OPTIMAL SHUTTER SPEED SEQUENCES FOR REAL-TIME HDR VIDEO

Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg

{guthier, kopf, effelsberg}@informatik.uni-mannheim.de
University of Mannheim, Germany

ABSTRACT

A technique to create High Dynamic Range (HDR) video frames is to capture Low Dynamic Range (LDR) images at varying shutter speeds. They are then merged into a single image covering the entire brightness range of the scene. While shutter speeds are often chosen to vary by a constant factor such as one stop, we propose an adaptive approach. The scene's histogram together with functions judging the contribution of an LDR exposure to the HDR result are used to compute a sequence of shutter speeds. This sequence allows for the estimation of the scene's radiance map with a high degree of accuracy. We show that, in comparison to the traditional approach, our algorithm achieves a higher quality of the HDR image for the same number of captured LDR exposures. Our algorithm is suited for creating HDR videos of scenes with varying brightness conditions in real-time, which applications like video surveillance benefit from.

Index Terms— HDR Video, Shutter Speed, Video Surveillance

1. INTRODUCTION

A recurring problem in video surveillance is the monitored scene having a range of brightness values that exceeds the capabilities of the capturing device. An example would be a video camera mounted 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 algorithms for face 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 LDR images captured in quick sequence at different shutter settings [1, 2]. Each LDR image then captures one facet of the scene's brightness range. When fused together, an HDR video frame is created that reveals details in dark and bright regions simultaneously.

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 LDR images using varying exposure settings, e.g., shutter speed or gain. 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 brightness values of the entire scene. As a last step, the HDR frame is tone mapped in order to be displayable on a regular LDR screen.

In a video surveillance scenario, all these steps must be performed in real-time. One way of speeding up the entire process is to only capture as few LDR images as necessary, that is, to optimally
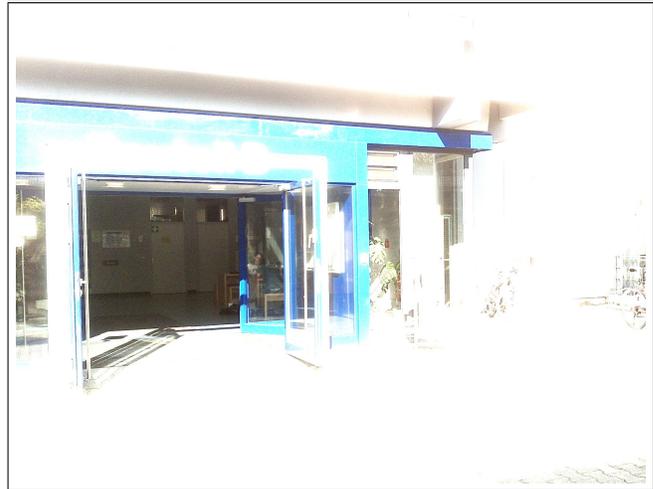
Fig. 1. The inside of the building is much darker than the outside. There is no shutter speed setting that exposes both correctly at the same time. A solution to this problem is using a sequence of shutter speeds and merging the images together.

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 monitored 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 the surveillance example above, where a camera is pointed at the entrance of a building, it may be a sensible choice to use one long shutter speed that suits the inside of the building and another shorter one adjusted to the outside. This

way, the whole scene can be covered with just two carefully chosen shutter speeds. Such a choice can only be made if the scene's brightness histogram is considered.

Barakat et al. [3] focus entirely on minimizing the number of exposures while covering the entire dynamic range of the scene. Only minimum and maximum of the scene's irradiance range are taken into account, and the least possible overlap of exposures is always 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.

In [4], the authors use a model of shot noise to determine the sequence of shutter values producing the highest SNR for a given number of exposures. The shutter speeds are obtained by solving a constrained optimization problem. For this purpose, a coarse approximation of the scene irradiance histogram is used. However, the computation is too costly to be done on-line. The authors do not employ a pixel weighting scheme, but always use the brightest pixel before saturation.

An approach to emulate an effective camera with a given response function and dynamic range was published in [5]. In an off-line process, a static table of exposure times is created that spans the desired dynamic range. The static table prevents adaptation to changes in scene brightness distribution, for example when large reflective surfaces like cars appear.

A very recent method to determine noise-optimal exposure settings uses varying gain levels [6]. 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. Again, computation of the exposure settings is too expensive to be used in a real-time scenario.

The authors of [7] developed a theoretical model for photons arriving at a pixel by estimating the parameters of a Gamma distribution. From the model, exposure values are chosen that maximize a criterion for recoverability of the radiance map. The focus lies on the impact of saturated pixels on the HDR result.

In [8], an algorithm for estimating optimal exposure parameters from a single image is presented. The brightness of saturated pixels is estimated from the unsaturated surrounding. Using this estimation, the expected quality of the rendered HDR image for a given exposure time is calculated. The exposures leading to the lowest rendering error are chosen.

In an HDR video, the histogram of scene brightness values is often a by-product of tone mapping the previous frames [9]. The novel approach we present in this paper thus uses the entire 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 SNR for a given number of exposures or minimizes the number of exposures required to achieve a desired SNR. We also give our definition of *contribution functions* to specify precisely what we mean by "well-exposed". An image pixel is a noisy measurement of physical radiance. The quality of this measurement is a function of the pixel value, with higher values generally leading to a more accurate measurement. This circumstance is modeled by our contribution functions. It is a concept similar to the noise models used in other methods. In order to be applicable to video, we consider bootstrapping and convergence to a stable shutter sequence. Additionally, we introduce a stability criterion for the shutter speeds to prevent flicker in the video.

Our main contributions presented in this paper are:

- A real-time algorithm for computing shutter speed sequences according to the scene's histogram,
- an increase in quality of the HDR result for the same number of exposures,
- bootstrapping and temporal smoothing of the shutter speed sequences for the use in HDR video, and
- contribution functions and their relationship to log brightness histograms to estimate "well-exposedness".

In the following section, we introduce weighting functions for LDR pixels and give our definition of *contribution functions* as a means of judging an exposure's impact on the HDR result. Section 3 then defines log radiance histograms and demonstrates a useful relationship between them and contribution functions which is exploited by our algorithm. The algorithm for finding optimal shutter speed sequences itself is described in Section 4. The quality of the HDR images produced by our optimal shutter sequences and the computational cost are analyzed in Section 5 of this paper. Section 6 concludes the paper.

2. WEIGHTING FUNCTIONS

An HDR image is a map of radiances contained 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:

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

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 $\tilde{E}_i$:

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

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. Note that without prior calibration, radiance values E computed like this only represent physical quantities up to an unknown scale factor. This is sufficient for our purpose. We thus use the terms *radiance* and *scaled radiance* interchangeably to denote the pixel values of an HDR frame.

Weighting functions are usually chosen to reflect noise characteristics of a camera, the derivative of its response function (i.e., the camera's sensitivity), and saturation effects. They are often found in the literature as parts of HDR creation techniques [1, 10, 11]. Even though various weighting functions exist, they often share a few common properties. Most notably, the extremes of the pixel range are always assigned zero weight. This means that pixels with these values contain no useful information about the real radiance. As an example, a white sheet of paper and a reflection of the sun in a window can – under certain exposure settings – both be represented by a pixel value of 255, even though the sun is several orders of magnitude brighter than the paper. The same reasoning applies to very
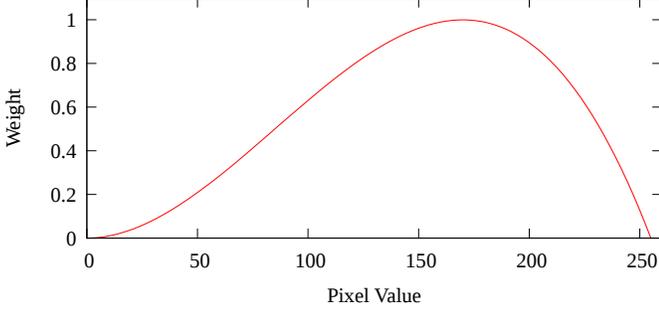
Fig. 2. The weighting function we use in our experiments. The weight of a pixel is its value multiplied by a hat function normalized to a maximum weight of 1.

dark pixels. Another common attribute of weighting functions is the location of their maximum. Pixels with a medium to high value are considered to be more faithful than dark pixels. This is due to the fact that a large portion of the image noise (e.g., quantization noise, fixed pattern noise) is independent of the amount of light falling onto the pixel. A bright pixel thus has a better signal-to-noise ratio than a dark one. Figure 2 shows an exemplary weighting function. In our experiments, we found that the function shown in the plot gives the best results, but our approach also works for any other choice.

For a given shutter speed Δt, we can thus 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 using the camera's response function f. The weighting function w then assigns a weighting to the pixel value. We define

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

as the *contribution* of an image captured at Δt to the estimation of a radiance value E. In the special case of a linear response function, $c_{\Delta t}$ looks like a shifted and scaled version of w. An example for a contribution function in the log domain is shown in Figure 4.

3. LOG RADIANCE HISTOGRAMS

When creating HDR videos in real-time, the scene's brightness distribution is known from the previous frames. Additionally, some tone mapping operators create histograms of scene radiance values as a by-product or can be modified to create them with little extra effort [9]. In this section, we describe how 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 Figures 3 and 4.

The histogram over the logarithm of scene radiance has M bins. Each bin with index $j = 1, ..., M$ corresponds to the logarithm of a discrete radiance value: $b_j = log(E_j)$. Bin j counts the number $H(j)$ of pixels in the HDR image having a log radiance of b_j. The bins have even spacing in the log domain, meaning that for any j, the log radiance values b_j and b_{j+1} of two neighboring bins differ by a constant $\Delta b = b_{j+1} - b_j$. The non-logarithmic radiance values corresponding to two neighboring bins thus differ by a constant

Fig. 3. Example of a tone mapped HDR image.
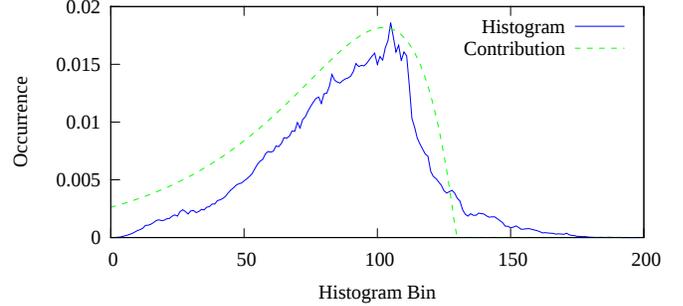
Fig. 4. The solid line depicts the log radiance histogram of our example scene (Figure 3). The dashed line is the contribution function in the log domain corresponding to the first shutter speed chosen by our algorithm. The exposure was chosen such that it captures the most frequently occurring radiance values best.

factor $exp(\Delta b) = exp(b_{j+1})/exp(b_j) = E_{j+1}/E_j$.

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. When considering log radiance histograms, the continuous contribution function is reduced to a discrete vector of contribution values. It has one contribution value for each radiance interval of the histogram. We can now exploit a useful relationship between the log radiance histogram and our contribution vector: Shifting the contribution vector by a number of s bins leads to

$$
\begin{aligned}
& c_{\Delta t}(exp(b_j + s\Delta b)) \\
= \; & w(f(exp(b_j + s\Delta b)\Delta t)) \\
= \; & w(f(exp(b_j)exp(\Delta b)^s\Delta t)) \\
= \; & w(f(exp(b_j)\Delta t')) \\
= \; & c_{\Delta t'}(exp(b_j)),
\end{aligned}
$$

where

$$\Delta t' = exp(\Delta b)^s \Delta t. \qquad (4)$$

This means that the contribution vector corresponding to shutter speed $\Delta t'$ is identical to a shifted version of the original vector. We thus easily obtain an entire series of contribution vectors for shutter speeds that differ by a factor of $exp(\Delta b)^s$. In other words, only the shift, but not the shape of the contribution function depends on the shutter speed in the log domain. This allows us to move the contribution function over a peak in the histogram and then derive the corresponding shutter speed using the above formula.

4. OPTIMAL SHUTTER SEQUENCE

In order to compute an optimal shutter speed sequence, we first calculate an initial contribution vector from the known camera response and a chosen weighting function. Camera response functions can be estimated as described in [1, 10, 11] The initial shutter speed Δt to compute $c_{\Delta t}$ can be chosen arbitrarily. For ease of implementation, we choose Δt such that the first histogram bin is mapped to a pixel value of 1, that is $f(exp(b_1)\Delta t) = 1$. Note that $f^{-1}(0)$ is not uniquely defined in general. The size of the contribution vector depends on the dynamic range of the camera, reflected in its response function. Reaching a certain scene radiance $E_{N+1} = exp(b_{N+1})$, the camera's pixels will saturate, resulting in $f(exp(b_j)\Delta t) = 255$ for $j \geq N + 1$ in case of an 8 bit sensor. It is safe to assume that any reasonable weighting function assigns zero weight to this pixel value. Hence, the contribution vector $c_{\Delta t}(E_j) = w(f(exp(b_j)\Delta t))$ consists of N nonzero values. It can be shifted to $M + N - 1$ possible positions in the log radiance histogram. Each shift position s corresponds to a shutter speed Δt_i, which can be calculated using Equation 4: $\Delta t_i = exp(\Delta b)^s \Delta t$. This equivalence between shutter and shift is utilized later.

Here, we explain how a new shutter speed is added to an existing shutter sequence. The first shutter can be determined analogously. So we assume that the sequence already consists of a number of shutter speeds Δt_i. To each Δt_i belongs a contribution vector $c_{\Delta t_i}(E_j)$, with $E_j = exp(b_j)$ being the radiance values represented by the histogram bins. See Figure 4 for an example. We now need to decide whether to add another shutter to the sequence or not, and find out which new shutter brings the biggest gain in image quality. For this purpose, we define a *combined contribution vector* $C(E_j)$ that expresses how well the radiances E_j are captured in the determined exposures. We make the assumption, that the quality of the measurement of a radiance value only depends on the highest contribution value any of the exposures achieves for it. The combined contribution is thus defined as the maximum contribution for each histogram bin

$$C(E_j) = \max_i \left(c_{\Delta t_i}(E_j) \right). \tag{5}$$

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:

$$C = \sum_{j=1}^{M} C(E_j)H(j). \tag{6}$$

This is essentially the same as the cross correlation between the two. 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. If the histogram is normalized such that its bins sum up to 1 and the weighting function has a peak value of 1, then C is in the range of

[0..1] and can be expressed as a percentage. $C = 1$ then means that for each radiance value in the scene, there exists an exposure which captures it perfectly.

Perfect coverage is not achievable in a realistic scenario. It is more practical to stop adding shutters to the sequence once a softer stop criterion is met. We came up with three different stop criteria: the total number of exposures, a threshold for C and a maximum sum of shutter speeds. The criterion that limits the *total number of exposures* is always active. It guarantees that the algorithm terminates after calculating a finite number of shutter speeds. We also use this criterion to manually choose the number of exposures for our evaluation for better comparability. This is described in more detail in Section 5.

The *threshold for the coverage value C* is a quality criterion. A threshold closer to 1 allows for a better estimation of scene radiance, but requires to capture more exposures. We chose $C \geq 0.9$ for our running system.

For the type of camera we employ, the capture time of a frame is roughly proportional to the exposure time. And since we are interested in capturing real-time video at 25 frames per second, the sum of all shutter speeds must not exceed 40 milliseconds. Note that the camera exposes new frames in parallel to the processing of the previous ones. So we have indeed nearly the full HDR frame time available for capturing. Our third stop criterion is an adjustable threshold for the *sum of shutter speeds*. However, it should be made clear that the algorithm has little control over meeting this requirement. In the example shots we took, only two exceeded the threshold. But they in turn overshot it by a large factor. We argue that it is the camera operator's responsibility to adjust aperture and gain or to use a different lens to cope with particularly dark scenes.

The algorithm described so far is greedy in that it does not reconsider the shutter speeds it already chose. We added a second iteration over the shutter sequence to allow for some hindsight refinement. All shutters but the first one are refined in the same way. The first shutter is treated differently as described in the next paragraph. The shutter to be refined is first removed from the sequence. The algorithm for finding the next best shutter according to the maximum increase of C is then applied again. In most cases, the resulting shutter value is similar, but slightly better than the previous choice with respect to coverage. This is because the algorithm is aware of the rest of the sequence at this point. Our experimental results support this claim.

So far, we described the algorithm to determine a sequence of shutter speeds for a single HDR frame based on a perfect histogram of the scene. However, there are two major problems that arise when applying this algorithm to HDR video directly: imperfect histograms and flicker.

Perfect histograms are not available in a real video. The available histograms are created from the previous frame which generally differs from the current one. Furthermore, the dynamic range covered by the histogram is only as high as the range covered by the previous exposure set. For example if the camera pans towards a window looking outside, the bright outdoor scene may be saturated even in the darkest exposure. This shows up as a thin peak at the end of the histogram of the previous frame (see Figure 5). How bright are these pixels really? To find out, the algorithm needs to produce a shutter sequence that covers a larger dynamic range than the histogram of the previous frame indicates. This allows the sequence to adapt to changes in the scene.

We accomplish this by treating the first shutter in the sequence differently. The special treatment is based on the observation that underexposed images contain more accurate information than overexposed ones. The dark pixels in an underexposed image are a noisy
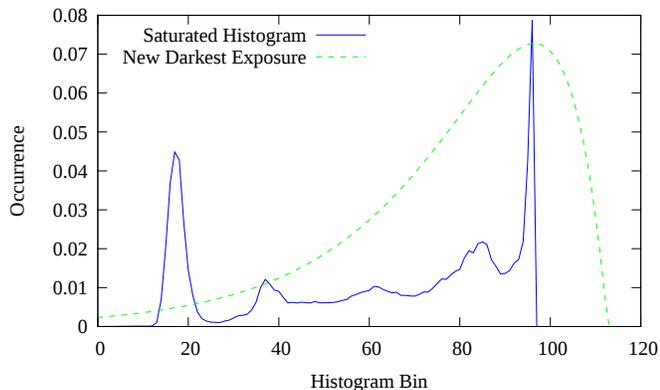
Fig. 5. Some areas of the scene are overexposed even in the darkest exposure. It shows up as a peak at the highest radiance value in the histogram. In the next frame, the algorithm chooses a shutter speed that covers the peak. By doing so, areas with a higher radiance than the previous maximum can still be captured faithfully.

estimate of the radiance in the scene. However, this noise is unbiased. Saturated pixels on the other hand always have the maximum pixel value, no matter how bright the scene actually is. As a consequence of this observation, the first shutter is chosen such that its contribution peak covers the highest radiance bin of the histogram. The peak of a weighting function is usually not located at the highest possible pixel value. This means that radiances beyond the peak – if existing in the next frame – are still represented by a non-saturated pixel. See Figure 5 for an example. This allows to faithfully record radiance values that are a certain percentage higher than the previous frame's maximum, and the sequence can adapt to brighter scenes. Change towards a darker scene is less critical, because underexposed pixels still contain enough information about the real radiance to calculate a new longer shutter time. With adaptation enabled, bootstrapping becomes straightforward. We can start with any set of shutter speeds and arrive at the correct values after a few frames. The speed of adaptation is evaluated in the experimental results section of this paper.

The second problem to deal with when applying our algorithm to HDR video is flicker. It is a side effect of changing the shutter sequence over time. Consider the following scenario: A bright saturated area like a white wall leads to a peak at the highest histogram bin. This gives rise to a darker exposure taken in the next frame as shown in Figure 5. The darker exposure causes the histogram peak to spread out over several bins. It may now cause too little extra coverage to justify the darkest exposure. In this situation, the algorithm oscillates between including the lowest shutter speed and omitting it. In the resulting video, the white wall would alternate between having texture and being completely saturated.

Another reason why stable shutter sequences are desirable is the way we operate our camera. 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 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*. If the number of shutters in the two sequences differs, then they are not similar. If it is the same, then we calculate the distance between their shutter values. The distance between two shutters is expressed as a percentage to model their exponential nature. For each value in the first sequence, the closest shutter speed in the second one is found. This search is necessary because the order of the lists is arbitrary. The distance between all closest shutter pairs is averaged. If the average is greater than a threshold (we use 20%), the sequences are not similar. Otherwise they are similar.

Using this definition, we achieve temporal stability by distinguishing between two states: *changing* and *static*. We always run our algorithm to determine a new shutter sequence. In the *changing* state, this new sequence is used directly and new camera parameters are transmitted. In the *static* state, the sequence is simply discarded and the parameters of the previous frame are kept. Change between the states occurs according to the following rules:

- When in the *static* state and the newly determined sequence is not similar to the previous one, increase a counter.
- If more than certain number of non-similar sequences occur in a row (3 in our system), transition to the *changing* state.
- A sequence similar to the currently used one always brings the algorithm back to the *static* state and resets the counter.

These rules have the effect that small variations in the shutter speeds are ignored. Once the scene actually changes, it takes three frames to react. Then the algorithm retains its original flexibility. It is able to adjust in each frame until a stable shutter speed sequence is found again. For fast bootstrapping, the system starts in the *changing* state.

5. EXPERIMENTAL RESULTS

This section presents the evaluation of our algorithm for optimal shutter speed sequences. Section 5.1 describes a subjective user study we conducted to assess the HDR image quality our approach achieves compared to the traditional way of choosing evenly spread shutters. For reasons described later in the section – most notably the unavailability of a perfect reference HDR video – only still images are used in this study. Section 5.2 contains a number of experiments to investigate the algorithm's behavior in a live video system. They include an analysis of the algorithm's adaptation to changing brightness conditions and of its processing time.

5.1. Subjective User Study

27 participants took part in our subjective user study. Five of them were familiar with HDR imaging algorithms. The study was done over a website that allows to rate the quality of HDR images.[1] See Figure 6 for a screenshot of the website. Its first page contains a brief introduction to HDR imaging and the problem of choosing suitable shutter speeds. The participants were told to base their rating on: The amount of under- and overexposure present, the amount of image noise, and quantization effects in color gradients. An example for each type of artifact was given. Variations in overall image brightness, contrast or color saturation were to be ignored as they may occur as a side-effect of tone mapping. The subjects were then shown twelve datasets of various scenes (see examples in Figure 7). Each dataset consisted of three HDR images: a reference image, an image created using shutter speeds from our approach and one where evenly spread shutters were used. The reference was always shown on the left side while the two survey images were shown in random order to avoid subjective bias. Each of the two images had to be rated using the five scores (numerical value in parentheses): Very Good (5), Good (4), Average (3), Poor (2), Very Poor (1).

---

[1]http://pi4.informatik.uni-mannheim.de/∼bguthier/survey/

Reference image   Survey image 1   Survey image 2

click on image to enlarge   Very Good   Good   Average   Poor   Very Poor   Very Good   Good   Average   Poor   Very Poor

next

Fig. 6. Screenshot of the website we used for our subjective user study. A reference image and two survey images are shown and participants can rate their quality.

We used an AVT Pike F-032C FireWire camera capable of capturing 208 VGA frames per second with an aperture of f/2.8. The twelve scenes we captured had dynamic ranges exceeding the camera's capabilities. To attain radiance values with high precision, we chose static scenes and used a tripod. Each scene was captured as a set of 79 LDR exposures with shutter speeds varying by a factor of $\sqrt[8]{2}$. An exposure set covers the entire range of our camera's shutter settings (37 μs to 81.9 ms). All 79 exposures were used to generate the reference image and the log radiance histogram of each scene. The reference image is assumed to be an accurate representation of the scene radiance.

To create our datasets, we manually selected a suitable number of LDR exposures to be used for the two survey HDR images of each scene. The number was chosen low enough for a discernible degradation of image quality to facilitate the rating process. For comparison, the default stop criterion for total coverage is $C \geq 90\%$, while the average coverage achieved for our datasets was 80.4% for optimal and 75.9% for equidistant shutters. The chosen number of exposures was used as the only stop criterion of our algorithm; a sequence of shutter speeds was created accordingly. Out of the 79 saved images of one dataset, those best matching the determined shutter speeds were merged to create the first HDR image. The second image was created using evenly spaced shutter speeds. To determine this sequence, the minimum and maximum scene radiance were considered, and the same number of exposures were spread evenly to cover the entire dynamic range. "Evenly" in this context means that the corresponding shutter speeds vary by a constant factor, i.e., a constant offset in the log domain. The shortest shutter speed was chosen in the same way as for our algorithm. The only exception are equidistant shutter sequences with only two shutters. For these, we found that choosing them closer to the center of the histogram gives better results. Due to the way we determined them, equidistant shutters also benefit from prior knowledge of the scene radiance, which is an advantage over plain exposure bracketing. This needs to be considered when comparing the achieved scores.

The main reason to use HDR still images instead of video for subjective quality assessment is the availability of a perfect reference image and with it the reproducibility of the results. Capturing 79 LDR exposures at varying shutter speeds allows to reconstruct the real scene radiance accurately. The shutter values are sufficiently close together to simulate arbitrary shutter sequences. Capturing the same amount of exposures for an HDR reference video is not feasible. Another reason is the difficulty to capture the optimal and the equidistant shutter video both at once. And lastly, HDR video may introduce various new artifacts like misalignment of the exposures or temporally inconsistent tone mapping. These additional artifacts may mask the difference between the two shutter speed choices.

The 27 participants rating 12 datasets resulted in a total of 324 pairs of scores, one for optimal and one for equidistant shutters. Seven pairs were invalid because at least one score was not specified by the subjects. This was explicitly allowed in order to not encourage the participants to enter bogus scores when wanting to skip datasets. Averaging the 317 valid ratings results in a score of 3.73 for the optimal shutter algorithm and 2.83 for the equidistant approach. Note that the absolute value of the score is meaningless as the survey images were intended to be flawed. As a second aggregation of the results, we counted the instances where either of the approaches scored better than the other. This leads to our approach achieving a better score in 70%, the same in 16% and a worse score in 14% of the ratings. Our approach got rated worse the most often in a dataset where it created a stronger quantization effect in the clouded sky. The sky only covers a relatively small area of the scene. It appears however that human observers pay more attention to it than its area indicates. We believe that this discrepancy between impact on the scene histogram and human attention poses a challenge for our algorithm. Tackling it exhaustively would require a costly visual attention analysis of the scene.

Figure 7 shows the reference images of all twelve scenes. The plot next to each image contains the log radiance histogram of the reference HDR image. It is normalized so that its bins sum up to 1. The plot also displays the combined contribution functions created by the two algorithms. It is calculated according to Equation 5. It can be seen, that the equidistant shutters disregard the brightness distribution of the scene and sometimes exposures are captured that add little to the coverage value. The achieved coverage values and the calculated shutter speeds are presented in Table 1. Due to the special treatment of the first shutter in our algorithm, its achieved coverage can be lower than for equidistant shutters. This effect is most prominent in scenes where only two exposures are used.

5.2. Objective Measurements

The experiments presented in this section were all conducted in a real-time HDR video system. Our shutter speed sequence algorithm uses the histogram of the current HDR frame as input. The histogram was created during tone mapping of the frame. The calculated shutter values are then used to capture the LDR exposures for the next frame. An appropriate subset of the following three scenarios was used for the measurements.

1. Mostly static indoor scene with no camera motion.

2. A busy road with moving cars but no camera motion.

3. Moving scene with many camera pans between dark indoor and very bright outdoor areas.

Unless stated otherwise, the measurements were taken over a period of 15 seconds ($\approx$ 375 HDR frames).
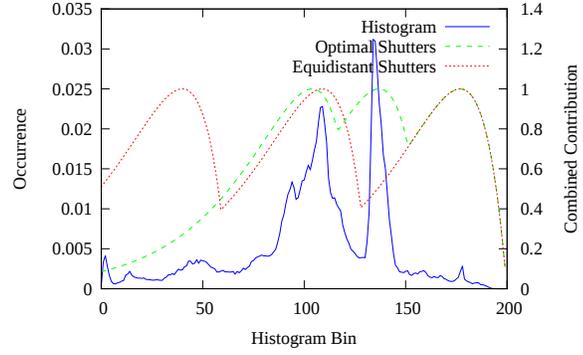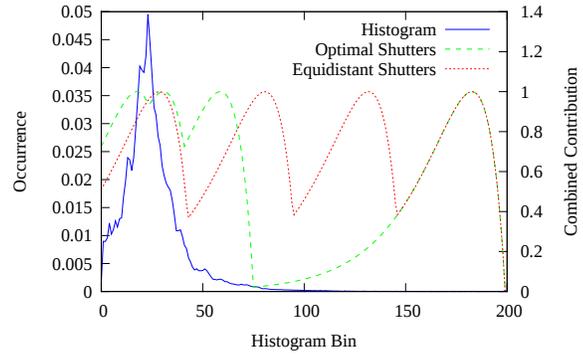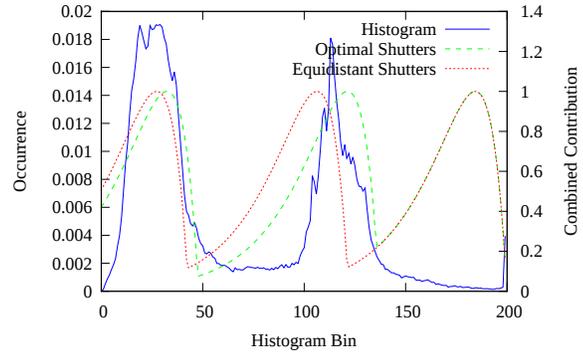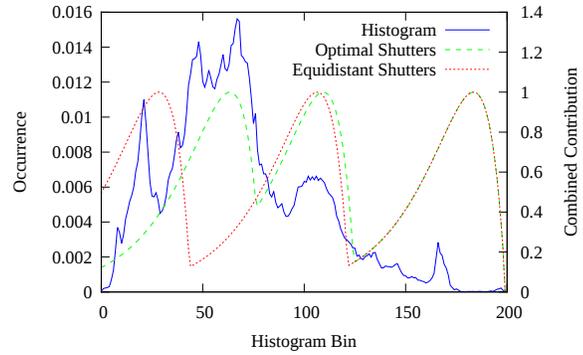
Fig. 7. The left column shows the reference images of the example scenes used in our subjective evaluation. The plots contain the corresponding normalized log radiance histogram. The dashed lines are the maximum of the contribution functions belonging to the shutter speeds determined by our algorithm and to the equidistant shutters.

(e)
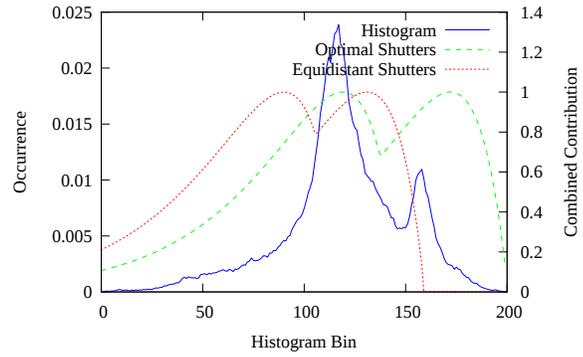
(f)
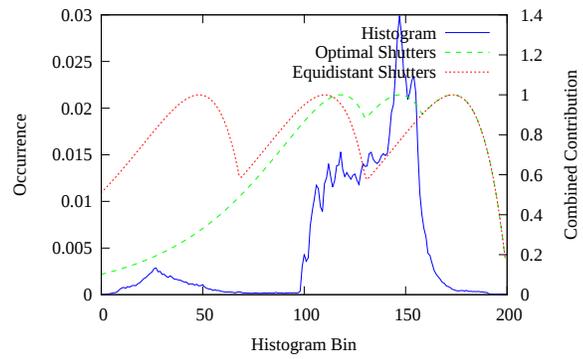
(g)
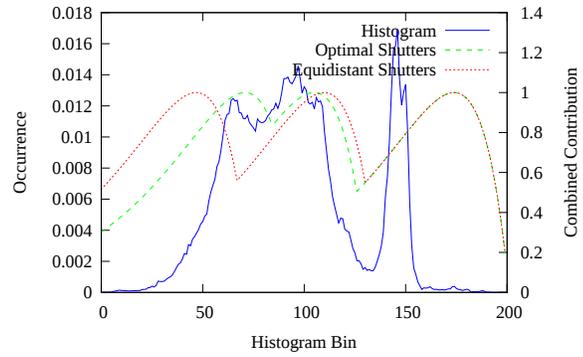
(h)
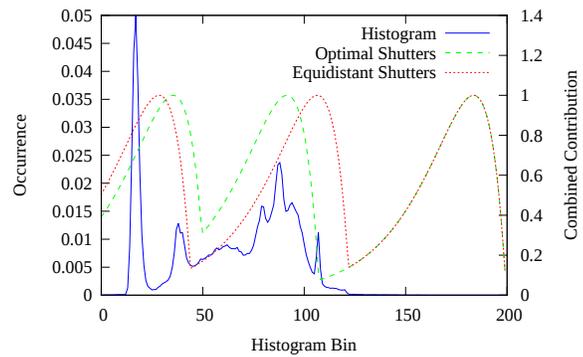
(i)
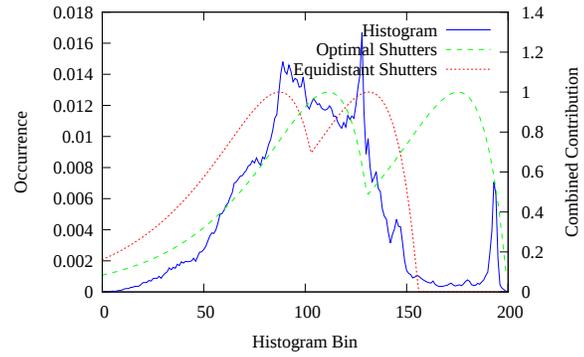
(j)
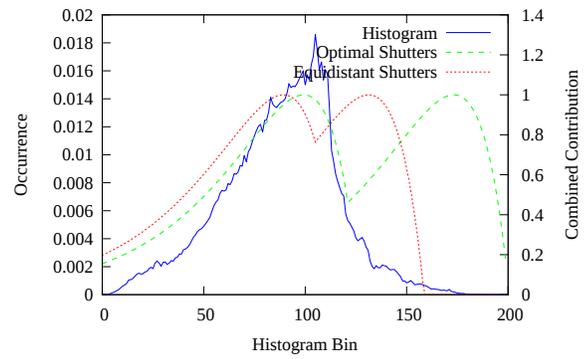
(k)
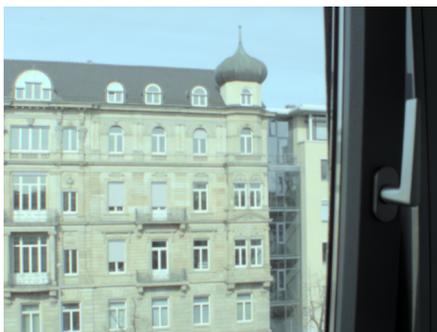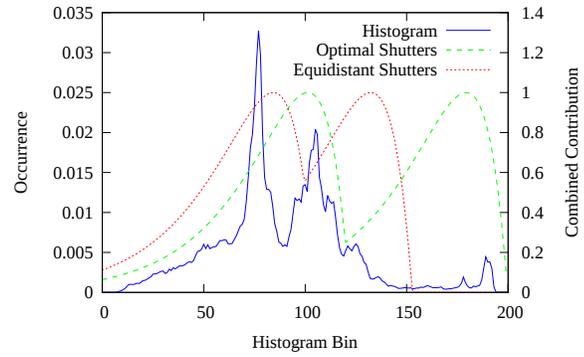
(l)
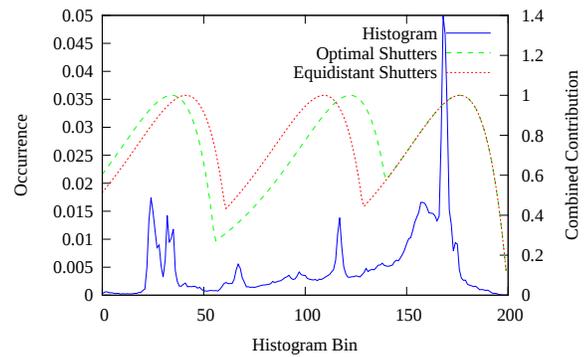
Scene	C_{OPT}	C_{EQ}	Shutters (OPT)	Shutters (EQ)
(a)	84.0%	77.2%	0.26 2.36 0.87	0.26 2.07 16.5
(b)	92.9%	83.8%	0.02 12 3.8 21	0.02 0.18 1.5 13
(c)	77.3%	69.8%	0.02 17.4 0.34	0.02 0.67 21.8
(d)	68.5%	53.0%	0.06 12.1 1.70	0.06 1.90 57.2
(e)	86.1%	77.0%	0.96 3.45	2.60 7.06
(f)	90.9%	81.9%	0.64 1.20 2.47	0.64 3.04 14.5
(g)	88.8%	82.0%	0.78 4.40 10.4	0.78 3.88 19.3
(h)	74.7%	64.8%	0.03 1.59 17.4	0.03 0.85 23.8
(i)	72.4%	82.0%	0.68 4.19	2.40 8.46
(j)	77.9%	83.2%	1.81 11.7	5.25 15.3
(k)	66.4%	73.0%	1.26 18.4	6.41 32.7
(l)	85.0%	83.9%	0.61 37.4 2.92	0.61 4.33 30.8

Table 1. The second and third column contains the coverage values C for the twelve scenes as achieved by the two algorithms: optimal shutters (OPT) and equidistant shutters (EQ). The third and fourth column show the calculated shutter speeds in milliseconds.

Scenario	Size Differs	Average Distance	Std. Dev.
1	0%	1.00%	0.89%
2	3.75%	2.79%	1.74%
3	18.87%	8.08%	9.11%

Table 2. Percentage of sequences with differing number of shutters, average distance between the sequences and the standard deviation of the distance. They were obtained from 15 second shots in the three aforementioned scenarios.

As described in Section 4, the shutters that were determined greedily are being refined in a second pass over the sequence. The goal of this is to improve the coverage value C which describes how well the chosen exposures overlap with the scene histogram. In order to evaluate the additional gain from the refinement step, we measured C before and after the refinement. This was done in the third (dynamic) scenario. Averaged over 15 seconds of video, the refinement achieved a 1.5% increase of C. To judge this result, one must consider two things: Firstly, the algorithm usually stops adding shutters to the sequence once $C \geq 0.9$. Because the maximum coverage is 1.0, there is not much room for improvement. Secondly, refinement does not add new shutters to the sequence, but adjusts the existing ones. Compared to capturing an extra frame to obtain a higher coverage, it is thus a rather cheap operation. We decided to include the refinement step into our running system, but omitting it is a viable option when processing time needs to be saved.

For our stability criterion, we defined the percentual distance between two shutter speed sequences. In order to get an understanding of this quantity and to decide upon a similarity threshold, we measured the distances between two sequences computed in two consecutive frames. This was done in all three scenarios and the stability criterion was ignored. The results are listed in Table 2. When the size of two sequences differs, they are always classified as non-similar. So the first column of the table counts how often the size changed during the 15 seconds of the video. It is given as a percentage of the frames. The second column contains the average distance between two consecutive sequences. The standard deviation is given in the third column.

These values can be used to determine a suitable threshold for the distance to distinguish similar and non-similar sequences. We make the following observations. The first scene is completely static. Therefore, the shutter speed sequence should remain the same at all

times. All measured distances should be considered as being similar. The second scene contains moving cars and the shutter sequence needs to adapt occasionally. In the third scenario, the sequence needs to change a lot to accommodate the varying brightness conditions. To meet these requirements, we set the threshold to 20%. Activating the stability criterion with this threshold, we repeated the experiments. During the 15 seconds, the algorithm was in the *changing* state 0% of the time in the first scenario, 0% in scenario 2, and 11.49% of the time in scenario 3. We found that these results were rather insensitive to changes in the threshold as long as it is high enough for a stable sequence most of the time. Once the scene's brightness actually changes noticeably, the size of the sequence often changes too and the distance between the sequences becomes very large.

In the experiment described in the following, we investigated the time it takes for our algorithm to adapt to changes in the scene. We did this by keeping the scene and the camera static, choosing extreme shutter speed sequences and measuring the number of frames it takes to stabilize. The scene and aperture of the camera were chosen such that the optimal shutter sequence consisted of four shutter values around the center of the camera's shutter range. By *center*, we mean the middle value in the log domain with the same *factor* to the lowest as to the highest shutter. For our camera, the shutter value of 1.74 ms is a factor of 47 higher than the minimum and lower than the maximum shutter. The algorithm was set to the *changing* state and three different starting sequences were set: the sequence consisting of only the shortest possible shutter, the longest shutter and a sequence covering the full shutter range with one stop between the shutters. We then measured the number of frames the algorithm stayed in the *changing* state. The values are averaged over 375 runs for each of the three starting sequences.

As expected, the full coverage sequence adjusted the fastest. It took 2.07 frames to stabilize. This means that the stable sequence could be directly calculated from the first HDR frame in almost all of the iterations. From only the shortest shutter value, it took exactly 3 frames to stabilize. The algorithm already calculated three shutters in the second frame and reached the final sequence in the third. It then switched to the *stable* state in the fourth frame, because the calculated sequence was similar. The worst adaptation speed was achieved when starting from only the longest shutter value, that is, from the brightest image. The lowest shutter in the sequence was approximately halved in every frame. In the average, the algorithm was in the *changing* state for 8.20 frames. This confirms our previous statement that convergence towards darker scenes (i.e., higher shutter values) is easier. It also justifies the special treatment of the first shutter in the sequence as described earlier.

Since it is our goal to perform shutter sequence computations in real-time to create HDR videos, we measured the processing time taken by our algorithm. As mentioned earlier, we assume that the histogram of the previous HDR frame was computed during tone mapping (e.g., Ward's histogram normalization technique [9]). Histogram creation is thus not included in these measurements. The system we used for this experiment has an AMD Athlon II X2 250 dual-core CPU. The scenario with dynamic camera and scene was used to cover a large variety of shutter sequence lengths. The experiment showed that 96.5% of our 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 0.30 ms per shutter value including refinement. For comparison, the entire process of creating a displayable HDR frame from 2 to 8 base exposures takes

6 to 15 ms on a GPU. In a 25 fps real-time HDR video system, there are 40 ms available for processing each frame. Our algorithm is thus fast enough to be used in this application.

6. CONCLUSIONS AND OUTLOOK

We presented an approach to computing shutter speed sequences for temporally bracketed HDR videos. Our goal is to maximize the achieved HDR image quality for a given number of LDR exposures. This is done by consecutively adding shutters to the sequence that contribute to the image quality the most. Choosing evenly spread shutters wastes too much time for capturing exposures which contribute little to the HDR result. We are thus able to save capturing and processing time over the traditional approach by being able to reduce the number of LDR exposures without impairing quality. Analysis of the algorithm's behavior in a real-time HDR video system showed that it is suitable for such a scenario and can be employed in video surveillance.

Using the histogram coverage as our criterion for optimization means focusing on the largest image areas first. We believe that being able to see as much as possible in a video is the main focus in surveillance. However, the user study showed that in certain situations, HDR images are also judged by *where* in the image the quality is achieved. We would like to take this into account in our future work.

7. REFERENCES

[1] Paul E. Debevec and Jitendra Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997.

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

[3] N. Barakat, A. N. Hone, and T. E. Darcie, "Minimal-bracketing sets for high-dynamic-range image capture," *IEEE Trans. on Image Processing*, vol. 17, no. 10, 2008.

[4] T. Chen and A. El Gamal, "Optimal scheduling of capture times in a multiple capture imaging system," in *Proc. of the SPIE Electronic Imaging Conference*, 2002.

[5] M.D. Grossberg and S.K. Nayar, "High dynamic range from multiple images: Which exposures to combine?," in *Proc. of the ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, 2003.

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

[7] K. Hirakawa and P.J. Wolfe, "Optimal exposure control for high dynamic range imaging," in *Proc. of the 17th IEEE International Conference on Image Processing (ICIP)*, 2010.

[8] D. Ilstrup and R. Manduchi, "One-shot optimal exposure control," in *Proc. of the 11th European Conference on Computer Vision (ECCV)*. Springer Berlin, Heidelberg, 2010.

[9] Gregory W. Larson, Holly Rushmeier, and Christine Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, 1997.

[10] S. Mann and R.W. Picard, "Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures," in *Proceedings of the IS&T 48th Annual Conference*, 1995.

[11] Tomoo Mitsunaga and Shree K. Nayar, "Radiometric self calibration," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.