Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Multi-Camera Platform for Panoramic Real-Time HDR Video Construction and Rendering

Vladan Popovic · Kerem Seyid · Eliéva Pignat · Ömer Çogal · Yusuf Leblebici

Received: date / Revised: date

Abstract High dynamic range (HDR) images are usually obtained by capturing several images of the scene at different exposures. Previous HDR video techniques adopted the same principle by stacking HDR frames in time domain. We designed a new multi-camera platform which is able to construct and render HDR panoramic video in real-time, with 1024×256 resolution and a frame rate of 25 fps. We exploit the overlapping fields-of-view between the cameras with different exposures to create an HDR radiance map. We propose a method for HDR frame reconstruction which merges the previous HDR imaging techniques with the algorithms for panorama reconstruction. The developed FPGA-based processing system is able to reconstruct the HDR frame using the proposed method and tone map the resulting image using a hardware-adapted global operator. The measured throughput of the system is 245 MB/s, which is, up to our knowledge, among the fastest HDR video processing system.

Keywords High dynamic range · Smart cameras · FPGA implementation · Tone mapping · Real-time systems

1 Introduction

Dynamic range in the digitally acquired images is defined as the ratio between the brightest and the darkest pixel in the image. Most modern cameras cannot capture sufficiently wide dynamic range to truthfully represent radiance of the natural scenes, which may contain several orders of magnitude from light to dark regions. This results in underexposed or overexposed regions in the taken image and the lack of local contrast. Fig. 1 shows three shots taken under different exposure settings of a camera. The underexposed and overexposed images show fine details in very bright and very dark areas, respectively. These details cannot be observed in the moderately exposed image.

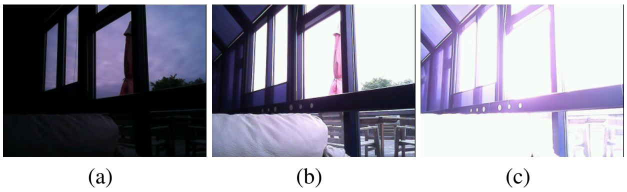

Fig. 1 A subset of images taken for recovering the camera response curve. The images are taken with (a) short, (b) medium and (c) long exposure time.

High dynamic range (HDR) imaging technique was introduced to increase dynamic range of the captured images. HDR imaging is used in many applications, such as remote sensing [1], biomedical imaging [2] and photography [3], thanks to the improved visibility and accurate detail representation in both dark and bright areas.

HDR imaging relies on encoding images with higher precision than standard 24-bit RGB. The most common method of obtaining HDR images is called exposure bracketing and it includes taking several low dynamic range (LDR) images, all under different exposures [4]. Debevec and Malik [5] developed an algorithm for creating wide range radiance maps from multiple LDR images. The algorithm included obtaining camera response curve, creation of HDR radiance map and storage in RGBE format [6].

This work has been partially funded by the Science and Technology Division of the Swiss Federal Competence Center Armasuisse. The authors gratefully acknowledge the support of XILINX, Inc., through the XILINX University Program.

Vladan Popovic · Kerem Seyid · Eliéva Pignat · Ömer Çogal · Yusuf Leblebici
Microelectronic Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 11, 1015 Lausanne, Switzerland

E-mail: vladan.popovic@epfl.ch

Other approaches based on a weighted average of differently exposed images were proposed in [7–9], with differently calculated weights. State-of-the-art algorithms for radiance map construction include camera noise model and optimization of the noise variance as the objective function [10, 11].

An exposure bracketed sequence can also be fused into the HDR image without the radiance map calculation. Exposure fusion method [12] is a pipelined fusion process where LDR images are combined based on saturation and contrast quality metrics. Thanks to direct image fusion, the exposure fusion is not a resource-demanding algorithm as there is no HDR radiance map to be stored, which significantly reduces the memory requirement. A similar principle is used for contrast enhancement using a single LDR image [13]. An alternate option to exposure bracketing is to use an adjustable camera response curve sensor, such as LinLog [14].

Besides capturing the natural scenes, another problem occurs when displaying them. The modern displays are limited to the low dynamic range, which causes inadequate representation of even standard LDR images. In order to avoid such problems, a tone mapping operation is introduced to map the real pixel values to the ones adapted to the display device. The purpose of tone mapping is to compress the full dynamic range in the HDR image, while preserving natural features of the scene.

Tone mapping operators can be divided into two main groups named global and local operators. Global operators are spatially invariant because they apply the same transformation to each pixel in the image. These algorithms usually have low complexity and high computational speed. However, such algorithms have problems preserving the local contrast in the images where the luminance is uniformly occupying the full dynamic range. The first complex global techniques were based on human visual system (HVS) model and subjective experiments [15, 16]. The latest global techniques are based on adaptive mapping. Drago et al. [17] introduced an adaptive logarithmic mapping which applies different mapping curves based on pixel luminosity. The curves vary from $\log_2$ for the darkest pixels, to $\log_{10}$ for the brightest. Similarly, Mantiuk et al. [18] have recently developed a tone mapping algorithm adaptive to the display device.

Opposite to the global operators, local operators are more flexible and adaptable to the image content, which may drastically improve local contrast in regions of interest. Since they differently operate on different regions of the image, they are computationally more expensive and resource-demanding. Reinhard et al. [19] introduced a local adaptation of a global logarithmic mapping. The adaption was inspired by photographic film development in order to avoid halo artifacts. Fattal et al. [20] proposed an operator in gradient domain which was computationally more efficient than other local operators. Nevertheless, both Reinhard and Fattal operators are very resource-demanding for large images, since they require a Gaussian pyramid decomposition and a Poisson equation solver, respectively. Durand and Dorsey [21] presented a fast bilateral filtering where high contrast areas are preserved in the lower spatial frequencies. However, the main disadvantage of this method is the significantly lower overall brightness.

Obtaining and reproducing the HDR video is a difficult challenge due to various issues. Majority of the techniques use exposure bracketed frames from a single camera, which results in high motion blur among frames. Furthermore, using frames from the same camera inherently lowers the effective frame rate of the system, independently of the tone mapping process. The display frame rate is further influenced by both the HDR imaging technique and the processing system. Majority of the systems are based on central processing units (CPU) or graphics processing units (GPU). Even though GPUs are targeted to process large amount of data in parallel, they often fail to meet the tight real-time timing constraints.

In this paper we present a new imaging system for HDR video construction and rendering. The key idea is to use a multi-camera setup to create a composite frame, where cameras with the overlapping field-of-view (FOV) are set to different exposure times. Such system reduces the motion blur, as there is no inter-frame gap time (which can be several hundreds milliseconds in the standard HDR cameras). Additionally, the frames are captured at the same moment by all cameras, which reduces the intra-frame motion of the scene objects to the difference interval of cameras' exposure times. We developed a hardware prototype customized for real-time video processing, utilizing the multi-camera setup. It is a high performance field programmable gate array (FPGA) based system which provides capability for real-time HDR frame construction and tone mapping.

Our contribution in this work is twofold: (1) We propose a single processing pipeline for real-time HDR radiance map construction and simultaneous rendering at 25 frames per second (fps) rate, with reduced intra-frame motion, and (2) we present a hardware prototype on which the pipeline is implemented.

2 Related Work

Exposure bracketing using a single video camera is the most widely used method for HDR video construction. Kang et al. [22] proposed a method of creating a video from an image sequence captured while rapidly alternating the exposure of each frame. Kalantari et al. [23] apply the identical principle and use the patch-based synthesis to deal with the fast movements in the scene. The HDR construction in both cases is realized in post-processing and does not have the real-time processing capability. Gupta et al. [24] recently proposed a way of creating HDR video using Fibonacci exposure bracketing. In this work they adapted a machine vision camera Miro M310 to quickly change exposures and thus reduce the inter-frame delay. However, this system still requires significantly long time to acquire a sequence of frames with desired exposures. Another approach is to use a complex camera with beam splitters [25]. A similar setup is also used in the work of Kronander et al. [26]. This spatially adaptive HDR reconstruction algorithm fits local polynomial approximation to the raw sensor data. However, the algorithm requires intensive processing to recover and display the HDR video.

Exposure bracketing can also be used in multi-camera or multi-view setups. Ramachandra et al. [27] proposed a method for HDR deblurring using already captured multi-view videos with different exposure times. Portz et al. [28] presented a high-speed HDR video using random per-pixel exposure times. This approach is a true on-focal-plane method which still needs to be implemented on a sensor chip.

High frame rate HDR imaging is a challenging problem, even with state-of-the-art processing units. Thus, many attempts have been made to develop a dedicated hardware processing system for this purpose. Hassan and Carletta [29,30] proposed an FPGA architecture for Reinhard [19] and Fattal [20] local operators. Even though the proposed implementations concern only the tone mapping operator, the designs require a lot of resources. This originates from the Gaussian pyramid and look-up table (LUT) implementation of the logarithm function [29] and a local Poisson solver [30]. Another FPGA system was implemented by Lapray et al. [31, 32]. They presented several full imaging systems on Virtex-5 FPGA platform as a processing core. The system uses a special HDR monochrome image sensor providing a 10-bit data output.

Apart from FPGA systems, GPU implementations of full HDR systems are also available. Akyüz [33] presented a comparison of CPU and GPU processing pipelines for already acquired bracketed sequences. Furthermore, real-time GPU implementations of different local tone mapping operators can be found in [34, 35].

3 Camera Prototype

A custom-made FPGA platform is designed for the practice of the real-time omnidirectional video system [36]. The assembled prototype is shown in Fig. 2(a). The designed prototype is an FPGA-based processing platform, which includes eight Xilinx XC5VLX110 Virtex5 FPGAs. One FPGA is targeted for the implementation of the central/master unit and the other seven are slaves used for camera interfacing and local processing on the camera level.

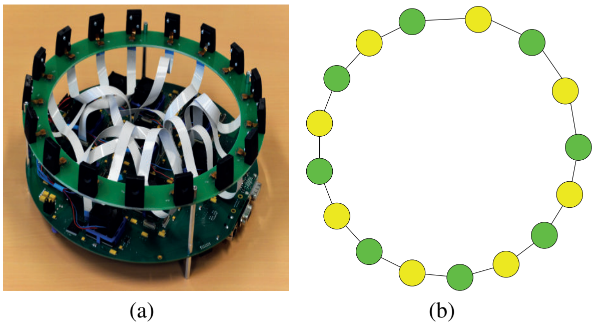

Fig. 2 (a) The built prototype with processing board at the bottom and the installed camera PCB ring. The diameter of the system is $2r = 30$ cm; (b) The graph representation of the camera arrangement. The yellow and green circles represent the cameras with long and short exposure times, respectively. The links between cameras are drawn and each camera can communicate (share pixel values) only with the differently exposed neighbors.

The central FPGA hosts the central processing unit of the system. It is designed to be in charge of system initialization, timing synchronization among the FPGAs, inter-FPGA communication control, video display, and Gigabit Ethernet and USB 2.0 links to a personal computer (PC). Role of the slave FPGA is to create a partial composite frame and send it to the central unit for display. Each slave FPGA is capable of interfacing seven imagers and seven 2 MB SRAM modules, due to limited number of available user I/O pins. Hence, each imager has a dedicated memory storage, which allows the processing unit to simultaneously access pixels from several cameras.

The board is capable of interfacing maximum forty-nine cameras to achieve the full hemispherical view. For the purpose of HDR video application, sixteen cameras are placed on a circular PCB ring. The PCB ring in Fig. 2(a) is $2r =$ 30 cm in diameter. Low-cost cell-phone VGA cameras, with the minimum FOV of 46°, are placed and operated at 25 fps. The graph representation of the camera connections is given in Fig. 2(b). Each camera is able to communicate, *i.e.* share pixel data, with at most two neighboring cameras. Thanks to the inter-FPGA connections, cameras are able to obtain information from a neighboring camera, even if they are not connected to the same FPGA. The communication between cameras is used for workload distribution of HDR frame compositing process, which will be detailed in Section 5.

4 HDR Video

The pixel streams coming from the cameras are processed in real-time; hence, HDR video is created as a stack of HDR frames in time domain. Construction of each frame can be divided into two independent processes: (1) construction of HDR composite frame, and (2) tone mapping the composite frame to achieve realistic rendering.

4.1 HDR Composite Frame

Panorama construction is a well-studied research topic with many proposed algorithms for image stitching and blending. Excellent image quality is especially obtained using spline-based blending techniques [37, 38]. A multi-camera system developed at Stanford University [39] is using such algorithm for panorama construction, and post-processing the differently exposed images to obtain an HDR image. A similar system is developed at EPF Lausanne [40, 41], with the capability of real-time construction and rendering.

Thanks to the circular arrangement of the cameras on this prototype, we adopted the similar approach as in [40], simplified to a two-dimensional case. The installed cameras were calibrated for their intrinsic and extrinsic parameters: focal length, frame center position, lens distortion and angular position in space (yaw, pitch, roll) with geometric center of the prototype as the origin point. The calibration is realized using Kolor Autopano software. To be able to reproduce the HDR image, the cameras are also color calibrated. The camera's response curve is recovered using a set of shots of the same scene with different exposure settings. Three out of twelve taken images are shown in Fig. 1. The response curve is recovered by applying the algorithm proposed by Debevec and Malik [5]. Only one camera is color calibrated, as we assumed that the response curve is identical for all installed cameras. Both calibrations are done only once, as the parameters do not change over time.

FOVs of the cameras overlap such that each point in space is observed by at least two cameras. We exploit this property and set the camera exposures to different values. During the camera initialization phase, all cameras are set to the auto-exposure mode. The camera with the longest exposure time, *i.e.* the one observing a dark region, is taken as a reference. In the following step, half of the cameras are set to the reference exposure t_{ref}, while other half is set to $t_{ref}/4$, such that two cameras with overlapping FOVs have different exposure times. The resulting diagram is shown in Fig. 2(b), where the yellow and green circles represent cameras with long and short exposure times, respectively.

The calibration data provides yaw, pitch and roll data for each camera. We are able to determine tessellation of the hemispherical projection surface according to the influence of the cameras, using these Euler angles and the focal length of the camera [42]. Each 3D region in the obtained tesselation denotes a solid angle in which the observed camera has dominant influence, whereas boundaries of these regions are lines of identical influence of two cameras. This tesselation is called the Voronoi diagram [43]. The most influential camera within a single tile is called the *principal camera*.

As the calibration parameters are known, the composite image is constructed by projection of the camera frames onto the hemispherical surface. In order to obtain the HDR radiance map, the color calibration data should be included as suggested in [5]. The pixel values C_i are expressed as:

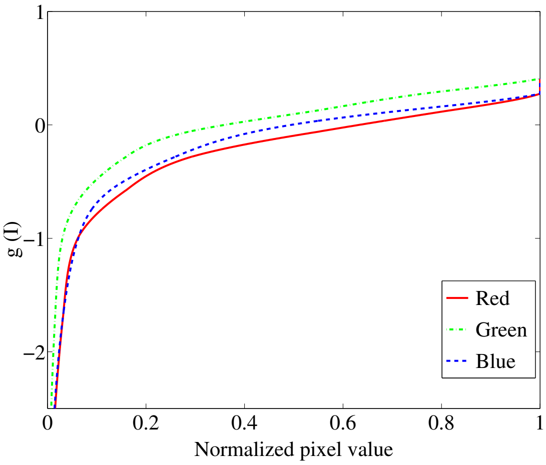

Fig. 3 Recovered response function $g(I)$ of a single camera. Three curves correspond to red, green, and blue pixels, as shown in the legend.

$$\ln C_i = \frac{\sum_j w(I_{j,i}) \cdot \left(g(I_{j,i}) - \ln t_{ref,j}\right)}{\sum_j w(I_{j,i})} \tag{1}$$

$$w(I_{j,i}) = \begin{cases} I_{j,i} - I_{j,min} & \text{, if } I_{j,i} \leq \frac{1}{2}(I_{j,min} + I_{j,max}) \\ I_{j,max} - I_{j,i} & \text{, otherwise} \end{cases} \tag{2}$$

where j is the camera index, i is the pixel position, C is the composite image, $I_{j,i}$ represents a set of pixels from contributing cameras, g is the camera response function, and $I_{j,min}$ and $I_{j,max}$ are minimum and maximum pixel intensities in the observed camera frame. The camera response function is recovered using the approach by Debevec [5], and it is shown in Fig 3.

The nature of the HDR imaging is to recover the irradiance using sensors with different exposures. Hence, we constrain the expression (1) by evaluating it using only two contributing cameras with mutually different exposures. The second camera is referred to as the *secondary camera*.

Even though the calibration by Autopano is precise, registration errors and visible seams are unavoidable. Hence, an additional blending process is required. The Gaussian blending method proposed in [40] is based on a weighted average among the cameras contributing to the observed direction. The weights are samples of the Gaussian function scaled by the distance of camera from the observation point in the projection plane. To include Gaussian blending in the HDR model, the piecewise linear weights $w(I_{j,i})$ from (2) are modified to include the physical position of the camera:

$$w'(I_{j,i}) = w(I_{j,i}) \cdot \frac{1}{r_j} \cdot e^{-\frac{d_i^2}{2\sigma_d^2}} \tag{3}$$

where the notation is kept identical to (1) and (2), with r_j as the distance of camera's projection from the observer, and d_i as the pixel distance from the frame center. High standard deviation σ_d increases region of influence of each camera; hence, relative influence of the *principal camera* is reduced. This results in a smoothly blended background and increased ghosting around edges of the objects in the scene. Thus, the standard deviation σ_d is empirically determined for the given camera setup in order to obtain the best image quality.

The result of applying equations (1) and (3) on the acquired data provides the composite HDR radiance map, which should be tone mapped for realistic display.

4.2 Tone Mapping

Yoshida et al. [44] made an extensive comparison of the tone mapping operators. The comparison was realized by human subjects grading several aspects of the constructed image, such as contrast, brightness, naturalness and detail reproduction. One of the best graded techniques in this review was the local operator by Drago et al. [17]. Therefore, this operator will be taken as a base for the development of an FPGA-suitable operator. Similar to majority of the global operators, this operator uses logarithmic mapping function expressed in (4), where displayed luminance L_d is derived from the ratio of world luminance L_w and its maximum L_{max}. The algorithm adapts the mapping function by changing the logarithm base t as a function of the bias parameter b, as shown in (5).

$$L_d = \frac{\log_t(L_w + 1)}{\log_{10}(L_{max} + 1)} \tag{4}$$

$$t(b) = 2 + 8 \cdot \left(\frac{L_w}{L_{max}}\right)^{\frac{\ln b}{\ln 0.5}} \tag{5}$$

Even though this mapping is created for interactive applications, its speed is very slow for video applications. The reported frame rate is below 10 fps, for 720 × 480 pixels image, without any approximations which decrease the image quality [17]. Calculation of the logarithm values is the most process-intensive part, whether the algorithm is implemented on CPU or GPU. We have derived an operator suitable for direct hardware implementation which shortens the calculation time.

Drago et al. [17] proposed changing logarithm base and calculating only natural and base-10 logarithms. However, fast logarithm calculations are very resource-demanding, because they require large pre-calculated LUTs. Hence, we approximate the logarithm of the form $\log(1+x)$ by the Chebishev polynomials of the first kind $T_i(x)$ [45]. This approximation needs only 6 integer coefficients to achieve 16-bit precision, which is enough for log-luminance representation in our prototype. The Chebyshev approximation can be applied to both natural and base-10 logarithm by only changing the coefficients. The coefficients for the natural logarithm are denoted as $c_e(i)$, while $c_{10}(i)$ are for base-10 in (6).

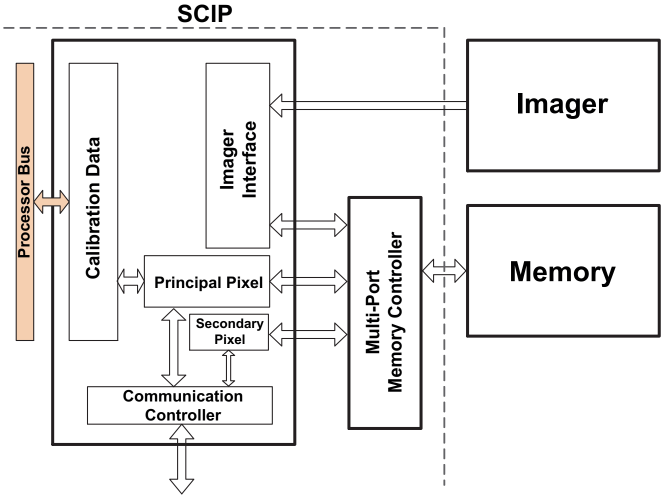

Fig. 4 Internal blocks of the smart camera IP used in the slave FPGAs

According to [17], the best visually perceived results are obtained for the bias parameter $b \approx 0.85$. Fast calculation of generic power functions, *e.g.* the one required in (5), is not possible. Hence, we fixed the parameter to $b = 0.84$, to relax the hardware implementation, without losing any image quality. The exponent is then 0.25, and the result can be evaluated by two consecutive calculations of the square root. The square root is also approximated by the Chebyshev polynomials. The expanded operator is expressed as:

$$L_d = \frac{\sum_{i=0}^{5} c_e(i) T_i(L_w)}{\sum_{i=0}^{5} c_{10}(i) T_i(L_{max}) \cdot \ln\left(2 + 8\left(\frac{L_w}{L_{max}}\right)^{\frac{1}{4}}\right)} \tag{6}$$

The natural logarithm term in the denominator cannot be precisely approximated by Chebyshev polynomials, due the arguments much higher than 1. A suitable approximation of the expression $\ln x$ is a fast convergence form of the Taylor series, which is expressed in (7). This expression needs only 3 non-zero coefficients to achieve a sufficient 16-bit precision, but the argument should be preconditioned as shown:

$$\ln x = 2 \sum_{k=1}^{3} \frac{1}{2k-1} \left(\frac{x-1}{x+1}\right)^{2k-1} \tag{7}$$

Algorithm 1 Smart Camera Processing

```
calculate calibration data
calculate weights
for all principal pixels do
    $p_m := read\_pixel\_from\_memory$
    $p_{s,in} := request\_pixel\_from\_secondary\_camera$
    $C := \frac{w'_m \cdot p_m}{w'_m + w'_s} + p_{s,in}$
    send $C$ to central unit
end for
for all secondary pixels do
    wait for request from principal camera
    $p_s := read\_pixel\_from\_memory$
    $p_{s,out} := \frac{w'_s \cdot p_s}{w'_m + w'_s}$
    send $p_{s,out}$ to principal camera
end for
```

The equations (6)-(7) describe the new tone mapping operator suitable for hardware implementation. The set of required mathematical operations is reduced to only addition, multiplication and division, which are suitable for fast implementation.

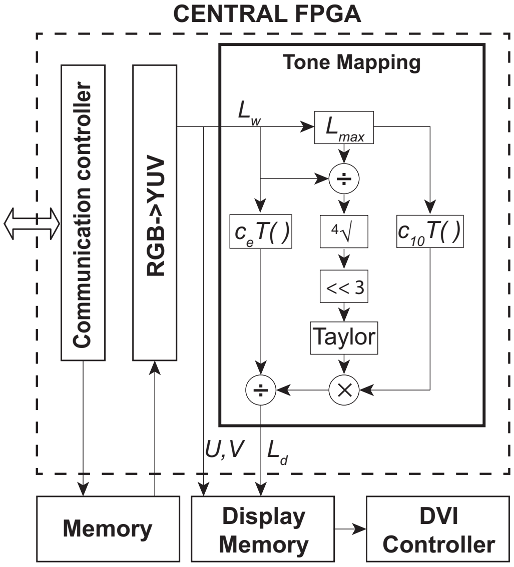

Fig. 5 Internal architecture of the central FPGA. Tone mapping block is emphasized as the core processing unit.

5 FPGA Implementation

5.1 Local Processing

The processing platform consists of seven slave FPGAs used for local image processing, and each slave unit can be connected to seven cameras, due to I/O pin availability. Local processing is realized on the camera level, utilizing the custom-made Smart Camera Intellectual Property (SCIP) shown in Fig. 4. SCIP is instantiated for each camera in the system, and it is in charge of creating a partial HDR composite within camera's FOV. Responsibilities of each SCIP are three-folded: (1) Acquire pixels from the imager and store them in memory, (2) Evaluate the HDR pixel value where the selected camera is the principal camera, and (3) Provide pixel value to the principal camera, when the selected camera is the secondary camera. The Imager Interface in Fig. 4 receives the pixel stream from the camera and stores in the memory. Calibration data block stores information about position of all cameras which are physically close to the observed camera. Thus, SCIP determines the local Voronoi tessellation, and calculates both principal and secondary weights for the camera. The distributed implementation of the algorithm from Section 4.1 is summarized in Algorithm 1.

The Principal pixel block is responsible for calculation of the final HDR pixel value. Using the calibration data, the block reads the appropriate pixel from memory, multiplies it with the weight, and requests the weighted pixel from the secondary camera. The secondary camera is not necessarily connected to the same FPGA. Thanks to the Communication controller, where camera connection graph is stored, the secondary pixel is obtained. The secondary pixel has already been multiplied by the HDR blending weight in the Secondary pixel block, thus only final addition is required. The resulting HDR pixel is further provided to the central unit.

The Secondary pixel block operates in the similar fashion. The block waits for the pixel request from the principal camera, reads the pixel from memory, multiplies by the weight and sends the value back to the principal camera. Both principal and secondary pixel blocks operate concurrently; hence, there is no wait time between principal and secondary pixel processing, which allows very fast calculation time and no loss in the frame rate.

5.2 Central Processing

The central FPGA acts as a global system controller. The received data comprises sixteen parts of the full HDR panorama, *i.e.* one part per SCIP. Besides pixel data, SCIPs send information about the correct position in the full-view panorama. Hence, the central unit decodes the position and places the HDR pixel at the appropriate location in the temporary storage memory. When all the pixels belonging to the same frame are received, tone mapping process starts.

The RGB pixel values are read from memory and transformed into the YUV color system, with 16-bit precision per channel. To be in accordance with the previous notation, the values of the pixel luminance channel Y will be denoted by L_w.

The tone mapping implementation consists of two parts: finding the maximum pixel luminance L_{max} and tone mapping curve implementation. Finding L_{max} consists of finding the maximum value in a sequence of the read luminances. L_{max} value is needed for the core tone mapping operation, as shown in (6). When HDR video stream is processed, L_{max} is taken from the previous frame, under the assumption that the scene illumination does not vary faster than response time of the HVS. The parameter is updated at the end of each frame.

Fig. 5 presents the block diagram of the central unit, with emphasized tone mapping block. Chebyshev and Taylor polynomials are evaluated using pipelined implementation of the Horner scheme. The fast Anderson algorithm [45] is used for division implementation.

Taylor series approximation of the logarithm is fast converging only around the center point of the expansion, *i.e.* $x = 1$ when expansion from (7) is used. Even though the luminance value is in the range [0,1], the logarithm argument in the denominator of the tone mapping function (6) varies in the range [2,10]. Hence, the argument needs to be brought as close as possible to 1. Since the luminance values are logical vectors (vectors of ones and zeros), the identity (8) is used. The number of leading “ones” in the fixed-point representation of the luminance determines the scaling factor y, and the division is implemented as the arithmetical bit-shift-right operation.

$$\begin{aligned} \ln x &= \ln(x/2^y \cdot 2^y) \\ &\approx \ln(x/2^y) + y \cdot 0.6931 \end{aligned} \tag{8}$$

The tone mapped luminance value is combined with the corresponding chrominance components and written into the DVI controller for display.

6 Results and Discussion

6.1 Image Quality

The installed cameras have a vertical FOV = 46° and capture 4.9 Mpixels/frame in total. Even though the ratio of vertical and horizontal FOV of the system is 1:8, we experienced that a panoramic strip of size 256 × 1024 pixels provides enough pixel information, without significant deformation of the objects. This panorama is fitted in the VESA standard XGA frame (768 × 1024 pixels) and displayed directly on screen using DVI connection. The XGA frame is chosen due to 36 Mb capacity of the dedicated display memory in Fig. 5.

In order to quantify the loss in image quality due to applied approximations, the peak signal-to-noise-ratio (PSNR) is calculated for images in the calibration set, whose subset is shown in Fig. 1. The HDR image is created and tone mapped in Matlab using approximated and non-approximated calculations. Non-approximated double-precision tone mapped image is taken as the ground truth. Resulting luminance of the approximated tone mapping from Section 4.2 is quantized as a 16-bit value and its PSNR is measured to be 103.61 dB. Thus, luminance of the resulting image does not lose its original 16-bit precision.

Three video screenshots are shown in Fig. 6. Fig. 6(a) depicts an indoor scene using the automatic exposure mode of the cameras. The measured dynamic range is 1:43. Inside objects are well visible, however, the window region is saturated due to strong light outside of the room. Fig. 6(b) shows the same scene rendered using the proposed HDR module. Even though overlap of FOVs is uneven for each camera pair, difference in color tone is not noticeable. Furthermore, the produced image shows details in previously saturated regions, such as the other buildings, while preserving visibility in the darker inside regions. The dynamic range of the reconstructed scene is increased to approximately 1:160, which results in 3.72 increase in dynamic range.

The indoor reconstructions suffer from ghosting of near objects due to parallax. The ghosting was expected, because the cameras were calibrated in an environment with no close objects. However, the observed ghosting is different from motion blur, which originates from the difference in exposures. Fig. 6(c) shows a rendered HDR outdoor scene, where the closest objects were approximately at 30 m distance. Hence, the edges in this images are significantly sharper than in the indoor environment. The motion blur is not visible around the moving crane or tree branches, thanks to negligible difference in exposure times.

Our HDR construction method does not provide as significant increase in dynamic range as some of the other methods, due to the use of only 2 f-stops. However, up to our knowledge, it is the only system which uses multiple cameras to create and render HDR radiance map simultaneously, and provides real-time HDR video signal at the output. The next step is to additionally improve the dynamic range by increasing the number of cameras, and using more than two different exposures per reconstructed pixel. Furthermore, image quality can be improved by using a more complex blending algorithm, such as [37]. However, real-time implementation of such algorithm requires a more powerful hardware setup.

6.2 System Performance

The chosen figure of merit for performance of real-time systems is the total processing bandwidth, which best describes the system’s capability. The figure of merit is calculated as:

$$BW = N_{pixels} \cdot F \cdot BPP \tag{9}$$

where N_{pixels} is the total number of processed pixels, F is the operational frame rate, and BPP is the number of bytes per

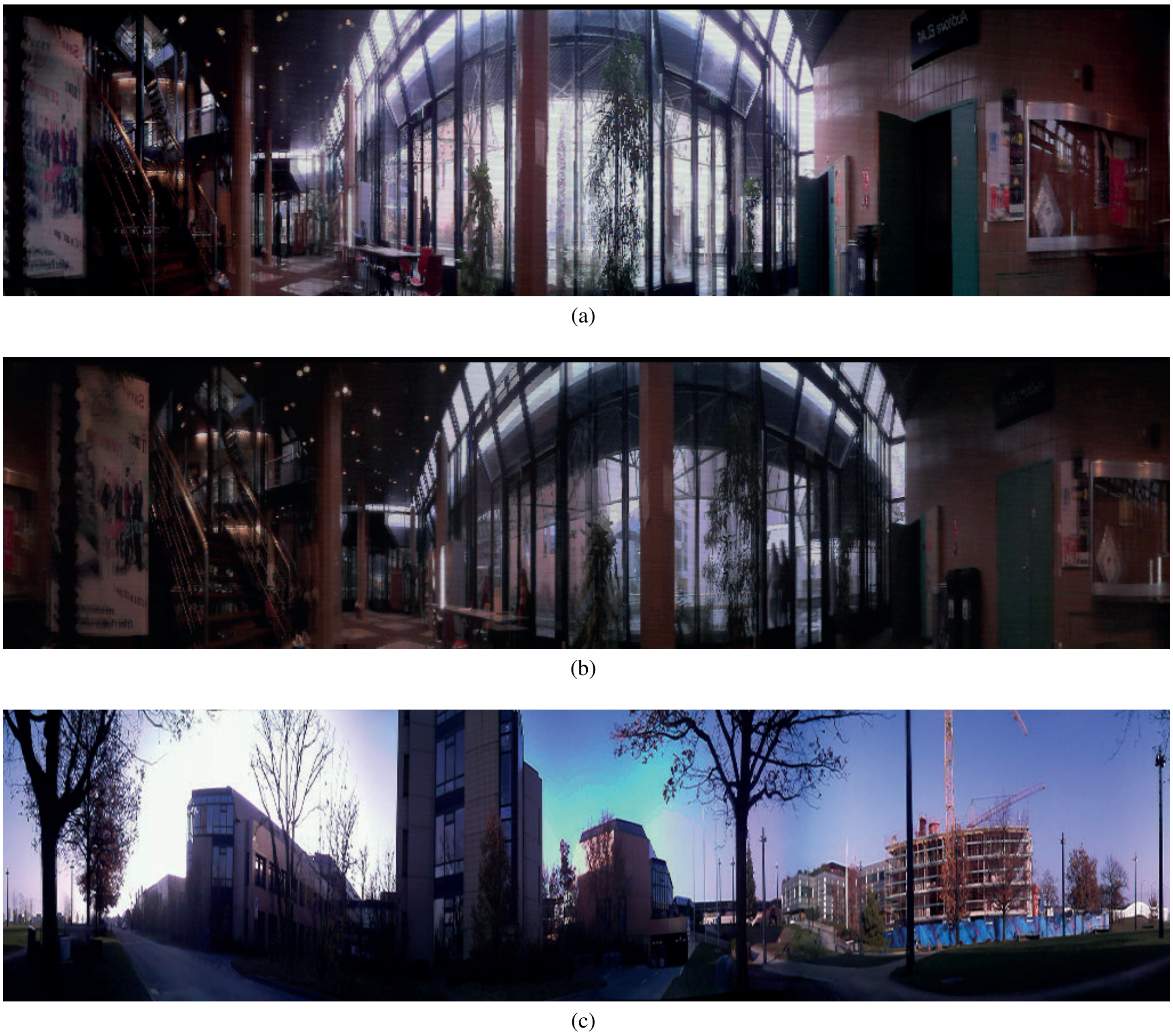

Fig. 6 Panoramic HDR reconstruction with a pixel resolution of 256 × 1024. The cameras were set (a) to automatic exposure mode, (b) such that two neighboring cameras have different exposure times, one four times shorter, and (c) one exposure time eight times shorter, to adapt to bright conditions of outdoor scenery. The blending weights are calculated using $\sigma_d = 300$, to provide sufficient influence of the secondary camera.

Table 1 Performance comparison of the related systems.

Type	Full HDR systems				Only tone mapping			
	This work	[32]	[26]	[24]	[17]	[34]	[30]	[35]
Bandwidth [MB/s]	245.7	196.6	112	45	37.8	74	104.85	214
Processing unit	Virtex-5	Virtex-5	GeForce 680	–	Fire GL X1	GeForce 8800	Stratix II	GeForce 8800 GTX
Real-time video	Yes	Yes	Yes	Yes	No	No	No	Yes

processed pixel. As equations (1)-(3) show that all pixels acquired by the presented system are processed, the number of processed pixels is equal to sixteen VGA (640×480) frames. The operational frame rate is $F = 25\ fps$ as input and output frame rates are equal. Each pixel is represented with $BPP = 2$ bytes in RGB format. The conversion to YUV in the central FPGA transforms each pixel into two bytes for luminance, and one byte per chrominance channel.

Comparison of the designed prototype and algorithm implementation with the related systems is given in Table 1. The numbers in the comparison are taken from the original publications if they are published, or calculated by equation (9) using the available publication data.

Performance comparison shows that the proposed system is superior to the state-of-the-art systems for HDR video construction. The only comparable work is of Slomp and

Table 2 Slave FPGA device utilization.

Module	Memory Controller	Communication Controller	Calibration Accelerator	HDR Composition	MicroBlaze	Total Used	Available
Slices LUTs	3899	18701	1839	32515	1372	63732	69120
Slice Registers	7182	13115	1909	11669	1441	40509	69120
BlockRAM/FIFO	7	48	2	0	32	89	128
DSP48Es	0	0	22	35	4	61	64

Table 3 Central FPGA device utilization.

Module	Memory Controller	Communication Controller	DVI + USB	HDR Tonemapping	MicroBlaze	Total Used	Available
Slices LUTs	1989	4419	828	4164	1234	18376	69120
Slice Registers	3131	4920	1021	2489	1381	17498	69120
BlockRAM/FIFO	4	48	1	3	32	88	128
DSP48Es	0	1	0	54	3	58	64

Oliveira [35] with 214 MB/s. However, this system uses the high-end GPU to implement only the tone mapping function. The main reason for high performance of our system is the fully pipelined operation which processes one pixel per clock cycle. Thus, frame rate is linearly dependent on the clock frequency. The platform allows higher frame rates and bandwidth, which can be achieved by installing faster, or higher resolution CMOS cameras.

Each FPGA uses a MicroBlaze soft processor for system initialization, control and calibration data calculation. The processor local bus frequency and operating frequency of the HDR construction peripherals are 108 MHz and 125 MHz, for the slave and central FPGAs respectively. The inter-FPGA communication controller operates on 216 MHz dual-data rate (DDR), providing 432 Mb/s data rate per LVDS pair. Even though maximum possible frequencies of HDR peripherals are higher, they are lowered to reduce the power consumption, since the 25 fps frame rate is still achieved. The power consumption of eight FPGAs is 31.72 W in total, which is lower than consumption of any commercially available GPU.

Detailed utilization summaries of slave and central FPGAs are given in Table 2 and Table 3. The utilization reports are provided for the complete system capable of supporting all forty-nine cameras. In addition, post-synthesis utilization estimates of the main logical sub-blocks are provided in the tables. The reports show that HDR composition and inter-camera communication controller blocks occupy the major part of the FPGA.

6.3 Real-Time Video Examples

Two video examples of the real-time HDR reconstruction are recorded and provided as the supplementary material. One example shows the improvement of our system compared to the automatic exposure mode of the used cameras. The second example shows the difference compared to manually reduced exposure time, in order to detect objects outside of the room. The blur around the object edges is due to parallax effect, which appears because the camera is calibrated for far objects. This issue can be resolved by having several different calibration parameter sets, which is one of the next steps in the platform development.

7 Conclusion and Future Work

In this paper, we proposed a new HDR video multi-camera system. The system produces a real-time HDR video using multiple low-cost cell-phone cameras, *i.e.* without rather expensive HDR sensors. It is able to simultaneously acquire LDR data, reconstruct an HDR radiance panoramic composite frame, and tone map for realistic display on screen. High system bandwidth and 25 fps frame rate make this prototype an excellent choice for real-time and HDR video applications.

The reconstruction algorithm utilizes the overlap in FOVs of the camera sensors, which are set to different exposure times. We exploit this setup to increase the dynamic range of the captured images and construct an HDR composite image. The HDR image is tone mapped using the fast pipelined global tone mapping algorithm, which was adapted for efficient FPGA implementation.

The next steps in the development of the presented HDR system include enhancement of image quality, further extension of dynamic range, and reduction of the prototype's size and processing power. The image quality will be improved by including the real-time implementation of multiresolution blending [46] into the design. Additionally, other geometrical placements are being considered, such as planar grid of cameras where each pixel is observed by more than two cameras. This arrangement will allow even higher dynamic range within a limited FOV. Apart from quality improvements, the FPGA processing units will be replaced by application-specific integrated circuit (ASIC), which will reduce size of the system and its cost.

Acknowledgements The authors would like to thank H. Afshari, S. Hauser and P. Bruehlmeier for their work on designing the hardware platform.

References

1. Chander, G., Markham, B.L., Helder, D.L.: Summary of current radiometric calibration coefficients for Landsat MSS, TM, ETM+, and EO-1 ALI sensors. In: *Remote Sensing of Environment* **113**(5), 893–903, 2009
2. Jungmann, J.H., MacAleese, L., Visser, J., Vrakking, M.J.J., Heeren, R.M.A.: High Dynamic Range Bio-Molecular Ion Microscopy with the Timepix Detector. In: *Analytical Chemistry* **83**(20), 7888–7894, 2011
3. Bloch, C.: The HDRI Handbook 2.0: High Dynamic Range Imaging for Photographers and CG Artists. Rocky Nook, 2013
4. Mann, S., Picard, R.W.: On Being 'Undigital' with Digital Cameras: Extending Dynamic Range by Combining Differently Exposed Pictures. In: Proceedings of IS&T. 1995, 442–448
5. Debevec, P.E., Malik, J.: Recovering High Dynamic Range Radiance Maps from Photographs. In: ACM SIGGRAPH 97. New York, NY, USA, 1997, 369–378
6. Ward, G.: Graphics Gems II, Academic Press, San Diego, CA, USA, chapter Real Pixels, 80–83. 1991
7. Mitsunaga, T., Nayar, S.: Radiometric Self Calibration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1999, volume 1, 374–380
8. Pattanaik, S.N., Reinhard, E., Ward, G., Debevec, P.E.: High Dynamic Range Imaging - Acquisition, Display, and Image-Based Lighting. Morgan Kaufmann, 2005
9. Robertson, M.A., Borman, S., Stevenson, R.L.: Estimation-theoretic approach to dynamic range enhancement using multiple exposures. In: *Journal of Electronic Imaging* **12**(2), 219–228, 2003
10. Granados, M., Ajdin, B., Wand, M., Theobalt, C., Seidel, H.P., Lensch, H.P.A.: Optimal HDR reconstruction with linear digital cameras. In: Proceedings of 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2010, 215–222
11. Hasinoff, S.W., Durand, F., Freeman, W.T.: Noise-Optimal Capture for High Dynamic Range Photography. In: Proceedings of 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2010, 553–560
12. Mertens, T., Kautz, J., Van Reeth, F.: Exposure Fusion. In: Pacific Conf. on Computer Graphics and Applications. 2007, 382–390. doi:10.1109/PG.2007.17
13. Saleem, A., Beghdadi, A., Boashash, B.: Image fusion-based contrast enhancement. In: *EURASIP Journal on Image and Video Processing* **2012**(10), 2012. doi:10.1186/1687-5281-2012-10
14. Martinez-Sanchez, A., Fernandez, C., Navarro, P.J., Iborra, A.: A Novel Method to Increase LinLog CMOS Sensors' Performance in High Dynamic Range Scenarios. In: *Sensors* **11**(9), 8412–8429, 2011. doi:10.3390/s110908412
15. Ward, G., Rushmeier, H., Piatko, C.: A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. In: *IEEE Trans. Vis. Comput. Graphics* **3**(4), 291–306, 1997. doi:10.1109/2945.646233
16. Pattanaik, S.N., Tumblin, J., Yee, H., Greenberg, D.P.: Time-dependent visual adaptation for fast realistic image display. In: ACM SIGGRAPH 00. New York, NY, USA, 2000, 47–54. doi:10.1145/344779.344810
17. Drago, F., Myszkowski, K., Annen, T., Chiba, N.: Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. In: *Computer Graphics Forum* **22**(3), 419–426, 2003. doi:10.1111/1467-8659.00689
18. Mantiuk, R., Daly, S., Kerofsky, L.: Display Adaptive Tone Mapping. In: *ACM Trans. Graph.* **27**(3), 68:1–68:10, 2008
19. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic Tone Reproduction for Digital Images. In: *ACM Trans. Graph.* **21**(3), 267–276, 2002. doi:10.1145/566654.566575
20. Fattal, R., Lischinski, D., Werman, M.: Gradient Domain High Dynamic Range Compression. In: *ACM Trans. Graph.* **21**(3), 249–256, 2002. doi:10.1145/566654.566573
21. Durand, F., Dorsey, J.: Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. In: *ACM Trans. Graph.* **21**(3), 257–266, 2002. doi:10.1145/566654.566574
22. Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High Dynamic Range Video. In: *ACM Trans. Graph.* **22**(3), 319–325, 2003. doi:10.1145/882262.882270
23. Kalantari, N.K., Shechtman, E., Barnes, C., Darabi, S., Goldman, D.B., Sen, P.: Patch-based High Dynamic Range Video. In: *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH Asia 2013)* **32**(6), 2013
24. Gupta, M., Iso, D., Nayar, S.: Fibonacci Exposure Bracketing for High Dynamic Range Imaging. In: IEEE International Conference on Computer Vision (ICCV). 2013
25. Tocci, M.D., Kiser, C., Tocci, N., Sen, P.: A Versatile HDR Video Production System. In: *ACM Trans. Graph.* **30**(4), 41:1–41:10, 2011. doi:10.1145/2010324.1964936
26. Kronander, J., Gustavson, S., Bonnet, G., Unger, J.: Unified HDR reconstruction from raw CFA data. In: Proceedings of IEEE International Conference on Computational Photography. 2013
27. Ramachandra, V., Zwicker, M., Nguyen, T.: HDR Imaging From Differently Exposed Multiview Videos. In: IEEE 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video. 2008, 85–88
28. Portz, T., Zhang, L., Jiang, H.: Random Coded Sampling for High-Speed HDR Video. In: IEEE International Conference on Computational Photography (ICCP). 2013. doi:10.1109/ICCPhot.2013.6528308
29. Hassan, F., Carletta, J.E.: An FPGA-based architecture for a local tone-mapping operator. In: *Journal of Real-Time Image Processing* **2**(4), 293–308, 2007. doi:10.1007/s11554-007-0056-7
30. Vytla, L., Hassan, F., Carletta, J.: A real-time implementation of gradient domain high dynamic range compression using a local Poisson solver. In: *Journal of Real-Time Image Processing* **8**(2), 153–167, 2013. doi:10.1007/s11554-011-0198-5
31. Lapray, P.J., Heyrman, B., Rosse, M., Ginhac, D.: HDR-ARtiSt: High Dynamic Range Advanced Real-time Imaging System. In: IEEE International Symposium on Circuits and Systems. 2012, 1428–1431. doi:10.1109/ISCAS.2012.6271513
32. Lapray, P.J., Heyrman, B., Ginhac, D.: HDR-ARtiSt: an adaptive real-time smart camera for high dynamic range imaging. In: *Journal of Real-Time Image Processing* 1–16, 2014. doi:10.1007/s11554-013-0393-7

33. Akyuz, A.O.: High dynamic range imaging pipeline on the GPU. In: *Journal of Real-Time Image Processing* 1–15, 2012. doi:10.1007/s11554-012-0270-9
34. Akil, M., Grandpierre, T., Perroton, L.: Real-time dynamic tone-mapping operator on GPU. In: *Journal of Real-Time Image Processing* **7**(3), 165–172, 2012. doi:10.1007/s11554-011-0196-7
35. Slomp, M., Oliveira, M.M.: Real-Time Photographic Local Tone Reproduction Using Summed-Area Tables. In: Computer Graphics International. Istanbul, Turkey, 2008, 82–91
36. Afshari, H.: A real-time multi-aperture omnidirectional visual sensor with interconnected network of smart cameras. Ph.D. thesis, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland, 2013. doi:10.5075/epfl-thesis-5717
37. Brown, M., Lowe, D.: Automatic Panoramic Image Stitching Using Invariant Features. In: *International Journal of Computer Vision* **74**(1), 59–73, 2007
38. Szeliski, R., Uyttendaele, M., Steedly, D.: Fast Poisson Blending using Multi-Splines. In: IEEE International Conference on Computational Photography (ICCP). 2011. doi:10.1109/ICCPHOT.2011.5753119
39. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High Performance Imaging Using Large Camera Arrays. In: *ACM Trans. Graph.* **24**, 765–776, 2005. doi:10.1145/1073204.1073259
40. Popovic, V., Afshari, H., Schmid, A., Leblebici, Y.: Real-time Implementation of Gaussian Image Blending in a Spherical Light Field Camera. In: Proceedings of IEEE International Conference on Industrial Technology. 2013, 1173–1178. doi:10.1109/ICIT.2013.6505839
41. Popovic, V., Seyid, K., Akin, A., Cogal, O., Afshari, H., Schmid, A., Leblebici, Y.: Image Blending in a High Frame Rate FPGA-based Multi-Camera System. In: *Journal of Signal Processing Systems* 1–16, 2013. doi:0.1007/s11265-013-0858-8
42. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2nd edition, 2004
43. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, 2nd edition, 2000
44. Yoshida, A., Blanz, V., Myszkowski, K., Seidel, H.P.: Perceptual Evaluation of Tone Mapping Operators with Real-World Scenes. In: SPIE Human Vision & Electronic Imaging X. 2005, 192–203. doi:10.1117/12.587782
45. Meyer-Baese, U.: Digital Signal Processing with Field Programmable Gate Arrays. Springer-Verlag, Berlin, Germany, 3rd edition, 2007
46. Popovic, V., Seyid, K., Schmid, A., Leblebici, Y.: Real-time Hardware Implementation of Multi-resolution Image Blending. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013, 2741–2745. doi:10.1109/ICASSP.2013.6638155