A Real-Time System for Capturing HDR Videos

Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg
Department of Computer Science IV
University of Mannheim
Mannheim, Germany
{guthier | kopf | effelsberg}@informatik.uni-mannheim.de

ABSTRACT

We present a system that is capable of capturing, processing and displaying High Dynamic Range (HDR) videos at 23 frames per second. For each frame in the video, a sequence of regular images is captured in quick succession at optimally chosen shutter speeds. Intermediate camera motion is compensated and the images are combined into an HDR frame which is tone mapped for display. Our system makes use of novel algorithms that are fast enough for processing the frames in real-time. The most time-consuming operations are parallelized and executed on a graphics card for a speed-up of 15 to 1 over the sequential version.

Categories and Subject Descriptors

I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture; I.4.3 [**Image Processing and Computer Vision**]: Enhancement—*Registration*

General Terms

Algorithms, Performance

Keywords

High Dynamic Range, Processing, Display

1. INTRODUCTION

A recurring problem when capturing videos is the scene having a range of brightness values that exceeds the capabilities of the capturing device. An example would be that of a scene captured by a camcorder which includes a very bright outside area but also dark shaded regions. Because of the potentially big brightness difference, it may not be possible to capture the details of both areas in the video using a single exposure setting. This results in under- and overexposed pixels in the video footage.

A low-cost solution to this problem is temporal exposure bracketing, i.e., using a set of video frames captured in quick sequence at different shutter settings [3]. Each frame then captures one facet of the scene's radiance range. When fused together, a high dynamic range video frame is created that reveals details in dark and bright regions simultaneously. Doing exposure bracketing and merging at a sufficiently fast rate results in an HDR video.

The process of creating a frame in an HDR video can be thought of as a pipeline where the output of each step is the input to the subsequent one. It begins by *capturing* a set of low dynamic range (LDR) images using varying shutter settings. Typically, the shutter speed is doubled or halved with each additional image captured. Next, the images are *registered* with respect to each other to compensate for camera motion during capture. The aligned images are then *stitched* together to create a single HDR frame containing accurate radiance values of the entire scene. As a last step, the HDR frame is *tone mapped* to the output range of a regular LDR screen for visualization. In this paper, we present a system for capturing HDR video in real-time. It is capable of capturing, processing and displaying 23 HDR video frames per second.

The rest of this paper is structured as follows. Section 2 presents previous work in the field of HDR imaging. In Section 3, we describe the major components of our system. Their quality and processing time is discussed in Section 4. Section 5 concludes the paper.

2. RELATED WORK

Although HDR video creation in our system consists of the four steps capturing, registration, stitching and tone mapping, most of the previous work focuses on one step only. Our goal is to combine these steps into an application which is able to capture and process HDR data in real-time. Such a combined system allows to use information calculated in previous steps to improve the processing time and the visual quality of the resulting video.

The most popular technique to create HDR images is using a set of LDR images captured in quick sequence at different exposure settings. Most works in this field focus on the estimation of the inverse camera response function to map pixel values onto scene radiance [3]. With known shutter speeds, multiple exposures can be stitched into an HDR frame by computing a weighted average over the pixel values after applying the inverse camera response function.

Several techniques have been proposed to determine suitable exposure settings for capturing. Barakat et al. [1] present an approach which minimizes the number of exposures while covering the entire dynamic range of the scene. Minimizing

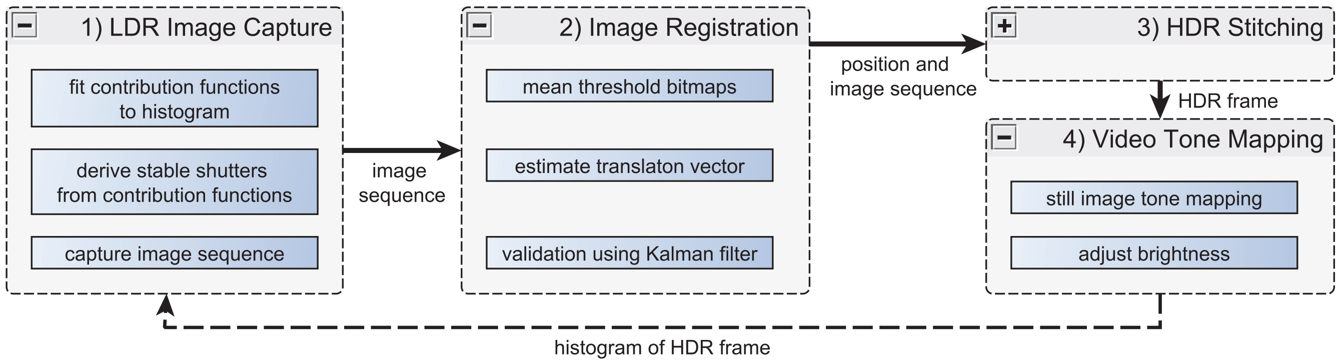

Figure 1: Overview of the HDR video processing system. As a first step, a sequence of LDR images is captured. The image sequence is passed to the registration module where camera motion in the sequence is compensated. The registered sequence is then stitched into a single HDR frame, which is finally tone mapped for display. HDR image statistics are passed back to the capturing module and used to determine the capture parameters for the next frame.

the number of LDR images to capture is desirable to save capture time. Minimum and maximum of the scene's irradiance range are taken into account, and the least possible overlap of exposures is chosen. A recent method to determine noise-optimal exposure settings using varying gain levels is proposed in [6]. For a given sum of exposure times, increasing gain also increases the SNR. However, the computation of the exposure settings in their approach is too expensive to be used in a real-time scenario.

Existing approaches to image registration often have difficulties coping with the large brightness difference between the LDR exposures. Only few techniques treat this problem specifically [5]. Ward [7] uses thresholded images that are robust to brightness variation and performs an efficient hierarchical search for translational camera motion.

Tone mapping operators map floating point valued radiance back to suitable 8-bit pixel values for display. Only few operators specific to HDR *video* content have been proposed. Benoit et al. [2] propose a model based on properties of the human retina. HDR video is enhanced by a non-separable spatio-temporal filter with added temporal constancy. The drawback of existing video tone mapping techniques is that they can only be used with a specific TM operator. We have developed a tone mapping technique for videos which removes flicker in a post-processing step and is applicable to all TM operators [4].

3. HDR VIDEO PROCESSING SYSTEM

In this section, we give an overview of our HDR video system, with novel approaches for acquisition and registration. The HDR video pipeline as implemented by us is shown in Figure 1. It consists of the four modules LDR image capture, image registration, HDR stitching, and video tone mapping. They are outlined in the following subsections.

3.1 LDR Image Capture

The capturing of LDR images constitutes the first step. The idea is to only use the shutter speeds that contribute the most information to the HDR frame. The fewer images are captured, the less time is needed to process them, leading to higher frame rates. Yet at the same time, the dynamic range of the scene may necessitate a certain minimum number of exposures so that all detail is captured properly. The goal is thus to get the most out of the recorded exposures.

An HDR frame is a map of physical radiance in the scene. Each pixel in the captured LDR images is a noisy measurement of this radiance. The quality of this measurement is a function of the pixel value, with higher values generally leading to a more accurate measurement. This relationship between pixel value and measurement quality is expressed in a weighting function (see Figure 2 for an example). It determines how much the radiance estimate from a pixel contributes to the corresponding HDR pixel. In other words, it judges a pixel's usefulness for recovering a radiance value based on its brightness value.

For a given shutter speed Δt, we can calculate how well a radiance value E can be estimated from an image captured at Δt. This is done by combining the camera's response function f, i.e., the mapping from brightness to pixel value, and the weighting function w, which is the mapping from pixel values to usefulness. A radiance value E is mapped to a pixel value using the camera's response function f. The weighting function w then assigns a weighting to the pixel value. We define

$$c_{\Delta t}(E) = w(f(E\Delta t)) \qquad (1)$$

as the *contribution* of an image captured at Δt to the estimation of a radiance value E. These *contribution functions* specify precisely what we mean by "well-exposed".

In our HDR video system, the histogram of scene radiance values is a by-product of tone mapping the previous frames. Our approach thus uses the available histogram to calculate a shutter speed sequence in real-time. The shutter speeds are chosen in a way such that frequently occurring radiance values are well-exposed in at least one of the captured LDR images. This increases the average image quality for a given number of exposures or minimizes the number of exposures required to achieve a desired quality. We calculate the optimal shutter speeds Δt_i such that the peaks of the contribution functions $c_{\Delta t_i}(E)$ of the LDR images coincide with the peaks in the histogram. That is, radiance values that occur frequently in the scene lead to LDR images to be captured which measure these radiance values accurately. This is illustrated in Figure 3.

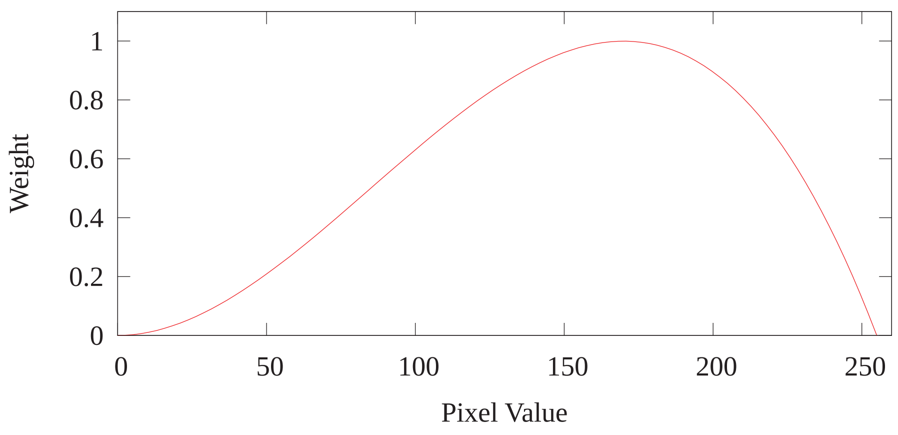

Figure 2: Example of a weighting function. The weight of a pixel is its value multiplied by a hat function normalized to a maximum weight of 1.

The contribution function corresponding to a different shutter speed $\Delta t'$ can be easily obtained by shifting the original function to another position in the histogram. This allows us to move the contribution function over a peak in the histogram and then derive the corresponding shutter speed. The process of finding new shutter speeds is iterated until the entire histogram is covered.

Determining a new shutter sequence for every frame in an HDR video can create visible flicker. Also, stable shutter sequences are more practical when operating the camera in the *sequence mode*. In this mode, a sequence of exposure parameters is sent to the camera. It then repeatedly captures exposures by cycling through the parameter list. This is done asynchronously by the camera and the captured exposures are buffered. Changing the shutter sequence requires a costly retransmission of the parameters, and the buffers are used suboptimally. For these reasons we impose a stability criterion upon the shutter sequence. We begin by defining whether two given shutter speed sequences are similar based on the percentual distance between their shutter values. Using this definition, we achieve temporal stability by distinguishing between two states: *changing* and *static*. The transition to *changing* only happens, when the calculated shutter sequence differs from the one currently set in the camera for a number of frames in a row. Only in the *changing* state, the new sequence is actually transmitted to the camera. Otherwise, the parameters are kept the same. Like this, small variations in the shutter speed sequence are ignored.

3.2 Histogram-based Image Registration

We address the challenge of estimating the camera motion between two LDR images in an efficient way. We argue that a purely translational camera motion model is sufficiently accurate for high frame rates ($\geq$ 200 frames per second). The goal of the registration algorithm is thus to estimate a translation vector between two LDR images captured at different exposure settings.

For estimating the translation vectors, we use *mean threshold bitmaps* (MTB) as described in [7]. A mean threshold bitmap is a black and white image that was created from the brightness channel of an image such that 50% of the image pixels are white and 50% are black. This threshold can be derived from the brightness histogram of the image. The advantage of an MTB compared to a regular grayscale image is that – within certain limits – two exposures depicting the same scene captured at two different exposure settings will result in approximately the same MTB. This fact is very desirable for image registration.

The horizontal and vertical component of the translation vector are estimated separately in a greedy algorithm. We start by counting the number of black pixels in each column of both MTBs to be aligned to create column histograms. A bin in the column histogram represents the number of black pixels in the corresponding column. This is demonstrated in Figure 4. By computing the NCC for all possible shifts between the column histograms of the two exposures and finding the maximum correlation, we estimate the horizontal component of the translation vector. Repeating this process for image rows allows us to estimate the vertical component in the same way.

As a last step, all resulting translation vectors are validated using a Kalman filter to incorporate knowledge of the camera motion in the previous frames into the estimation. A certainty criterion is used to determine the weighting between using the computed translation directly and extrapolating it from the preceding trajectory.

3.3 HDR Stitching

The registered image sequence is merged into a single frame. We do not contribute own ideas to the field of HDR stitching and simply use the techniques proposed by Debevec and Malik [3]. They apply the inverse camera response function to the pixel values of the LDR images to convert them back to exposure. Since we employ an industrial FireWire camera with linear response, the response function f is the identity function in our case. By dividing the exposure by the shutter time, values that are proportional to scene radiance are obtained. Each pixel value in the HDR frame is then determined by computing a weighted average over the radiance at the same pixel position in all LDR images of the sequence.

3.4 Video Tone Mapping

In order to be displayable, the large radiance ranges of an

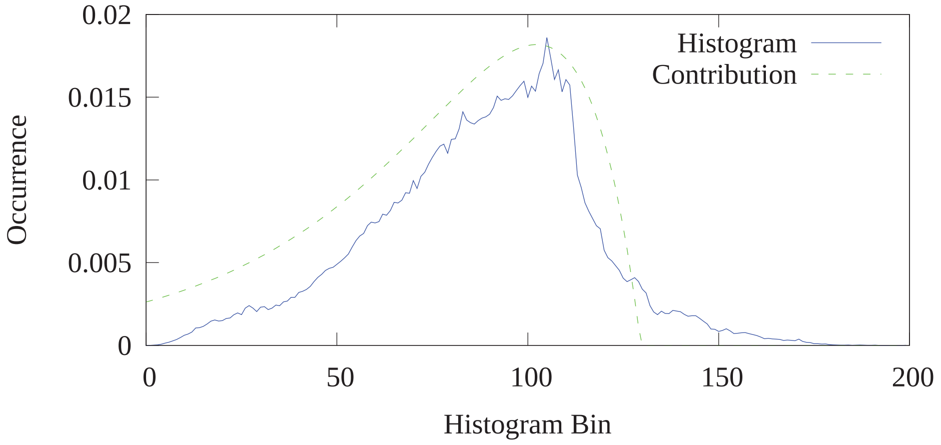

Figure 3: The solid line depicts an example log radiance histogram. The dashed line is the contribution function in the log domain corresponding to the first shutter speed chosen by our algorithm. The exposure was chosen such that it captures the most frequently occurring radiance values best.

HDR frame need to be compressed to the output range of a regular screen. Preferably, the compression is done in a way that maintains as much of the gained HDR information as possible. This process is called *tone mapping*. It is our goal to perform tone mapping of HDR videos using standard operators designed for still images. When doing so, temporal changes of the minimum, maximum, or average scene radiance lead to flicker in the tone mapped video.

We propose a generic method for the automatic detection and removal of flicker. It is implemented as a post-processing step in order to be independent of the tone mapper used. Flicker is caused by large changes in the average image brightness from one tone mapped frame to the next. To reduce the visibility of flicker, we adjust the average image brightness after tone mapping if it differs from the preceding frame by too much. This is done using image normalization and clamping to the output range, which is included as a last processing step in many tone mapping operators anyway. A frame that was mapped to a much darker average than its predecessor is adjusted to a level closer, but still darker than the previous frame. After a few frames of convergence, the same average brightness the operator would maintain without our intervention is reached again.

4. EVALUATION

To evaluate the quality of results produced by using our *LDR image capturing*, we performed a subjective user study with 27 participants. The subjects were shown twelve datasets, each consisting of a reference, an HDR result created using shutter speeds from our approach and one where evenly spread shutters were used. Each of the two results was rated using five scores from very good (5) to very poor (1). Averaging the ratings results in a score of 3.73 for the optimal shutter algorithm and 2.83 for the equidistant approach. Our approach achieved a better score in 70%, the same in 16%, and a worse score in 14% of the ratings.

For the evaluation of our *image registration*, all frames of five test videos were registered manually first. The resulting translation vectors constitute the ground truth. As the criteria for our evaluation, we use mean and standard deviation of the distance between the estimate and the ground truth over all frames of the videos. We compare it to our implementation of Ward's still image algorithm [7]. By incorporating knowledge of the motion in the previous frames, our algorithm achieves a lower registration error. Table 1 shows this result.

Video #	Ward	our approach
1	1.56 (3.46)	1.12 (2.60)
2	1.05 (2.21)	1.13 (0.89)
3	1.37 (4.05)	0.78 (0.78)
4	2.27 (4.70)	1.38 (1.37)
5	3.96 (6.33)	2.77 (2.89)

Table 1: Average image registration error (and standard deviation in brackets). We compared Ward's algorithm to our registration approach.

It is our goal to create HDR video in real-time. We thus made use of the parallel processing capabilities of a graphics processing unit (GPU) to improve the performance of our system. Among the most time-consuming operations are: the computation of brightness and row/column histograms for registration, computing the weighted average during HDR stitching, color space conversion and applying the tone mapping operator. These tasks are therefore performed on a GPU. Our system achieved an average rate of 23 HDR frames per second at VGA resolution in a 30 seconds test scenario. Compared to a sequential implementation, our parallelized version processes the frames 15 times faster on the average.

5. CONCLUSIONS

We presented a system for creating and displaying HDR video in real-time. It makes use of novel approaches for determining optimal shutter speeds, registering differently exposed LDR images and removing flicker from the tone mapped video. They achieve a higher image quality than existing techniques and are fast enough to create HDR video at 23 frames per second.

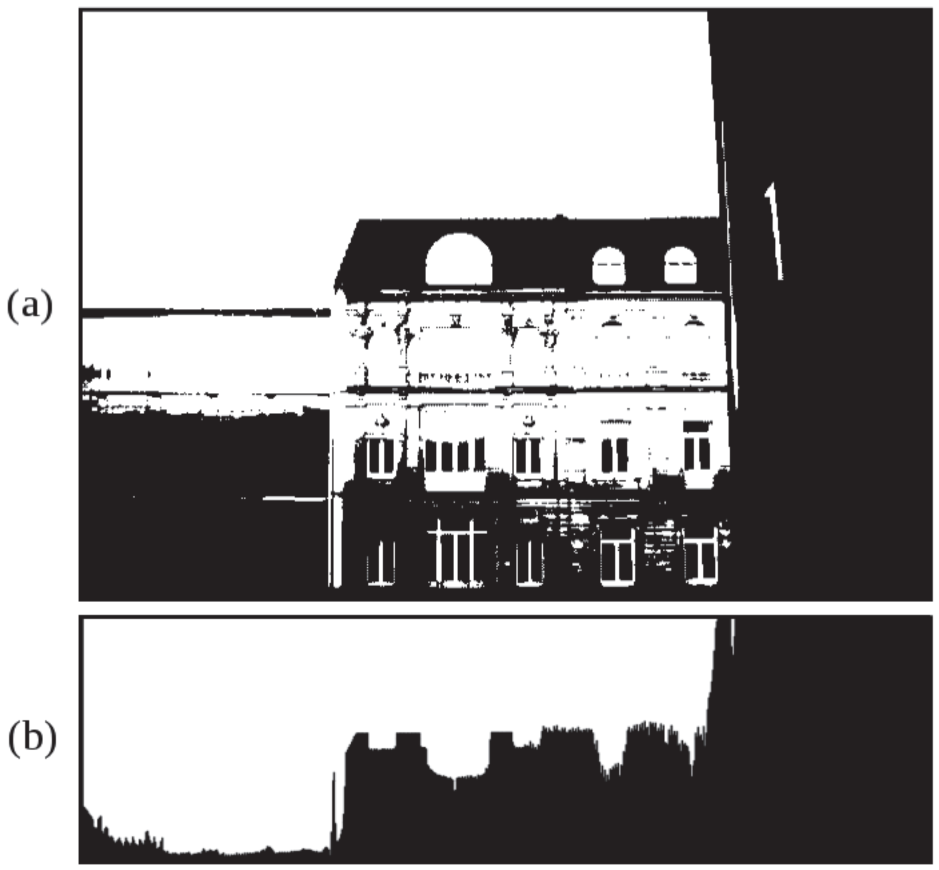

Figure 4: Mean Threshold Bitmap of an LDR frame (a) and its corresponding column histogram counting black pixels (b).

6. REFERENCES

[1] N. Barakat, A. Hone, and T. Darcie. Minimal-bracketing sets for high-dynamic-range image capture. *IEEE Trans. on Image Processing*, 17(10):1864 –1875, 2008.

[2] A. Benoit, D. Alleysson, J. Herault, and P. Callet. *Spatio-temporal Tone Mapping Operator Based on a Retina Model*, chapter 2, pages 12–22. Springer, 2009.

[3] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. of the 24th annual conference on computer graphics and interactive techniques*, pages 369–378, 1997.

[4] B. Guthier, S. Kopf, M. Eble, and W. Effelsberg. Flicker reduction in tone mapped high dynamic range video. In *Proc. of IS&T/SPIE Electronic Imaging (EI) on Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, volume 7866, pages 78660C:01 – 78660C:15, January 2011.

[5] B. Guthier, S. Kopf, and W. Effelsberg. Histogram-based image registration for real-time high dynamic range videos. In *Proc. of the 17th IEEE International Conference on Image Processing (ICIP)*, pages 145 – 148, September 2010.

[6] S. Hasinoff, F. Durand, and W. Freeman. Noise-Optimal Capture for High Dynamic Range Photography. In *Proc. of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 553–560, 2010.

[7] G. Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8(2):17–30, 2003.