



**HAL**  
open science

# HDR-ARtiSt: An Adaptive Real-time Smart camera for High Dynamic Range imaging

Pierre-Jean Lapray, Barthélémy Heyrman, Dominique Ginhac

► **To cite this version:**

Pierre-Jean Lapray, Barthélémy Heyrman, Dominique Ginhac. HDR-ARtiSt: An Adaptive Real-time Smart camera for High Dynamic Range imaging. 2012, pp.1-12. hal-00794471v1

**HAL Id: hal-00794471**

**<https://hal.science/hal-00794471v1>**

Submitted on 26 Feb 2013 (v1), last revised 14 Jan 2014 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# HDR-ARtiSt: An Adaptive Real-time Smart camera for High Dynamic Range imaging

Pierre-Jean Lapray, Barthélemy Heyrman and Dominique Gin hac, *Member, IEEE*

**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) video stream from multiple image acquisitions. 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 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.

**Index Terms**—High dynamic range, smart camera, pipeline processing, real-time systems, specific memory management core, video signal processing.

## I. 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 sunlit and shaded areas. When capturing such a scene, many cameras are unable to store the full Dynamic Range (DR) resulting in low quality images 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 images with higher than standard 24-bit RGB format, and then increasing the range of luminance that can be stored. Reinhard et al. [1] 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 [2]–[4]. Similarly, HDR techniques should facilitate object tracking [5] or automotive applications [6], [7] under uncontrolled illumination. Recent research programs on machine vision have clearly demonstrated the benefits of real-time HDR vision [8], [9]. Finally, medical applications require high precision images and HDR lighting techniques to improve the rendering quality of imaging systems [10], [11]. This is particularly true for biological imaging [12], for fluorescence-based imaging systems [13], [14], or capsule endoscopes [15], [16].

There are two major approaches to create HDR images: to develop new HDR sensors or to combine

Manuscript submitted December, 2012.

This work was supported by the DGCIS (French Ministry for Industry) within the framework of the European HiDRaLoN project.

Pierre-Jean Lapray, Barthélemy Heyrman and Dominique Gin hac are with the Laboratory of Electronic, Computing and Imaging Sciences, Le2i UMR 6306, Univ Burgundy, Dijon, France.

Corresponding author: Dominique Gin hac (dgin hac@u-bourgogne.fr).

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 [17] for a comparative analysis). Most of these sensors are tailor-made and provide dedicated pixel-processing units to extend DR [18]–[23].

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 [24]–[27]. 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 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 [28]–[30] 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 hardware system dedicated to real-time HDR imaging from multiple exposure capturing to display, through radiance maps and tone mapping. The remainder of the paper is as follows: in Section II, we briefly review the literature on existing HDR systems. Section III 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 Section IV. Based on a detailed study of the visual quality of this implementation, an improved three-frame solution is described in Section V. Finally,

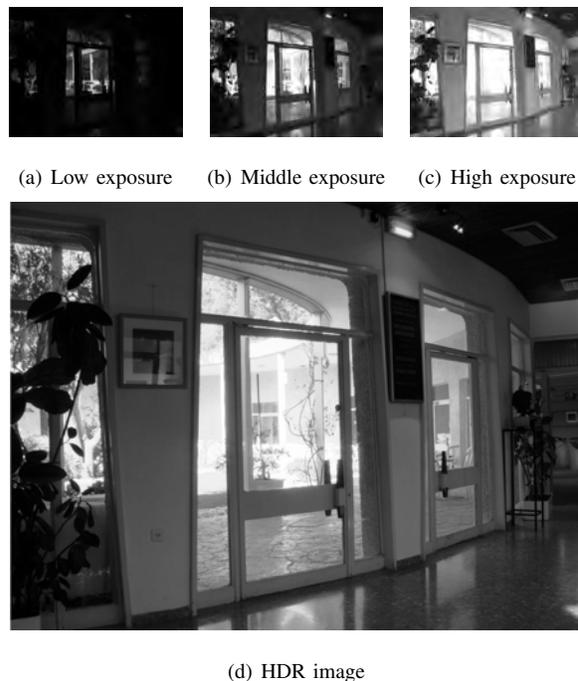


Fig. 1. HDR frame of a real scene captured with 3 different exposure times.

Section VI concludes this paper and outlines directions for future work.

## 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 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 images remains an interesting challenge.

### A. Existing hardware architectures

In 2007, Hassan [31] 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 [32]. 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 high dynamic range still images from the Debevec library [24]. 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. 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 in [33]. They use the Fattal et al. local algorithm [34]. 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.

Ching-Te et al. [35] 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  $1,024 \times 768$ -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. [27] 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 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 seconds (8 seconds for the radiance mapping and 2 seconds for the tone mapping). Based on Kang's algorithms, Youm et al. [36] 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 seconds for each  $640 \times 480$ -pixel frame on a 1.53 GHz AMD Athlon machine.

Finally, Ke et al. [37] 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.

### B. Efficient algorithms for HDR video

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

TABLE I  
LIST OF COMMONLY USED TERMS AND VARIABLES.

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

1) *Step 1: HDR creating:* Digital cameras have limited dynamic range since they can only capture from 8-bit to 14-bit images. This do not cover the full dynamic range of real scenes. The most common method to generate HDR content is to capture multiple images of a same scene with different exposure times. In the following, we will assume that the scene is static and that the captures are fast enough that light changes can be safely ignored. 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 and we have to estimate the nonlinear 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 et al. [24], Mitsunaga et al. [25], and Robertson et al. [38]. According to the detailed description of these methodologies [39] and the comparison of their real-time software implementations [40], we decided to use the Debevec's method, in which the CTF function is evaluated from the film reciprocity equation  $f$ :

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

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

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

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]:

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

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

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

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. Cadik et al. [30] show that the global part of a tone mapping operator is essential to obtain goods results. A psychophysical experiment by Yoshida et al. [42], based on a direct comparison among the appearances of real-world HDR images shows that global methods like Drago et al. [43] or Reinhard et al. [44] 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. According to our temporal and hardware constraints, the best compromise is the global tone mapper from Duan et al. [45]. 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:

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

$$with C = \frac{\log(E_{ij} + \tau) - \log(E_{ij(min)} + \tau)}{\log(E_{ij(max)} + \tau) - \log(E_{ij(min)} + \tau)} \quad (5)$$

where  $E_{ij(min)}$  and  $E_{ij(max)}$  are the minimum and maximum luminance of the scene and  $\tau$  is inversely linked to the brightness of the mapped image. Increasing  $\tau$  makes darker images while lower values give brighter images.

### III. ARCHITECTURE OF THE HDR-ARTiST PLATFORM

To perform 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 images 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 images.

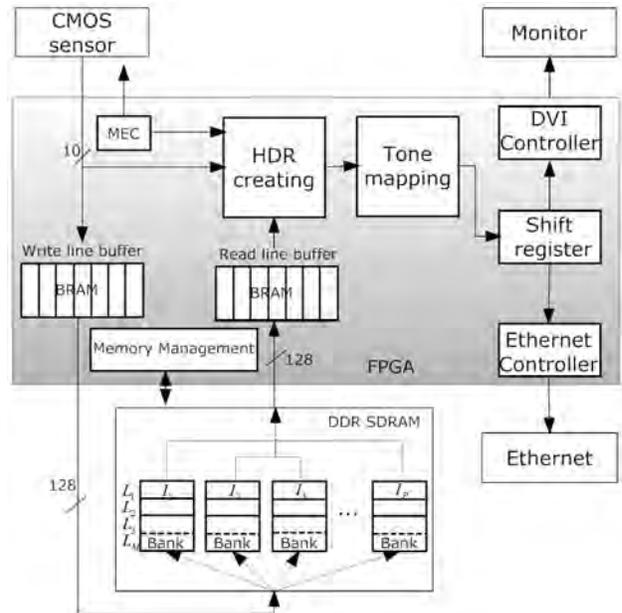


Fig. 2. Overview of the HDR-P video system architecture.

#### A. Global hardware architecture

The HDR-ARTiSt platform is a smart camera built around a Xilinx ML507 board, equipped with a Xilinx Virtex-5 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 a LCD monitor. It also implements an ethernet controller to store frames on a host computer.

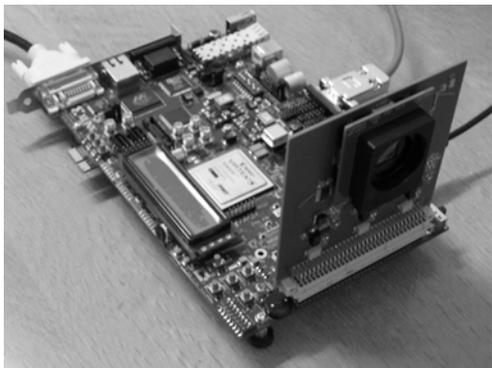


Fig. 3. HDR-P video hardware is prototyped on a Xilinx Virtex-5 ML507 FPGA board and a daughter card with the EV76C560 1.3-Mpixel sensor.

A custom-made PCB extension board has been designed and plugged into the FPGA board to support the Ev76c560 image sensor, a  $1,280 \times 1,024$ -pixel CMOS sensor from e2v [46]. 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 image data stream header.

### B. 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 performed. The main drawback is the limited output framerate. As illustrated on 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 performed from the frames  $I_1$ ,  $I_2$ , and  $I_3$ , the second HDR frame  $H_2$  from  $I_4$ ,  $I_5$ , and  $I_6$ .

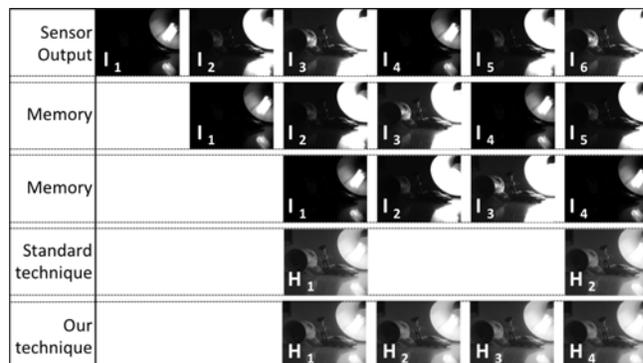


Fig. 4. Framerates of the HDR standard technique and our technique.

To overcome this limitation, we propose a specific Memory Management Unit (MMU) able to continuously perform a new HDR frame at the sensor framerate from the  $P$  previous frames. As seen on Fig. 4, the HDR frame  $H_2$  is performed 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  $P - 1$  frames with the sensor output in order 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 perform temporary data storage and to handle with 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 performing any HDR computation, the mem-

ory 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 during inter-frame and buffered into BRAMs to be quickly available for the HDR creating process.

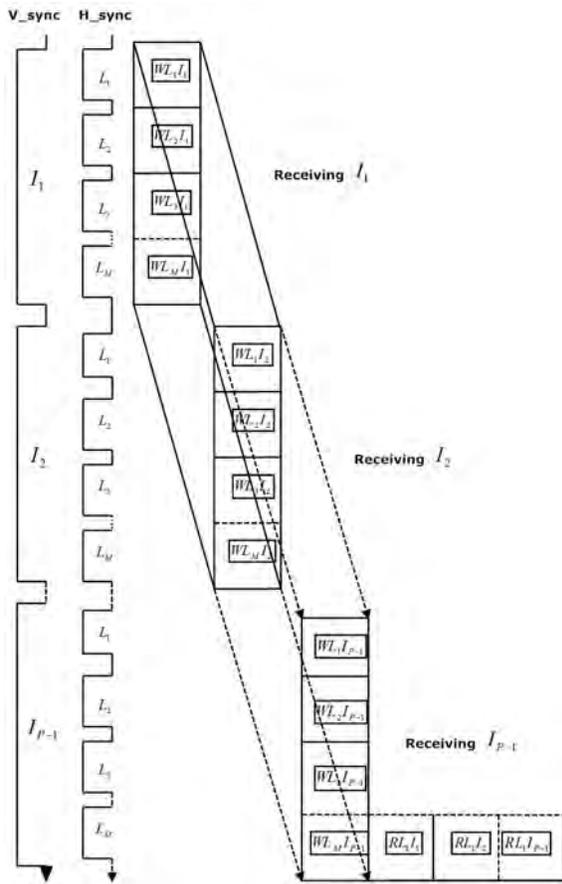


Fig. 5. Initialization of the Memory Management Unit with storage of the first  $P-1$  frames.

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  $t = t_{HDR_1}$  (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  $t = t_{HDR_2}$ , etc.

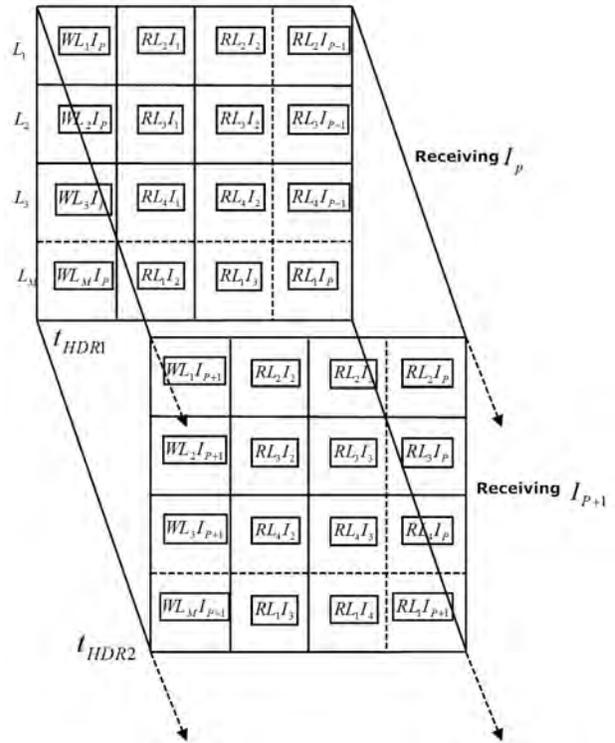


Fig. 6. Memory Management Unit with parallel acquisition of new lines of pixels and reading of previously stored lines.

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 done, our system is synchronized with the

sensor framerate (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. The computation of an HDR row needs  $P$  memory accesses (one write and  $P-1$  read operations during one row interval). The traditional technique requires to store  $P$  frames in memory. Its 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 row number of each frame).

#### IV. 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 2 images to generate each HDR frame.

##### A. 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 [27] and the Alston's double-exposure system [47]. In our case, the exposure settings alternate between two different values that are continuously adapted to reflect the scene changes. The estimation of the best exposure times is performed from scene statistics provided automatically by the sensor in the data-stream header of each frame. These parameters are the 64-level histogram and the numbers of low-level saturated pixels and high-level saturated pixels for each frame. Following

Gelfand et al. [48], 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 can not be considered for an HDR video. So, in our approach, we decide to continuously update the set of exposure times from frame to frame in order to instantaneously handle any change of the light conditions.

##### B. 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 125MHz. 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 2048-depth and 10-bit wide, in order to manage a full 1,280-pixel row. A full row of 1,280 pixels captured by the sen-

sor is written into the memory in  $\frac{1,280 \times 10}{128 \times 125 \cdot 10^6} = 0.8 \mu s$ . Similarly, the time needed for reading a 1,280-pixel row from the SDRAM is identical. These two operations are then low time-consuming and take place during the inter-row interval.

### C. Algorithm simplifications for an efficient hardware implementation

In order to meet both the temporal constraints and platform requirements, some simplifications of the algorithms described in Section II-B are proposed.

*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 preliminary evaluated by a dedicated PC software from a sequence of images, and then stored into a Look-Up Table (LUT) on the FPGA. Moreover, in order to reduce the computation complexity and to optimize the hardware resources, some other mathematical operations, such as neperian 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 of the number of bracketed images used to create one HDR image. (see Fig. 7 for the HDR-2 configuration). 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.

*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,

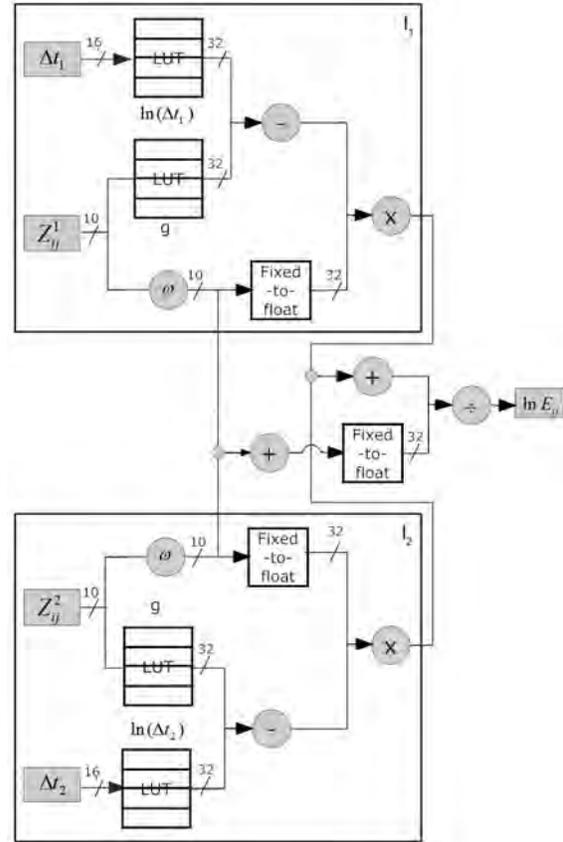


Fig. 7. HDR creating pipeline for HDR-2 video, using LUTs tree.

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. 8. Moreover, from a pragmatic point of view, we decide to set  $\tau = 0$  in order to simplify the hardware implementation. We also set  $D_{max} = 255$  and  $D_{min} = 0$  since standard monitors use a 8-bit format for each channel. 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 images with higher resolutions could be easily produced by increasing  $D_{max}$  and changing the final Float-to-Fixed operator.

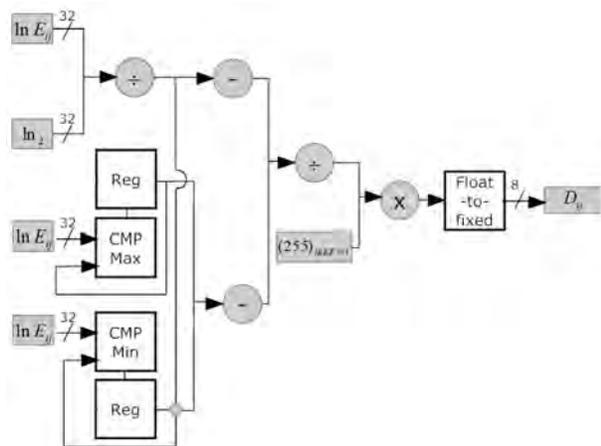


Fig. 8. Tone mapping computation.

#### D. Results

Simulation and validation phases of the HDR-2 architecture have been performed using Modelsim. Then, it has been implemented on the HDR-ARTiST platform. Table II is a summary of the whole design and Table III 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 processors, and consequently, consumes many DSP blocks. This is not the case for our implementation. It can be seen in Table III 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

TABLE II  
DESIGN SUMMARY.

Clock Domains			
Sensor interface	SDRAM controller	HDR processing	DVI controller
114Mhz	125Mhz	114Mhz	25Mhz

System Parameters			
P	Resolution	Throughput	Framerate
2	1,280 × 1,024	78.6 Mpixels/sec	60

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 post processing step is also a significant consuming task (7% of the LUTs and 4% of the flips flops). This task embeds a dedicated DVI controller designed to display camera data on a LCD monitor. Both LDR unprocessed pixel streams and computed HDR video can be displayed. 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 can not be considered as a real built-in part of the application. It is only used to stream output data into a LCD monitor where the frames from different stages

TABLE III  
HDR-2: SUMMARY OF HARDWARE SYNTHESIS REPORT

	Used	Available	Utilization
<b>Logic utilization</b>			
Number of Occupied Slices	5,272	11,200	47.1%
<b>Complexity distribution</b>			
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%

of the logic (typically, LDR frames after pre-processing and HDR frames 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.

## V. THE HDR-3 VIDEO SYSTEM

### A. HDR-2 limitations

The HDR-2 system has been evaluated both on images from the Debevec's database (Fig. 9), and on real-scene images captured by the HDR-ARTiSt platform (Fig. 10 and 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 captured with exposure times ranging from 1/1,024 to 32s.



(a) Low exposure

(b) High exposure



(c) HDR image

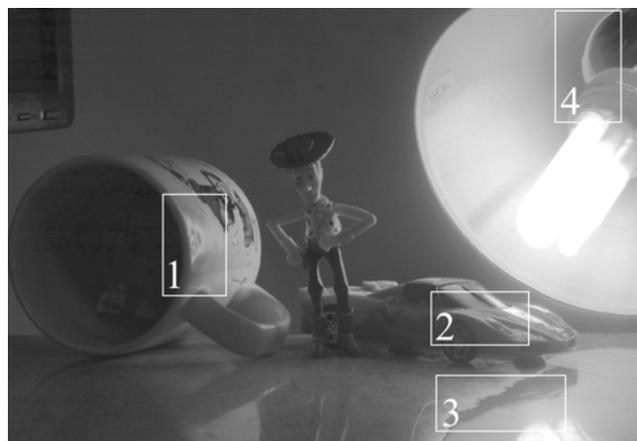
Fig. 9. Visual evaluation of the HDR-2 application.

Sets of two images have been selected to evaluate our hardware implementation. Fig. 9 depicts an example of an HDR image performed by the HDR-2 architecture from a low exposure ( $\Delta t_p = 0.5s$ ) and a high exposure ( $\Delta t_p = 16s$ ). It should be noted that the 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

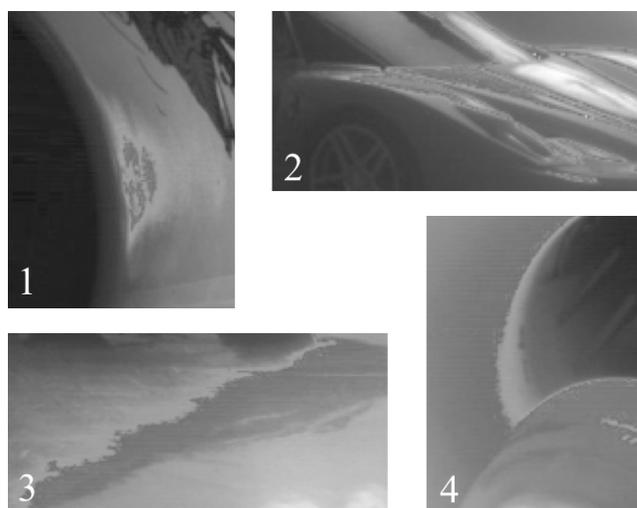
artifact.

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 Fig.10 and Fig. 11). 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  $0.15\text{ ms}$  and  $1.9\text{ ms}$  for the low and high exposures) because of the blinding source of light.

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. 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. 10, 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 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



(a) HDR-2 video system limitations in extreme lighting conditions



(b) Zoom on different areas

Fig. 10. HDR-2 system limitations in extreme light conditions.

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 details. 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.

### B. 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. The tone mapped image of the experimental scene using this new implementation is shown in Fig. 11. It is seen that the various artifacts present in Fig. 10 have disappeared in this new image.



Fig. 11. Visual result of the HDR-3 implementation.

In Table IV, 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 IV, 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 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

TABLE IV  
HDR-3: SUMMARY OF HARDWARE SYNTHESIS REPORT.

	Used	Utilization	Variation with HDR-2
<b>Logic utilization</b>			
Number of Occupied Slices	5,891	52.6%	+11.7%
<b>Complexity distribution</b>			
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%

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.

### C. 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 [31], [33] and the last one on a GPU platform [49]. However, they implement only the tone mapping, using standard HDR images as inputs. Indeed, hardware vision systems for capturing HDR images are



(a) Gold standard.

(b) Our system.

(c) Hassan et al. [31]



(d) Vytla et al. [33]



(e) Slomp et al. [49]

Fig. 12. Comparison of the proposed system output with different real-time tone mapping methods.

at a adolescence stage [50] 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 V 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  $1.280 \times 1.024$ -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.

TABLE V  
PERFORMANCE COMPARISON WITH EXISTING REAL-TIME TONE MAPPING IMPLEMENTATIONS

	This work	FPGA [31]	FPGA [33]	GPU [49]
Device	Virtex 5	Stratix II EP2S130F780	Stratix II EP2S15F484	Nvidia GeForce 8800 GTX
Frame rate (fps)	60	60	60	100
Frame size (pixels)	1,280×1,024	1,024×768	1,024×1,024	1,024×1,024
MSE	286.95	668.01	386.46	586.07
PSNR (dB)	23.55	19.88	22.26	20.45
UQI	0.90	0.71	0.67	0.70

We have also compared the image quality of the tone-mapped images produced by the different architectures using three metrics: mean square error (MSE), peak signal-to-noise ratio (PSNR) and universal quality index (UQI) [51]. 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. Our gold standard is the tone-mapped image of the Stanford memorial church with the Drago's method [43]. This technique is described as the most natural method and also the most detailed method in dark region [42]. 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. 12. Evaluating the output images visually, we see that the HDR-3 system gives comparable results to the gold standard. This visual feeling is reinforced by the three metrics. The HDR-3 architecture has the lowest MSE and the highest PSNR. Moreover, with an UQI value of 0.90, 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.

## VI. 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 video fulfills drastic real-time constraints while satisfying image quality requirements. It embeds a full HDR pipeline, performing successively multiple captures, HDR creating, tone mapping, and streaming of the HDR video to a 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 parameterized 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 framerate. It performs a new HDR image for each new sensor capture and delivers displayable data with a bandwidth of 629 Mb/s. Such a memory unit also limits drastically motion blur and ghosting artifacts in the HDR images 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 perform the best HDR frame.

## REFERENCES

- [1] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski, *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*, 2nd ed., ser. The Morgan Kaufmann series in Computer Graphics. Burlington, MA: Elsevier, 2010.
- [2] M. Sayed and J. Delva, "An efficient intensity correction algorithm for high definition video surveillance applications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 11, pp. 1622–1630, nov. 2011.
- [3] A. Boschetti, N. Adami, R. Leonardi, and M. Okuda, "An optimal Video-Surveillance approach for HDR videos tone mapping," in *19th European Signal Processing Conference EUSIPCO 2011*, Barcelona, Aug. 2011.
- [4] S. Mangiat and J. Gibson, "Inexpensive high dynamic range video for large scale security and surveillance," in *Military Communications Conference, 2011 - MILCOM 2011*, nov. 2011, pp. 1772–1777.
- [5] S.-H. Lee, H. Woo, and M. G. kang, "Global illumination invariant object detection with level set based bimodal segmentation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 4, pp. 616–620, april 2010.
- [6] M. Schanz, C. Nitta, A. Bussmann, B. Hosticka, and R. Wertheimer, "A high-dynamic-range cmos image sensor for automotive applications," *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 7, pp. 932–938, july 2000.
- [7] X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 10, pp. 1522–1533, oct. 2011.
- [8] S. Hrabar, P. Corke, and M. Bosse, "High dynamic range stereo vision for outdoor mobile robotics," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 430–435.
- [9] P.-F. Ruedi, P. Heim, S. Gyger, F. Kaess, C. Arm, R. Caseiro, J.-L. Nagel, and S. Todeschini, "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*, feb. 2009, pp. 46–47,47a.
- [10] M. Leflar, O. Hesham, and C. Joslin, "Use of high dynamic range images for improved medical simulations," in *Modelling the Physiological Human*, ser. Lecture Notes in Computer Science, N. Magnenat-Thalmann, Ed. Springer Berlin Heidelberg, 2009, vol. 5903, pp. 199–208.
- [11] H.-G. Graf, C. Harendt, T. Engelhardt, C. Scherjon, K. Warkentin, H. Richter, and J. Burghartz, "High dynamic range cmos imager technologies for biomedical applications," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 281–289, jan. 2009.
- [12] M. Abolbashari, G. Babaie, F. Magalhes, M. V. Correia, F. M. Arajo, A. S. Gerges, and F. Farahi, "Biological imaging with high dynamic range using compressive imaging technique," in *Proc. SPIE 8225, Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues X*, vol. 8225, 2012, pp. 82251X–82251X–7.
- [13] M. Kfoury, O. Marinov, P. Quevedo, N. Faramarzpour, S. Shirani, L.-C. Liu, Q. Fang, and M. Deen, "Toward a miniaturized wireless fluorescence-based diagnostic imaging system," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 14, no. 1,

- pp. 226–234, jan.-feb. 2008.
- [14] F. Leblond, S. C. Davis, P. A. Valds, and B. W. Pogue, “Pre-clinical whole-body fluorescence imaging: Review of instruments, methods and applications,” *Journal of Photochemistry and Photobiology B: Biology*, vol. 98, no. 1, pp. 77–94, 2010.
- [15] N. B. MacKinnon and U. Stange, “Apparatus and methods relating to expanded dynamic range imaging endoscope systems,” US Patent 8 100 826, Jan, 2012.
- [16] T. Khan and K. Wahid, “Low power and low complexity compressor for video capsule endoscopy,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 10, pp. 1534–1546, oct. 2011.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] L. Lindgren, J. Melander, R. Johansson, and B. Miller, “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.
- [22] S. Morfu, P. Marquié, B. Nofiélé, and D. Ginjac, “Nonlinear systems for image processing,” in *Advances in Imaging and Electron Physics*, P. W. Hawkes, Ed. Elsevier, 2008, vol. 152, pp. 79–151.
- [23] J. Dubois, D. Ginjac, 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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, 2007, pp. 65–71.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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 0.13 μm technology on an ARM SoC platform,” *Journal of Signal Processing Systems*, vol. 64, no. 1, pp. 93–107, 2010.
- [36] S.-J. Youm, W. ho Cho, and K.-S. Hong, “High dynamic range video through fusion of exposed-controlled frames,” in *Proceedings of IAPR Conference on Machine Vision Applications*, 2005, pp. 546–549.
- [37] 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, 2009, pp. 2665–2668.
- [38] 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(2), pp. 219–228, 2003.

- [39] 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.
- [40] W. S. G. Yourganov, “Acquiring high dynamic range video at video rates,” Dept. of Computer Science, York University, Tech. Rep., 2001.
- [41] 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.
- [42] A. Yoshida, V. Blanz, K. Myszkowski, and H. P. Seidel, “Perceptual evaluation of tone mapping operators with real-world scenes,” in *Human Vision & Electronic Imaging X*. SPIE, 2005, pp. 192–203.
- [43] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive logarithmic mapping for displaying high contrast scenes,” *Computer Graphics Forum*, vol. 22, no. 3, pp. 419–426, 2003.
- [44] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic tone reproduction for digital images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 267–276, 2002.
- [45] 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.
- [46] e2v technologies, “Ev76c560 BW and colour CMOS sensor,” 2009. [Online]. Available: <http://www.e2v.com/products-and-services/high-performance-imaging-solutions/imaging-solutions-cmos-ccd-emccd/>
- [47] L. Alston, D. Levinstone, and W. Plummer, “Exposure control system for an electronic imaging camera having increased dynamic range,” US Patent 4 647 975, June, 1987.
- [48] N. Gelfand, A. Adams, S. H. Park, and K. Pulli, “Multi-exposure imaging on mobile devices,” in *Proceedings of the international conference on Multimedia*, New York, USA, 2010, pp. 823–826.
- [49] M. Slomp and M. M. Oliveira, “Real-time photographic local tone reproduction using summed-area tables,” in *Computer Graphics International*, 2008, pp. 82–91.
- [50] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli, “Artifact-free high dynamic range imaging,” in *IEEE International Conference on Computational Photography (ICCP)*, April 2009.
- [51] Z. Wang and A. C. Bovik, “A universal image quality index,” *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.



**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 Electronical 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 PhD in electronics and image processing from Burgundy University, France, in 2005.

He is currently 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** (M'05) received his Masters Degree in Engineering (1995) followed by a Ph.D 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 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.