Smart camera design for realtime High Dynamic Range imaging

Pierre-Jean Lapray*, Barthélémy Heyrman*, Matthieu Rossé* and Dominique Ginhac*†
*LE2I UMR 5158, Univ Burgundy, Dijon, France
Email: (pjlapray,heybar,dginhac,mrosse)@u-bourgogne.fr
†Corresponding author

***Abstract*—Many camera sensors suffer from limited dynamic range. The result is that there is a lack of clear details in displayed images and videos. This paper describes our approach to generate high dynamic range (HDR) from an image sequence while modifying exposure times for each new frame. For this purpose, we propose an FPGA-based architecture that can produce a real-time high dynamic range video from successive image acquisition. Our hardware platform is build around a standard low dynamic range CMOS sensor and a Virtex 5 FPGA board. The CMOS sensor is a EV76C560 provided by e2v .This 1.3 Megapixel device offers novel pixel integration/readout modes and embedded image pre-processing capabilities including multiframe acquisition with various exposure times. approach consists of a pipeline of different algorithmic phases: automatic exposure control during image capture, alignment of successive images in order to minimize camera and objects movements, building of an HDR image by combining the multiple frames, and final tonemapping for viewing on a LCD display. Our aim is to achieve a realtime video rate of 25 frames per second for a full sensor resolution of $1,280 \times 1,024$ pixels.**

I. Introduction

A standard camera is able to capture only a fraction of the visual information that is visible to the human eye. This is specifically true for real scenes that can simultaneously include areas of low and high illumination due to transitions between sunlit and shaded areas. When capturing such a scene, the camera is unable to store the full dynamic range (i.e. the ratio between the lightest and darkest pixel) resulting in low quality images where details are concealed in shadows or washed out by bright lights.

High Dynamic Range (HDR) imaging techniques appear as a solution to overcome this major issue by extending the precision of the digital images. HDR imaging encodes images with higher than standard 24-bit RGB format, increasing the range of luminance that can be digitally stored. HDR images can be created in two different ways. The first method requires the development of specific HDR sensors that can capture the entire scene dynamic range. Several sensors development have been done with techniques such as well-capacity adjusting [1], time-to-saturation [2], or self-reset [3] (see [4] for a comparative analysis of these sensor architectures). To perform these functions, most of these sensors provide a processing unit at chip level or at column level [5]–[7], and even at pixel level [8]–[12]. The second method relies on conventional Low Dynamic Range (LDR) sensors to capture HDR data by recording multiple exposures of the same scene. By limiting the exposure time, the image loses low-light detail in exchange for improved details in areas of high illumination. A contrario, by increasing the exposure time, the resulting image contains the details in the dark areas but none of the details in the bright areas due to pixel saturation. Complex imaging algorithms enable to build a single HDR image (called radiance map) that covers the full dynamic range by combining the details of the successive acquisitions. An example of an HDR image obtained from two images (respectively acquired with exposure times of 0.9 ms and 9 ms) is shown in Fig. 1.

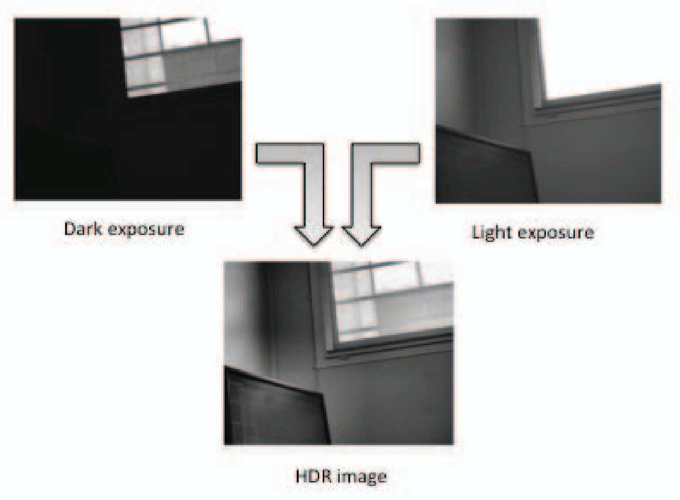

Fig. 1. Multiple exposures combined into one HDR image

HDR imaging techniques have received a lot of attention in the computer graphics community. Devebec et al. [13], Mitsunaga et al. [14], Robertson et al. [15], and Kang et al. [16] have developed various algorithms for producing HDR images from multiple exposures LDR images captured with a standard camera. Simultaneously, works on tone mapping operators [17]–[19] have been conducted in order to compress the HDR resulting images to the displayable range of 8-bit/channel monitors in such a way that the visual sensation of the real scene is faithfully reproduced. A deep study of literature, both on HDR synthesis and tone mapping techniques, highlights efficient and complex software operators that have been implemented mostly on workstations with no timing constraint. There is a need to produce real-time HDR imaging that can be embedded into a smart camera. The work

978-1-4577-1707-9/11/$26.00 ©2011 IEEE

described in this paper develops a complete algorithm pipeline for HDR imaging from multiple exposures acquisition, through radiance maps and tone mapping, to display. We present a simplified and practical hardware solution to produce real time HDR video.

The remainder of the paper is as follows: in section II, we briefly review the literature on real-time HDR imaging solutions. Section III is a theoretical introduction to HDR pipeline implementation and deals with specific issues such as automatic scene exposure, motion correction, radiance map and local/global tone mapping. From this pipeline, we propose a dedicated hardware architecture in section IV. Conclusions are provided in Section V.

II. Related work

The problems of capturing the complete dynamic range of a real scene and reducing this dynamic range to a viewable range have drawn the attention of many authors. However, the main part of the proposed algorithms have been developed without taking into account the specificities and the difficulties inherent to hardware implementations. Unfortunately, these works are not generally suitable for efficient real-time implementation on smart cameras. As a consequence, generating real-time HDR images still remains a interesting challenge.

In 2007, Hassan [20] described an FPGA-based architecture for local tone mapping of gray scale HDR images, able to generate $1,024 \times 768$-pixel image at 60 frames per second. The architecture is based on a modification of the nine-scale Reinhard's operator [21]. Several limitations can be noticed. First of all, This works focuses only on the tone-mapping process and does not care about the HDR capture, using a set of high dynamic range still images from the Debevec library [13]. Secondly, the tone mapping operator requires to store a full image before evaluating the logarithmic average of the image,leading to a video latency. This limitation can be overcome by using the logarithmic average of the previous image to normalize the current image. Thirdly, using the Gaussian pyramid requires a lot of bits per pixel, increasing the amount of onboard memory. Another real-time hardware implementation of tone mapping has been recently proposed in [22]. They use the Fattal et al. local algorithm [23]. This operator is less complex than the Reinhard's operator, then requiring less onboard memory. The key point of this works is the inexpensive hardware implementation of a simplified Poisson solver for Fattal's operator. It gives a real-time tone mapping implementation on a Stratix II FPGA operating at 100 frames per second on one megapixel images.

Ching-Te et al. [24] describe a methodology for the development of a tone-mapping processor of optimized architecture using an ARM SOC platform, and illustrate the use of this novel HDR tone-mapping processor for both photographic and gradient compression. Based on this methodology, they develop an integrated photographic and gradient tone-mapping processor that can be configured for different applications. This newly-developed processor can process $1,024 \times 768$ images at 60 fps, runs at 100 MHz clock and consumes a core area of $8.1mm^2$ under TSMC 0.13 μm technology.

Kang et al. [16] describe an algorithmic solution performing both video capture and HDR synthesis, and able to generate HDR video from an image sequence of a dynamic scene captured while rapidly varying the exposure of each frame (alternating light and dark exposures). For this purpose, the approach consists of three main parts: automatic exposure control during capture, HDR stitching across neighboring frames, and tone mapping for viewing. The implemented technique produces video with increased dynamic range while handling moving parts in the scene. However, the implementation on a 2GHz Pentium 4 machine does not reach the real-time constraint because the processing time for each video frame $(768 \times 1,024)$ is about 10 seconds (8 seconds for the radiance mapping and 2 seconds for the tone mapping). Based on Kang's algorithms, Youm et al. [25] create an HDR video by merging two images from different exposures acquired by a stationary video camera system. Their methodology mainly relies on the simple strategy of automatically controlling exposure times and effectively combines bright and dark areas in short and long exposure frames. Unfortunately, they do not reach real-time processing with approximately 2.5 seconds for each (640×480) pixel frame on a 1.53 GHz AMD Athlon XP 1800+ machine.

Finally, Ke et al. [26] propose an innovative method to generate HDR video that is displayable on a conventional monitor. This method is drastically different from state-of-the-art works because only one LDR image is enough to generate HDR-like images, with fine details and uniformly-distributed intensity. They implement a hardware-efficient virtual HDR image synthesizer that includes virtual photography and local contrast enhancement. This system has much lower complexity compared to traditional HDR generation and tone-mapping algorithm. Under UMC90nm technology, it achieves real time for 720 × 480 video frames at 60 fps.

III. Global HDR algorithm pipeline

In this section, we focus on the global HDR algorithm pipeline. Generating an HDR video sequence as well as HDR still images consists of automatically determining the optimal exposures for multiple frames capture, motion correcting between successive images, computing radiance maps, calibrating images and local/global tone mapping for viewing as depicted on Fig. 2.

A. Auto exposure control

The Auto Exposure Control (AEC) of a camera is able to measure the brightness of the scene and then computes an appropriate exposure in order to maximize the number of well-exposed pixels. However, most real scenes have a so great dynamic range that can not be captured by a camera. Therefore, some information of the scene is lost because some pixels are saturated and other ones are under-exposed. In order to acquire a wider dynamic range, a solution is to precisely monitor exposure times and capture multi exposure frames.

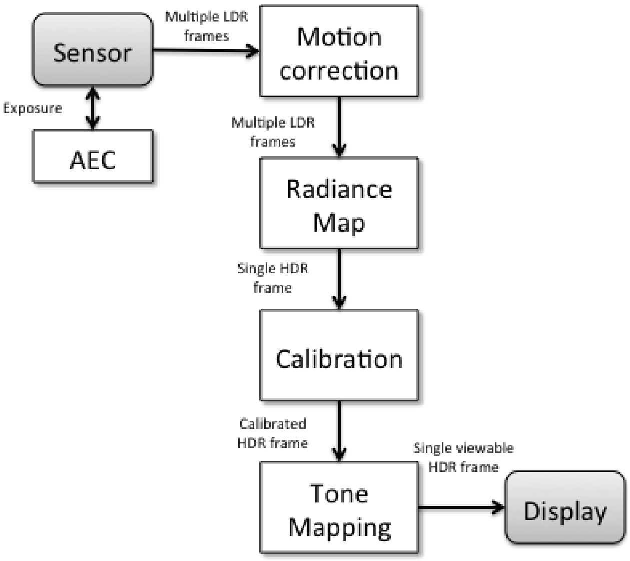

Fig. 2. Processing steps required in producing an HDR video sequence

A typical example is the autobracketing system implemented on single-lens reflex cameras, where the camera captures the same image several times with different exposure settings, both over exposed (brighter image) and under exposed (darker image) compared to a given exposure (medium image). For HDR imaging, the AEC is a crucial step in terms of recorded information and needs to precisely and continuously evaluate exposure settings in order to capture a maximum of information in the scene. The candidate exposures must be automatically computed from previous captures on the basis of pixels statistics, such as histograms, ratio of dark or bright pixels.

B. Motion estimation and image alignment

HDR imaging based on multiple exposure frames implies some limitations, especially in the context of video. Correct construction of HDR from multiple images requires that each of the images represents precisely the same scene. This is obviously the case for static scenes. However, for dynamic scenes due either to camera movement or motion of objects, merging techniques of LDR images inevitably leads to motion artifacts and ghosting. For example, if somebody is moving during the frame sequence acquisition, he may appear as a ghost in the combined HDR image due to his multiple locations in the successive frames. To improve quality, global and local displacements in LDR frames must be realigned using for instance optical flow estimation [27]. The technique described by Kang et al. [16] handles both camera movement and object movement in a scene and is based on a variant of the Lucas and Kanade motion estimation technique [28]. Another technique [29] uses a median threshold bitmap and is about 10 times faster.

The motion estimation and image alignment is a complex algorithmic step in the complete HDR pipeline and state-of-the-art works highlight difficulties to give real-time results. Consequently, in the remainder of this paper, we will only focus on static scenes that do not require motion estimation and image alignment. This optional phase will be later included in our hardware pipeline.

C. HDR creating

The pixel values of an image are the result of a non linear function of the exposure. This function is linked to the characteristic curve (the response to the variations in exposure) of the sensor. So, creating an HDR image from multiple LDR images, i.e producing a radiance map requires two major stages:

- Recovering the sensor response function;
- Reconstructing the HDR radiance map.

From literature, three most popular algorithms for recovering the response curve can be extracted: Debevec et al. [13], Mitsunaga et al. [14], and Robertson et al. [30]. A technical paper [31] compares the first two algorithms in a real-time implementation in software. The results of time calculation for an image are significantly identical. For a detailed description of these methodologies, see Granados et al. [32]. In the remainder of this paper, we only presents the original algorithm developed by Debevec et al. [13]. According to these authors, the film reciprocity equation is:

$$Z_{ij} = f(E_i \Delta t_j) \tag{1}$$

Where E_i is the irradiances, Z_{ij} is the pixel value of pixel location number i in image j and Δt_{ij} is the exposure duration. The function g is defined as $g = \ln f^-1$. The response curve g can be determined by resolving the following quadratic function:

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2 \tag{2}$$

Where $g(z)$ is the log exposure corresponding to pixel value z and λ is a weighting scalar depending on the amount of noise expected on g. This leads to 1024 values of $g(z)$, minimizing the equation 2. The response curve can be evaluated before the capture process, from a sequence of several images. The curve can be used to determinate radiance values in any image acquired during the video streams. For Schubert et al. [33], the HDR reconstruction requires only two images for the reconstruction HDR. The following equation can be used:

$$\ln E_i = \frac{\sum_{j=1}^{P} \omega(Z_{ij})(g(Z_{ij}) - \ln \Delta t_{ij})}{\sum_{j=1}^{P} \omega(Z_{ij})} \tag{3}$$

Where $\omega(z)$ is the weighting function. It is a simple hat equation corresponding to the weighting function $\omega(z)$:

$$\omega(z) = \begin{cases} z - Z_{min} \text{ for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z \text{ for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \tag{4}$$

D. Calibration

Reinhard et al. [34] explain that a supplementary step dedicated to calibration is needed before performing the tone mapping operation. Indeed, because human visual response to lighting conditions is non linear, tone reproduction operators must have precise information about the light conditions to differentiate for example between day scene or night scene. Then, HDR images must be expressed in real world units (candelas per square meter) and not in arbitrary units. For this purpose, the log average luminance found in the image can be evaluated:

$$L_{av} = \exp\left(\frac{1}{N}\sum_{x,y}\log(L_W(x,y))\right) \tag{5}$$

Where N is the number of pixels, $L_W(x,y)$ is the radiance value of the pixel (x,y).

From L_{av}, we can estimate f and α with the following equations:

$$f = \frac{2\log_2 L_{av} - \log_2 L_{min} - \log_2 L_{max}}{\log_2 L_{max} - \log_2 L_{min}} \tag{6}$$

$$\alpha = 0.18 \times 4^f$$

f is the computed distance of the log average luminance L_{av} relative to the minimum luminance L_{min} and the maximum luminance L_{max}. The α parameter can be seen as an unitless number that relates to the overall light level. α is finally used to estimate the calibrated pixels $L'_W(x,y)$ of the HDR image from the uncalibrated pixels $L_W(x,y)$:

$$L'_W(x,y) = \frac{\alpha}{L_{av}} L_W(x,y) \tag{7}$$

E. Tone mapping

Tone mapping is the last step of the HDR pipeline. To reproduce HDR radiance maps on LDR display devices, it's necessary to convert the HDR values to 8-bit integer values in such a way that all the details are still faithfully reproduced. Many tone mapping algorithms for compressing and displaying HDR images have been proposed. They are generally classified into two category: global tone mapping [21], [35], [36], which applies a single fixed mapping function to each pixel in an image, and local tone mapping [23], [37], [38], which adapts the mapping function used for a pixel depending on the neighborhood of the pixel. As explained into [36], global tone mapping operators are computationally very simple and preserve the intensity orders of the original scenes. However, global tone mapping operators generally cause loss of details in the mapped images, so local tone mapping operators are applied, at a higher cost, to better preserve the details and local contrast in the images, as local tone mapping technologies consider pixel neighborhood information in the mapping processing for each individual pixel.

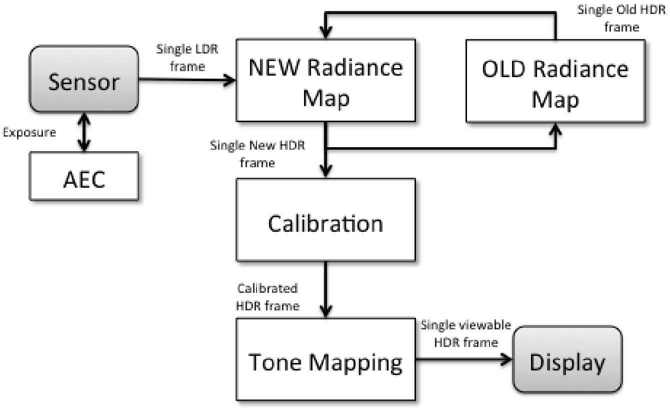

Fig. 3. The proposed hardware pipeline for HDR video computation.

IV. Proposed hardware architecture

The block diagram of our HDR imaging hardware architecture is shown in Fig. 3 and is made up of 5 main blocks. At the input, the AEC block communicates with the sensor in order to acquire the multi exposure LDR frames. Each single LDR frame is then combined with the old radiance map into a new radiance map, that is further calibrated and tone mapped for display. The new radiance map will become the old radiance map for the next iteration.

This block diagram is significantly different of the above-mentioned HDR pipeline in the Section III because of its costly processing time that can not perform HDR video in real time. We must simplify the algorithms pipeline and propose innovative solutions that are compatible with hardware resources to achieve a real-time HDR video at 25 frames per second. For this purpose, our hardware pipeline does not require the storage of different multi exposure LDR frames before reconstructing the HDR frame. The new radiance map block continuously processes the incoming flow of pixel from the sensor to produce an HDR frame. This new radiance map is then stored in a specific memory and will serve as the old radiance map at the next iteration.

To sum up, our processing pipeline does not reconstruct an HDR image from several LDR stored frames as classically shown in the literature. It rather aims to deal with pixel flow processing by continuously updating the current radiance map with new data acquired from the sensor.

A. High dynamic range capabilities sensor

The EV76C560 sensor [39] provided by e2v is a 1.3 million pixels (1280 × 1024) CMOS image sensor. It is suitable for many different types of applications where superior performance are required, specifically in low-light conditions. It offers a high 10-bit digital readout speed at 60 fps in full resolution. It includes some features like multiple regions of interest, defective pixels correction, sub-sampling,... It also embeds some basic image processing functions such as image histograms, evaluation of the number of low saturated pixels (at 0 value) and high saturated pixels (at 1023 value). The dynamic range is approximatively 62dB, so this sensor can be considered as a low dynamic range imaging system. To improve its HDR capabilities, this sensor is able to successively acquire 2 images with different integration times, the second acquisition simultaneously occurring with the first frame readout, as depicted in Fig. 4. Such integrated features minimize artefacts such as blurring for dark scenes or ghosts for moving scenes.

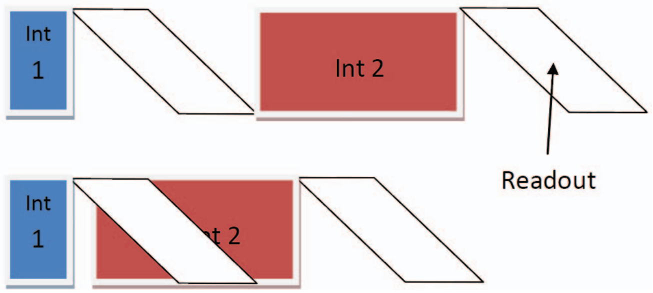

Fig. 4. High dynamic range capabilities of the e2v sensor. The standard integration is on top and the high dynamic integration is at the bottom.

B. Auto exposure control

Our exposure control algorithm is based on pixels statistics such as the histograms of the two successive images. The EV76C560 is able to automatically computes a 64 16-bit categories histogram, the number of dark pixels (S_{low}), and the number of saturated pixels (S_{high}) for each acquired frame. Such complementary information can be read before two frames readout. Our aim is to calculate the number of pixels in the upper and lower part of the histograms of the two images as:

$$P_{t',s} = \sum_{h=1}^{h=32} \frac{H_{t',s}(h)}{N - S_{low}} \tag{8}$$

$$P_{t',l} = \sum_{h=33}^{h=64} \frac{H_{t',l}(h)}{N - S_{high}} \tag{9}$$

P is the proportion of pixels located in a part of the histogram. s meens the short exposure, l meens the long exposure, H is the histogram of the image, h is the position of the categorie (1 - 64), S is the number of saturated pixels (dark pixel at level 0 or bright pixel at level 1023) and N is the total number of pixels. The better exposure time for each image can be evaluated as follows:

$$\Delta t_{t',s} = \Delta t_{t'-1,s} \cdot (\frac{\Delta t_{opt,s}}{P_{t',s}}) \tag{10}$$

$$\Delta t_{t',l} = \Delta t_{t'-1,l} \cdot (\frac{\Delta t_{opt,l}}{P_{t',l}}) \tag{11}$$

$\Delta t_{t'}$ is the exposure time at time t'. Δt_{opt} depends on the number of saturated pixels. It is not possible to precisely evaluate the best exposure for a real scene because the tonemapped HDR image rendering depends on many factors (number of images, type of the scene, sensor sensitivity, etc.). In order to evaluate as precisely as possible the best values of the integration times, the following assumption was made: 80% of the pixels in the lower part of the histogram are present in the low exposure image, and 80% of the pixels in the upper part of the histogram are present in the high exposure image.

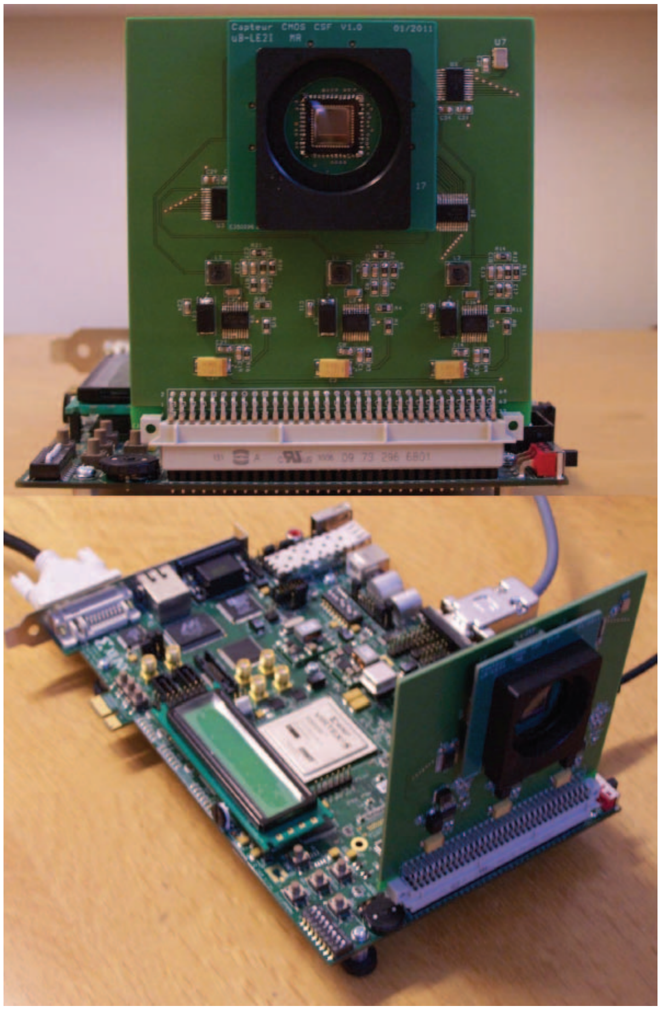

Fig. 5. The hardware architecture: an FPGA (ml507 Xilinx card) and a daugther card with e2v sensor.

To prevent an useless updating system for every frames, we introduce a threshold so that the system is not too slow for our architecture:

$$\Delta t_{sensor} = \begin{cases} \Delta t_{t'} \text{ for } \Delta t_{t'} - \Delta t_{t'-1} \geq \text{threshold} \\ \Delta t_{t'-1} \text{ for } \Delta t_{t'} - \Delta t_{t'-1} < \text{threshold} \end{cases} \tag{12}$$

C. Radiance map flow system

The goal of our algorithm is to perform HDR creating and tonemapping directly from the entering pixel stream (see Fig. 3). Our exposure control algorithm computes and programs the sensor with the two exposure times. The sensor sequentially sends to the FPGA two images with two different exposure times.

An initialization step is used to calculate the response curve of our system. This curve will allow us to convert each pixel in radiance value. An HDR image from two exposures is then calculated. This radiance map is stored in DDRII memory. Once these initialization steps are completed, we calculate a new value of radiance map from the old radiance map previously calculated and an LDR image provided by the sensor (as shown in Fig. 3). For this purpose, each pixel from the sensor is converted into radiance values with the following equation:

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j \tag{13}$$

The system response curve g previously calculated allows the pixel to be converted into the radiance value. Then, the pixel is associated with the pixel of the previous frame using a weighting system. The construction of the new value of the pixel radiance is then processed. This pixel is represented by a value with a large number of bits for each pixel. In our algorithm pipeline, the pixel will be tonemapped to reduce the pixel size to 8bits.

The advantage of our technique is to store only one image (HDR) in memory, and avoid the waiting of the two images to compute one HDR image. With this technique, one HDR frame needs 2 memory accesses (one read and write operation) whereas the traditional technique requires at least four memory accesses (two read and write operations).

D. Tone mapping

Before performing the tone mapping operation, the calibration of each pixel is processed as seen earlier in Section III-D. L_{av}, L_{max} and L_{min} are evaluated from the pixel flow of the previous image (for details, see equations 5, 6, and 7).

Cadik et al. [19] show that the global part of a tone mapping operator is most essential to obtain goods results. Moreover, a global tone mapper is the easiest way to implement the algorithm in real time, because local operators require complex computations, and also may generate halo artifacts. The choice of a candidate tone mapping operator has been done after comparing some global algorithms applied to a radiance map constructed from two images. Several global algorithms have been extensively tested in C++ (see Fig. 6). According to our temporal and hardware constraints, the best compromise is the global tone mapper introduced by Duan et al. [36]. This tone mapper is defined with the following equation:

$$\begin{aligned} D(I) &= C * (D_{max} - D_{min}) + D_{min} \\ with\ C &= \frac{\log(I+\tau) - \log(I_{min}+\tau)}{\log(I_{max}+\tau) - \log(I_{min}+\tau)} \end{aligned} \tag{14}$$

where D is the dispayable luminance and I a radiance value of the HDR frame. I_{min} and I_{max} are the minimum and maximum luminance of the scene and τ is the overall brightness control of the tonemapped image.

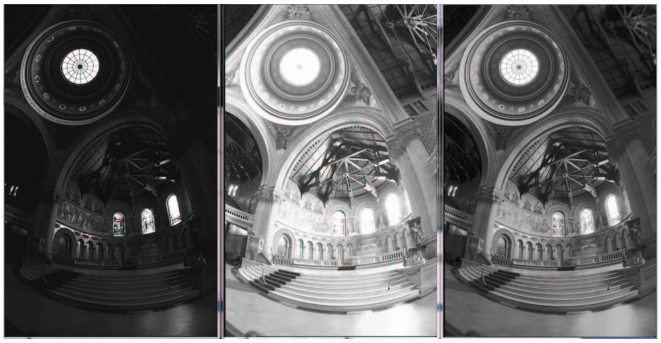

Fig. 6. Simulation of the global Duan et al. algorithm [36] in C++. Two consecutive exposures captured at immediate time steps t1, t2 contain different luminance ranges of a scene. The result on the right is the tonemapped image.

The evaluation of this equation require to have a full HDR stored frame. However, as our hardware pipeline processes HDR imaging on the pixel stream, the terms I_{min}, I_{max}, D_{min}, D_{max} are computed from the previous HDR frame. This choice can be easily justifiable because the vision system takes some time to adapt to the global brightness of the scene.

E. Dedicated embedded camera with FPGA

The complete hardware architecture of the smart camera is built around the e2v sensor and a Xilinx ML507 development board [40] (including a Virtex-5 FPGA). A daughter card hosting the e2v sensor has been specifically designed and is connected with the headers of the FPGA card (see Fig. 5). A DDRII SDRAM memory is used to store the HDR images. The DVI output of the FPGA board is connected to an external LCD monitor to display the HDR images.

V. Conclusion

In this paper, we present a complete hardware pipeline dedicated to HDR video that can fulfill drastic realtime constraints while satisfying image quality requirements.

After a detailed study of the conventional algorithms pipeline of HDR imaging, we proposed an innovative pipeline architecture, with its simplicity and high operating frequency. The key point of this new vision system is that it directly operates on the stream of pixels without storing multiple LDR frames. Secondly, we also propose a hardware vision system based on a intrinsically high dynamic range image sensor associated with a FPGA development board.

Development of the HDR pipeline is currently in a simulation phase and a first hardware implementation is underway. We are also planning to develop and benchmark other several state-of-the-art algorithms both for computing radiance maps, calibrating HDR images, and local/global tone mapping. Development of motion correction and alignment of images are also considered. To sum up, this work forms the basis of a powerful realtime architecture dedicated to HDR video.

References

[1] S. Decker, D. McGrath, K. Brehmer, and C. Sodini, "A 256 × 256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2081 –2091, Dec 1998.

[2] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi, and G.-F. Dalla Betta, "A 138 dB dynamic range CMOS image sensor with new pixel architecture," in *IEEE International Solid-State Circuits Conference (ISSCC), Digest of Technical Papers*, vol. 1, Feb 2002, pp. 40–442.

[3] L. McIlrath, "A low-power low-noise ultrawide-dynamic-range CMOS imager with pixel-parallel A/D conversion," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 5, pp. 846 –853, May 2001.

[4] S. Kavusi and A. El Gamal, "A quantitative study of high dynamic range image sensor architectures," in *Proceedings of the SPIE Electronic Imaging '04 Conference*, vol. 5301, Jan 2004, pp. 264–275.

[5] O. Yadid-Pecht and A. Belenky, "In-Pixel Autoexposure CMOS APS," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 8, pp. 1425–1428, Aug 2003.

[6] P. Acosta-Serafini, M. Ichiro, and C. Sodini, "A 1/3" VGA linear wide dynamic range CMOS image sensor implementing a predictive multiple sampling algorithm with overlapping integration intervals," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1487–1496, Sept 2004.

[7] M. Sakakibara, S. Kawahito, D. Handoko, N. Nakamura, M. Higashi, K. Mabuchi, and H. Sumi, "A high-sensitivity CMOS image sensor with gain-adaptative column amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 5, pp. 1147–1156, May 2005.

[8] A. Krymsky and T. Niarong, "A 9-V/Lux 5000-frames/s 512 × 512 CMOS sensor," *IEEE Transactions on Electron Devices*, vol. 50, no. 1, pp. 136–143, Jan 2003.

[9] G. Cembrano, A. Rodriguez-Vazquez, R. Galan, F. Jimenez-Garrido, S. Espejo, and R. Dominguez-Castro, "A 1000 FPS at 128 × 128 vision processor with 8-bit digitized I/O," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1044–1055, Jul 2004.

[10] L. Lindgren, J. Melander, R. Johansson, and B. Mller, "A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1350–1359, Jun 2005.

[11] Y. Sugiyama, M. Takumi, H. Toyoda, N. Mukozaka, A. Ihori, T. kurashina, Y. Nakamura, T. Tonbe, and S. Mizuno, "A high-speed CMOS image sensor with profile data acquiring function," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2816–2823, Dec 2005.

[12] J. Dubois, D. Ginhac, M. Paindavoine, and B. Heyrman, "A 10 000 fps CMOS sensor with massively parallel image processing," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 706–717, Mar 2008.

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

[14] T. Mitsunaga and S. Nayar, "Radiometric Self Calibration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun 1999, pp. 374–380.

[15] 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, 2003.

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

[17] K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer, "Tone reproduction and physically based spectral rendering," in *Eurographics 2002*. Eurographics Association, Sep 2002, pp. 101–123.

[18] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, pp. 13 – 24, 2005.

[19] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi, "Evaluation of HDR tone mapping methods using essential perceptual attributes," *Computers & Graphics*, vol. 32, pp. 330–349, 2008.

[20] F. Hassan and J. E. Carletta, "A real-time FPGA-based architecture for a Reinhard-like tone mapping operator," in *Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, Aire-la-Ville, Switzerland, Switzerland, 2007, pp. 65–71.

[21] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 267–276, 2002.

[22] L. Vytla, F. Hassan, and J. Carletta, "A real-time implementation of gradient domain high dynamic range compression using a local poisson solver," *Journal of Real-Time Image Processing*, pp. 1–15, 2011, doi: 10.1007/s11554-011-0198-5.

[23] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 249–256, July 2002.

[24] C.-T. Chiu, T.-H. Wang, W.-M. Ke, C.-Y. Chuang, J.-S. Huang, W.-S. Wong, R.-S. Tsay, and C.-J. Wu, "Real-time tone-mapping processor with integrated photographic and gradient compression using 0.13μm technology on an ARM SoC platform," *Journal of Signal Processing Systems*, vol. 64, no. 1, pp. 93–107, 2010.

[25] S.-J. Youm, W. ho Cho, and K.-S. Hong, "High dynamic range video through fusion of exposured-controlled frames," in *Proceedings of IAPR Conference on Machine Vision Applications*, 2005, pp. 546–549.

[26] W.-M. Ke, T.-H. Wang, and C.-T. Chiu, "Hardware-efficient virtual high dynamic range image reproduction," in *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP'09)*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2665–2668.

[27] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proceedings of the Second European Conference on Computer Vision (ECCV 92)*. London, UK: Springer-Verlag, 1992, pp. 237–252.

[28] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, vol. 2. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[29] G. Ward, "Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures," *Journal of Graphics, GPU, and Game tools*, vol. 8, no. 2, pp. 17–30, 2003.

[30] 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, 2003.

[31] W. S. G. Yourganov, "Acquiring high dynamic range video at video rates," Dept. of Computer Science, York University, Tech. Rep.

[32] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H. Seidel, and H. Lensch, "Optimal hdr reconstruction with linear digital cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 215–222.

[33] F. Schubert, K. Schertler, and K. Mikolajczyk, "A hands-on approach to high-dynamic-range and superresolution fusion," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2009, pp. 1–8.

[34] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. San Francisco, CA, USA: Morgan Kaufmann Publishers, December 2005.

[35] F. Drago, W. L. Martens, K. Myszkowski, and H.-P. Seidel, "Perceptual evaluation of tone mapping operators," in *ACM SIGGRAPH 2003 Sketches & Applications*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 1–1.

[36] J. Duan, M. Bressan, C. Dance, and G. Qiu, "Tone-mapping high dynamic range images by novel histogram adjustment," *Pattern Recognition*, vol. 43, pp. 1847–1862, May 2010.

[37] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, pp. 257–266, July 2002.

[38] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, "A perceptual framework for contrast processing of high dynamic range images," in *Proceedings of the 2nd symposium on Applied perception in graphics and visualization (APGV)*, New York, NY, USA, 2005, pp. 87–94.

[39] e2v technologies. (2009) Ev76c560 BW and colour CMOS sensor. [Online]. Available: http://www.e2v.com/products-and-services/high-performance-imaging-solutions/industrial-imaging/industrial-imaging-sensors/datasheets—sensors/

[40] Xilinx. (2009) Virtex-5 fxt fpga ml507 evaluation platform. [Online]. Available: http://www.xilinx.com/products/devkits/HW-V5-ML507-UNI-G.htm