KUWS03] KANG S. B., UYTTENDAELE M., WINDER S., SZELISKI R.: High dynamic range video. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM, pp. 319–325. 1

[MN99] MITSUNAGA T., NAYAR S.: Radiometric Self Calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 1999), vol. 1, pp. 374–380. 1

[MPC09] MIN T.-H., PARK R.-H., CHANG S.: Histogram based ghost removal in high dynamic range images. In *ICME'09: Proceedings of the 2009 IEEE international conference on Multimedia and Expo* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 530–533. 1

[Mys08] MYSZKOWSKI K.: *High Dynamic Range Video*. Morgan and Claypool Publishers, 2008. 1

[RBS99] ROBERTSON M., BORMAN S., STEVENSON R.: Dynamic range improvement through multiple exposures. In *Proceedings of the IEEE International Conference on Image Processing* (Kobe, Japan, Oct. 1999), vol. 3, IEEE, pp. 159–163. 1

[Vel07] VELHO L.: Histogram-based hdr video. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters* (New York, NY, USA, 2007), ACM, p. 62. 1

[War03] WARD G.: Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *JOURNAL OF GRAPHICS TOOLS 8* (2003), 17–30. 1