

# HDR-ARtiSt: an adaptive real-time smart camera for high dynamic range imaging

Pierre-Jean Lapray · Barthélémy Heyrman ·  
Dominique Ginjac

Received: 2 September 2013 / Accepted: 20 December 2013  
© Springer-Verlag Berlin Heidelberg 2014

**Abstract** This paper describes a complete FPGA-based smart camera architecture named HDR-ARtiSt (High Dynamic Range Adaptive Real-time Smart camera) which produces a real-time high dynamic range (HDR) live video stream from multiple captures. A specific memory management unit has been defined to adjust the number of acquisitions to improve HDR quality. This smart camera is built around a standard B&W CMOS image sensor and a Xilinx FPGA. It embeds multiple captures, HDR processing, data display and transfer, which is an original contribution compared to the state-of-the-art. The proposed architecture enables a real-time HDR video flow for a full sensor resolution (1.3 Mega pixels) at 60 frames per second.

**Keywords** Smart camera · High dynamic range, memory management core · Parallel processing · FPGA implementation

## 1 Introduction

Standard cameras capture only a fraction of the information that is visible to the human eye. This is specifically true for natural scenes including areas of low and high illumination due to transitions between sunlight and shaded areas. When capturing such a scene, many cameras are unable to store the full Dynamic Range (DR) resulting in low-quality video where details are concealed in shadows or washed out by sunlight. High Dynamic Range (HDR) imaging

techniques appear as a solution to overcome this issue by encoding digital video with higher than standard 24-bit RGB format, and then increasing the range of luminance that can be stored. Reinhard et al. [35] provide an exhaustive review, highlighting various fields of applications of HDR cameras. For example, HDR capturing techniques are essential for outdoor security applications in which unpredictable illumination changes may affect performance algorithms [5, 28]. Similarly, HDR techniques should facilitate object tracking or automotive applications [24, 39, 42] under uncontrolled illumination. Recent research programs on machine vision have clearly demonstrated the benefits of real-time HDR vision [20, 37]. Finally, medical applications require high-precision images and HDR lighting techniques to improve the rendering quality of imaging systems [17, 25].

There are two major approaches to create HDR content: to develop new HDR sensors or to combine multiple exposure frames captured by conventional low dynamic range (LDR) sensors. First, several HDR sensors have been designed with techniques such as well-capacity adjusting, time-to-saturation, or self-reset (see [22] for a comparative analysis). Most of these sensors are tailor-made and provide dedicated processing units to extend DR [1, 7, 12, 26, 30, 38].

The second method relies on conventional off-the-shelf LDR sensors to capture the HDR data by recording multiple acquisitions of the same scene while varying the exposure time [9, 21, 29, 36]. By limiting the exposure time, the image loses low-light detail in exchange for improved detail in bright areas. By increasing the exposure time, the image is only detailed in the dark areas because of the pixel saturation in bright areas. Each pixel is at least properly exposed in one image and is under or overexposed in other images of the sequence. The images are then

---

P.-J. Lapray · B. Heyrman · D. Ginjac (✉)  
Laboratory of Electronic, Computing and Imaging Sciences,  
Le2i UMR 6306, University of Burgundy, Dijon, France  
e-mail: dginjac@u-bourgogne.fr



**Fig. 1** HDR frame of a real scene captured with three different exposure times (sequences by Fattal et al. [14]). **a** Low exposure, **b** middle exposure, **c** high exposure, **d** HDR image

combined into a single HDR frame (i.e., a radiance map). Finally, since current display technology has a limited DR, HDR images need to be compressed by tone mapping operators [2, 3, 6, 10, 27, 33] in such a way that the visual sensation of the real scene is faithfully reproduced, as depicted in Fig. 1.

This paper presents a complete FPGA-based smart camera architecture named HDR-ARtiSt (High Dynamic Range Adaptive Real-time Smart camera). This smart camera is able to provide a real-time HDR live video from multiple exposure capturing to display through radiance maps and tone mapping. The main contribution of this work is the generation of a new FPGA-embedded architecture producing an uncompressed B&W  $1,280 \times 1,024$ -pixel HDR live video at 60 fps. An embedded DVI controller is also provided to display this HDR live video on a standard LCD monitor. The HDR-ARtiSt camera could obviously embed some complex image processing applications onto the FPGA or could be connected to a more standard PC managing the video stream.

The remainder of the paper is as follows: in Sect. 2, we briefly review the literature on existing HDR systems. Section 3 describes our proposed hardware architecture, highlighting the multi-streaming memory management unit designed to address the computation capacity and memory bandwidth limitations. A first implementation using a two-frame acquisition is presented in Sect. 4. Based on a

detailed study of the visual quality of this implementation, an improved three-frame solution is described in Sect. 5. Finally, Sect. 6 concludes this paper and outlines directions for future work.

## 2 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 has been developed without taking into account the specifications 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 live video remains an interesting challenge.

### 2.1 Existing hardware architectures

In 2007, Hassan and Carletta [19] 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 [34] operator. Several limitations can be noticed. Firstly, this work focuses only on the tone-mapping process and does not care about the HDR capture, using a set of HDR still images from the Debevec library [9]. 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 using the logarithmic average of the previous image to normalize the current image. Finally, 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 by Vytla et al. [47]. They use the Fattal et al. [14] local algorithm. This operator is less complex than the Reinhard's operator, then requiring less onboard memory. The key point of this work 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 with one megapixel image resolution.

Chiu et al. [8] 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

**Table 1** List of commonly used terms and variables

$P$	Number of frames to create one HDR frame
$I_p$	$p$ th frame in the sequence of $P$ frames
$M$	Frame height (i.e., row number)
$N$	Frame width (i.e., column number)
$Z_{ij}^p$	Luminance of the pixel $(i, j)$ in the $p$ th frame
$E_{ij}$	Luminance (or radiance) of the HDR pixel $(i, j)$
$g$	Camera transfer function (CTF)
$\Delta t_p$	Exposure time of the $p$ th frame
$D_{ij}$	Tone-mapped pixel
$D_{\min}, D_{\max}$	Minimum and maximum values of the display devices
$\tau$	Overall brightness of the mapped frame
$L_m$	Line index
W&R	Memory Write and Read operations

process  $1,024 \times 768$ -pixel images at 60 fps, runs at 100 MHz clock and consumes a core area of  $8.1 \text{ mm}^2$  under a TSMC  $0.13\text{-}\mu\text{m}$  technology.

Kang et al. [21] describe an algorithmic solution computing both video capture and HDR synthesis, and able to generate HDR video from an image sequence of a dynamic scene captured while varying the exposure at 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 neighbouring 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 2 GHz Pentium 4 machine does not reach the real-time constraint because the processing time for each  $1,024 \times 768$ -pixel video frame is about 10 s (8 s for the radiance mapping and 2 s for the tone mapping). Based on Kang’s algorithms, Youm et al. [51] 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 tactic 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 about 2.5 s for each  $640 \times 480$ -pixel frame on a 1.53 GHz AMD Athlon machine.

Finally, Ke et al. [23] propose an innovative method to generate HDR video. This method differs drastically with the above-mentioned state-of-the-art works because only one LDR image is enough to generate HDR-like images, with fine details and uniformly distributed intensity. To obtain such a result, they implement a hardware-efficient virtual HDR image synthesizer that includes virtual photography and local contrast enhancement. Under a UMC 90-nm CMOS technology, it achieves real time for  $720 \times 480$ -pixel video frames at 60 fps.

## 2.2 Efficient algorithms for HDR video

Generating an HDR live video stream as well as HDR still images consists of automatically determining the optimal exposures for multiple frames capture, computing radiance maps, and local/global tone mapping for viewing on a standard LCD monitor.

Table 1 is the list of common terms and variables used in the equations of the following sections.

### 2.2.1 Step 1: HDR creating

Digital cameras have limited dynamic range since they can only capture from 8-bit to 14-bit images, mainly due to the limitations of the analog to digital converters in terms of noise levels. The most common method to generate HDR content is to capture multiple images of a same scene with different exposure times. If the camera has a linear response, we can easily recover the HDR luminance  $E_{ij}$  from each luminance  $Z_{ij}^p$  and exposure times  $\Delta t_p$  stored in each frame  $p$ . Unfortunately, cameras do not have a linear response (i.e.,  $Z_{ij}^p$  is not proportional to  $E_{ij}$  and  $\Delta t_p$ ), and we have to estimate the non-linear camera transfer function (CTF) called  $g$  to combine properly the different exposures.

Three popular algorithms for recovering this camera transfer function can be extracted from literature: Debevec and Malik [9], Mitsunaga and Nayar [29], and Robertson et al. [36]. According to the detailed description of these methodologies and the comparison of their real-time software implementations, we decided to use the Debevec’s method. The main advantage of this approach is that there is very little constraint about the response function (other than its invertibility). Moreover, the proposed algorithm has proved to be quite robust and easy to use, due to the simplicity of equations [15, 18]. The CTF function is evaluated from the film reciprocity equation  $f$ :

$$Z_{ij}^p = f(E_{ij}\Delta t_p) \tag{1}$$

The CTF function  $g$  is defined as  $g = \ln f^{-1}$  and can be obtained by minimizing the following function:

$$\begin{aligned} \mathcal{O} = & \sum_{i=1}^M \sum_{j=1}^N \sum_{p=1}^P [g(Z_{ij}^p) - \ln E_{ij} - \ln \Delta t_p]^2 \\ & + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} g''(z)^2, \end{aligned} \tag{2}$$

where  $\lambda$  is a weighting scalar depending on the amount of noise expected on  $g$ ,  $Z_{\min}$  and  $Z_{\max}$  are, respectively, the lowest and the greatest pixel values. The evaluation of  $g$  only requires the evaluation of a finite number of values between  $Z_{\min}$  and  $Z_{\max}$  (typically 1,024 values for a 10-bit precision sensor). These values can be preliminary evaluated from a sequence of several images, then stored

in the camera, and reused further to convert pixel values. For recovering the HDR radiance value of a particular pixel, all the available exposures of this pixel are combined using the following equation [41]:

$$\ln E_{ij} = \frac{\sum_{p=1}^P \omega(Z_{ij}^p) [g(Z_{ij}^p) - \ln \Delta t_p]}{\sum_{p=1}^P \omega(Z_{ij}^p)}, \tag{3}$$

where  $\omega(z)$  is a weighting function giving higher weight to values closer to the middle of the function:

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

Originally, the Debevec’s method has been developed for photography but according to Yourganov and Stuerzlinger [15], this method can be easily applied to digital video, both for static and dynamic scenes, if captures are fast enough that light changes can be safely ignored. Consequently, such a method is widely used to produce HDR video, by capturing frames with alternating bright and dark exposures, as pointed by Tocci et al. [46].

### 2.2.2 Step 2: Tone mapping

The HDR pixels are represented by a high bit-depth conflicting with the standard display devices, requiring a high to low bit-depth tone mapping. Čadež et al. [6] show that the global part of a tone mapping operator is essential to obtain good results. A psychophysical experiment by Yoshida et al. [50], based on a direct comparison among the appearances of real-world HDR images, shows that global methods like Drago et al. [13] or Reinhard et al. [34] are perceived as the most natural ones. Moreover, a global tone mapper is the easiest way to reach real-time constraints because local operators require more complex computations. The choice of a candidate tone mapping operator has been done after comparing and intensively testing several C++ global algorithms applied to a radiance map constructed from two or three images. Results are provided in Table 2. According to our temporal and hardware constraints, the best compromise is the global tone mapper from Duan et al. [11]. As an illustration, the HDR image depicted on the Fig. 1 has been obtained with this algorithm. This tone mapper compresses the luminance of the HDR pixel  $E_{ij}$  to a displayable luminance  $D_{ij}$  with the equation:

$$D_{ij} = C * (D_{\max} - D_{\min}) + D_{\min}$$

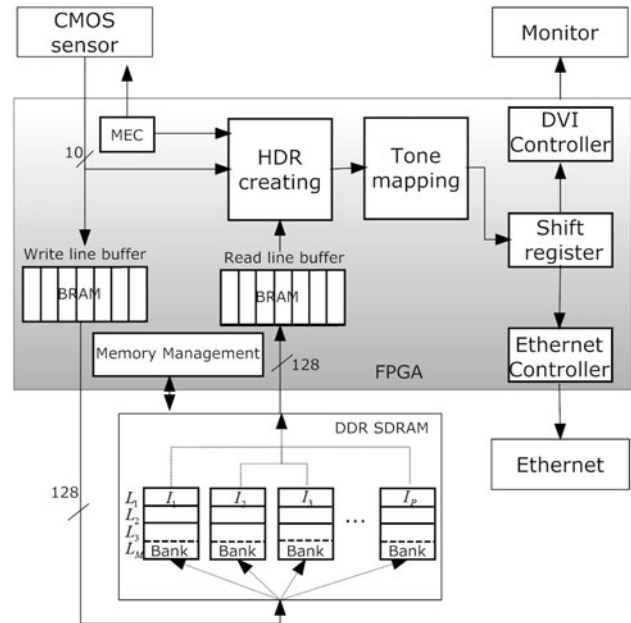
$$\text{with } C = \frac{\ln(E_{ij} + \tau) - \ln(E_{ij(\min)} + \tau)}{\ln(E_{ij(\max)} + \tau) - \ln(E_{ij(\min)} + \tau)}, \tag{5}$$

where  $E_{ij(\min)}$  and  $E_{ij(\max)}$  are the minimum and maximum luminance of the scene and  $\tau$  is inversely linked to the

**Table 2** Comparison metrics for TMOs algorithms applied to a common HDR image (memorial by Debevec) constructed from three images

TMO	PSNR	UQI	SSIM	NRMSE	Time (s)
Drago et al. [13]	34.93	0.54	0.58	0.1	5.43
Duan et al. [11]	42.34	0.97	0.91	0.01	5.71
Reinhard et al. [34]	29.11	0.89	0.8	0.32	5.52
Schlick [40]	40.71	0.43	0.56	0.03	5.50
Pattanaik et al. [32]	42.01	0.14	0.29	0.02	5.59

Exposure times used are 32, 1 s and 31, 25 m s



**Fig. 2** Overview of the HDR-P video system architecture

brightness of the mapped image. Increasing  $\tau$  makes darker images while lower values give brighter images.

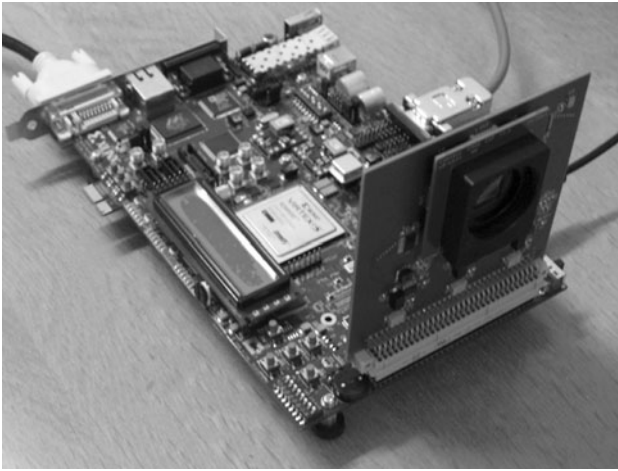
## 3 Architecture of the HDR-ARTiSt platform

To compute real-time HDR algorithms, a dedicated FPGA-based smart camera architecture was designed to address the computation capacity and memory bandwidth requirement (see Fig. 2 for an overview). This architecture does not put any restriction on the number of frames used for HDR creating. In the remainder of this paper, this generic architecture will be shortly called HDR-P, where  $P$  is the number of frames.

### 3.1 Global hardware architecture

The HDR-ARTiSt platform is a smart camera built around a Xilinx ML507 board, equipped with a Xilinx Virtex-5





**Fig. 3** HDR-P video hardware is prototyped on a Xilinx Virtex-5 ML507 FPGA board and a daughter card with the EV76C560 1.3-Mpixel sensor

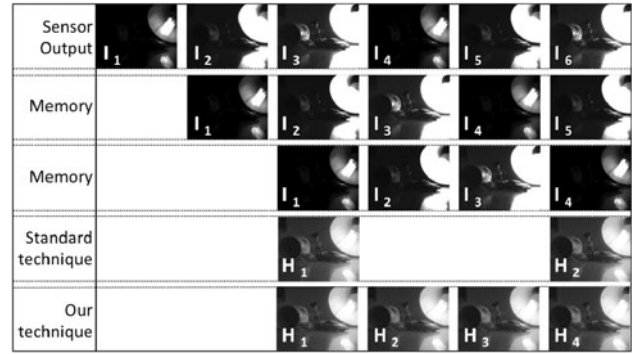
XC5VFX70T FPGA (see Fig. 3). The motherboard includes a 256 MB DDR2 SDRAM memory used to buffer the multiple frames captured by the sensor. Several industry-standard peripheral interfaces are also provided to connect the system to the external world. Among these interfaces, our vision system implements a DVI controller to display the HDR video on an LCD monitor. It also implements an Ethernet controller to store frames on a host computer.

A custom-made PCB extension board has been designed and plugged into the FPGA board to support the Ev76c560 image sensor, a 1, 280 × 1, 024-pixel CMOS sensor from e2v [45]. It offers a 10-bit digital readout speed at 60 fps in full resolution. It also embeds some basic image processing functions such as image histograms, evaluation of the number of low and high saturated pixels. Each frame can be delivered with results of these functions encoded in the video data stream header.

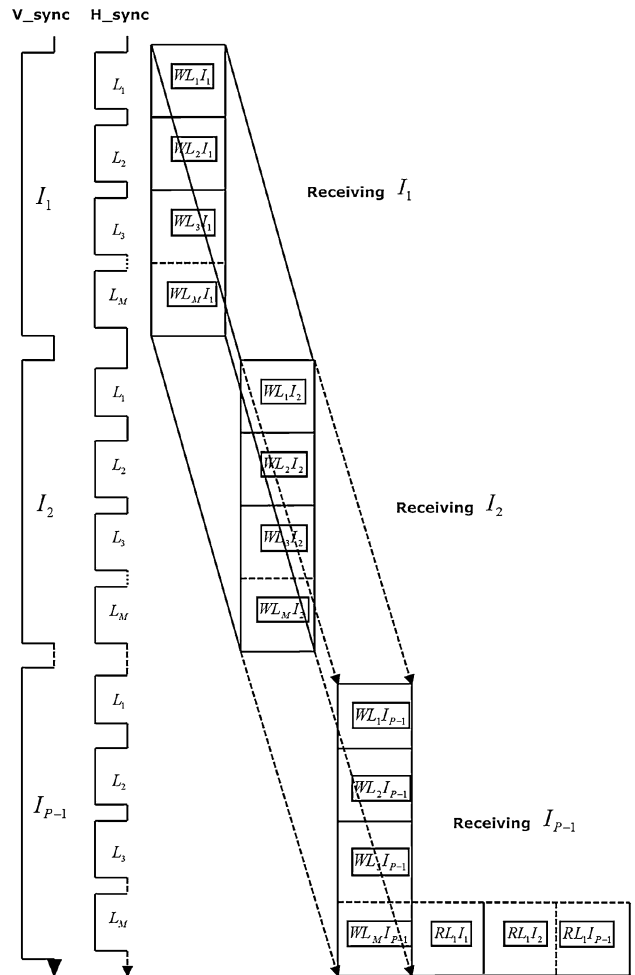
### 3.2 Multi-streaming memory management unit

Standard HDR techniques require two sequential steps: (1)  $P$  single frames must be captured and stored into memory, and (2) the HDR frame can be computed. The main drawback is the limited output frame rate. As illustrated in Fig. 4, with  $P = 3$  single frames (low, middle and high exposures) captured at 60 fps, the resulting HDR video is dramatically limited to 20 fps: the first HDR frame  $H_1$  is computed from the frames  $I_1, I_2,$  and  $I_3$ , the second HDR frame  $H_2$  from  $I_4, I_5,$  and  $I_6$ .

To overcome this limitation, we propose a specific Memory Management Unit (MMU) able to continuously build a new HDR frame at the sensor frame rate from the  $P$  previous frames. As seen in Fig. 4, the HDR frame  $H_2$  is

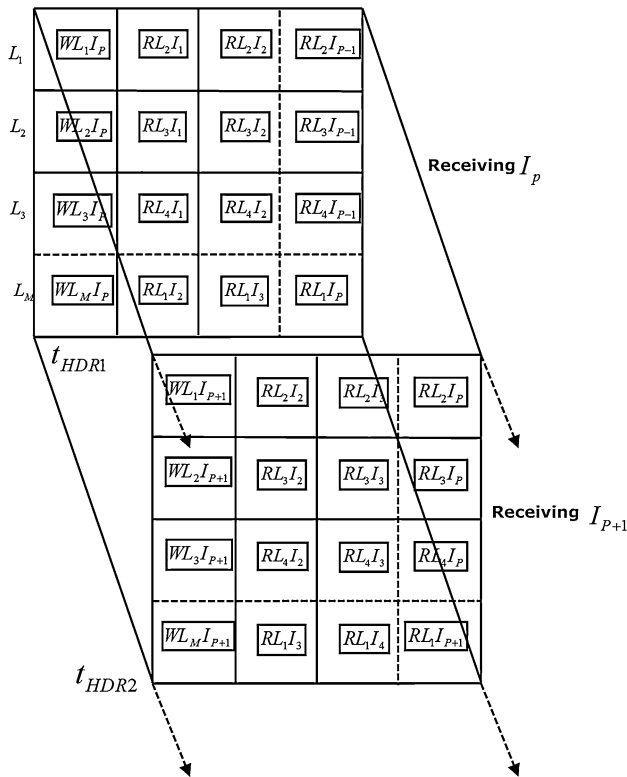


**Fig. 4** Frame rates of the HDR standard technique and our technique



**Fig. 5** Initialization of the Memory Management Unit with storage of the first  $P - 1$ th frames

built from the frames  $I_2, I_3,$  and  $I_4$ ; the HDR frame  $H_3$  from  $I_3, I_4,$  and  $I_5$ , etc. This multi-streaming MMU (called MMU-P, with  $P$  the number of frames) continuously manages the storage of  $P - 1$  frames, the oldest frame being systematically replaced with the new acquired frame. Simultaneously, the MMU-P manages the reading of these



**Fig. 6** Memory Management Unit with parallel acquisition of new lines of pixels and reading of previously stored lines

$P - 1$  frames with the sensor output to feed the HDR creating process. For this purpose, a time-sharing strategy is used to store and read back the different video streams into the DDR2 memory. Small size buffers implemented in FPGA embedded Block RAMs (BRAMs) are required to store temporary data and to handle the sharing of SDRAM data bus. To meet the challenge of real-time constraints and to minimize the buffer sizes, the MMU-P performs row-by-row read/write operations to transfer the different streams.

Before any HDR computation, the memory unit needs to be set up (see Fig. 5). Indeed, when the vision system has just been turned on, an initialization step captures and stores row-by-row  $P - 1$  frames into the SDRAM ( $WL_m I_p$  with  $0 < m \leq M$  and  $0 < p < P$ ). After receiving the last row  $L_M$  of the frame  $I_{p-1}$ , the first rows  $L_1$  of each previously stored frame ( $RL_1 I_p$  with  $0 < p < P$ ) are read between frames and buffered into BRAMs to be quickly available for the HDR creating process.

The second step is managed by the MMU-P core and starts at the beginning of the  $p$ th capture. During each row interval, the current row  $L_m$  is written to the SDRAM memory while rows  $L_{m+1}$  of the  $P - 1$  previous frames are read and buffered into BRAMs as depicted in Fig. 6. With such a technique, when the sensor delivers a new row, the HDR creating process has a simultaneous and easy access to the current row and the corresponding rows of the

previous frames. Then, the first HDR frame is obtained at the end of the capture of the frame  $I_p$  at  $t = t_{HDR1}$  (see Fig. 6). This process is then reiterated with the frame  $I_{p+1}$  and the computation of the second HDR frame from  $I_2, I_3, \dots, I_{p+1}$  at  $t = t_{HDR2}$ , etc.

To summarize, the MMU-P is able to capture and store the current stream of pixels from the sensor, and delivers simultaneous  $P - 1$  pixel streams previously stored to the HDR creating process. With such a memory management, we avoid waiting for the capturing of new  $P$  frames before computing any new HDR data. Once the initialization is done, our system is synchronized with the sensor frame rate (i.e., 60 fps) and is able to produce a new HDR frame for each new capture. Moreover, in terms of memory, the MMU-P requires to store only  $P - 1$  frames, because the oldest captured frame is read and overwritten by the current frame acquired by the sensor. For reasons of efficiency, the MMU-P reads and stores lines of pixels. The computation of an HDR line requires  $P$  memory accesses: one write operation for the current line captured by the sensor and  $P - 1$  read operations for the  $P - 1$  lines previously stored in memory during the row interval. On the opposite, the traditional technique requires to store  $P$  frames in memory. It requires at least  $P + 1$  memory accesses (one write and  $P$  read operations). Finally, to generate an HDR image, the MMU-P saves  $M$  writing memory access operations (with  $M$ , the line number of each frame).

#### 4 Implementation of the HDR-2 video system

The HDR-P has been first prototyped on the HDR-ARtiST platform with the limitation  $P = 2$ , using only two frames to generate each HDR frame.

##### 4.1 Multiple exposure control

The auto exposure bracketing system implemented in digital cameras is a very useful technique to automatically capture multiple exposures at fixed multiples around an optimum exposure. Our multiple exposure control (MEC) for low- and high-exposure images is slightly different because our objective is rather to evaluate each individual exposure to get the maximum of well-exposed pixels. Our work is relatively close to the Kang's exposure control [21] and the Alston's double-exposure system [4]. In our case, the exposure settings alternate between two different values that are continuously adapted to reflect the scene changes. Following Gelfand et al. [16], we require that fewer than 10 % of the pixels are saturated at high-level for the short-exposure frame. If too many pixels are bright, the exposure time is decreased for the subsequent short exposure captures. Similarly, we require that fewer than 10 %

pixels are saturated at low-level for the long exposure. If too many pixels are dark, the exposure time is increased. Gelfand et al. stop this iterative process when the two exposures are stable, and use these values for the capture of the full resolution photography. Such an approach is optimal to capture a single HDR image but cannot be considered for an HDR video.

In our approach, we decide to continuously update the set of exposure times from frame to frame to minimize the number of saturated pixels by instantaneously handling any change of the light conditions. The estimation of the best exposure times is computed from the 64-level histogram ( $q$ ) provided automatically by the sensor in the data-stream header of each frame. For each low-exposure frame ( $I_L$ ) and each high exposure ( $I_H$ ), we evaluate  $Q_L$  and  $Q_H$  that are, respectively, the ratio of pixels in the four lower levels and the four higher levels of the histogram:

$$Q_L = \sum_{h=1}^{h=4} \frac{q(h)}{N} \quad Q_H = \sum_{h=60}^{h=64} \frac{q(h)}{N}, \quad (6)$$

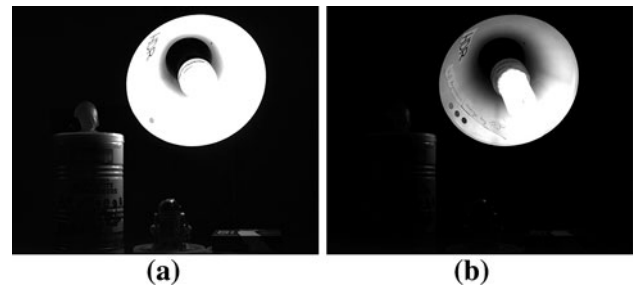
where  $q(h)$  is the amount of pixels in each histogram category  $h$  and  $N$  the total number of pixels. A series of decisions can be performed to evaluate the low-exposure time ( $\Delta t_{L,t+1}$ ) and the high-exposure time ( $\Delta t_{H,t+1}$ ) at the next iteration ( $t + 1$ ) of the acquisition process.

$$\Delta t_{L,t+1} = \begin{cases} \Delta t_{L,t} + 10x & \text{if } Q_L > Q_{L,req} + thr_{Lp} \\ \Delta t_{L,t} - 10x & \text{if } Q_L < Q_{L,req} - thr_{Lp} \\ \Delta t_{L,t} + 1x & \text{if } Q_L > Q_{L,req} + thr_{Lm} \\ \Delta t_{L,t} - 1x & \text{if } Q_L < Q_{L,req} - thr_{Lm} \end{cases} \quad (7)$$

$$\Delta t_{H,t+1} = \begin{cases} \Delta t_{H,t} - 10x & \text{if } Q_H > Q_{H,req} + thr_{Hp} \\ \Delta t_{H,t} + 10x & \text{if } Q_H < Q_{H,req} - thr_{Hp} \\ \Delta t_{H,t} - 1x & \text{if } Q_H > Q_{H,req} + thr_{Hm} \\ \Delta t_{H,t} + 1x & \text{if } Q_H < Q_{H,req} - thr_{Hm} \end{cases}, \quad (8)$$

where  $x$  is the integration time of one sensor row.  $Q_{L,req}$  and  $Q_{H,req}$  are, respectively, the required number of pixels for the low levels and the high levels of the histogram. To converge to the best exposure times as fast as possible, we decide to use two different thresholds for each exposure time:  $thr_{Lm}$  and  $thr_{Lp}$  are thresholds for the low-exposure time whereas  $thr_{Hm}$  and  $thr_{Hp}$  are for the high-exposure time. In our design, values of  $Q_{L,req}$  and  $Q_{H,req}$  are fixed to 10 % pixels of the sensor. Values of  $thr_{Lm}$  and  $thr_{Hm}$  are fixed to 1 % whereas  $thr_{Lp}$  and  $thr_{Hp}$  are fixed to 8 %.

Figure 7 illustrates how the MEC algorithm automatically estimates the new exposure times while rapidly varying illumination in the scene. When switching on the lamp, the number of white saturated pixels significantly increases (15.3 % for Fig. 7a). So, the MEC rapidly decreases the exposure time to get less than 10 % saturated pixels for the high-exposure time  $\Delta t_3$  (5.4 % for Fig. 7a). A similar approach is used to estimate the low-exposure time  $\Delta t_1$ .



**Fig. 7** **a** Has 15.3 % of white saturated pixels when the lamp is switched on. After MEC, **b** has 5.4 % of white pixels. **a** Before MEC, **b** after MEC

#### 4.2 Memory interface implementation

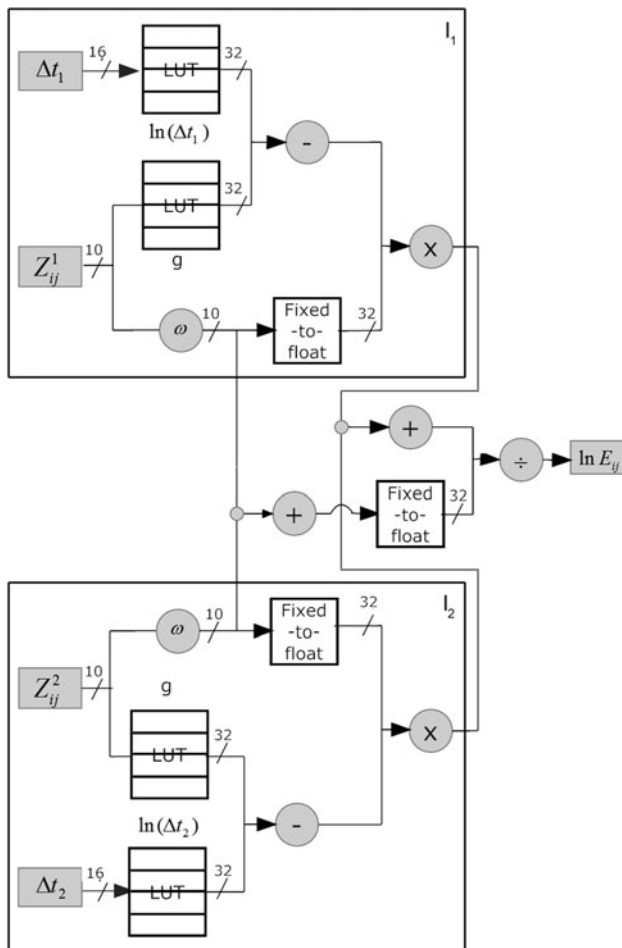
The HDR-ARTiST platform embeds a 256 MB memory with a 128-bit wide interface. Memory access operations are managed by a custom SDRAM controller specifically generated by the Xilinx Memory Interface Generator. The operation frequency of the SDRAM has been fixed to 125 MHz. The e2v sensor captures 60 fps with an inter-frame time of 236  $\mu$ s and a row time of 14.1  $\mu$ s (10  $\mu$ s for the 1,280 pixels and an inter-row time of 4.1  $\mu$ s). BRAMs are used on input and output of the SDRAM, and act as data row buffers to support DDR read and write burst operations. A set of  $P$  different BRAMs is required: one BRAM used to feed the SDRAM with the incoming sensor data and a set of  $P - 1$  BRAMs supplied with the parallel streams of the previously captured frames. These block memories manage independent clocks and support non-symmetric aspect ratios for IO operations. Each BRAM is 2,048 deep and 10-bit wide, to manage a full 1,280-pixel row. A full row of 1,280 pixels captured by the sensor is written into the memory in  $\frac{1,280 \times 10}{128 \times 125 \times 10^6} = 0.8 \mu$ s. Similarly, the time needed for reading a 1,280-pixel row from the SDRAM is identical. These two operations are low time-consuming and take place during the inter-row interval.

#### 4.3 Algorithm simplifications for an efficient hardware implementation

To meet both the temporal constraints and platform requirements, some simplifications of the algorithms described in Sect. 2.2 are proposed.

##### 4.3.1 HDR creating module

The evaluation of the CTF function  $g$  has not been implemented on the platform because it needs to be computed only once. So, the parameters of the CTF are preliminarily evaluated by a dedicated PC software from a sequence of images, and then stored into a Look-Up Table (LUT, 1,024-word memory) on the FPGA. Moreover, to



**Fig. 8** HDR creating pipeline for HDR-2 video, using LUTs tree

reduce the computation complexity and to optimize the hardware resources, some other mathematical operations, such as Napierian logarithms, are also pre-calculated and registered in LUTs.

Finally, the hardware cost of the implementation of the HDR creating described in Eq. 3 only depends on the number of bracketed frames used to create one HDR video frame (see Fig. 8 for HDR-2). Each frame requires two 32-bit arithmetic operators (1 subtractor, 1 multiplier) and one transition from 10-bit to IEEE754 32-bit wide (*Fixed-to-Float*). Other hardware resources (a 10-bit adder, a 32-bit adder, a 32-bit divider and a Fixed-to-Float operator) are also required, independently of the number of frames. A pipeline structure is implemented to speed-up the processing.

Table 3 compares the hardware resources required for the arithmetic operators used in the HDR pipeline. It reveals that a 32-bit floating-point format requires significantly less resource, specifically for complex operations such as division, square root and even multiplication. Only addition and subtraction are better in fixed-point format. Since division and multiplication are common operations in our design, we decided to use floating-point format.

**Table 3** Comparison of resource estimations for arithmetic operators (Xilinx IPs) in fixed-point and floating-point format

Operator	Fixed point		Floating point	
	LUTs	DSPs	LUTs	DSPs
Add/sub	75	0	477	0
Mult.	0	1	287	2
	696	0	659	0
Divide 1 cycle	132	1	107	3
	1,377	0	780	0
Divide 25 cycles	–	–	187	0
Square root 1 cycle	1,550	0	533	1
Square root 25 cycles	–	–	170	1

### 4.3.2 Tone mapping module

The hardware implementation of the Eq. 5 requires the preliminary estimation of the terms  $E_{ij(\min)}$  and  $E_{ij(\max)}$  relative to the HDR frame before any tone mapping operation. However, as our hardware pipeline computes HDR imaging and tone mapping on the incoming pixels stream, such an approach is inconceivable. We make the assumption that the light conditions do not vary significantly between two consecutive frames captured at 60 fps. So, these two terms are computed from the current HDR frame (CMP Min and CMP Max operators), stored in registers (Reg) and then used as the Min/Max values for the tone mapping of the next HDR frame, as seen in Fig. 9.  $\tau$  controls the overall brightness of the mapped frame. According to Duan et al. [11], a larger  $\tau$  makes the mapped frame darker and smaller  $\tau$  makes the mapped frame brighter. Several values have been tested in simulation and we decide to set  $\tau = 0$  because it simplifies the hardware implementation. We also set  $D_{\max} = 255$  and  $D_{\min} = 0$  since standard monitors use a 8-bit format for each channel. The multiplication by 255 is easily done by adding 8 to the exponent in floating point. This may change in the future due to successful efforts in building higher resolution displays. Our design will be able to follow this evolution because tone-mapped frames with higher resolutions could be easily produced by increasing  $D_{\max}$  and changing the final Float-to-Fixed operator.

### 4.4 Results

Simulation and validation phases of the HDR-2 architecture have been made using Modelsim. Then, it has been implemented on the HDR-ARTiST platform. Table 4 is a summary of the whole design and Table 5 describes details of the synthesis report. Usually, FPGA-based image processing requires many specific devices such as SRAM memory, multi-port memory, video direct memory access, dedicated



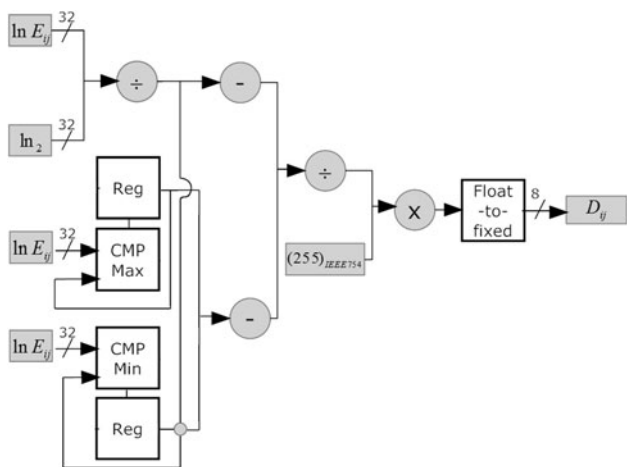


Fig. 9 Tone mapping computation

Table 4 Design summary

Clock domains			
Sensor	SDRAM	HDR processing	DVI
114 MHz	125 MHz	114 MHz	25 MHz
System parameters			
$P$	Resolution	Throughput	Frame rate
2	1,280 × 1,024	78.6 Mpixels/s	60

processors and, consequently, consumes many DSP blocks. There is no DSP blocks in our implementation because some functions are performed with LUTs and others are floating point operators. It can be seen in Table 5 that the implementation results in a relatively low hardware complexity since the number of slice LUTs is 12,859 (about 29 % of the device) and the number of slice flip-flops is 13,612 (i.e., 30 % of the device) for the HDR-2 configuration.

A significant part of the complexity is due to the MMU-P managing the fine synchronization of the pixel streams between external SDRAM memory and internal BRAMs (about 5 % of the LUTs and 6 % of the flip flops). Only 15.5 % of FIFO/BRAMs are used: BRAMs are mainly used to store the  $P$  lines of pixels required by the HDR creating process while FIFO are dedicated to the data synchronization between the different steps of the HDR pipeline. The HDR creating and the tone mapping processes consume, respectively, 9.4 and 7.7 % of the LUTs and 11.4 and 9 % of the flip flops due to the implementation of IEEE754 32-bit floating-point arithmetic operators (specifically dividers and multipliers). The entire HDR pipeline process has a global latency of 136 clock rising-edges (i.e., 1.2  $\mu$ s for a 114 MHz system clock). Such a low latency is not perceived by human eye and then, the pipeline process is able to deliver a full HDR video stream at a frame rate of 60 fps.

Table 5 HDR-2: summary of hardware synthesis report

	Used	Available	Utilization (%)
<i>Logic utilization</i>			
Number of Occupied Slices	5,272	11,200	47.1
<i>Complexity distribution</i>			
Number of Slice LUTs	12,859	44,800	28.7
Memory management	2,149		4.8
HDR creating	4,211		9.4
Tone mapping	3,433		7.7
Post-processing	3,066		6.8
Number of Slice Flip Flops	13,612	44,800	30.4
Memory management	2,682		6
HDR creating	5,107		11.4
Tone mapping	4,045		9
Post-processing	1,778		4
Number of FIFO/BRAM	23	148	15.5

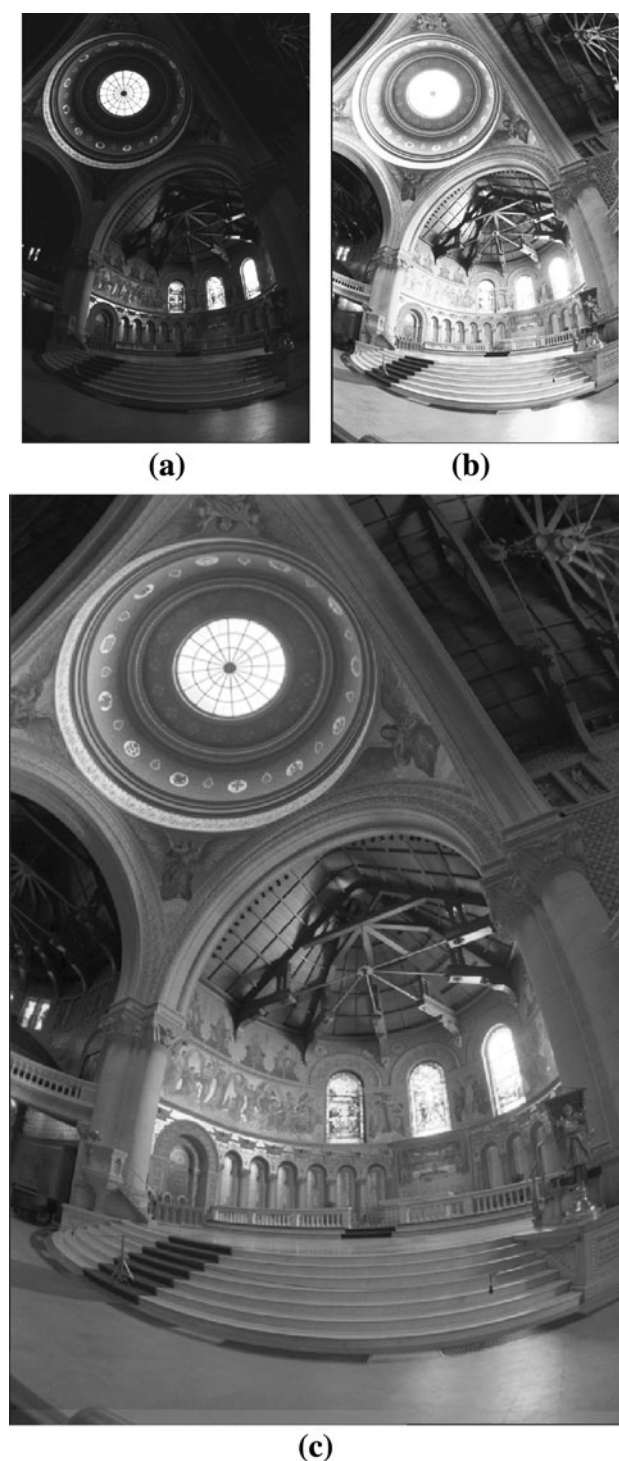
The post-processing step is also a significant consuming task (7 % of the LUTs and 4 % of the flip flops). This task embeds a dedicated DVI controller designed to display camera data on an LCD monitor. Both live video streams, LDR unprocessed pixel streams and computed HDR video can be displayed without latency noticeable for the viewer. For this purpose, the horizontal and vertical synchronization signals ( $H_{sync}$  and  $V_{sync}$  in Fig. 5) are regenerated from the output tone-mapped data and used to synchronize pixels by the DVI encoder. Even if this DVI controller consumes significant resources, it cannot be considered as a real built-in part of the application. It is only used to stream output data into an LCD monitor where the frames from different stages of the logic (typically, LDR frames after pre-processing and HDR fames after post-processing) can be visually inspected. Thus, ignoring the DVI controller resources, our implementation of the HDR application on the HDR-ARTiSt platform results in a relatively low hardware complexity. This opens the interesting possibility to implement other image processing applications on the FPGA. One of the best candidate applications should be motion analysis and artifacts correction to provide a higher quality HDR live video platform.

### 5 The HDR-3 video system

#### 5.1 HDR-2 limitations

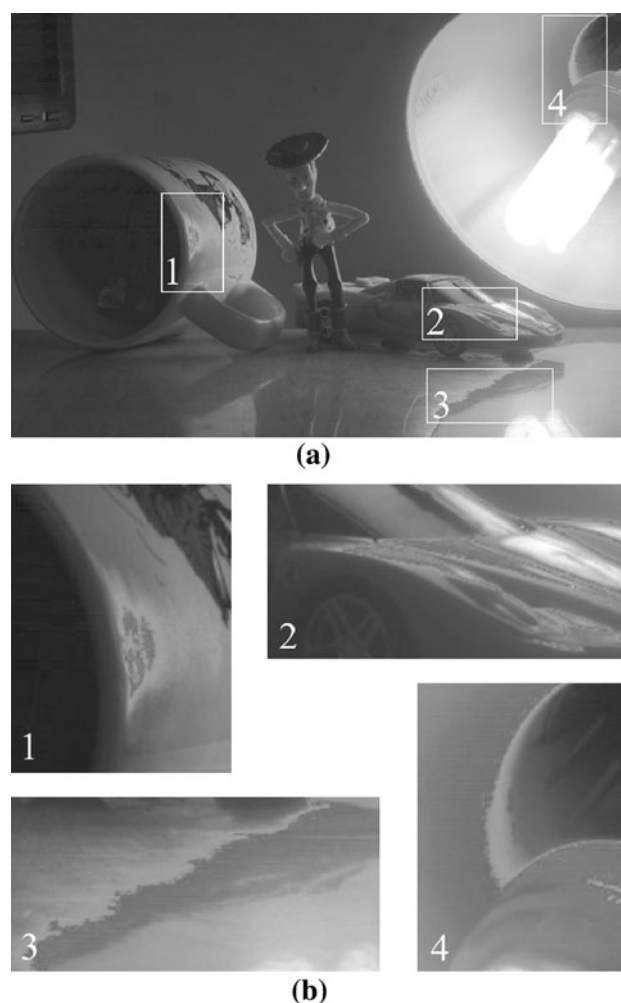
The HDR-2 system has been evaluated both on images from the Debevec’s database (Fig. 10), and on real-scene frames captured by the HDR-ARTiSt platform (Fig. 11).

Firstly, the classical database set to test HDR techniques is the Stanford Memorial Church sequence provided by Debevec. It includes a series of images



**Fig. 10** Visual evaluation of the HDR-2 application. **a** Low exposure, **b** high exposure, **c** HDR image

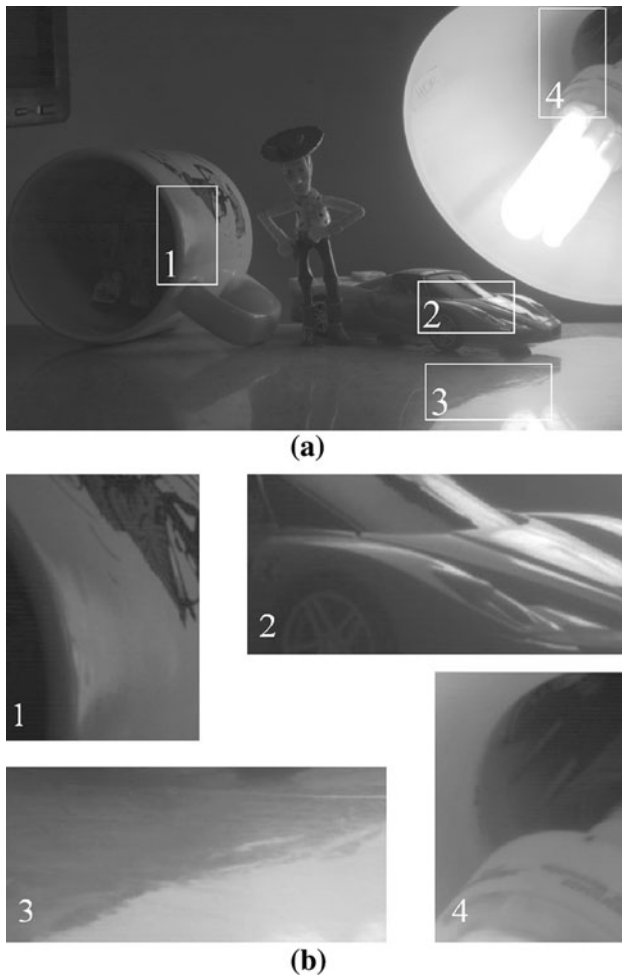
captured with exposure times ranging from 1/1,024 to 32 s. Sets of two images have been selected to evaluate our hardware implementation. Figure 10 depicts an example of an HDR image built by the HDR-2 architecture from a low exposure ( $\Delta t_p = 0.5$  s) and a high exposure ( $\Delta t_p = 16$  s). It should be noted that the



**Fig. 11** HDR-2 system limitations in extreme light conditions. **a** HDR-2 video system limitations in extreme lighting conditions, **b** zoom on different areas

exposures are relatively high, due to low-level light conditions inside the church. The tone-mapped image reproduces both details in dark and bright areas without noticeable visual artifact.

Secondly, the HDR-2 implementation has been also evaluated with real-world scenes captured by the HDR-ARtiSt platform. The experimental scene is a poorly illuminated desk on which we can find a cup of coffee and some toys (a toy car, a Toy Story figurine, a R2-D2 robot inside the cup of coffee). A bright lamp has been placed behind this scene to significantly enhance the dynamic range (see Figs. 11, 12). Obviously, the main difficulty for any HDR technique is to simultaneously render the bright area near the light bulb and the dark area inside the cup of coffee, while preserving accuracy on intermediate levels. Note that the exposures are here very low (respectively, 0.15 and 1.9 m s for the low and high exposures) because of the blinding source of light.



**Fig. 12** HDR-3 video system improvements in extreme lighting conditions. **a** HDR-3 video system improvements in extreme lighting conditions, **b** zoom on different areas

A first visual analysis shows that the results are similar to those obtained previously with the memorial church giving a high-quality reproduction of fine details in dark and bright areas. As examples, the R2D2 figure can be detected even the area inside the cup is particularly dark. On the other side, the word “HDR” written in the lampshade can be read.<sup>1</sup> However, a deeper analysis reveals a lack of details on the medium-lightened areas, highlighting some limitations of the HDR-2 implementation. Four different areas have been selected to highlight possible artifacts. These areas are zoomed in and displayed in the bottom part of Fig. 11, revealing some discontinuities along edges in the cup of coffee (area 1), in the hood of the car (area 2), in the shade of the car (area 3), and in the lampshade (area 4). These observable phenomena are mainly due to the wide dynamic range of

<sup>1</sup> The R2D2 figure and the word “HDR” are only visible in electronic version of the paper and are almost hidden in the print version.

the scene exceeding 100,000:1. In such extreme lighting conditions, it seems almost impossible to capture simultaneously information on the light bulb that illuminates the scene, information on the dark part of the scene, and all the intermediate levels with only two frames. Practically, for capturing details in the upper part of the radiance histogram (lamp), the exposure time of the first frame needs to be decreased severely. On the other side, for very dark areas, the exposure time of the second frame is increased drastically. So, with only two captures, it is not possible to capture the whole scene in detail. Since the two captures focus on the extreme parts of the illuminance, this implies inevitably a lack of information in the middle area of the histogram, leading to the above-mentioned artifacts.

### 5.2 HDR-3 implementation

To overcome this limitation, the most natural solution is to increase the number of frames. With a complementary middle exposure, the HDR-3 is able to capture the missing data in the median area of the illuminance. However, we limit ourselves to a 3-frame acquisition process because too many artifacts appear when four or more exposures are used in the case of real scenes with highly moving objects. The HDR-3 appears as the optimal solution able to satisfy the trade-off between HDR with a capture of about 120 dB dynamic range, low artifacts and high quality.

The middle exposure time  $\Delta t_M$  is computed from the equation of the exposure value  $EV_M$ :

$$EV_M = \log_2 \frac{f\text{-number}^2}{\Delta t_M} \quad \text{with } EV_M = \frac{EV_L + EV_H}{2} \quad (9)$$

where f-number is the aperture. Considering f-number = 1,  $\Delta t_M$  can be computed using the following equation:

$$\Delta t_M = \frac{1}{2^{EV_M}} = \frac{1}{2^{\frac{EV_L + EV_H}{2}}} = \frac{1}{\sqrt{2^{EV_L} \times 2^{EV_H}}} = \sqrt{\Delta t_L \times \Delta t_H}. \quad (10)$$

The tone-mapped frame of the experimental scene using this new implementation is shown in Fig. 12. It is seen that the various artifacts present in Fig. 11 are now widely reduced in this new frame.

In Table 6, a summary of the hardware complexity of the different blocks is given. The HDR-3 implementation obviously results in a slightly higher complexity than the HDR-2 implementation (+11.7 %). Among the submodules listed in Table 6, the amount of the logic resources consumed by the MMU (+31.6 % in terms of LUTs) and the HDR creating (+35.9 % of LUTs and +11.2 % if flip flops) are increased. Moreover, when the number of frames

**Table 6** HDR-3: Summary of hardware synthesis report

	Used	Utilization (%)	Variation with HDR-2 (%)
<i>Logic utilization</i>			
Number of Occupied Slices	5,891	52.6	+11.7
<i>Complexity distribution</i>			
Number of Slice LUTs	15,281	34.11	+18.8
Memory management	2,829	6.3	+31.6
HDR creating	5,722	12.8	+35.9
Tone mapping	3,433	7.7	0
Post-processing	3,297	7.4	+7.5
Number of Slice Flip Flops	15,134	33.8	+11.2
Memory management	2,682	6	+0.1
HDR creating	6,071	13.5	+18.9
Tone mapping	4,045	9	0
Post-processing	2,333	5.2	+31.2
Number of FIFO/BRAMs	30	20.3	+30.4

raises from 2 to 3, there is a significant rise in the usage of block RAMs (+30.4 %), mainly used by the MMU. In contrast to these significant increases, the tone mapping module does not undergo any change and consumes the same resources. The other modules of the system are relatively less affected, except the number of flip flops used by the post-processing submodule. Indeed, this module embeds some additional resources (registers, multiplexers for example) to manage the 3 LDR and the HDR streams and route them to the DVI output.

### 5.3 Extending the architecture to color data

The current B&W HDR-ARTiSt platform could be easily enhanced with a color sensor. In a traditional HDR workflow, HDR creating, tone mapping and other rendering steps are applied after demosaicing. However, as explained in Tamburrino et al. [44], the retinal processing of the human visual system (HVS) performs most of the adaptation operations on the cone mosaic before demosaicing. According to this principle, Tamburrino et al. [44] propose a new HDR workflow in which the tone mapping is applied directly on the color filter array (CFA) image, instead of the already demosaiced image. Demosaicing is the last step of the HDR workflow. Such an approach is then directly compatible with the HDR pipeline implemented on the HDR-ARTiSt platform. Indeed, the HDR creating and the tone mapping operation can be applied on the pixels of the mosaiced image using the three different response curves of the sensor (for the red, green and blue channels), before applying demosaicing. In terms of computation, it does not increase computational complexity because there is only one RAW data stream to process and not three RGB

streams. The demosaicing step, which converts the RAW to RGB data, is handled by a specific hardware block (named “Color Filter Array Interpolation”) provided by Xilinx. This block generates the missing color components associated with the commonly used Bayer pattern in digital camera systems. Additionally, a white balance algorithm IP provided by Xilinx can also be implemented in our platform to render colors as natural as possible. In terms of hardware resources on the HDR-ARTiSt platform, it only requires two more LUTs to store the response curves and the demosaicing/white balance operators provided by Xilinx. According to Debevec and Malik [9], we can also, for simplicity, use an identical single curve for both R and G components because these two response curves are very consistent.

### 5.4 Comparison with state-of-the-art existing architectures

In addition to performance results, a comparison of the HDR-3 architecture with three other systems has been conducted. Two of them are on a FPGA-based architecture [19, 47] and the last one on a GPU platform [43]. However, they implement only the tone mapping, using standard HDR images as inputs. Indeed, hardware vision systems for capturing and rendering HDR live video are at a adolescence stage [31] and we failed to find in the literature a full HDR system implementing all the steps from capturing to rendering. The HDR-3 architecture, on the other hand, is an innovative camera prototype implementing the full HDR pipeline. From this purely algorithmic point of view, our real-time implementation outstrips the computational capabilities of the three other implementations.

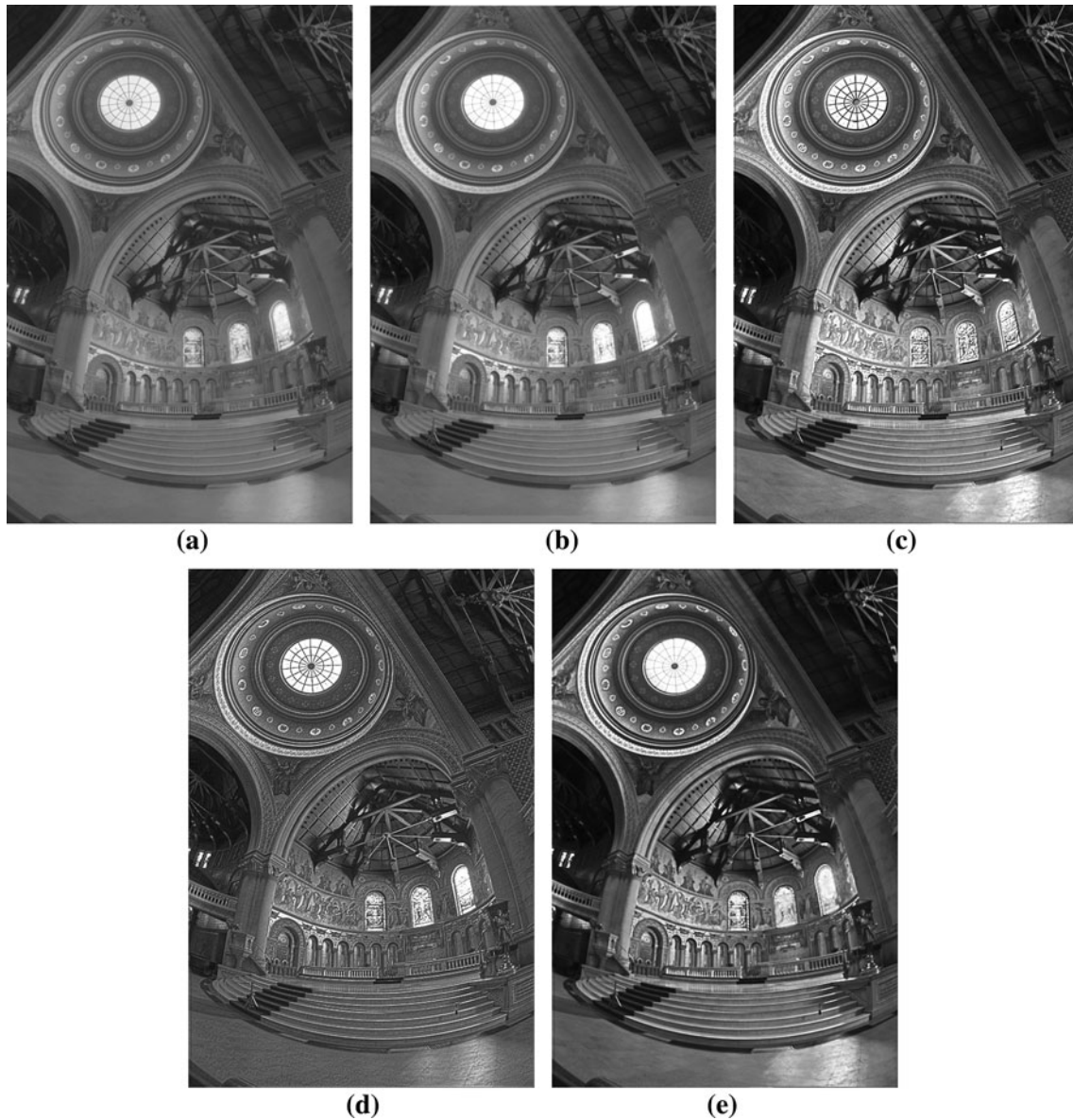
Table 7 summarizes the comparison results in terms of processing performance (fps, resolution). From a raw performance point of view, our architecture runs at 60 fps on a 1, 280 × 1, 024-pixel resolution, giving an overall throughput of 78.6 megapixels per second. This performance is significantly higher than the two other FPGA-based architectures and lower than the GPU implementation. Nevertheless, the Nvidia 8800 GTX processor used in this GPU alternative is not well suited to low-power embedded systems such as smart cameras.

We have also compared the image quality of the tone-mapped images produced by the different architectures using five metrics: mean square error (MSE), normalized root mean square error (NRMSE), peak signal-to-noise ratio (PSNR), Universal Quality Index (UQI [48]), and the Structural SIMilarity (SSIM [49]). UQI roughly corresponds to the human perception of distance among images. The value of the UQI between two images is in the [− 1, 1] range, and is 1 for identical images, 0 for uncorrelated images, and −1 for completely anticorrelated images. The



**Table 7** Performance comparison with existing real-time tone mapping implementations

	This work	FPGA [19]	FPGA [47]	GPU [43]
Device	Virtex 5	Stratix II EP2S130F780	Stratix II EP2S15F484	Nvidia GeForce 8800 GTX
Frame rate	60	60	60	100
Frame size (pixels)	$1,280 \times 1,024$	$1,024 \times 768$	$1,024 \times 1,024$	$1,024 \times 1,024$
MSE	$2.8695 \times 10^2$	$6.6801 \times 10^2$	$3.8646 \times 10^2$	$5.8607 \times 10^2$
NRMSE	$1.7 \times 10^{-1}$	$3.8 \times 10^{-1}$	$1.3 \times 10^{-1}$	$1.9 \times 10^{-1}$
PSNR (dB)	$2.355 \times 10^1$	$1.988 \times 10^1$	$2.226 \times 10^1$	$2.045 \times 10^1$
SSIM	$9.3 \times 10^{-1}$	$6.9 \times 10^{-1}$	$5.5 \times 10^{-1}$	$7.0 \times 10^{-1}$
UQI	0.90	0.71	0.67	0.70

**Fig. 13** Comparison of the proposed system output with different real-time tone mapping methods. **a** Gold standard, **b** our system, **c** Hassan et al. [19], **d** Vytla et al. [47], **e** Slomp et al. [43]

Structural SIMilarity (SSIM) index is a method for measuring the similarity between the two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. It is an improved version of the UQI index.

Our gold standard is the tone-mapped image of the Stanford memorial church with the method by Drago et al. [13]. This technique is described as the most natural method and also the most detailed method in dark region [50]. Considering the gold standard to be the reference signal, and the difference between this reference and the images processed by the other architectures to be the noise, the MSE, the PSNR and the UQI have been calculated for each architecture. The set of images is shown in Fig. 13. The method from Hassan and Carletta [19] seems to provide more contrast and image details than our proposed method. This is mainly due to the fact that they use as inputs HDR images with 28 bits per pixel, obtained off-line from a set of 16 multiple exposures of the same scene. However, the visual evaluation of the different images reveals that the HDR-3 system gives comparable results to the gold standard. This visual feeling is reinforced by different quality metrics. The HDR-3 architecture has the lowest MSE, the lowest NRMSE and the highest PSNR. Moreover, with an UQI value of 0.90 and a SSIM of 0.93, this indicates that the HDR images provided by our architecture are the closest from the gold standard and then, that the visual rendering of these images can be considered as the most natural one.

## 6 Conclusion

In this paper, we present a complete hardware vision system called HDR-ARtiSt based on a standard image sensor associated with a FPGA development board. This smart camera dedicated to real-time HDR live video fulfils drastic real-time constraints while satisfying image quality requirements. It embeds a full HDR pipeline, doing successively multiple captures, HDR creating, tone mapping, and streaming of the HDR live video to an LCD monitor.

The HDR-ARTist system has been built as an adaptive platform. From a purely hardware point of view, our HDR pipeline can be easily adapted to many conventional CMOS sensors. Using a new sensor only requires the design of a specific sensor board to be plugged onto the FPGA motherboard and the offline evaluation of the camera transfer function for the new sensor. From an application point of view, the HDR pipeline can be parametrized in terms of number of frames, depending on

the scene dynamic range. In this paper, two different real-time implementations, using, respectively, 2 and 3 frames have been discussed. For each implementation, we obtain high-performance results due to a finely tuned implementation of the original Debevec's algorithm and a global tone mapping from Duan. The comparison with other state-of-the-art architectures highlights a high-visual quality, close to Drago's algorithm, known as one of the best tone mappers. Moreover, to achieve high temporal performance, the HDR-ARtiSt platform embeds a dedicated memory management unit. This memory unit has been specifically designed for managing multiple parallel video streams to feed the HDR creating process. It significantly contributes to the whole performance of the system. Indeed, the HDR pipeline is synchronized on the image sensor frame rate. It builds a new HDR frame for each new sensor capture and delivers a live stream of displayable content with a bandwidth of 629 Mbits/s. Such a memory unit also limits motion blur and ghosting artifacts in the HDR frames because of the continuous acquisition process.

All these results open interesting avenues for future exploration both on hardware and software issues. New releases of the HDR-ARTist platform embedding FPGA from Xilinx Virtex-6 and Virtex-7 families are currently in development. They will give us opportunity to implement the most recent algorithms for computing radiance maps, and for local/global tone mapping. We also plan to automatically evaluate the real dynamic range of the scene and then to dynamically adapt the number of required captures to build the best HDR frame.

Finally, HDR video is prone to ghosting artifacts, which can appear with motion in the scene during the multiple captures. So, we intend to study and implement onto the FPGA dedicated ghost detection techniques to provide a real-time ghost-free HDR live video. A HW/SW approach based on reprogrammable Zynq architecture is under construction, which will simplify certain operations, and help to add some patches as the correction of local movements.

**Acknowledgments** This work was supported by the DGCIS (French Ministry for Industry) within the framework of the European HiDRaLoN project and by grants from the Conseil Regional de Bourgogne.

## References

1. Acosta-Serafini, P., Ichiro, M., Sodini, C.: A 1/3" VGA linear wide dynamic range CMOS image sensor implementing a predictive multiple sampling algorithm with overlapping integration intervals. *IEEE J. Solid-State Circuits* **39**(9), 1487–1496 (2004)

2. Akil, M., Grandpierre, T., Perroton, L.: Real-time dynamic tone-mapping operator on GPU. *J. Real-Time Image Process.* **7**(3), 165–172 (2012)
3. Akyuz, A.: High dynamic range imaging pipeline on the GPU. *J. Real-Time Image Process.*, pp. 1–15. doi:[10.1007/s11554-012-0270-9](https://doi.org/10.1007/s11554-012-0270-9) (2012)
4. Alston, L., Levinstone, D., Plummer, W.: Exposure control system for an electronic imaging camera having increased dynamic range. US Patent 4647975 A, Cambridge, Mass: U.S. Patent and Trademark Office (1987)
5. Boschetti, A., Adami, N., Leonardi, R., Okuda, M.: An optimal Video-Surveillance approach for HDR videos tone mapping. In: 19th European Signal Processing Conference EUSIPCO 2011, Barcelona (2011)
6. Čadík, M., Wimmer, M., Neumann, L., Artusi, A.: Evaluation of HDR tone mapping methods using essential perceptual attributes. *Comput. Graph.* **32**, 330–349 (2008)
7. Cembrano, G., Rodriguez-Vazquez, A., Galan, R., Jimenez-Garrido, F., Espejo, S., Dominguez-Castro, R.: A 1000 fps at  $128 \times 128$  vision processor with 8-bit digitized I/O. *IEEE J. Solid-State Circuits* **39**(7), 1044–1055 (2004)
8. Chiu, C.T., Wang, T.H., Ke, W.M., Chuang, C.Y., Huang, J.S., Wong, W.S., Tsay, R.S., Wu, C.J.: Real-time tone-mapping processor with integrated photographic and gradient compression using  $0.13 \mu\text{m}$  technology on an ARM SoC platform. *J. Signal Process. Syst.* **64**(1), 93–107 (2010)
9. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 369–378 (1997)
10. Devlin, K., Chalmers, A., Wilkie, A., Purgathofer, W.: Tone reproduction and physically based spectral rendering. In: Eurographics 2002, Eurographics Association, pp. 101–123 (2002)
11. Duan, J., Bressan, M., Dance, C., Qiu, G.: Tone-mapping high dynamic range images by novel histogram adjustment. *Pattern Recognit.* **43**, 1847–1862 (2010)
12. Dubois, J., Ginhac, D., Painsavoine, M., Heyrman, B.: A 10 000 fps CMOS sensor with massively parallel image processing. *IEEE J. Solid-State Circuits* **43**(3), 706–717 (2008)
13. Drago, F., Myszkowski, K., Annen, T., Chiba, N.: Adaptive logarithmic mapping for displaying high contrast scenes. *Comput. Graph. Forum* **22**, 419–426 (2003)
14. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. *ACM Trans. Graph. (TOG)* **21**, 249–256 (2002)
15. Yourganov, G., Stuerzlinger, W.: Acquiring high dynamic range video at video rates. Technical report, Department of Computer Science, York University (2001)
16. Gelfand, N., Adams, A., Park, S.H., Pulli, K.: Multi-exposure imaging on mobile devices. In: Proceedings of the International Conference on Multimedia, New York, USA, pp. 823–826 (2010)
17. Graf, H.G., Harendt, C., Engelhardt, T., Scherjon, C., Warkentin, K., Richter, H., Burghartz, J.: High dynamic range CMOS imager technologies for biomedical applications. *IEEE J. Solid-State Circuits* **44**(1), 281–289 (2009)
18. Granados, M., Ajdin, B., Wand, M., Theobalt, C., Seidel, H., Lensch, H.: Optimal HDR reconstruction with linear digital cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 215–222 (2010)
19. Hassan, F., Carletta, J.: 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, pp. 65–71 (2007)
20. Hrabar, S., Corke, P., Bosse, M.: High dynamic range stereo vision for outdoor mobile robotics. In: IEEE International Conference on Robotics and Automation, 2009. ICRA '09, pp. 430–435 (2009)
21. Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High dynamic range video. Technical report, Interactive Visual Media Group, Microsoft Research, Redmond, WA (2003)
22. Kavusi, S., El Gamal, A.: A quantitative study of high dynamic range image sensor architectures. In: Proceedings of the SPIE Electronic Imaging '04 Conference, vol. 5301, pp. 264–275 (2004)
23. Ke, W.M., Wang, T.H., Chiu, C.T.: Hardware-efficient virtual high dynamic range image reproduction. In: Proceedings of the 16th IEEE International Conference on Image Processing (ICIP'09), Piscataway, NJ, USA, pp. 2665–2668 (2009)
24. Lee, S.H., Woo, H., Kang, M.G.: Global illumination invariant object detection with level set based bimodal segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **20**(4), 616–620 (2010)
25. Leflar, M., Hesham, O., Joslin, C.: Use of high dynamic range images for improved medical simulations. In: Magnenat-Thalmann, N. (ed.) *Modelling the Physiological Human*. Lecture Notes in Computer Science, vol. 5903, pp. 199–208. Springer, Berlin (2009)
26. Lindgren, L., Melander, J., Johansson, R., Moller, B.: A multi-resolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging. *IEEE J. Solid-State Circuits* **40**(6), 1350–1359 (2005)
27. Liu, J., Hassan, F., Carletta, J.: A study of hardware-friendly methods for gradient domain tone mapping of high dynamic range images. *J. Real-Time Image Process.*, pp. 1–17. doi:[10.1007/s11554-013-0365-y](https://doi.org/10.1007/s11554-013-0365-y) (2013)
28. Mangiat, S., Gibson, J.: Inexpensive high dynamic range video for large scale security and surveillance. In: Military Communications Conference, 2011—MILCOM 2011, pp. 1772–1777 (2011)
29. Mitsunaga, T., Nayar, S.: Radiometric self calibration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 374–380 (1999)
30. Morfu, S., Marquié, P., Nofiélé, B., Ginhac, D.: Nonlinear systems for image processing. In: Hawkes, P.W. (ed.) *Advances in Imaging and Electron Physics*, vol. 152, pp. 79–151. Elsevier, Amsterdam (2008)
31. Gallo, O., Gelfand, N., Chen, W., Tico, M., Pulli, K.: Artifact-free high dynamic range imaging. Technical report, University of California (2009)
32. Pattanaik, S.N., Tumblin, J., Yee, H., Greenberg, D.P.: Time-dependent visual adaptation for fast realistic image display. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 47–54 (2000)
33. Reinhard, E., Devlin, K.: Dynamic range reduction inspired by photoreceptor physiology. *IEEE Trans. Vis. Comput. Graph.* **11**, 13–24 (2005)
34. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. *ACM Trans. Graph. (TOG)* **21**(3), 267–276 (2002)
35. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., Myszkowski, K.: *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*, 2nd edn. The Morgan Kaufmann Series in Computer Graphics. Elsevier, Burlington (2010)
36. Robertson, M.A., Borman, S., Stevenson, R.L.: Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *J. Electron. Imaging* **12**(2), 219–228 (2003)
37. Ruedi, P.F., Heim, P., Gyger, S., Kaess, F., Arm, C., Caseiro, R., Nagel, J.L., Todeschini, S.: An SoC combining a 132dB QVGA



- pixel array and a 32b DSP/MCU processor for vision applications. In: *Solid-State Circuits Conference—Digest of Technical Papers*, 2009. ISSCC 2009. IEEE International, pp. 46–47,47a (2009)
38. Sakakibara, M., Kawahito, S., Handoko, D., Nakamura, N., Higashi, M., Mabuchi, K., Sumi, H.: A high-sensitivity CMOS image sensor with gain-adaptive column amplifiers. *IEEE J. Solid-State Circuits* **40**(5), 1147–1156 (2005)
  39. Schanz, M., Nitta, C., Bussmann, A., Hosticka, B., Wertheimer, R.: A high-dynamic-range CMOS image sensor for automotive applications. *IEEE J. Solid-State Circuits* **35**(7), 932–938 (2000)
  40. Schlick, C.: Quantization techniques for visualization of high dynamic range pictures. In: Sakas, G., Maller, S., Shirley, P. (eds.) *Photorealistic Rendering Techniques, Focus on Computer Graphics*. Springer, Berlin, pp. 7–20 (1995)
  41. Schubert, F., Schertler, K., Mikolajczyk, K.: A hands-on approach to high-dynamic-range and superresolution fusion. In: *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 1–8 (2009)
  42. Sérot, J., Ginhac, D., Chapuis, R., Dértin, J.P.: Fast prototyping of parallel-vision applications using functional skeletons. *Mach. Vis. Appl.* **12**, 271–290 (2001)
  43. Slomp, M., Oliveira, M.M.: Real-time photographic local tone reproduction using summed-area tables. In: *Computer Graphics International*, pp. 82–91 (2008)
  44. Tamburrino, D., Alleysson, D., Meylan, L., Suesstrunk, S.: Digital camera workflow for high dynamic range images using a model of retinal processing. In: DiCarlo, J., Rodricks, B. (eds.) *Digital Photography IV, Proceedings of SPIE*, vol 6817, Conference on Digital Photography IV, San Jose, CA, 28–29 Jan 2008 (2008)
  45. e2v Technologies Ev76c560 BW and colour CMOS sensor. <http://www.e2v.com/products-and-services/high-performance-imaging-solutions/imaging-solutions-cmos-ccd-emccd/> (2009)
  46. Tocci, M.D., Kiser, C., Tocci, N., Sen, P.: A versatile HDR video production system. *ACM Trans. Graph.* **30**(4), 41:1–41:10 (2011)
  47. Vytla, L., Hassan, F., Carletta, J.: A real-time implementation of gradient domain high dynamic range compression using a local poisson solver. *J. Real-Time Image Process.* **8**(2), 153–167 (2013)
  48. Wang, Z., Bovik, A.C.: A universal image quality index. *IEEE Signal Process. Lett.* **9**(3), 81–84 (2002)
  49. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
  50. Yoshida, A., Blanz, V., Myszkowski, K., Seidel, H.P.: Perceptual evaluation of tone mapping operators with real-world scenes. In: *Human Vision and Electronic Imaging X, SPIE*, pp. 192–203 (2005)
  51. Youm, S.J., Cho, W.H., Hong, K.S.: High dynamic range video through fusion of exposure-controlled frames. In: *Proceedings of IAPR Conference on Machine Vision Applications*, pp. 546–549 (2005)



**Pierre-Jean Lapray** received his Masters degree in embedded electronics engineering from the Burgundy University. He is currently working towards a Ph.D. in the Electrical Department of LE2I (Laboratory of Electronic, Computing and Imaging Sciences) at the University of Burgundy. His research interests include image enhancement techniques, embedded systems and real-time applications. He is expected to graduate in Fall 2013.



**Barthélémy Heyrman** received his Ph.D. in electronics and image processing from Burgundy University, France, in 2005. He is currently an associate professor at the University of Burgundy, France and member of LE2I UMR CNRS 6306 (Laboratory of Electronic, Computing and Imaging Sciences). His main research topics are system on chip smart camera and embedded image processing chips.



**Dominique Ginhac** received his Masters Degree in Engineering (1995) followed by a PhD in Computer Vision (1999) from the Blaise Pascal University (France). He then joined the University of Burgundy as an assistant professor (2000) and became member of Le2i UMR CNRS 6306 (Laboratory of Electronic, Computing and Imaging Sciences). In 2009, he was promoted professor and became head of the electrical engineering department until 2011. He is currently a deputy director of the Le2i laboratory. His research activities were first in the field of rapid prototyping of real-time image processing on dedicated parallel architectures. More recently, he has developed an expertise in the field of image acquisition, hardware design of smart vision systems and implementation of real-time image processing applications.