This is a preliminary version of an article published in
Proc. of the 10th IEEE International Workshop on Multimedia Signal Processing (MMSP), Cairns, Australia, pp. 241 – 246, October 2008
by Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg

Capturing High Dynamic Range Images with Partial Re-exposures

Benjamin Guthier, Stephan Kopf, Wolfgang Effelsberg

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

***Abstract*—In this paper we present an optimized approach to capture high dynamic range (HDR) images. It is based on existing methods of creating HDR images by fusing a set of differently exposed low dynamic range (LDR) images. We optimize the capturing process of LDR images towards improved capture speed by using partial re-exposures. That is, we make use of the idea that it is not always necessary to capture full size images when only small portions of the scene require HDR. By analyzing captured images for badly exposed regions and re-exposing selectively, we save overall capture time and increase the frame rate when image sequences are recorded.**

I. Introduction

Natural scenes usually have a range of brightness values that exceeds the capabilities of digital capturing devices by far. Traditional "low dynamic range" (LDR) imaging devices often cannot capture their full dynamic range, especially that of back-lit scenes. This leads to under- and overexposed pixels in the captured images, and information on brightness differences between these pixels is lost. A variety of methods have been proposed to capture "high dynamic range" (HDR) images to overcome these limitations [1]. Most of them can be classified into three categories: temporally varying exposure, multiple capture devices, and extended dynamic range of the image sensor. All categories have their individual pros and cons when used for the creation of HDR images.

The most popular approach is using a set of LDR images captured in quick sequence at different exposure settings [2][3][4][5]. Each LDR image then captures one facet of the scene's dynamic range. While low exposure images show more contrast in bright areas, the images captured with high exposure reveal all details in darker regions. When fused together, an HDR image is created that covers the full dynamic range of the scene. An obvious disadvantage of this approach is the increase of capture time required to record a scene. Also, this class of methods is only suited for scenes with little motion where the image sequence can be merged into one consistent picture. The amount of scene motion allowed is directly related to the increase in capture time: The longer it takes to capture all the required images, the stronger the aliasing effects of motion between the shots will be.

Another approach to HDR image capturing uses multiple LDR capture devices simultaneously [6][7][8]. Beam splitters are used to allow an array of LDR cameras to view the same scene at the same time. The shutter speeds of the cameras differ from each other in order to achieve an effect similar to the temporally varying exposures described before. With multiple cameras, an entire set of LDR images that covers the scene's full dynamic range can be captured at once, leading to increased capture speed compared to the previously described method. As a drawback, this leads to a significant increase of hardware costs.

In contrast to the two classes described above, a third class of HDR capturing methods attempts to extend the dynamic range of the image sensor itself, rather than relying on multiple images or multiple sensors. This can be achieved, for example, by repeatedly resetting the sensor cells during integration, effectively leading to an increase in full-well capacity. Similar to this, logarithmic image sensors can be used that increase the sensor's dynamic range via logarithmic compression[1]. Other approaches use spatially varying exposures to sacrifice spatial resolution for dynamic range [9], adaptive pixels [10] or use the time taken for a pixel to saturate as a measure of scene radiance [11]. Common to all approaches that lie within this category is the raise in costs for sophisticated imaging hardware. Their clear advantage though is their ability to capture images with an extended dynamic range directly, without the necessity of LDR image fusion.

In this paper, we present an improved approach to capture a sequence of LDR images with varying exposures using a single camera. Instead of capturing LDR images at fixed shutter settings (fixed exposures), we adapt these settings and the number of exposures to the dynamic range of the given scene. Additionally, we make use of the idea that it might not always be necessary to capture a full image at one exposure setting if only a few image areas require HDR. We have developed an algorithm that detects badly exposed regions in an image and triggers the camera to re-capture only these potentially much smaller regions selectively. Reducing the image size like this decreases the overall capture time of an image significantly while avoiding the disadvantages of the other approaches.

This paper is structured as follows. In Section 2, we analyze the relationship between image capture parameters (e.g., image size and shutter speed) and the resulting capture time. We describe our new algorithm for selecting image regions of interest and doing re-exposure in the following section. Additionally, we show our way of improving the capture performance

[1]Examples for extended dynamic range image sensors are: SMaL camera, Pixim, SpheroCam HDR, LadyBug camera

by parallelizing image capturing and image analysis. The experimental results in Section 4 illustrate that the overall time to capture images can be reduced significantly by our approach. We end the paper with conclusions and an outlook.

II. Properties of "true partial scan"

Many industrial FireWire CCD cameras have a feature called "true partial scan"[2]. It allows the definition of a rectangular sub-area of cells on the CCD sensor – a *region of interest* (ROI) – 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 CCD sensor and transmit the image data over the FireWire bus is reduced, leading to a higher frame rate at lower image sizes.

We first identify relevant camera parameters that influence capture speed and infer general rules for the choice of parameters in Section II-A. We then apply the model exemplarily to a specific camera in Section II-B.

A. Parameters and general rules

The relevant parameters in our scenario are:

- shutter speed setting,
- position of the ROI on the CCD sensor, and
- width and height of the ROI to be captured.

The *shutter speed setting* determines how long the sensor cells are exposed to light before being read out and transmitted to the PC. This value imposes a delay in the capturing process and is added to the overall capture time.

The *position of the ROI* on the CCD sensor has no significant influence on capture time. Any ROI of a given size requires the same amount of time to be read out and transmitted, no matter where it is located on the sensor.

The *ROI size* has the most interesting influence on 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 at a constant frequency, 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. The ROI width merely determines the number of bytes per read-out cycle produced by the camera. This data is split into packets of a fixed size and sent over the bus at the bus' own cycle frequency. Since the bus packet size can only be set in discrete steps, a slight *increase* in capture time is perceived when decreasing the ROI width.

From these considerations, we can conclude the following:

- The position of ROIs to be captured is irrelevant.
- Given that the camera's data rate does not exceed the bandwidth of the FireWire bus, it is most efficient to capture images at full width.
- The height of a ROI to be captured should be chosen as small as possible.

Additionally, some CCD cameras require a minimum total image size to capture efficiently. Since the image width is fixed, this requirement results in a lower bound for the ROI height h_{min}.

At full image width, the total capture time in milliseconds at a given shutter setting s and image height h can be modeled as

$$T(s,h) = a + s + c + v * h, \tag{1}$$

where a is the time to set up an image buffer of appropriate size and to allocate bandwidth on the bus. c and v are the camera-specific constant and variable capture costs. a, c and v need to be determined experimentally as described in Section II-B.

Note that the constant cost of image capturing $(a + s + c)$ can exceed the variable cost $v * h$ 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. According to the above considerations, the distance between two ROIs can be expressed by the number d of image rows between them. It is more efficient to merge two regions and capture d additional rows rather than capturing twice and doubling the constant cost if

$$v * d < a + s + c \quad \Leftrightarrow \quad d < \frac{a + s + c}{v}. \tag{2}$$

B. Estimating capture costs

In our experiments, we used an AVT Marlin F-145B2 FireWire camera with a maximum resolution of 1392 by 1040 pixels, B/W. It allows shutter values ranging from 0.02 ms to 81.9 ms and features "true partial scan".

To estimate the camera-specific coefficients, we captured images at different heights and measured the time taken. We kept the image width constant at 1392 pixels and varied the height from 10 to 1040 in steps of 10, measuring each size five times for an average. The shutter speed was set to the minimum possible value of 0.02 ms and subtracted later. The time to allocate image buffers and bus bandwidth and to trigger the camera was averaged over all image sizes, resulting in a value of $a = 20.36$ ms. For each individual exposure, we started to measure the capture time after triggering and stopped when the image was fully received. The results are shown in Figure 1. As can be seen from the plot, it is inefficient to capture ROIs with a height of less than 48 rows – this is where the total image size falls below 64 kB. As a consequence of this characteristic trait, we set h_{min} to 48 for our camera. In order to estimate c and v, we fit a regression line to the sample data starting from $h = 50$ and obtained the values $c = 25.63$ ms and $v = 0.09778$ ms per row.

For our camera, the total time to capture an image of full width, height h and shutter speed s can therefore be calculated as

$$T(s,h) = 20.36 + s + 25.63 + 0.09778 * h. \tag{3}$$

III. The partial HDR algorithm

Our algorithm to capture HDR images using partial re-exposures of poorly exposed regions can be divided into the following steps:

[2] Sometimes terms like "selectable region of interest" and others are used.

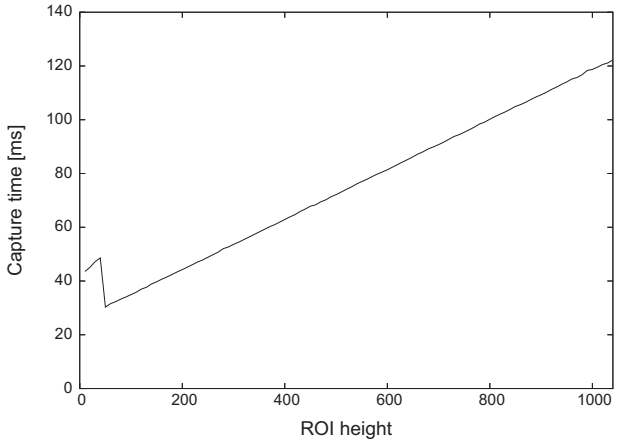


Fig. 1. Measured capture time in milliseconds for ROIs of a given height and full width.

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) If no more under- or overexposed regions have been found, create an HDR image from the set of exposures.

The algorithm explores the base image and all subsequently captured partial images iteratively and captures at only as many shutter speeds as necessary to cover the full dynamic range of the scene. As a side effect, it is insensitive to changes of the initial shutter setting. We use an initial setting of 20.48 ms throughout this paper.

In order to search captured images for under- or overexposed pixels, we first need a criterion that describes being "well exposed". We introduce a simple criterion based on the brightness value of an image pixel: A pixel is *valid* if its brightness value p lies within an interval $[p_{min}, p_{max}]$ and is *invalid* otherwise. In other words, very dark or very bright pixels are poorly exposed (invalid) and are considered for re-exposure. We use an 8-bit industrial camera with very little dark noise and choose $[p_{min}, p_{max}] = [10, 254]$. The choice for p_{min} is more or less arbitrary and can be adjusted to the needs of the particular application.

We make the assumption that the scene is static during the entire capturing procedure and that the camera is stationary.

A. Determining ROIs for re-exposure

This Section describes the main part of our work: The analysis of an image for invalid pixels and the identification of rectangular image areas to be captured again at different shutter speeds. We derive the latter mostly from the results of Section II.

The ROI detection process starts with the first captured image – the base image. It is the only image that is searched for both under- and overexposed pixels. All subsequently recorded images have either lower or higher shutter speeds than the original image and are only analyzed towards their corresponding direction. The two directions "higher shutter speeds" and "lower shutter speeds" are performed independently but in parallel to some degree, as can be seen later in Section III-C.

Our considerations in Section II have shown that 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. Note that two histograms must be created for the base image. For all later images, *one* histogram for either under- or overexposed pixel counts is sufficient.

From now on, only row histograms counting invalid pixels are considered, reducing the problem of finding ROIs to a one-dimensional one. As a preprocessing step, a morphological closing is done to the row histogram to achieve a preliminary grouping of nearby rows with high numbers of invalid pixels. A threshold r_{max} is then applied to the 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. By changing the parameter r_{max}, it is possible to adjust the trade-off between capture speed and image quality: Setting r_{max} to a lower value results in more rows to be marked for re-exposure, leading to a lower number of invalid pixels that remain in the final HDR image, but also to increased costs for capturing. The influence of r_{max} on the image capturing process is further examined in Section IV.

Next, the thresholded row histogram is searched for contiguous runs of marked rows. These constitute the basic ROIs for re-exposure. Before being pushed into the image capture queue, they are expanded to a minimum size of h_{min} rows, and ROIs that are closer together than d rows are merged into single regions to accomodate the properties of the camera's "true partial scan" feature, discussed in Section II.

Lastly, the detected ROIs are pushed into the queue of images to be captured. Depending on whether the image was analyzed for under- or overexposed pixels, the regions will be re-exposed with either longer or shorter shutter speeds respectively. In our approach, we double or halve the shutter speeds. By doing so, there will be enough image pixels that are valid in both of two consecutive exposures, so they can be used for planned image registration purposes. As soon as no more invalid pixels are found in any of the newly captured images, or the required shutter speed exceeds the camera's limits, the algorithm terminates. Therefore, no more shutter speeds than necessary to capture the scene's dynamic range are used.

B. HDR stitching

Once all LDR images have been captured successfully, using the mechanism described in the previous sections, they need to be fused together into one single HDR image. We refer to the process of combining LDR images into an HDR image as "HDR stitching". Numerous previous work is dedicated to this fusion process [2][3][4][12]. In our paper, we put the focus on image acquisition and use a simple method for HDR stitching. The SNR of pixels with high brightness values is higher than that of dark pixels due to the higher influence of quantization noise in dark regions. For each pixel in the HDR image, we thus simply compute its radiance value using the value of the brightest, non-saturated pixel across all exposures. The search for the brightest, non-saturated pixel can be limited to the set of exposures that include the respective pixel, and computation time can be saved.

C. Implementation issues

Our experiments in Section II revealed a phenomenon that can be utilized for an efficient implementation of the partial HDR algorithm. From an implementation point of view, the image capturing process can be divided into two distinct parts. The first part is the setup phase where an image buffer is allocated, bandwidth on the bus is reserved, and the camera is triggered to record an image. In the second phase, the camera exposes the sensor to the light of the scene, reads out the currents accumulated in its CCD cells, and sends the image data over the FireWire bus to the PC where it is written into memory via "Direct Memory Access" (DMA). During this second phase, until the image is fully received from the camera, the CPU is idle. We can therefore use this idle CPU time to analyze the captured images without adding to the overall capture time.

In our implementation, we make use of this fact in the following manner: First the base image I_0 is captured. Next, the base image is analyzed for overexposed regions which are then put into a re-exposure queue. After these two initial sequential steps, the algorithm can be parallelized: While the overexposed regions are re-captured with a shorter exposure time to create a new image I_{+1}, we use the idle CPU time to analyze I_0 again, this time for underexposed regions. Then, during the process of capturing the brighter image I_{-1}, the darker image I_{+1} is analyzed, and so on.

Our way of implementing this is by using two queues. One queue contains the images to be analyzed and the other contains camera settings for images to be captured. In each step, one element from *each* queue is considered and both are processed in parallel. 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 in the capture queue, the analysis can thus be performed for free and will not add to the overall capture time. As a future step of improvement, it is imaginable to use idle time during the capturing of a base image – for example to compute the HDR result from the set of previously captured images in case HDR image sequences are to be produced.

TABLE I
MEASURED TIME IN MILLISECONDS TAKEN FOR IMAGE CAPTURING, IMAGE ANALYSIS AND HDR STITCHING. FOR EACH SCENARIO THE TWO APPROACHES FULL HDR (F) AND PARTIAL HDR (P) WERE EXAMINED.

Scenario	Capturing	Analysis	Stitching	Total
Hallway (f)	1263	0	365	1628, 100%
Hallway (p)	954	77	274	1305, 80%
Indoor (f)	515	0	174	689, 100%
Indoor (p)	367	26	126	519, 75%
LEDs (f)	1936	0	545	2481, 100%
LEDs (p)	981	41	254	1276, 51%
PCB (f)	1493	0	451	1944, 100%
PCB (p)	1078	46	285	1409, 72%
Telephone (f)	897	0	256	1153, 100%
Telephone (p)	642	30	169	841, 73%
Window (f)	1074	0	327	1401, 100%
Window (p)	739	87	246	1072, 77%

IV. EXPERIMENTAL RESULTS

We've conducted experiments to evaluate the performance of our algorithm. The two performance criteria we considered were: 1) Time taken to create an HDR image, and 2) the quality of the resulting image with regard to the number of invalid pixels that remain in the final result. Both quantities were evaluated for our partial image HDR approach with adaptive numbers of exposures as well as for the traditional approach of creating HDR images using full images. For simplicity we refer to the two compared approaches as "partial HDR" and "full HDR". To compare the results, we first ran our algorithm on the test data to determine the set of exposures used and then measured the full HDR approach using the same set.

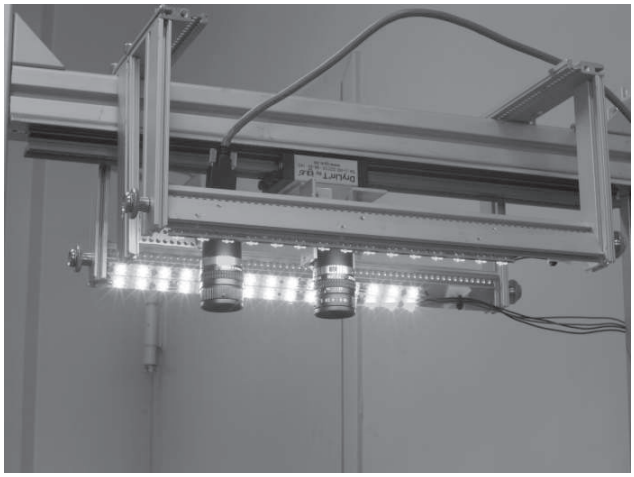
As test data to base these results on, we created six scenarios we refer to as: Hallway, Indoor, LEDs, PCB, Telephone and Window. HDR images of these can be seen in Figure 2. Each scenario consists of twelve saved exposures of the same static scene captured with a static camera. The shutter speeds were set to $2^i * 0.02ms, i = 1, .., 12$ in accordance with all possible shutter settings requested by our algorithm.

A compromise had to be made between making our measurements reproducible and measuring time taken to capture images as realistically as possible. The former suggests using saved images while the latter requires capturing live images. We chose to conduct our experiments on saved image data. Instead of recording live images with the camera, we copy ROIs from saved images and use the results of Section II to assess the required capture time. As described in Section III-C, the computation time that can be scheduled in parallel to an image acquisition is neglected.

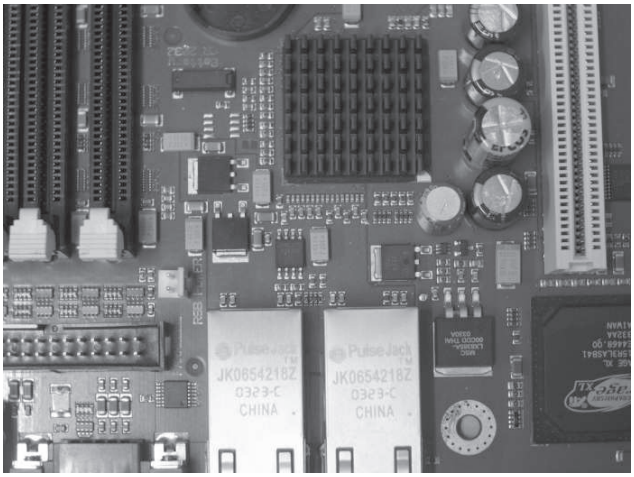
Table I shows the results of the speed measurements in each of the six scenarios using full and partial HDR. In this experiment, the parameter r_{max} was set to 0.7%. It can be seen that using partial re-exposures in these scenarios saves 20-49% of the time to create an HDR image. This leads to an increase in achievable frame rates by 25-96% in case that a sequence of HDR images is recorded. As expected, the LED example (see Figure 2) leads to the biggest performance gain because the bright areas cover only a small portion of the scene.

(a) Hallway

(b) Indoor



(c) LEDs



(d) PCB

(e) Telephone

(f) Window

Fig. 2. HDR images of the six scenarios we used for our experiments. (a) and (b) were chosen as worst case examples for our approach: (a) has very large overexposed areas due to the window in the background and reflections on the floor, and (b) displays an indoor scene with a rather low dynamic range. A high performance gain is to be expected in scenario (c) where the bright LEDs cover only a small portion of the image.

TABLE II

INFLUENCE OF r_{max} ON IMAGE QUALITY AND CAPTURE SPEED. THE "INVALID" COLUMN DISPLAYS THE PERCENTAGE OF INVALID PIXELS IN THE HDR IMAGE. "TOTAL TIME" SHOWS THE TOTAL TIME TAKEN TO CAPTURE, ANALYZE AND STITCH IMAGES IN RELATION TO THE SPEED ACHIEVED BY THE FULL HDR APPROACH AS SHOWN BEFORE.

Scenario	r_{max}	**Invalid**	**Total time**
Hallway	0%	0%	87%
	0.7%	0.02%	80%
	5%	0.31%	67%
	10%	3.68%	49%
PCB	0%	0%	89%
	0.7%	0.04%	72%
	5%	1.04%	42%
	10%	3.78%	37%

As a "worst case" example, the Hallway scenario contains overexposed regions that comprise a large area of the scene (due to reflections on the floor). Throughout all scenarios, the overhead introduced through image analysis accounts for approximately 5% of the overall duration.

The parameter r_{max} influences the process of detecting invalid regions in an image. It determines the maximum allowed percentage of invalid pixels in an image row. Rows that contain at most r_{max} percent invalid pixels are not re-exposed and may generate under- or overexposed pixels in the final HDR image. As stated before, r_{max} is an optimization parameter allowing to adjust the trade-off between capture speed and image quality. The influence of r_{max} on these two performance criteria is illustrated in Table II. It shows the percentage of invalid pixels in the created HDR image and the overall time taken to capture, analyze and stitch images. The overall time is expressed as the percentage of the full HDR capture time, analog to Table I. To avoid clutter, we chose two exemplary scenarios and four different settings for r_{max} each. The results in the other scenarios were similar.

V. CONCLUSIONS AND FUTURE WORK

We have shown a technique to capture HDR images more efficiently than by capturing images with varying exposure at full resolution. By capturing partial images and selecting the range of shutter settings used adaptively, we were able to increase the frame rate by 25-96%. We also showed how our algorithm can be parallelized, so that image analysis is performed while exposing a new image. Images can therefore be analyzed "for free" while waiting for the next image. Capturing partial images also reduces the amount of redundant data that is inputted to HDR stitching which allows for more time to be saved.

A limitation of our approach is the relatively high constant cost of the capturing process. In our scenario, only roughly one half of the total capture time was dependant on the image size, setting an upper bound to the achievable performance gain.

Future work will be done on relaxing the requirements of a static camera and scene. We will enhance our algorithm through a motion compensation process to make full use of the increased frame rate. When capturing videos instead of single images, it is also possible to use information on the radiance distribution of previous frames to adjust the algorithm's parameters.

REFERENCES

[1] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2006.

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

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

[4] T. Mitsunaga and S. K. Nayar, "Radiometric self calibration," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1, Jun. 1999, pp. 374–380.

[5] M. A. Robertson, S. Borman, and R. L. Stevenson, "Estimation-theoretic approach to dynamic range enhancement using multiple exposures," *Journal of Electronic Imaging*, vol. 12, no. 2, pp. 219–228, Apr. 2003.

[6] M. Aggarwal and N. Ahuja, "Split Aperture Imaging for High Dynamic Range," *International Journal of Computer Vision*, vol. 58, no. 1, pp. 7–17, 2004.

[7] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, 2005.

[8] E. Ikeda, *Image data processing apparatus for processing combined image signals in order to extend dynamic range.* U.S. Patent 5801773, Sep. 1998.

[9] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: spatially varying pixel exposures," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Jun. 2000, pp. 472–479.

[10] S. K. Nayar and V. Branzoi, "Adaptive dynamic range imaging: optical control of pixel exposures over space and time," in *Proc. of the ninth IEEE International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1168 – 1175.

[11] V. Brajovic and T. Kanade, "A sorting image sensor: an example of massively parallel intensity-to-time processing for low-latency computational sensors," in *Proc. of the IEEE ICRA*, vol. 2, Mar. 1996, pp. 1638–1643.

[12] B. C. Madden, "Extended intensity range imaging," University of Pennsylvania, Tech. Rep. 248, Dec. 1993.