This is a preliminary version of an article published by
B. Guthier, S. Kopf, W. Effelsberg:
Algorithms for a Real-Time HDR Video System.
ELSEVIER Pattern Recognition Letters (PATREC), Vol. 34 (1), Pages 25-33, January 2013.

Algorithms for a Real-Time HDR Video System

Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg

Department of Computer Science IV, University of Mannheim, Mannheim, Germany

Abstract

When the dynamic range of radiance values in a scene exceeds the capabilities of a camera, a single picture can only capture one brightness range of the scene faithfully at a time. We propose a system for creating high dynamic range (HDR) videos that overcomes this limitation. It acquires a number of images under varying exposure settings from dark to bright, each containing new scene radiance information. The camera motion between the images is compensated and they are fused into a single HDR frame. For visualization on regular displays, the video frame is tone mapped to the output range of the display. We introduce algorithms for reduced redundancy acquisition, efficient registration and visualization of HDR video that are fast enough to be used in real-time.

Keywords: HDR video, multi-spectrum video acquisition, image registration, video tone mapping

1. Introduction

A recurring problem when capturing videos, e.g., for surveillance purposes, is the scene having a range of brightness values that exceeds the capabilities of the capturing device. An example would be a video camera situated in a bright outside area, directed at the entrance of a building. Because of the potentially big brightness difference, it may not be possible to capture details of the inside of the building and the outside simultaneously using just one shutter speed setting. This results in under- and overexposed pixels in the video footage, impeding the use of pattern recognition algorithms like face

Email address: {guthier, kopf, effelsberg}@informatik.uni-mannheim.de (Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg)

Figure 1: The inside of the building is much darker than the outside. There is no shutter speed setting that exposes both correctly at the same time. A solution to this problem is using a sequence of shutter speeds and merging the images together.

recognition and human tracking. See Figure 1 for an example. 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 [1, 2]. Each frame then captures one facet of the scene's radiance range. When fused together, a high dynamic range (HDR) 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 aligned with respect to each other to compensate for camera and scene motion during capture. The aligned images are then merged 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.

HDR video can be understood as a form of multi-spectrum video. We use the term "multi-spectrum" in a more loosely defined sense here. Originally, it refers to measuring light intensities at specific *wavelength ranges* and combining the measurements. The fact that each wavelength range has different characteristics makes fusing them challenging. Once this is done though, information from all ranges can be harnessed simultaneously. In our

scenario, the wavelengths are restricted to the three color channels red, green and blue. However, we measure the intensity of the light in each spectrum more accurately by combining exposures that cover different *intensity ranges* (e.g., from dark to bright). The problem of differing characteristics between the ranges to be fused remains. For example, areas that contain structure in one image might be completely saturated in another (see Figure 1). The resulting HDR frame includes information from all intensity ranges. Pattern recognition algorithms can later work directly on the fused HDR data and take advantage of the increased bit depth and the improved visibility of detail.

In this paper, we present a system for acquisition, registration and visualization of HDR video. We introduce two separate methods for fast capturing of the LDR sequences required to create an HDR frame. Both aim at reducing the redundancy when capturing multiple images of the same scene. Furthermore, we present an image registration technique that is both robust to extreme brightness differences and fast enough to be used on real-time video. For the visualization of HDR video, we show an extension of existing still-image tone mapping techniques to video. It mainly focuses on the removal of flickering artifacts arising in this situation.

The rest of this paper is structured as follows. Section 2 presents previous work in the field of HDR images and videos. In Section 3, we give an overview of the components of the proposed system. Sections 4, 5 and 6 contain the details of our image capture, image registration, and tone mapping techniques. Their quality and processing time is discussed in Section 7. Section 8 concludes the paper.

2. Related Work

In the last few years, several approaches have been proposed that combine data from different light spectra in video surveillance scenarios. Torresan et al. use a rule-based decision model to fuse data from the infrared and the visible spectrum for pedestrian tracking [3]. A general problem of rule-based approaches is the handling of inconsistency between the two channels. Conaire et al. describe a system for object segmentation by fusing infrared and regular image data [4]. To avoid inconsistencies between the segmented objects in different channels, a "transferable belief model" is used to combine conflicting information. The video surveillance system proposed by Chen and Wolf focuses on object tracking and information fusion of both channels [5].

This system also handles object merge, split, or occlusion by using a hierarchical information fusion approach from pixel level to object level. Kumar et al. use fuzzy logic and Kalman filtering to track objects more reliably [6]. Liu and Laganiere focus on registration techniques to align images from the infrared and the visible spectrum [7]. They are aligned by analyzing the differences in adjacent frames caused by a moving person.

Several of the aforementioned challenges are very similar to high dynamic range video. In case of HDR, all frames are captured in the visible spectrum and the images are more similar to each other than to infrared. Nevertheless, only a small amount of pixels contain useful data when comparing under- and overexposed images, and classical feature based image registration techniques fail. Our system focuses on the special challenges in the context of HDR video. Besides the question of how to fuse inconsistent data, efficient algorithms to capture and process frames are discussed.

The HDR video creation pipeline consists of four steps: capturing, LDR image registration, merging LDR frames into an HDR frame, and tone mapping. Most of the previous work focuses on one step only. Our goal is to combine all steps in an application which is able to handle the data in real-time. Such a combined system allows to use information calculated in the other steps to improve the visual quality and the speed of the overall system.

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 [1, 8, 9, 10]. An obvious disadvantage is the increase of capture time required to record a scene. Alternative approaches use sophisticated hardware like beam splitters that allow an array of LDR cameras to view the same scene at the same time [11, 12]. The shutter speeds of the cameras differ from each other. With multiple cameras, an entire set of LDR images that covers the scene's full dynamic range can be captured at once, leading to high capture speed. The major disadvantage is the significantly increased hardware cost.

Several techniques have been proposed to determine suitable **exposure settings**. Barakat et al. [13] present an approach which minimizes the number of exposures while covering the entire dynamic range of the scene. Minimum and maximum of the scene's irradiance range are taken into account, and the least possible overlap of exposures is chosen. They do not consider the SNR of the HDR result during the choice of exposure times, that is, each pixel is considered to contribute the same amount to the result regardless of

its value. The algorithm is a fast heuristic suitable for real-time use. A recent method to determine noise-optimal exposure settings uses varying gain levels is proposed in [14]. For a given sum of exposure times, increasing gain also increases the SNR. The authors define SNR as a function over log radiance values. However, they only consider the worst-case SNR, i.e., the minimum of the SNR function and ignore the average SNR of the HDR result. Only the extrema of the scene's brightness are considered. The computation of the exposure settings is too expensive to be used in a real-time scenario.

Existing approaches to **image registration** often have difficulties coping with the high brightness difference between LDR exposures [15]. Only few techniques treat this problem specifically. Kang et al. propose a method for estimating camera and scene motion, but its computational cost is too high to be used in real-time [16]. Ward uses thresholded images that are robust to brightness variation and performs an efficient hierarchical search for translational camera motion [17]. In previous work, we have proposed a fast registration algorithm that extends Ward's technique [18].

Tone mapping (TM) operators map radiance values back to suitable 8-bit pixel values for display. They are classified as spatially invariant, global operators and spatially variant, local operators.

Global operators are non-linear functions based on the content of an image as a whole, using statistical values such as average luminance to estimate optimal mapping parameters for a particular image. These operators are simple and fast, but are limited in their ability to process very high dynamic ranges. The histogram adjustment method proposed by Ward et al. [19] applies a monotonic tone reproduction curve to all pixels. The idea is to allocate most of the displayable dynamic range to luminance ranges that are represented by many pixels. Thus pixels in less frequent brightness levels are compressed more strongly. Additionally, human visual limitations such as glare or visual acuity are regarded in further processing steps.

Local operators like the photographic operator published by Reinhard et al. [20] consider a set of neighboring pixels for estimation of the parameters of a transformation function. Each pixel of an image is mapped differently, based on the local features of its neighborhood. Because the human visual system is sensitive to local contrast, high quality images spanning high dynamic ranges are possible with this method. Due to the more complex nature of these operators, computation time increases and artifacts such as halo effects can occur.

Only few TM operators *specific to HDR video content* have been pro-

posed. Benoit et al. [21] propose a model based on properties of the human retina. HDR video content is enhanced by a non-separable spatio-temporal filter with added temporal constancy. This is done by imitating the retina's luminance compression and additional temporal information processing. A general model for temporal luminance adaptation was proposed by Krawczyk et al. [22]. In accordance with the human visual system, that reacts to temporal changes in luminance conditions, a time constant for the speed of the adaptation is introduced. The drawback of existing video tone mapping techniques is that they can only be used with a specific TM operator. We have developed a TM technique for videos which removes flicker in a post-processing step and is applicable to all TM operators [23].

3. HDR Video Processing Pipeline

The process of creating a frame in an HDR video is a pipeline where the output of each step is the input to the subsequent one. This is shown in Figure 2. It consists of four modules: LDR image capture, image registration, HDR stitching, and video tone mapping. We have made isolated contributions to the fields of capturing, registration and tone mapping. In this article, we integrate our previous contributions into a complete HDR video system and tackle the arising challenges. Our additions include:

- GPU-based color conversion of the image material between the steps of the pipeline.
- Calculation of an initial shutter setting for the partial re-exposure module from the average pixel value of the HDR frame created during HDR stitching.
- GPU implementations of the most time-consuming parts of the HDR pipeline and performance evaluation of the system as a whole.
- An alternative approach for capturing image sequences based on optimal shutter speeds (see Section 4.2).

Step 1: LDR Image Capture
The capturing of LDR images constitutes the first step. We present two alternative methods for capturing with reduced redundancy. The decision about which one to use is based on the capabilities of the capturing camera as well as the preferred optimization. The first one minimizes the amount of

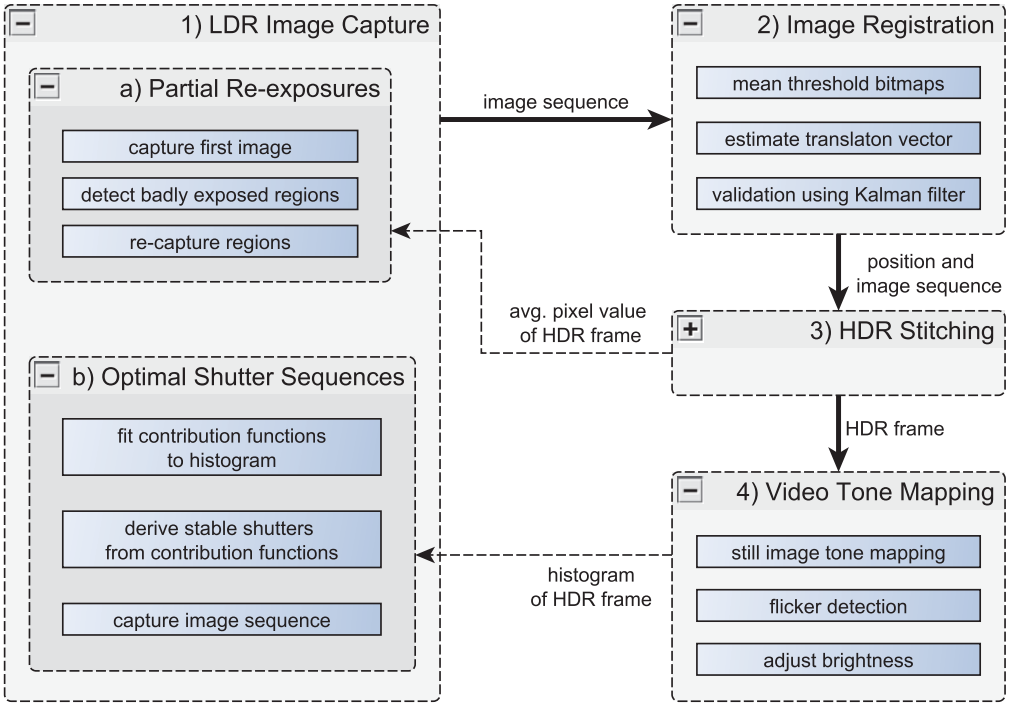

Figure 2: Overview of the HDR video processing pipeline. As a first step, a sequence of LDR images is captured using one of the two presented methods. The image sequence is passed to the registration module where camera motion in the sequence is compensated. The registered sequence is then stitched into 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.

image data by only re-capturing the potentially small badly exposed areas of a base LDR image. These areas are detected during the capturing process and new images are triggered one by one. The second approach presented here has not been published yet. The idea is to only use the shutter speeds that contribute the most information to the HDR frame. In this case, the shutter sequence is determined in advance using the histogram of the previous frame. It is transmitted to the camera which then captures all images in one go.

Step 1 (a): Partial Re-Exposures

Many industrial FireWire CCD cameras have a feature called *true partial scan*. It allows the definition of a rectangular sub-area of cells on the CCD sensor – a region of interest – to be read out while all other cells are being

discarded. As a result, the time needed to read out the relevant parts of the sensor and to transmit the image data over the FireWire bus is reduced, leading to a higher frame rate at lower image sizes.

In our partial re-exposure approach, we do not capture a fixed number of LDR images with varying shutters, but adapt the number to the dynamic range of the scene. Additionally, we make use of the idea that it might not always be necessary to capture a full image at another exposure setting if only few image areas require a higher dynamic range. We developed an algorithm that detects badly exposed regions in an already captured image and triggers the camera to re-capture only these regions. Reducing the image size decreases the overall capture time of an image significantly. Capturing partial images also reduces the amount of redundant data that is used to merge the LDR images into an HDR image which saves additional processing time.

Step 1 (b): Optimal Shutter Sequences
Another way of speeding up capturing is to optimally choose shutter speeds at which to capture. The fewer images are captured, the less time is taken 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. So the goal is to get the most out of the recorded exposures.

In an HDR video, the histogram of scene radiance values is often a by-product of tone mapping the previous frames [19]. This second 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 brightness values are well-exposed in at least one of the captured LDR images. This increases the average signal-to-noise ratio (SNR) for a given number of exposures or minimizes the number of exposures required to achieve a desired SNR.

Step 2: Image Registration
We address the challenge of estimating the camera motion between two partial LDR frames in an efficient way. We argue that a purely translational camera motion model is sufficiently accurate for high frame rates. This assumption is supported by [17]. At 200 frames per second, 5 ms pass between two consecutive LDR frames. Assuming that a visually pleasing camera pan takes 5 seconds to pan across the entire width of a frame, the motion from one exposure to the next is only one thousandth of the frame. Registration

inaccuracy due to a simple translational model is not visible on such a small scale. Local object motion, however, is not accounted for in our global model. Object motion in the context of HDR has been determined in [16] by estimating the optical flow in an offline process. It is too slow to be performed in real-time.

The goal of the registration algorithm is thus to estimate a translation vector between two LDR frames captured at different exposure settings. We improve upon the approach based on mean threshold bitmaps [17]. The hierarchical 2D search is replaced by two separate exhaustive 1D searches to speed up the computation. We start by counting the number of dark pixels in each column of both frames to be aligned to create column histograms. By using a normalized cross correlation between the two column histograms, we estimate the horizontal component of the translation vector. Repeating this process for image rows allows us to estimate the vertical component, respectively. The resulting vector is then validated using a Kalman filter to incorporate knowledge of the prior motion into the estimation.

Step 3: HDR Stitching
The registered image sequence is then merged into a single frame. The weighting functions shown here are used again in Section 4.2 to determine optimal shutter speeds.

An HDR frame is a map of radiances in a scene. In order to reconstruct this radiance map from the pixel values of the captured LDR images, the camera's response function f must be known [1]. For the duration Δt that the camera's shutter is open, a pixel on the CCD sensor integrates the scene radiance E, resulting in a total exposure of $E\Delta t$. The camera's response function then maps the exposure to a pixel value $I = f(E\Delta t)$, usually in the range of $[0, 255]$. When the shutter speeds Δt_i used to capture the LDR images are known, the inverse of the response function can be used to make an estimate $\tilde{E}_i$ of the original radiance from pixel value I_i in LDR image i:

$$\tilde{E}_i = \frac{f^{-1}(I_i)}{\Delta t_i}. \tag{1}$$

A good approximation of the radiance value at a pixel in the HDR image is then obtained by computing a weighted average over all estimates $\tilde{E}_i$:

$$E = \frac{\sum_i w(I_i)\tilde{E}_i}{\sum_i w(I_i)}. \tag{2}$$

The weighting function w determines how much the radiance estimate $\tilde{E}_i$ from a pixel I_i contributes to the corresponding HDR pixel E. In other words, it judges a pixel's usefulness for recovering a radiance value based on its brightness value.

Step 4: Video Tone Mapping
In order to be displayable on a regular screen, the large radiance ranges of an HDR frame need to be compressed to 8-bit values. 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. Flicker is detected by large changes in the average image brightness from one tone mapped frame to the next. To reduce flicker, we adjust the image brightness after tone mapping by normalization and clamping. The brightness variation is smoothed over several frames, becoming less disturbing. The advantage of this approach is that it is applicable to all tone mapping operators.

4. LDR Image Capture

4.1. Capturing with Partial Re-exposures

When capturing low dynamic range (LDR) sequences with a camera that allows to read out a smaller region of interest (ROI) on the image sensor, the size of the region influences the capture speed. Increasing the height of the ROI leads to a linear increase in capture time. This is obvious because CCD sensors are usually read out row by row, while rows that are not to be captured are discarded completely. Contrary to this, no time can be saved by decreasing the ROI width because the read-out time of a row on the sensor is constant.

Parts of the costs of capturing an image are constant for all ROI sizes, e.g., triggering the camera and exposing the sensor to the light, while others depend linearly on the height of the ROI. The constant cost of image capturing can exceed the variable cost for small ROI heights by far. Instead of capturing two close but distinct ROIs, it can therefore be more efficient to capture both regions and the area in between in one step.

Our algorithm to capture HDR frames using partial re-exposures of poorly exposed regions can be divided into the following steps (see Figure 3):

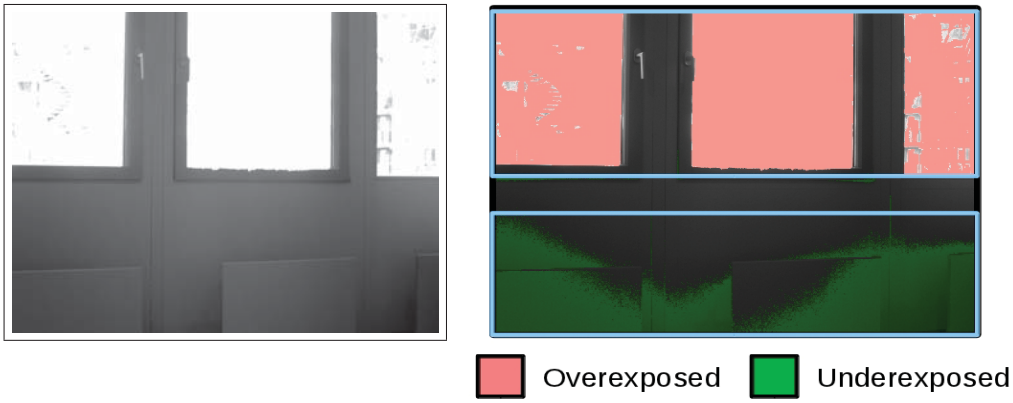

Figure 3: The left image shows the base image of an LDR image sequence. Some areas of the base image are badly exposed. Only the rectangular ROIs are captured again with a shorter (top) and longer (bottom) shutter speed.

1. Capture a base image of the scene at full resolution and an initial shutter setting,
2. Search the captured image for under- or overexposed pixels,
3. Group these pixels into ROIs for re-exposure and determine an appropriate shutter speed setting,
4. Re-Capture all ROIs from the previous step with different shutter settings and repeat from 2 using each newly captured image,
5. Stop if no more under- or overexposed regions are found.

The initial shutter setting to capture the base image is determined from the average radiance of the previous HDR frame. It is chosen such that the average radiance is mapped to the center of the pixel value range (e.g., 128). The algorithm explores the base image and all subsequently captured partial images iteratively and captures only as many images as necessary to cover the full dynamic range of the scene. In order to search captured images for under- or overexposed pixels, we define that a pixel is *valid* (well exposed) if its brightness value p lies within an interval $[p_{min}, p_{max}]$ and is *invalid* otherwise.

No performance gain can be achieved by capturing images at less than full width. We therefore restrict the set of possible ROIs to those with a width equal to the full width of the CCD sensor. Such a region is fully described

by the location of the first row belonging to the ROI and its height. Thus, as a first step in determining areas for re-exposure, a histogram is created with as many bins as the number of rows in the image to be considered. Each bin stores the number of invalid pixels found in its corresponding image row.

From now on, only row histograms counting invalid pixels are considered, reducing the problem of finding ROIs to a one-dimensional one. A threshold r_{max} is then applied to the smoothed histogram, marking those image rows having an invalid pixel count of more than r_{max} percent. Marked rows in the histogram are the ones to be considered for re-exposure.

Next, the thresholded row histogram is searched for contiguous runs of marked rows. They are expanded to a minimum size of h_{min} rows, which is a characteristic of the camera used. When two ROIs are close enough together for it to be faster to capture both at once, they are merged into a single region. Lastly, the detected ROIs are pushed into the image capture queue. Depending on whether the image was analyzed for under- or over-exposed pixels, the regions will be re-exposed with either longer or shorter shutter speeds respectively. In this approach, we vary the shutter speeds by a constant factor between an exposure and a partial re-exposure. As soon as no more invalid pixels are found in any of the newly captured images, the algorithm terminates. Therefore, no more exposures than necessary to capture the scene's dynamic range are used.

From the moment the camera's sensor is exposed to the light of the scene, until the image is fully received from the camera, the CPU is idle. We can use this idle CPU time to analyze the captured images without adding to the overall capture time. We found that in our setup, capturing even the smallest possible image took longer than analyzing a full image. As long as there are more images to re-expose, the analysis can thus be performed for free and does not add to the overall capture time.

4.2. Capturing with Optimal Shutter Sequences

In this section, we present the alternative capturing approach that makes use of the camera's *sequence mode*. A shutter sequence is determined and transmitted to the camera, which then captures the image sequence asynchronously. This approach makes use of the weighting functions introduced in Section 3 (see Figure 4 for an example).

For a given shutter speed Δt, we can calculate how well a radiance value E can be estimated from an image captured at Δt by combining the response and the weighting function. A radiance value E is mapped to a pixel value

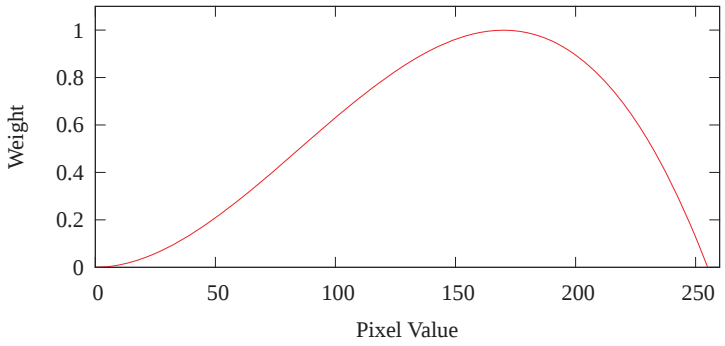

Figure 4: Example of a weighting function. The weight of a pixel is its value multiplied by a hat function normalized to a maximum weight of 1.

using the camera's response function f. The weighting function w then assigns a weighting to the pixel value. We define

$$c_{\Delta t}(E) = w(f(E\Delta t)) \tag{3}$$

as the *contribution* of an image captured at Δt to the estimation of a radiance value E.

When creating HDR video in real-time, the scene's radiance histogram is known from the previous frames, e.g., as a by-product of tone mapping [19]. Each histogram bin with index $j = 1, ..., M$ counts the number $H(j)$ of pixels in the HDR image having a log radiance of $b_j = log(E_j)$. A log radiance histogram can be used to calculate a sequence of shutter speeds Δt_i which allows the most accurate estimation of the scene's radiance. We do this by choosing the 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 5.

Equation 3 states that, for a given shutter speed Δt and an LDR image captured using Δt, the value of $c_{\Delta t}(exp(b_j))$ indicates how accurately log radiance b_j is represented in the LDR image. The continuous contribution function sampled at b_j results in a discrete vector of contribution values. The contribution vector corresponding to a different shutter speed $\Delta t'$ can

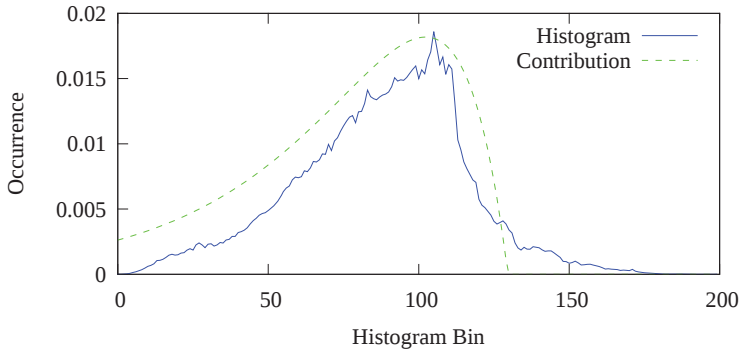

Figure 5: The solid line depicts an example log radiance histogram. The dashed line is the contribution function in the log domain corresponding to the first shutter speed chosen by our algorithm. The exposure was chosen such that it captures the most frequently occurring radiance values best.

be easily obtained by shifting the original vector 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.

Here, we explain how a new shutter speed is added to an existing shutter sequence. The first shutter can be determined analogously. We assume that the sequence already consists of a number of shutter speeds Δt_i. To each Δt_i belongs a contribution vector $c_{\Delta t_i}(E_j)$. To find out which new shutter brings the biggest gain in image quality, we define a *combined contribution vector* $C(E_j)$ that expresses how well the radiances E_j are captured in the determined exposures. We define it as the maximum contribution for each histogram bin

$$C(E_j) = \max_i \left(c_{\Delta t_i} \left(E_j \right) \right). \tag{4}$$

This definition can now be used to calculate a single *coverage value* C to estimate how well-exposed the pixels in the scene are in the exposures. C is obtained by multiplying the frequency of occurrence of a radiance value $H(j)$ by the combined contribution $C(E_j)$ and summing up the products:

$$C = \sum_{j=1}^{M} C(E_j) H(j). \tag{5}$$

The algorithm tries out all possible shifts between a new contribution vector and the log histogram. The shutter speed corresponding to the shift that leads to the biggest increase of C is added to the sequence.

We stop adding shutters to the sequence once one of three stop criteria is met:

1. A maximum number of exposures is reached,
2. the coverage value C is above a threshold, meaning that scene radiance can be estimated sufficiently well from the exposure sequence, or
3. the sum of shutter speeds exceeds the time available for exposure in a video frame.

Changing the shutter sequence for every frame when creating 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. Like this, small variations in the shutter speed sequence are ignored.

5. Image Registration

This section describes our histogram-based algorithm for image registration. The input to our algorithm is a set of n exposures consisting of one full resolution base frame I_0 and possibly smaller re-exposures I_i for $i = 1, ..., n-1$ captured at different exposure settings. Each re-exposure was initiated by badly exposed regions in an already captured parent frame. The base frame is the root of the whole set. The output of image registration is a two-dimensional integer translation vector $\vec{v}_i$ describing the shift between each

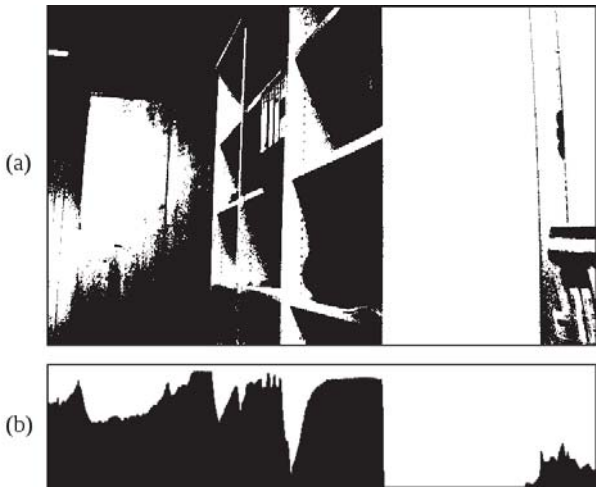

Figure 6: Mean Threshold Bitmap of an LDR frame (a) and its corresponding column histogram counting black pixels (b).

exposure I_i and its parent I_{i-1}. Our algorithm performs no image registration on the base frame of an exposure set.

For estimating the translation vectors, we use *mean threshold bitmaps* (MTB) as described in [17]. 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. The threshold m_i that achieves this ratio is calculated from the brightness histogram. 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.

We estimate a two-dimensional shift $\vec{v}_i = (x_i, y_i)$ between two exposures I_{i-1} and I_i by estimating two one-dimensional shifts x_i and y_i separately. We start by estimating the horizontal shift x_i. The first step in doing so is to build column histograms over the full image I_i and the overlapping image area of I_{i-1}. A bin in the column histogram represents the number of black pixels in the corresponding column of the exposure's MTB. This is demonstrated in Figure 6.

Let w_i and h_i be the width and height of I_i. The column histogram $B_i^x(j)$ of exposure I_i counting black pixels is a function of the column index

$j = 1, ..., w_i$ and is defined as

$$B_i^x(j) = |\{I_i(j,k) < m_i \ ; \ k = 1, ..., h_i\}| \tag{6}$$

where $I_i(j,k)$ is the pixel value at position (j,k) and $|\cdot|$ denotes the number of elements in the set. The histogram for I_{i-1} is defined accordingly.

The horizontal shift x_i is now estimated using these two histograms. We let the shift s assume all possible integer values within a search range (e.g., -64 to 64 pixels) and compute the *normalized cross correlation* (NCC) between the histograms of exposures I_{i-1} and I_i under the given shift. The s producing the highest correlation value is then used as the estimate for x_i. Using row histograms, the vertical shift y_i can be estimated analogously. Our experiments show that the choice of which dimension to start with and the number of iterations have little effect on the final result. We therefore only estimate x_i and y_i once and set $\vec{v}_i = (x_i, y_i)$ as the resulting translation vector. In addition to NCC, we also experimented with different metrics for comparing histograms. The advantage of simpler ones like the sum of absolute/squared differences and histogram intersection is their lower computational complexity. We found, however, that NCC achieves the best results. The computational effort of calculating the NCC between two histograms with at most 640 bins is negligible compared to the computation of brightness, row and column histograms over a full image.

As a last step, all resulting vectors are validated using a Kalman filter to incorporate knowledge of the prior motion 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.

The computation of the brightness histogram to determine the median threshold and the creation of row and column histograms are the most time-consuming steps in our algorithm. We thus implemented them in a parallel way to be executed on a graphics processing unit (GPU). The image is first subdivided into rectangular areas of size 32×64 pixels. A separate histogram is created for each area in parallel. All histograms are then successively added up to one final histogram over the entire image.

6. Video Tone Mapping

When using existing still image tone mapping operators to visualize HDR videos, temporal incoherence of the minimum, maximum, or average scene

Figure 7: Three tone mapped frames of an example video. As soon as the window enters the camera's field of view, the scene's minimum and maximum luminance changes greatly, leading to a visible brightness difference of the tone mapped frames.

radiance leads to image flicker. An example would be an HDR video with a camera turn from a dark indoor area towards a window showing a light outdoor scene. The tone mapping operator now attempts to map the suddenly increased radiance range to the same output values, leading to an overall much darker image. When this transition from light to dark happens too quickly, it is perceived as flicker. This is illustrated in Figure 7.

Such flicker artifacts are sufficiently well detected by computing the geometric average image brightness of a tone mapped frame and comparing it to the average of the previous frame. The biggest challenge here is finding a suitable threshold for the difference of the averages of two consecutive frames. We made use of a model found in the literature on the human visual system called Stevens' power law [24]. It uses the notion of a *just noticeable difference* ΔR, which depends on a given background luminance R, and introduces an adjustable parameter k. For a given luminance level (in our case the log average of the previous frame), this model allows the computation of a maximum luminance change that will remain unnoticed to a human observer. Even though the setting for which this law was developed slightly differs from ours, it serves as a perceptional basis for our criterion. In its general form, it is given by

$$\Delta R = kR^{\alpha}, \tag{7}$$

where $\alpha \approx 0.33$ when considering brightness. A suitable value for the parameter k was determined by us experimentally. ΔR is then used as threshold for the geometric average image brightness to detect flicker.

A robust flicker detection makes flicker removal straightforward. If flicker occurs in a frame, we iteratively adjust its brightness until it is within the tolerable threshold. We use the example at the beginning of this section to

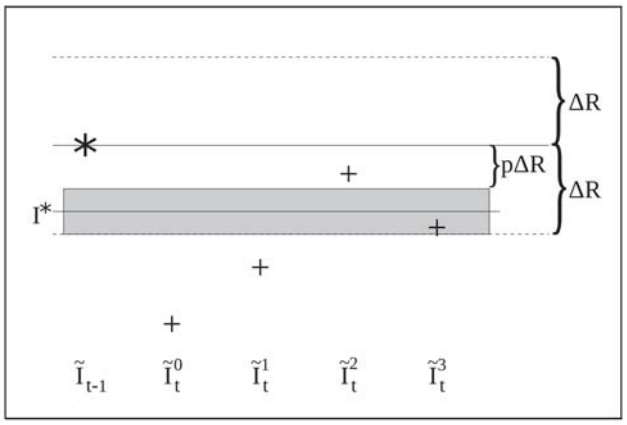

Figure 8: Frame t is too dark after tone mapping ($\tilde{I}_t^0 < \tilde{I}_{t-1} - \Delta R$). Its brightness is thus iteratively adjusted towards the target value I^*. After three iterations, it falls within the tolerable brightness range drawn in gray.

explain flicker removal. The algorithm is implemented as a post-processing step and works with any tone mapper. We start by tone mapping the current frame t with the chosen operator and settings. Next, the log average pixel value $\tilde{I}_t$ of the frame is computed. Then we calculate the maximum allowable brightness difference ΔR to the previous frame using Stevens' power law (Equation 7):

$$\Delta R = k(\tilde{I}_{t-1})^{0.33}, \tag{8}$$

where $\tilde{I}_{t-1}$ is the log average of the previous frame. Now, we check whether $|\tilde{I}_{t-1} - \tilde{I}_t| > \Delta R$. If it is not, then the frame is likely not to be a flickering frame. In our example however, it is assumed that the current frame is much darker than the previous one ($\tilde{I}_{t-1} - \Delta R > \tilde{I}_t$). The goal is now to increase the frame's brightness so that it falls within a tolerable range. The lower end of this range is given by $\tilde{I}_{t-1} - \Delta R$, meaning that there shall be no detectable flicker. It is also desirable to maintain the original brightness produced by the TM operator as well as possible. After adjustment, $\tilde{I}_t$ should therefore be close to the lower end of the range. To accommodate this fact, we set the upper bound to $\tilde{I}_{t-1} - p\Delta R$, where p is a percentage we set to 50% in our implementation. The next step is to iteratively adjust the frame's brightness, producing a sequence $\tilde{I}_t^0, \tilde{I}_t^1, ..., \tilde{I}_t^i$, until it falls into the desired

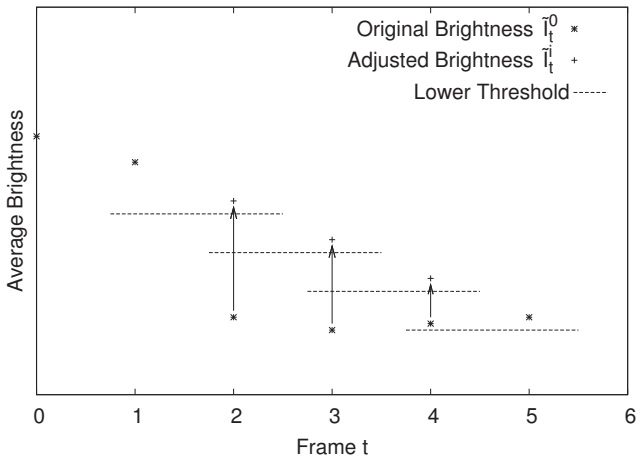

Figure 9: This plot shows a rapid decrease of average brightness between frames 1 and 2. Frame 2 is adjusted to reduce the brightness gap. The amount of adjustment needed decreases with each subsequent frame. In frame 5, the adjusted brightness has converged towards the value achieved by tone mapping with standard parameters. No more adjustment is required.

range of $[\tilde{I}_{t-1} - \Delta R, \tilde{I}_{t-1} - p\Delta R]$. As an explicit target value I^*, we aim for the range's center. The process of iteratively approaching the desired brightness is depicted in Figure 8.

The adjusted brightness $\tilde{I}_t^i$ of frame t is now slightly lower than $\tilde{I}_{t-1}$ (more specifically: $\tilde{I}_t^i \approx I^* < \tilde{I}_{t-1}$), and the difference is within a range we consider to be unobtrusive. The next frame $t+1$ is then tone mapped with standard parameters again. If there is no more rapid scene histogram change between frames t and $t+1$, $\tilde{I}_{t+1}$ is now closer to $\tilde{I}_t$ than $\tilde{I}_t$ was to $\tilde{I}_{t-1}$ and the amount of adjustment required is smaller. After a few frames, the difference approaches a value less than ΔR and no further adjustment is necessary. Figure 9 illustrates this convergence towards standard parameters.

7. Evaluation

In this section, we evaluate the presented algorithms. It is split up into two parts. The first discusses the achieved quality of the algorithms according to our metrics. In the second part, we evaluate the processing times of the presented subparts.

For all experiments, we used a desktop PC with an AMD Athlon II X2 250 64-bit CPU with two cores running at 3 GHz and a total of 4 GB of RAM. The installed graphics card is an Nvidia GeForce GTX 480 with 15 multicores running at a clock rate of 1.4 GHz and 1.5 GB of dedicated memory. Each multicore can process 32 threads at once. Our camera is an AVT Pike F-032C FireWire camera capable of capturing 208 frames per second in VGA resolution. It can capture in the sequence mode and uses a Bayer color filter array to acquire color images. Unless stated otherwise, we use five HDR video sequences in VGA resolution for our evaluation. Each is about 10 seconds long. The LDR base image sequences are stored for each frame.

7.1. Quality of the Results

When capturing LDR sequences with *partial re-exposures*, we leave out redundant areas that are already well exposed in a previously captured image. However, we only consider an image row for re-exposure, when it contains more than r_{max} badly exposed pixels. Depending on the choice for this threshold, a certain percentage of pixels remain invalid in the final HDR frame. When setting r_{max} to 0%, 0.7%, 5% and 10% of the image width, the percentages of invalid pixels in the HDR frame are 0%, 0.03%, 0.66% and 3.73% respectively. We chose $r_{max} = 0.7\%$ for our running system.

To evaluate the quality of results produced by using our *optimal shutter speed sequences*, we performed a subjective user study. We believe that a subjective evaluation is superior to an objective one, since the HDR images created from our optimized sequences are targeted at human observers. To our knowledge, there exists no objective metric that is specific to judging the quality of an HDR image. Comparing the results to a perfect reference HDR image by using standard metrics like the PSNR is heavily biased in favor of underexposure. An underexposed image exhibits quantization noise with pixel values differing only by small amounts from the reference. The true brightness of a saturated pixel however can be arbitrarily large. A reflection of the sun appearing as a small white disk might be acceptable, even though the displayed brightness is orders of magnitude lower than the real one.

In our subjective study, the 27 participants 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 had to be rated using five scores ranging 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. Note that

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.

the HDR material was intended to be flawed for better comparison. 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 the sequences 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 [17]. By incorporating knowledge of the motion in the previous frames, our algorithm thus achieves a better registration accuracy. Table 1 shows this result.

The HDR test sequences were tone mapped with three different operators [25, 19, 20] without using our proposed *flicker reduction*. In a subjective user study with 10 participants, a total of 50 frames were marked as flickering by 50% or more of the participants. From the marked frames, we determined the value for the adjustable parameter k in Stevens' power law. We set the parameter in a way such that the power law produced only one false negative (missed flicker frame) and 87 false positives (erroneously detected flicker). False positives are acceptable, since additional tone mapping of a non-flicker frame merely increases the computational effort. From the number of false negatives and false positives for our choice of k, we can conclude that exceeding the threshold is a *necessary criterion* for a flickering frame. That is, if our detector classifies a frame as non-flickering, it is very likely to actually be a non-flicker frame. Adjusting a frame's brightness until our detector stops reporting flicker thus removes flicker with a high likelihood.

7.2. Processing Time

We've conducted experiments to compare the time taken to capture an LDR sequence using our *partial re-exposure* algorithm to the time of the traditional approach of creating HDR images using full images. Averaged over all sequences, our approach saved 29% of the total capturing time. The amount of saving varied from 49% in a scene with only one small saturated light source to 20% in a scene with large reflecting surfaces. Throughout all test sequences, the overhead introduced by image analysis accounts for approximately 5% of the overall duration.

When computing *optimal shutter sequences*, the histogram of the previous HDR frame is available from tone mapping. Histogram creation is thus not included in the timing measurements. Our experiment showed that 96.5% of our shutter speed algorithm's processing time is spent for trying out all possible shifts between contribution vector and histogram to find the next shutter speed with the best coverage value. As a consequence, the processing time is roughly proportional to the number of shutters in the sequence. We measured 0.14 ms per shutter value.

The time taken to *register two images* of the LDR sequence depends linearly on the size of the smallest image. We always capture at full image width. Varying the image height from 100 to 480 pixels resulted in processing times from 2 to 8ms on a CPU. The overall processing time was decreased by a factor of 5.9 in the average by running parts of the algorithm on a GPU.

For judging the computational effort of our *flicker reduction algorithm*, we assume that the log average brightness of a frame is obtained from the tone mapping operator as a by-product. This is the case for our GPU implementation of the tone mapper presented in [19]. The cost of flicker detection is thus negligible. Hence, the additional computational effort produced by our flicker reduction algorithm is mainly due to the repeated normalization of flickering frames. The brightness of 4.16 frames was adjusted in the average to smooth the brightness variance of each detected flicker frame. Each adjustment took 1.317 iterations of normalization until the desired target brightness was met.

8. Conclusions

We presented a system for acquisition, registration, and visualization of high dynamic range videos. Each HDR video frame is created using a sequence of LDR images. A combined HDR frame then contains more infor-

mation than each individual exposure. The improved HDR algorithms suit the needs of an HDR video system like high frame rates and temporal coherence. They benefit from knowledge obtained from previous frames. We showed that in the average, capturing with partial re-exposures saves 29% of the time for exposure acquisition compared to capturing full images while only introducing 0.03% of under- or overexposed pixels into the HDR result. Adapting shutter speeds to the scene instead of simply using equidistant shutters gave better subjective quality in 70% and the same quality in 16% of the cases. In our test sequences, the still-image registration algorithm introduced in [17] produced an average error of 2.0 pixels. Our approach is simpler and more efficient to compute, but achieves a lower error of 1.4 pixels by making use of prior knowledge in a video. Its average processing time was decreased by a factor of 5.9 by running parts of the algorithm on a GPU. 49 out of 50 flickering frames were detected and removed by our video tone mapping techniques. With these improvements, our system makes HDR video possible for real-time applications.

References

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

[2] B. Guthier, S. Kopf, W. Effelsberg, Capturing high dynamic range images with partial re-exposures, in: Proc. of the 10th IEEE International Workshop on Multimedia Signal Processing (MMSP), 2008, pp. 241–246.

[3] H. Torresan, B. Turgeon, C. Ibarra-Castanedo, P. Hebert, X. P. Maldague, Advanced surveillance systems: combining video and thermal imagery for pedestrian detection, in: Proceedings of SPIE on Thermosense, Vol. 5405, SPIE, 2004, pp. 506–515.

[4] C. Conaire, N. O'Connor, E. Cooke, A. Smeaton, Multispectral object segmentation and retrieval in surveillance video, in: IEEE International Conference on Image Processing, 2006, pp. 2381–2384.

[5] C.-Y. Chen, W. Wolf, Background modeling and object tracking using multi-spectral sensors, in: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, VSSN '06, ACM, New York, NY, USA, 2006, pp. 27–34.

[6] P. Kumar, A. Mittal, P. Kumar, Fusion of thermal infrared and visible spectrum video for robust surveillance, in: P. Kalra, S. Peleg (Eds.), Computer Vision, Graphics and Image Processing, Vol. 4338 of Lecture Notes in Computer Science, Springer Heidelberg, 2006, pp. 528–539.

[7] Z. Liu, R. Laganiere, Registration of IR and EO Video Sequences based on Frame Difference, in: Fourth Canadian Conference on Computer and Robot Vision (CRV), 2007, pp. 459–464.

[8] S. Mann, R. Picard, Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures, in: Proc. of IS&T 48th Annual Conference, 1995, pp. 422–428.

[9] T. Mitsunaga, S. K. Nayar, Radiometric self calibration, in: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., Vol. 1, 1999, pp. 374–380.

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

[11] M. Aggarwal, N. Ahuja, Split Aperture Imaging for High Dynamic Range, International Journal of Computer Vision 58 (1) (2004) 7–17.

[12] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, High performance imaging using large camera arrays, ACM Transactions on Graphics (TOG) 24 (3) (2005) 765–776.

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

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

[15] B. Guthier, S. Kopf, W. Effelsberg, High-resolution inline video-AOI for printed circuit assemblies, in: Proc. of IS&T/SPIE Electronic Imaging (EI) on Image Processing: Machine Vision Applications II, Vol. 7251, 2009, pp. 725104:01 – 725104:12.

[16] S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High dynamic range video, ACM Transactions on Graphics (TOG) 22 (3) (2003) 319 – 325.

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

[18] B. Guthier, S. Kopf, 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), 2010, pp. 145 – 148.

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

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

[21] A. Benoit, D. Alleysson, J. Herault, P. Callet, Spatio-temporal Tone Mapping Operator Based on a Retina Model, Springer, Berlin, Heidelberg, 2009, Ch. 2, pp. 12–22.

[22] G. Krawczyk, K. Myszkowski, D. Brosch, HDR Tone Mapping, Springer Series in Advanced Microelectronics (26), Springer, Heidelberg, 2007, Ch. 11, pp. 147–178.

[23] B. Guthier, S. Kopf, M. Eble, 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, Vol. 7866, 2011, pp. 78660C:01 – 78660C:15.

[24] J. C. Stevens, S. S. Stevens, Brightness function: Effects of adaptation, Journal of the Optical Society of America 53 (3) (1963) 375–385.

[25] G. Ward, A contrast-based scalefactor for luminance display, Graphics Gems IV (1994) 415–421.