VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

HDR IMAGE ARTIFACT COMPENSATION

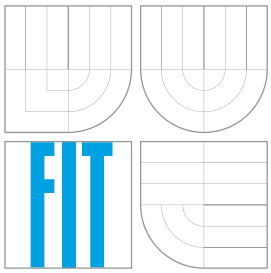
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VĚRA MÜLLEROVÁ

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

KOMPENZACE OBRAZOVÝCH ARTEFAKTŮ V HDR OBRAZE

HDR IMAGE ARTIFACT COMPENSATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VĚRA MÜLLEROVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MARTIN MUSIL

BRNO 2015

Abstrakt

Tato diplomová práce se zabývá syntézou HDR obrazu (High Dynamic Range Imaging). HDRI technologie se stala v posledních letech velice populární. Běžný a nejvíce používaný způsob vytvoření HDR obrazu je spojení více snímků stejné scény pořízených pomocí různých expozičních časů. Tato technika funguje správně pouze v případě, že se jedná o statickou scénu. Pokud je však ve scéně nějaký pohyb ve chvíli, kdy se pořizují snímky dané scény, výsledný HDR obraz obsahuje artefakty zvané jako duchy. V této práci jsou prezentovány základní informace o HDRI se zaměřením na metody odstraňující artefakty z HDR obrazů. Práce shrnuje již existující metody a dvě z nich - tzv. bitmap movement detection a histogram based ghost detection - představuje jako vhodné pro použití v real-time skládání HDR obrazu a pro implementaci na FPGA (Field-Programmable Gate Array) architektuře. Tyto metody jsou v práci implementovány v programovacím jazyce C++ jako prototypy. Navíc je zde navržena modifikace metody založené na výpočtu histogramu pro jednodušší a efektivnější implementaci na FPGA architektuře.

Abstract

This thesis deals with a synthesis of high dynamic range imaging (HDRI). HDRI technology is becoming increasingly popular in recent years. A standard and most common approach to obtain an HDR image is a multiple exposures fusion that consists of combining multiple images of the same scene captured with different exposure times. This technique works perfectly only on static scenes. However, if there is a motion in the scene during a sequence acquisition, a resultant HDR image contains ghosting artefacts due to moving objects in the captured scene. Basic information about HDRI are presented in this thesis. The main focus is given to de-ghosting methods that are reviewed and two of them - a bitmap movement detection based on a median threshold and a histogram based ghost detection - are presented as suitable techniques for a real-time video capturing and implementation on FPGA (Field-Programmable Gate Array) architecture. These two methods are implemented in C++ programming language as prototypes. Moreover, a modification of histogram-based ghost detection is proposed, implemented and discussed to simplify its implementation on FPGA architecture.

Klíčová slova

HDR obraz, artefakty, FPGA, real-time HDR, HDR video

Keywords

HDR image, de-ghosting, ghost artifact, FPGA, real-time HDR, HDR video

Citace

Věra Müllerová: HDR Image Artifact Compensation, diplomová práce, Brno, FIT VUT v Brně, 2015

HDR Image Artifact Compensation

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením pana inženýra Martina Musila. V práci jsem uvedla všechny literární prameny a publikace, ze kterých jsem čerpala.

.....
Věra Müllerová
May 25, 2015

Poděkování

Chtěla bych poděkovat svému vedoucímu práce inženýru Martinu Musilovi z Vysokého učení technického v Brně. On a pan Ing. Petr Musil mi poskytli mnoho zajímavých nápadů a rad ohledně tématu studovaného v této práci a také v oblasti psaní technické zprávy. Také bych chtěla poděkovat svým rodičům, protože bez nich bych na této univerzitě nemohla studovat.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When a photograph of a scene with a big range of brightness is taken, bright areas tend to be over-exposed while dark regions tend to be under-exposed. It is caused by a fact that common digital cameras can only capture a limited luminance dynamic range. The human visual system (HVS) can adapt to a dynamic range of up to 10,000:1 for parts of a scene and over $10^{12} - 10^{14} : 1$ of magnitude in total. In contrast to HVS, digital cameras have a much lower dynamic range of typically less than 1000:1. Therefore, a very interesting and powerful technique has been developed during the last two decades to capture wider dynamic range by conventional cameras called high dynamic range imaging (HDRI). There is a special hardware which allows to take high dynamic range images directly. However, this specialized hardware is very expensive and for commercial use only. This fact encourages a lot of researches in the field of HDRI.

The most common and widely used method to obtain HDR images is a multiple exposures combination. The sequence of single-exposure low dynamic range (LDR) images of the same scene are captured at different exposure times by traditional camera technology and combined into a final HDR image. Each image in the sequence of multiple exposures will have different pixels properly exposed, under-exposed or over-exposed. However, individual parts of images in the sequence must overlap for the successful restoration of the dynamic range. Therefore, it is possible to ignore very dark and very bright pixels from computations of the resulting image.

The biggest limitation of the multiple exposures combination is a requirement of a completely static scene when the sequence of images is being captured, because any object motion in the scene during capturing can cause ghosting artefacts in the final image.

Various methods to detect and remove ghosting artifacts from HDR images have been developed during the last decade. These methods are called de-ghosting algorithms. The algorithms are from extremely elementary, logically understandable and easy to implement but often with poor results through more sophisticated to very complicated, hard to understand, computationally challenging and with time-consuming implementation but with excellent outputs on the other hand.

This thesis discusses de-ghosting methods in HDRI. Moreover, this work is a part of a research project dealing with the real-time HDR FPGA (Field Programmable Gate Array) video camera. The de-ghosting methods described below will be evaluated based on the use in this project and the implementation included in this thesis will follow the FPGA architecture requirements.

The chapter following after this introduction (2) presents a theoretical basis about high dynamic range imaging. There are explanations of technical terms connected with HDRI. A pipeline of an HDR acquisition and its elements are presented in Chapter 2 as well. This chapter also includes a review of existing de-ghosting algorithms. Chapter 3 describes a theoretical basis about FPGA which is necessary for a correct solution design. A reader can find an explanation of FPGA programming principles in this chapter as well. Chapter 4 focuses on a specification of thesis's goals. A realization and implementation of two selected algorithms and proposed algorithm are described in the fifth chapter (5). This chapter also includes results of the implemented methods. Finally, the last chapter 6 summarizes acquired information and shows benefits of this master's thesis.

Chapter 2

Acquisition of High Dynamic Range Images

Camera sensors have a limitation in a dynamic range of luminance which can be captured since the first camera has been invented. In other words, it is impossible to make a photograph of a scene with a big range of brightness where bright as well as dark regions will be well exposed. The extreme areas will appear saturated in the image. There are a few very expensive specialized cameras which allow to take bigger range of brightness than common digital cameras. However, they are for commercial use only and they are still in a development phase more than in practice. These reasons have caused a lot of research that resulted in a creation of a method called high dynamic range imaging (HDRI).

This chapter introduces a necessary background knowledge to understand high dynamic range imaging. It begins with an explanation of basic terms such as high dynamic range imaging, dynamic range, exposure value and radiance. Then the chapter continues with a brief description about light and color spaces. A simple high dynamic range pipeline is presented as well as a deeper explanation of some substantial parts of this pipeline. Capturing, the first component of the pipeline, is a subject of Chapter 2.1. A ghost detection problem is closely connected with this part hence the following Chapter 2.2 focuses on de-ghosting algorithms. Storing, the second element of the pipeline, is presented in Chapter 2.3 where a problem about encodings of high dynamic range images and its formats is discussed. Processing, the next component of the pipeline, includes image processing and much more but this part is not relevant for this thesis as well as HDR monitors. If a reader is interested in HDR monitors there is a whole chapter dealing with this topic in [33]. Finally, tone mapping which is connected with LDR monitors, is introduced in Chapter 2.4.

Interesting books focus on an explanation of issues about HDRI are *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting* [76], *Advanced High Dynamic Range Imaging* [6], *High-Dynamic-Range Vision* [33] and *The HDRI Handbook: High Dynamic Range Imaging for Photographers and CG Artists* [8]. These books were used as basic sources of technical information for this chapter.

High Dynamic Range Imaging

High Dynamic Range Imaging (HDRI) is a set of techniques which reproduces greater dynamic range of luminosity between the lightest and the darkest point of the image than it is possible by using standard digital imaging techniques. HDR images can represent greater range of luminance levels found in real-world scenes from direct sunlight to absolute darkness. This technology is used in photography, computer graphics and image processing. Non-HDR cameras may take photographs only with a limited exposure range which is a complication for a capture of HDR images.

Dynamic Range

Dynamic range (DR) is the biggest overall contrast which is possible to find in an image. The DR is defined as a logarithmic ratio called contrast ratio between the largest and the smallest readable signal. The dynamic range always depends on two factors - complete range of brightness and its smallest noticeable step. For a camera, the DR is a ratio of luminance that just saturates a sensor and a luminance that lifts a camera response to one standard deviation above a noise level. The dynamic range is also a unit of sound because the first use of the DR was in signal processing. The technical dynamic range is defined as logarithmic ratio between the biggest readable signal and noise in background. Its unit is a decibel. The DR is measured as a ratio, as a base-10 or base-2 logarithmic value. But photographers measure dynamic range using difference of exposure values in various parts of the image [8].

Exposure Value

Exposure value (EV) is a photographic measure which defines how much light gets through camera lens and strikes to a photosensitive layer of camera sensor. Amount of this light depends on a speed of camera shutter and a size of camera aperture. The EV is a combination of both these numbers [8].

The exposure value and contrast ratio formulate the same thing. But contrast ratio has linear unit and the EV is in a logarithmic scale. Therefore a transfer equation is $2^{exposure\ values} = constrast\ ratio$.

Radiance

Radiance measures the amount of light passing through, emitted from, leaving, or arriving at a particular point in a given direction. For a digital camera, the radiance values correspond to a physical quantity of light incident on each element of a sensor array. The radiance, a radiometric quantity, is measured in watts per square meter per steradians [76].

HDRI is a new area where a lot of disciplines, such as photography, computer graphics, programming, sciences dealing with specific aspects of light and human vision, meet each other. The essence of all in HDRI is a light. HDRI is a finite digital representation of light. It may contain more light than any other earlier technique [8].

The biggest challenge for HDRI is to display more realistic images on a computer. This is connected with human vision because people are main consumers of resultant outputs. Therefore it is necessary to know about light perception to understand HDRI concept.

Light

Light is very important in HDRI because a perception of a scene by human vision system depends on lighting conditions. Visible light is radiant energy measured in joules. Light travels in space, air and water, interacts with materials where can be refracted, reflected, transmitted or absorbed. Light can reach human eyes and stimulate them to produce visual sensations. It depends on a light wavelength [6, 76]. An electromagnetic spectrum is in Figure 2.1.

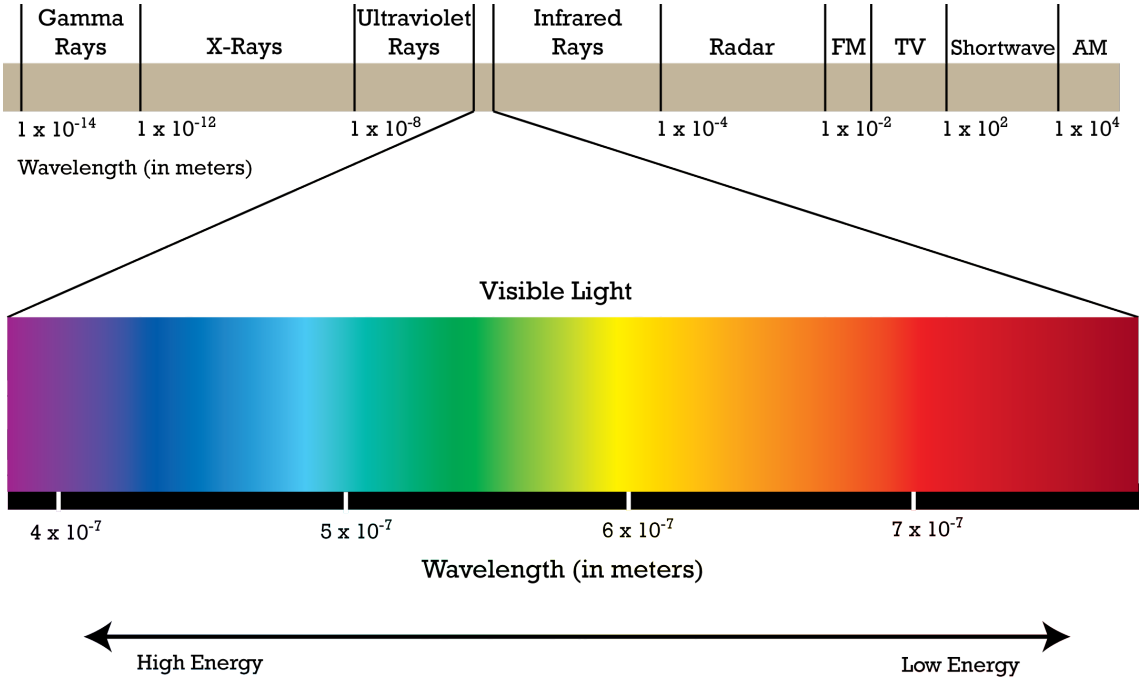


Figure 2.1: Electromagnetic spectrum[1]. A visible light is approximately between 390 and 790nm and a human eye sensitivity is between approximately 400 to 760nm.

There are two possible expressions of light measures. Radiometric quantities are defined in Table 2.1 and main photometric units are presented in Table 2.2.

Color Spaces

Color spaces are mathematical descriptions for representing colors. Color spaces are represented by a set of formulas, that define a relationship between color vector, or they are standard CIE (Commission Internationale de l'Éclairage) XYZ color spaces. A parameter Y in CIE XYZ color space defines brightness. The brightness (illuminance) may be computed from RGB as $Y = 0.2126R + 0.7152G + 0.0722B$ [76]. The chromaticity is specified by two derived parameters x and y. The color space defined by x, y and Y is called CIE xyY and it is represented by chromatic diagram in Figure 2.2. The parameters x and y may be computed from all trichromatic parts X, Y and Z as it is shown in Equation 2.1.

$$x = \frac{X}{X+Y+Z}, y = \frac{Y}{X+Y+Z} \tag{2.1}$$

[1]Source: http://www.pion.cz/en/article/electromagnetic-spectrum

Quantity	Unit	Definition	Description
Radiant energy	J (joule)	Q_e	The basic unit for light
Radiant power	W (watt) $= J \cdot s^{-1}$	$P_e = \frac{dQ_e}{dt}$	The amount of energy that flows per unit of time
Radiant intensity	$W \cdot sr^{-1}$	$I_e = \frac{dP_e}{d\omega}$	The amount of Radiant Power per unit of direction
Radiant exitance	$W \cdot m^{-2}$	$M_e = \frac{dP_e}{dA_e}$	The amount of Radiant Power per unit of area to all directions of the hemisphere from a point
Irradiance	$W \cdot m^{-2}$	$E_e = \frac{dP_e}{dA_e}$	The amount of Radiant Power per unit of area from all directions of the hemisphere at a point
Radiance	$W \cdot sr^{-1} \cdot m^{-2}$	$L_e = \frac{d^2 P_e}{dA_e \cos\theta d\omega}$	The amount of Radiant Power arriving/leaving at a point in a particular direction

Table 2.1: Main radiometric units [76, 6].

Quantity	Unit	Description
Luminous power (P_v)	lm (lumens) $= cd \cdot sr$	The weighted Radiant Power
Luminous energy (Q_v)	$lm \cdot s^{-1}$	Analogous to the Radiant Energy
Luminous intensity (I_v)	cd (candela) $= lm \cdot sr^{-1}$	The Luminous Power per direction
Luminous exitance (M_v)	$lux = lm \cdot m^{-2}$	Analogous to Radiant exitance
Illuminance (E_v)	$lux = lm \cdot m^{-2}$	Analogous to Irradiance
Luminance (L_v)	$cd \cdot m^{-2} = lm \cdot m^{-2} \cdot sr^{-1}$	The weighted Radiance

Table 2.2: Main photometric units [76, 6].

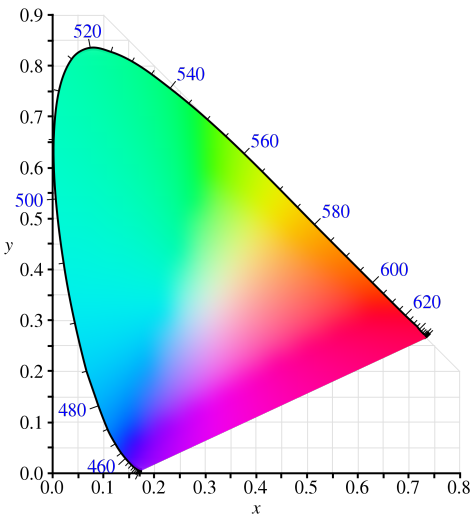


Figure 2.2: Chromatic diagram CIE xyY[2]

[2]Source: http://cs.wikipedia.org/wiki/CIE_XYZ

There are two classes of color spaces - device dependent and device independent. The color space CIE XYZ is typical representative of device independent color spaces. A typical device dependent one is the RGB color space. This color space is represented by three additive primaries - Red, Green and Blue. The RGB color space is very popular in HDRI. However, many computations are also calculated in the luminance channel Y from XYZ color space. This channel is often referred as L (luminance) in HDRI [6].

HDR Pipeline

A procedure of HDR image acquisition from capture to displaying may be summarized by a basic HDR pipeline shown in Figure 2.3.

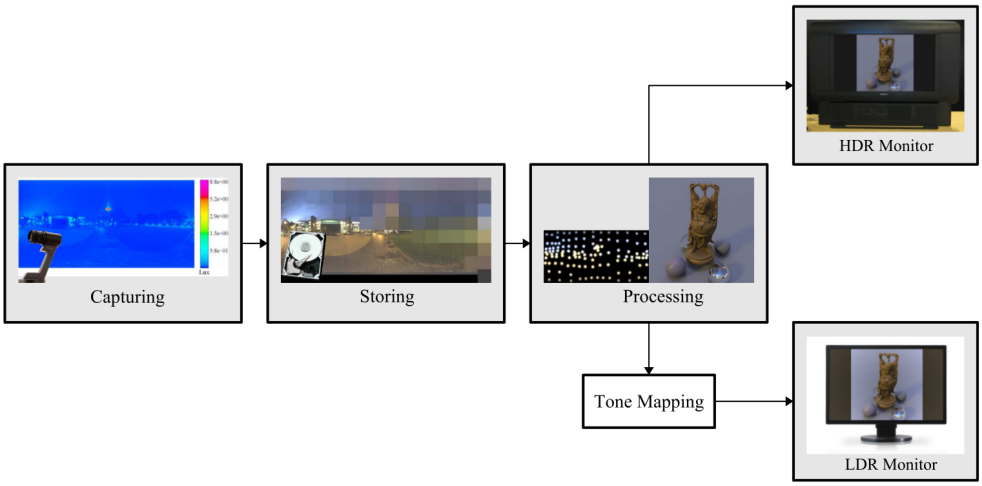


Figure 2.3: Basic HDR pipeline (taken from [6])

2.1 HDR Image Capture

Currently the biggest problem in a capture of HDR images is a possibility to record a full spectral content and dynamic range from a real world scene. An image sensor in a digital camera is exposed to a color and dynamic range of a scene and a lens is only a passive element which refocuses the incoming light onto the image plane. However, there are limitations in the image sensor design which prevent a camera to capture all of the information [76].

Therefore, there are a few ways to generate HDR images. Nowadays, the most common methods are multiple exposures composition, using of specialized hardware, expanding single-exposure LDR and computer rendering.

2.1.1 Expanding Single-Exposure LDR

An LDR content can be partly transformed into an HDR content using operators commonly called expansion operators (EO). This software technique allows an LDR image to be enhanced for viewing on HDR monitors and for using in HDR applications such as image-based lighting. This technique is analogous to the colourisation of black and white images. A number of methods for expanding an LDR image has been already proposed. Common steps in an expansion operator are [6]:

1. *Linearisation* - Creates a linear relationship between real-world radiance values and recorded LDR pixels.

2. *Expansion of pixel values* - Increases the dynamic range of the image. It means that low values are compressed, mid values are kept as they are and high values are expanded.

3. *Over/under-exposed reconstruction* - Generates the missing content in the over-exposed and under-exposed areas in the LDR image.

4. *Artefacts reduction* - Decreases artefacts resulting quantization or image compression, which can be visible after the expansion of pixel values.

5. *Color correction* - Keeps colors as in the original LDR image because colors are desaturated during expansion of pixel values.

2.1.2 Specialized Hardware

In recent years, some companies, such as Canon, Nikon, Sony, Sigma, etc., provide HDR cameras based on automatic multiple exposure capturing using features such as multi-exposure capturing or automatic exposure bracketing connected with automatic exposure merging. While others develop digital sensors that can natively record bigger dynamic range of a scene in a single shot.

There are a few researches dealing with sensors for HDR capture. For example High Dynamic Range CMOS (HDRC) and a family of SuperCCD with SuperCCD HR (High Resolution), SuperCCD SR (Super Dynamic Range) and SuperCCD EXR [8]. The sensors can be based on either CMOS (Complementary Metal-Oxide-Semiconductor), such as HDRC, or CCD (Charge-Coupled Device) technology like SuperCCD.

Numerous attempts have been tried to expand the dynamic range of CMOS image sensors. Generally, these methods can be divided into three categories [53]:

Logarithmic response HDR CMOS image sensor
These sensors use logarithmic response pixels or circuits to non-linearly extend the dynamic range which is possible because the photo-current flowing through a resistor with logarithmic current-voltage characteristic. Generally, this resistor can be implemented by a MOS transistor operating in weak inversion mode. Figure 2.4 shows its structure. The most disappointed disadvantage of the logarithmic image sensor is due to its working theory, non-linear response, which is not preferred in most of applications [53].

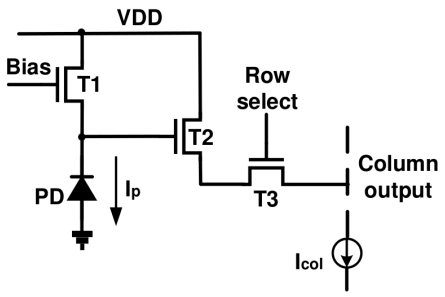


Figure 2.4: Logarithmic response pixel structure (taken from [46])

Lateral overflow integration capacitor implemented HDR sensor
These sensors apply a lateral overflow capacitor to improve the operation range. This type of HDR CMOS imager achieves from a 100dB up to 180dB dynamic range. The biggest advantage of such types of HDR image sensors is their linear response which is preferred in developing the color processing for wide dynamic range imaging data. A basic structure of the lateral overflow integration capacitor implemented HDR sensor is shown in Figure 2.5. Disadvantages of this pixel are mainly situated around the transfer gate. A partially opened TX transistor during the exposure could introduce additional dark current source resulting in higher dark current shot noise. Besides, the complex drive circuits are also full of challenges [53].

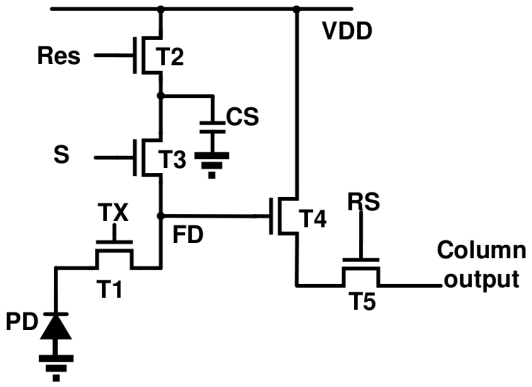


Figure 2.5: Lateral overflow capacitor HDR pixel (taken from [2])

Multiple exposure-time HDR sensor
This type of HDR sensors adopts multiple exposure-times to expand the dynamic range. This approach is increasing the read out speed to realize multiple exposures within one frame period. This method can provide a linear response and over 120dB dynamic range at the expense of complexity of an external system and frame memories. In fact, compared to a normal CMOS image sensor, the extended dynamic range is contributed by the high-speed read out circuitry. Thus, any type of pixel can be used. The most undesired problem of such a method is the discontinuous signal to noise ratio (SNR) at the transition points between different integration times. It is because the photo-diode shot noise, which is depended on illumination level, becomes the dominated noise source with the increase of illumination level [53].

Some of these sensors have even being marketed, but only a few integrated solutions are commercially available and very expensive:

High Dynamic Range CMOS (HDRC) [38]
HDRC has been developed by the Institut für Mikroelektronik Stuttgart (IMS CHIPS). This camera sensor pixel has a logarithmic response. The pixel value is digitalized into 10 bits. It means, the camera is able to capture a dynamic range of more than $10^7 : 1$. There is no need to change the exposure time or aperture settings with such large dynamic range. A disadvantage of this camera is a low resolution and more noise compared with LDR cameras. Figure 2.6 displays the dynamic range of HDRC and other sensors.

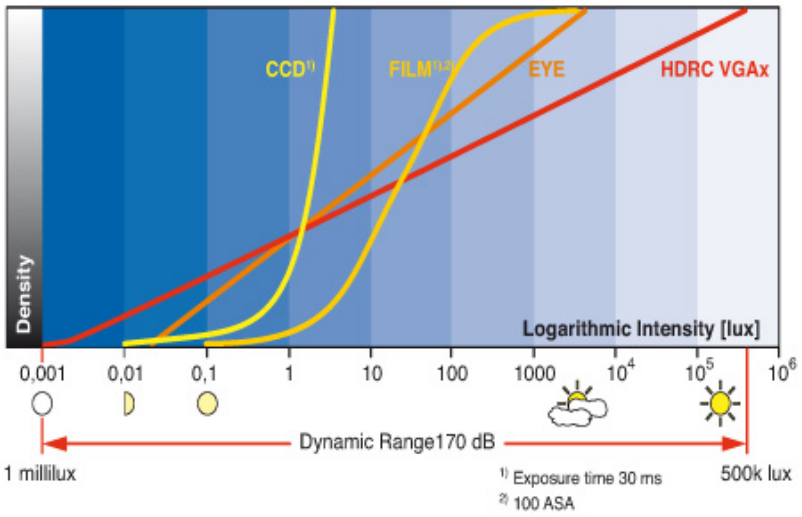


Figure 2.6: Dynamic range of HDRC and other sensors (taken from [38])

Silicon Vision Lars III [55]
This camera works on a principle of local self-adaptation. Each pixel has its exposure time and the pixel value consists of the exposure value and time-stamp determining the length of exposure. Figure 2.7 shows structure of LARS pixels.

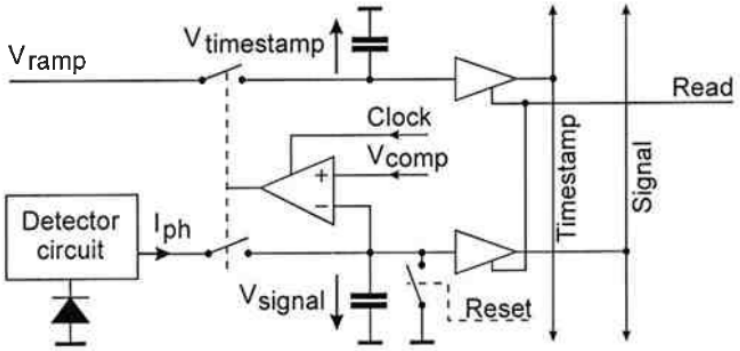


Figure 2.7: Block diagram of LARS pixels (taken from [55])

2.1.3 Multiple Exposures Fusion

Available consumer cameras are limited because they can record only a 12-bit linear or 8-bit sRGB color space into RAW format which does not cover a full dynamic range used in the real world. Therefore, the HDR image is generated by combining multiple single-exposure LDR images of the same scene which were captured at different exposure times by traditional camera technology.

Each image in the sequence of multiple exposures will have different pixels properly exposed, under-exposed or over-exposed as it is displayed later on in Figure 2.13. However, individual parts of images in the sequence must overlap for the successful restoration of dynamic range. Therefore, it is possible to ignore very dark and very bright pixels from computations of the resultant image.

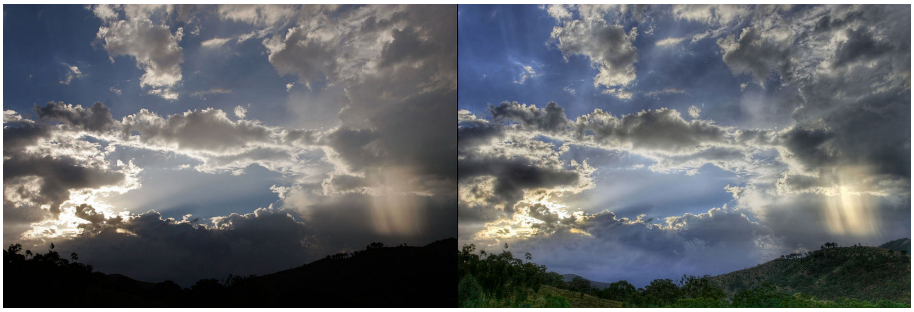


Figure 2.8: Comparison of LDR (on the left) and HDR (on the right) images[3]

Taking an Image by Digital Camera

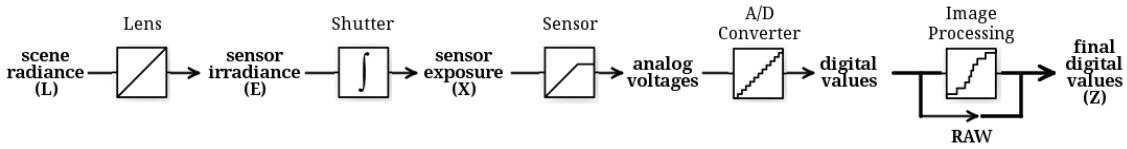


Figure 2.9: Image acquisition pipeline (inspired by [15])

Figure 2.9 shows a process of taking an image using a digital camera. The image is projected on the image sensor in the camera using the lens. The camera image sensor is formed by CCD or CMOS chip and has a linear response with a few exceptions:

- Level of black - The level of black color has not be exactly zero. The level can be shifted upwards because of the chip design, charge amplifiers, A/D converter and its noise. It means the whole color scale in the image is shifted and the black color has a greater value than zero. To eliminate this effect, the level of black has to be determined and the found value has to be subtract from the image data.
- Quantization - The quantization arises during the transfer of data from anolog to digital form. It causes bigger inaccuracies in the dark values of the image. The absolute error value can be reduced by using a converter with a larger number of bits.
- Saturation - The image sensor has a level of saturation, i.e. the maximal value of output data, because the sensor photo-diodes have a limited capacity charge. It means the camera gives the same numerical response for all values of the input luminance above the level of saturation. Pixels with this same response are called saturated and can not be used for HDR image composition. Thus, they are eliminated in the fusion methods by different ways, e.g. by using a weighting function.

The analog output data from the image sensor are converted to digital form using an A/D (analog-digital) converter which is also a linear process if the effects of quantization are ignored. The further processing follows in the camera. This further processing has the biggest effect on the camera response function. Some cameras support storing images in the RAW format which stores the output data directly from the sensor without an image processing.

[3]Source: http://en.wikipedia.org/wiki/Wikipedia:Featured_picture_candidates/Sunset_hdr

Multiple exposures fusion assumes that LDR images used to capture an HDR image are perfectly aligned, there is no movement in the scene, a noise of the CCD or CMOS sensor in the camera is not a problem and the camera is a device with the perfectly linear light measurement. These conditions are very rare in the real world [6, 76]. However, the mentioned problems with motion can be reduced by adapting alignment, ghost and noise removal algorithms from image processing which are mentioned in the next Section 2.2. To resolve the problem with a non-linear data from a camera, there is necessary to know and apply so-called a camera response curve.

Camera Response Function

Camera response function is a curve showing a relationship between amount of incoming light and image pixes values of a camera. Even if an optical sensor in a camera works quite linear, the resulting pixel values are non-linear. This may have several reasons such as the A/D (analog-digital) converter may not work linearly or there is a further processing on captured data in the camera. Common modifications are a gamma correction, white balance application and some corrections to make the image more visually appealing. The non-linear working A/D converter applies to all kinds of images but the other reasons apply only to non-RAW images. The camera response function is an important issue in the HDRI because the linear input data are necessary to compose a resultant HDR image. An inverse camera response function is used to provide linear values. A badly estimated camera response curve (non-linear data) will result in banding in contrast gradients. It is possible to deduce this function from the image exposures sequence [18].

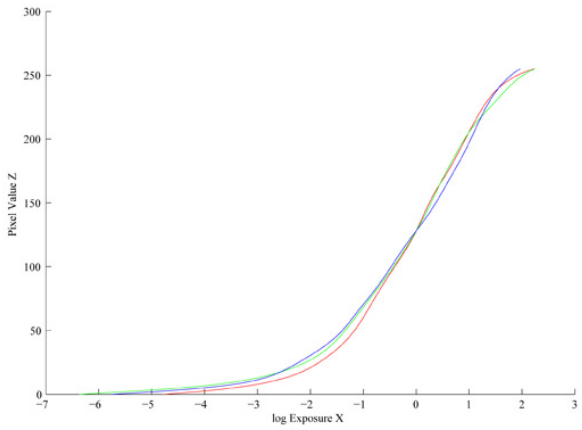
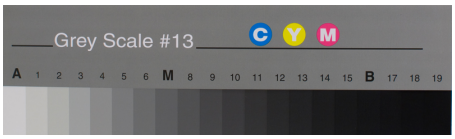


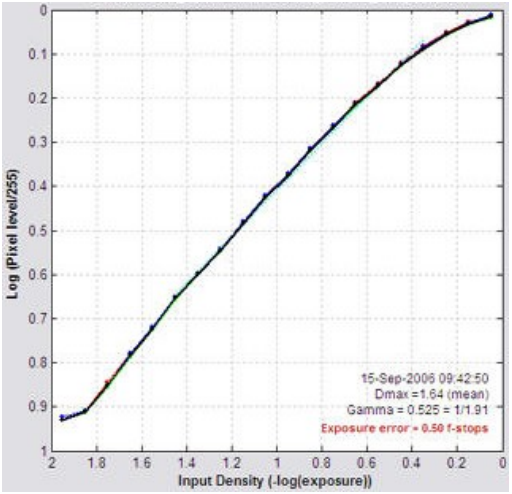
Figure 2.10: Camera response function of Canon EOS 350D[4]

A basic method to determine a camera response function is to use a table composed of several fields in grayscale. The measurement is complicated because the table has to be uniformly illuminated. Any inequalities cause a deviation in an identified response from a real camera response. This method returns only a relative linear response. An illustration of this method is shown in Figure 2.11. First, a photograph of the table is capture by a camera. Then, the camera response function is reconstructed based on known gray values in the table. This method is also used as a reference to test below mentioned methods.

[4]Source: http://cybertron.cg.tu-berlin.de/eitz/hdr/



(a) Table of gray scale Kodak Q13



(b) Measured response

Figure 2.11: Illustration of camera response function calibration[5]

There are also several methods to obtain a camera response function without a use of measuring tables. These methods use a collection of difference exposures and known relationship between them to reconstruct a camera response function.

Debevec

This algorithm was presented by Debevec et al. in [15]. The algorithm is based on exploiting a physical property of imaging systems, both photochemical and electronic, known as reciprocity. The reciprocity equation which defines the relationship between E_i, the film irradiance values, and Z_{ij}, the values of each pixel, is:

$$Z_{ij} = f(E_i \Delta t_j), \tag{2.2}$$

where i is a one dimensional index specifying the position of the pixel in the image, j is an index across different exposure times Δt_j and f is an unknown camera response function. However, since we assume the camera response function f is monotonous, it is invertible, and we can define a function g as the nature logarithm inverse to the function f:

$$g = \ln f^{-1}. \tag{2.3}$$

Then, we get an equation in the form:

$$g(Z_{ij}) = \ln E_i + \Delta t_j. \tag{2.4}$$

Unknown values are E_i and function g. Finding g implies recovering a finite number of values of $g(z)$, where $z =< z_{min}; z_{max} >$ is a finite set of values that pixels can take. Then, the problem is reduced to a search of finite number of values of $g(z)$ and N values of $\ln E_i$, which minimizes the following quadratic objective function:

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2 \tag{2.5}$$

[5]Source: http://www.imatest.com/docs/

Robertson
This method proposed by Robertson et al. [77] does not give any restrictions to the shape of the resulting camera response function. The method assumes the Gaussian weighting function to find a camera response function. The authors search for solutions of an objective function using the Gauss-Siedel iteration. The objective function is defined by authors.

Mitsunaga
This method presented in [64] approximates the camera response function by a polynom of N-th degree. The authors define an error function and they minimize this function to find coefficients of the polynom. The advantage of this method is an ability to determine the exact ratios of exposures.

Other methods
There exist also other methods, such as a histogram-based method [27] and method based on an attempt to deduce the response function from a single image [52].

Once each image is processed by a camera response function and they are in the same unit of measurement, corresponding pixels may be averaged across exposures excluding under-exposed and over-exposed pixels into the resultant HDR image [76].

The fusion of multiple exposures is the most commonly used method until recently and uses software methods to capture an HDR image. The fusion can be done in a radiance domain or image domain (see Figure 2.12).

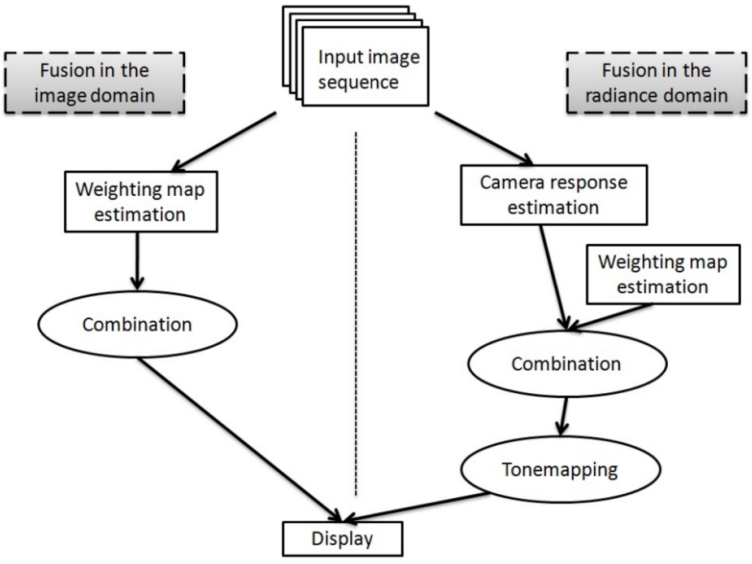


Figure 2.12: HDR image generation process (taken from [82])

Fusion in the Radiance Domain

This type of fusion was proposed by Debevec et al. [15] and consists of three steps. First, the camera response function is recovered to bring the pixel brightness values into the radiance domain. Next, the radiance maps are combined into an HDR image encoded

specially to store the pixel values that span the entire tonal range of the scene. Finally, a tone mapping operator is used to make the HDR image to be able to display on common LDR monitors [82].

Methods that combine exposures in the radiance domain give a true HDR radiance map which might be useful for later processing or displaying applications. A precision of these methods highly depends on an accurate estimation of the camera response function, which is sensitive to image noise and misalignment. Moreover, these methods require tone mapping operators.

Fusion in the Image Domain

Second type of methods presented by Mertens et al. [61] combine multiple exposures directly without a knowledge of a camera response function. These methods take only the best parts of each exposure. The resultant HDR image is obtained as a weighted average of pixel values across exposures:

$$I_{uv}^{C} = \sum_{k=1}^{N} w(Z_{uv}^{k}) Z_{uv}^{k}, \tag{2.6}$$

where I^C is a composite image, Z_{uv}^k is a pixel value at the position (u, v) and $w(Z_{uv}^k)$ is a weight of a pixel at the position (u, v).

Methods that combine exposures in the image domain are more efficient than the previously mentioned technique since they avoid the estimation of the camera response function and do not require tone mapping. These methods directly produce HDR images which can be displayed on LDR monitors.

There have been already proposed several algorithms which implement the HDR image acquisition from multiple LDR images with different exposures in the radiance or image domain. The only assumption, pixels with a linear dependency to the light in the scene, was discussed above.

Method by Mann

Probably the first known algorithm was presented by Steve Mann in [57] in 1993 and this method was more explained by himself and Rosalind W. Picard in [59] in 1995. At first, the authors offer a procedure to find the camera response curve. Then they compute a weighted average of all input images. The weighting function is based on the certainty functions. This approach provides a gradual transition between the images, where the shadow detail comes primarily from the lighter image and the highlight detail comes primarily from the darker image [59]. The paper does not present any specific equations.

Method by Debevec

This method was presented in [15] in 1997 by Paul E. Debevec and Jitendra Malik. At first, they use an amount of exposures to recover the camera response function. Then the algorithm fuses these exposures into a single HDR radiance map whose pixel values are proportional to the true radiance values in the scene. The algorithm uses all the available exposures for better robustness. For this purpose, it applies the weighting function:

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}, \tag{2.7}$$

where Z_{min} and Z_{max} are the least and greatest pixel values in the input images. The resultant pixel p in the HDR image is calculated from:

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}, \tag{2.8}$$

where E_i is the irradiance, P is the number of photographs and g is the camera response function. The algorithm assumes the exposure time Δt_j is known.

An advantageous secondary effect of the multiple exposures composition is a reducing of noise in the recovered radiance values. It also reduces the effects of imaging artifacts such as a film grain.

Method by Robertson
This method was discribed in [77] in 2003. This algorithm uses a probabilistic model and maximum likelihood method. The resultant pixels of HDR image are calculated as:

$$x_j = \frac{\sum_{i=1}^{P} w(Z_{ij}) t_i g(Z_{ij})}{\sum_{i=1}^{P} w(Z_{ij}) t_i^2}, \tag{2.9}$$

where x_j is the resultant pixel value j in the HDR image, P is the number of photographs, g is the camera response function, w is the weighted function and t is the exposure time which is known for each exposure i.

In this algorithm data from images with longer exposure times are weighted more heavily as indicated by t_i. It has an advantage of the reduction of noise.

Longest Unsaturated Exposure
This method selects resulting pixels from a single exposure which has the longest exposure time but still unsaturated pixel value at this exposure time. The resulting pixel value is calculated as:

$$L_p = Z_{longest_p}/t_{longest}, \tag{2.10}$$

where L_p is the resulting pixel value p in the resultant HDR image, the pixel value $Z_{longest_p}$ is the value of the pixel from the image with the longest exposure time where pixel is not saturated and $t_{longest}$ is the exposure time of this image.

2.2 Ghost Detection and Removal Algorithms

When the most common method for HDR acquisition, multiple exposures fusion, is used and there is some motion in the captured scene, ghost artefacts appear in the final image. There are already a lot of research studies dealing with a detection and removal of these motion artefacts in HDR images. A couple of these studies, such as [82, 28, 29, 86], make a review of the proposed methods in this area. These were used as basic research sources for this chapter.

The most common technique to obtain an HDR image is a combination of multiple different exposures. However, this method has two main problems regarding motion artefacts in resultant image. First one is a global camera motion called as misalignment. This can be solved by placing a camera on a tripod or by using an image registration method. For

example, Greg Ward in [88] proposed an efficient solution by using the median threshold bitmap technique. There is also a possibility to use methods based on key-points extraction and matching as well, such as Harris corners detector [31] or SIFT features detection [54]. The main goal of this thesis is to focus on the second problem: moving objects in the scene while capturing the images called as ghost artefacts. Figure 2.13 shows multiple exposures images with motion and a resultant HDR image with a ghost artefact.

(a)

(b)

Figure 2.13: (a) Six images with different exposures and (b) The resultant HDR image with ghost artefacts (both taken from [82])

Several methods which solving ghost artefacts in HDR images have been already developed. Most of them consists of two steps: a ghost detection, the detection of regions where ghosts appeared, and a ghost removal.

2.2.1 Ghost Detection Methods

Ghost detection methods detect motion in a sequence of exposures where a moving object can appear on a static background or on a moving background with static or dynamic objects. The following methods, mostly taken from [82], can detect both or only the first mentioned type of motion.

Variance-Based Ghost Detection

Variance based ghost detection method published in [76] by Reinhard et al. detects regions with moving objects based on weighted variance measure. First, the camera response function and the radiance maps for each LDR are computed. Then a Variance Image (VI) is generated by evaluating the variance of radiance values at each spatial location (u, v):

$$VI_{uv} = \frac{\sum_{k=1}^{N} w(Z_{uv}^k)(E_{uv}^k)^2 / \sum_{k=1}^{N} w(Z_{uv}^k)}{(\sum_{k=1}^{N} w(Z_{uv}^k)E_{uv}^k)^2 / (\sum_{k=1}^{N} w(Z_{uv}^k))^2} - 1, \tag{2.11}$$

where Z_{uv}^k = pixel value at the position (u, v) in exposure L_k, E_{uv}^k = estimated radiance value at the position (u, v) in exposure L_k and the weighting function is defined as:

$$w(Z_{uv}^k) = \begin{cases} Z_{uv}^k & \text{if } Z_{uv}^k \leq 127 \\ 255 - Z_{uv}^k & \text{if } Z_{uv}^k > 127 \end{cases}. \tag{2.12}$$

The resultant VI can be used as a likelihood measure for intra-image movements because regions inclusive motion exhibit high variance. Regions where the local variance measure is above a defined threshold, are detected as ghost regions:

$$G_{uv} = \begin{cases} 1 & \text{if } VI_{uv} \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}. \tag{2.13}$$

This method has weak results and cannot be used if moving objects have similar colors as a background. Therefore, Jacobs et al. in [40] proposed another measure derived from entropy.

Entropy-Based Ghost Detection

First, a local neighbourhood based entropy map is computed for each LDR image. For each pixel (u, v) in L_k, the entropy is calculated from a local histogram computed in the window of size $r \times r$ around (u, v), where r is an odd number bigger than 1:

$$H_{uv}^k = -\sum_{x=0}^{B-1} P(X = x) \log\left(P(X = x)\right), \tag{2.14}$$

where B is the total number of bins of the histogram and the probability $P(X = x)$ is obtained from the normalized histogram. Then, an Uncertainty Image (UI) is derived from the weighted difference of the precomputed entropy image as follows:

$$UI_{uv} = \sum_{k=1}^{N} \sum_{l=1}^{l<k} \frac{v^{kl}}{\sum_{k=1}^{N} \sum_{l=1}^{l<k} v^{kl}} h_{uv}^{kl}, \tag{2.15}$$

where $h_{uv}^{kl} = |H_{uv}^k - H_{uv}^l|$ and $v^{kl} = \min(w(Z_{uv}^k), w(Z_{uv}^l))$. The weighting function is defined as:

$$w(Z_{uv}^k) = \begin{cases} (Z_{uv}^k \times 0.9/127) + 0.05 & \text{if } Z_{uv}^k \leq 127 \\ ((255 - Z_{uv}^k) \times 0.9/127) + 0.05 & \text{if } Z_{uv}^k > 127 \end{cases}. \tag{2.16}$$

The resultant UI is used to find ghost regions based on thresholding:

$$G_{uv} = \begin{cases} 1 & \text{if } UI_{uv} \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}. \tag{2.17}$$

Prediction Based Ghost Detection

Prediction based method proposed by Thorsten Grosch in [26] uses the deviation between the predicted intensity value of a pixel and the actual intensity as a measure to find ghost pixels. The deviation is calculated from two images L_k and L_l using the estimated camera response function:

$$Z_{uv}^{l} = f(\frac{\Delta t_l}{\Delta t_k} f^{-1}(Z_{uv}^{k})), \tag{2.18}$$

where f is the camera response function and Δt_l and Δt_k are the exposure times of L_k and L_l, respectively.

Pixels which show a significant difference between the predicted and the actual value for each pair of consecutive input LDR images, are marked as ghost pixels:

$$G_{uv} = \begin{cases} 1 & \text{if } |Z_{uv}^{l} - Z_{uv}^{k}| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} . \tag{2.19}$$

Pixel Order Relation

It is possible to relate pixel values to radiance values using the camera response function $f()$ as published by Sidibé et al. in [80]:

$$Z_{uv}^{k} = f(E_{uv}^{k} \Delta t_k). \tag{2.20}$$

An increase in radiance values always produces an increased or equal recorded pixel values. The pixel order relation method uses this evidence to assume that $f()$ is monotonic. Then, the intensity values for each pixel location (u, v) in different exposures must satisfy:

$$Z_{uv}^{k} \leq Z_{uv}^{l}, \text{ if } \Delta t_k < \Delta t_l. \tag{2.21}$$

The creation of ghost map ensues from the previous equation as:

$$G_{uv} = \begin{cases} 0 & \text{if } Z_{uv}^{1} \leq Z_{uv}^{2} \leq \ldots \leq Z_{uv}^{N} \\ 1 & \text{otherwise} \end{cases} . \tag{2.22}$$

Bitmap Movement Detection

This method uses the median threshold bitmap (MTB) algorithm which was introduced by Ward et al. in [88] for a purpose of image alignment and taken over by Pece and Kautz [69] to detect ghost artefacts in dynamic scenes. The MTB technique helps to compare images that are taken under different exposures by effectively removing most of illumination differences between images. This method relies on the fact that if a pixel is not affected by ghost, then its relation to the median intensity of the image must be the same in all taken LDR images.

A binary bitmap B_k is computed by applying a threshold to the image L_k based on its median intensity pixels value. If the values in the image L_k are less than or equal to its median intensity pixels value, pixels of B_k are black. On the other hand white regions of B_k indicate the pixels whose values are greater than the median intensity pixels value. The obtained bitmap B_k reveals image features while removing intensity differences between different exposures (see Figure 2.14).

(a) Original Exposures: +2 and -2 stops respectively.

(b) Bitmaps generated by MTB.

Figure 2.14: Bitmap similarity using MTB technique (taken from [69])

By summing up all computed bitmaps into image M^*, the pixels affected by movement are detected because each pixel preserves its bit value across all B_k in the static scene. The morphological operations (dilation and erosion) are applied on the image M^* to reduce noise. Then, any pixel in the M^* that is neither 0 nor N (N is a number of exposures) is classified as a movement:

$$M^*_{uv} = \begin{cases} 0 & \text{if } S_{uv} = 0 \text{ or } S_{uv} = N \\ 1 & \text{otherwise} \end{cases}, \tag{2.23}$$

where S_{uv} is the sum of the bitmaps values at location (u, v):

$$S_{uv} = \sum_{k=1}^{N} B^k_{uv}. \tag{2.24}$$

M^* is converted into a cluster map M where each identified cluster has a different label which is computed using connected component labelling [16]. An overview of this technique is illustrated in Figure 2.15.

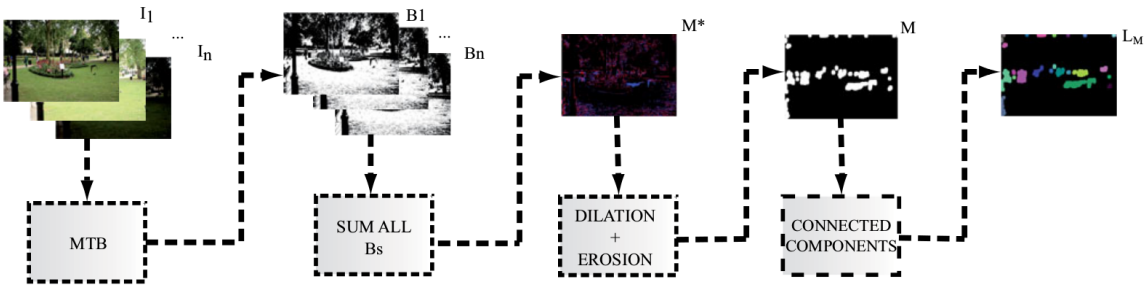


Figure 2.15: Overview of bitmap movement detection algorithm (taken from [69])

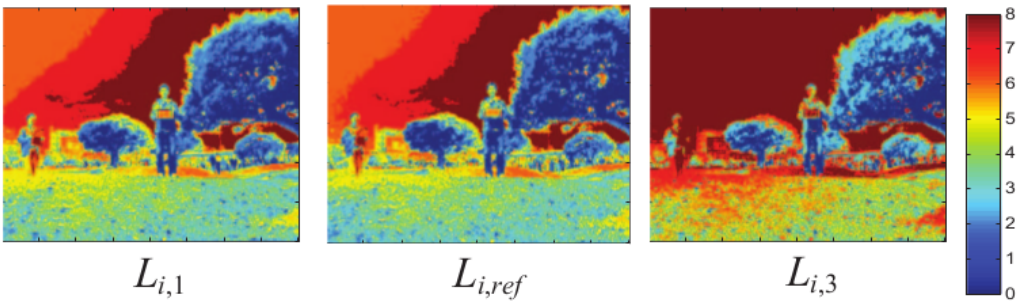
This method uses a fusion in an image domain but it is possible to integrate it also into a radiance domain fusion. The algorithm works well on a large variety of movement configurations. Moreover, the method is faster than other de-ghosting algorithms, relies only on simple binary operations and thus it can be easily implemented directly on a camera hardware [69].

Histogram Based Ghost Detection

This method proposed by Min et al. [62] calculates ghost maps based on multi-level threshold maps which are extended from the MTB. It takes an advantage of a condition that grey levels at a particular pixel location must exhibit an increasing or equal property when the images are captured from lowest to highest exposures.

First, each image z_j is divided into N levels which gives a set of N threshold values $T_{j,k}$, where each level has the same number of pixels. Then, the multi-level threshold maps L_j are computed by classifying the intensity value of z_j into N levels using these thresholds. Figure 2.16b shows the multi-level threshold maps L_j, $1 \leq j \leq 3$, $N = 8$, extracted from LDRIs in Figure 2.16a.

(a) LDRI sequence with three different exposures



(b) Multi-level threshold maps

Figure 2.16: Motion detection by histogram based method (taken from [62])

Finally, the ghost maps are estimated using the computed multi-level threshold maps for each LDRI excluding the mid-exposure LDR which is taken as the reference image:

$$G_{i,j} = \begin{cases} 1 & \text{if } |L_{i,ref} - L_{i,j}| \geq 1, j \neq ref \\ 0 & \text{otherwise} \end{cases} . \tag{2.25}$$

Thus, the method produces $j - 1$ ghost maps, where j is a number of input exposures. This method generates the radiance map based on Debevec et al. [15] and incorporates computed ghost maps into their weighting factor.

Lee et al. proposed an improvement of this algorithm in the Improved Histogram Based Ghost Detection in [49]. Later on Ahirwal et al. also built on this method with their Ghost-Free High Dynamic Range Imaging Using Histogram Separation and Edge Preserving Denoising in [1].

Patch-Based Ghost Detection

This algorithm was presented in [79] by Sen et al. This is a patch-based energy minimization formulation. The algorithm uses the reference image L_{ref} which is an LDR image that contains most well exposed pixels. The resultant HDR image contains as much information as possible from this reference image L_{ref}. In places, where the reference image L_{ref} is not well exposed, every patch in the image H at a given exposure should have a similar patch in one of the LDR images after exposure adjustment (coherence). Also, every exposure adjusted patch in all L_k images should be contained in H at exposure k (completeness). The iterative approach performs joint optimization of image alignment and HDR merge process until all the exposures are correctly aligned to the reference exposure and a good quality HDR result is produced [29].

Hu et al. [36] present another patch-based method in 2013 based on an HDR generation from exposure stacks. This algorithm uses an iterative approach to register LDR images to a reference image. The algorithm produces a registered stack from a sequence of misaligned images of dynamic scenes. The algorithm propagates the intensity and gradient information during HDR reconstruction. This approach performs well even if the reference image has large saturated areas [29].

RANSAC Based Ghost Detection

This method [22] belongs to a group of patch-based methods. The method is base on the fact that the intensity values at any location (u, v) in any two input images L_k and L_l satisfy the following condition:

$$\frac{Z_{uv}^k}{\Delta t_k} = \frac{Z_{uv}^l}{\Delta t_l}. \tag{2.26}$$

Besides saturated pixels, the above rule is broken only at locations affected by ghost. The processing is performed on a patch level in order to be robust to noise. At first, the least saturated image is selected as the reference L_{ref}. Then, log intensities of an $r \times r$ patch in L_k are plotted against the log intensities of the corresponding patch in the reference image L_{ref} in order to find patches of L_k affected by ghost. A best fit line through the plot is obtained by the RANSAC procedure [21] and the percentage number of outliers is calculated using a distance threshold. If the percentage is greater than the threshold, the tested patch includes a ghost.

Graph-Cuts Based Ghost Detection

Heo et al. [32] use joint probability density functions between exposure images to get global intensity transfer functions to roughly detect ghost regions. These regions are further refined using energy minimization based on graph-cuts methods [11]. First, joint intensity histograms are constructed for the color channel $c \in R, G, B$ between the reference image L_{ref} and another image L_k:

$$P_{ij}^{ref,k|c} = \sum_{u=1}^{U} \sum_{v=1}^{V} G_{uv}^k \times T[(i,j) == (Z_{uv}^k, Z_{uv}^{ref})], \tag{2.27}$$

where T is 1 if the argument is true and 0 otherwise, and k is a number of exposure. A ghost map is defined for each exposure by:

$$G_{uv}^{k} = \begin{cases} 1 & \text{if } P_{ij}^{ref,k|c}(Z_{uv}^{ref|c}, Z_{uv}^{k|c}) \leq threshold \\ 0 & \text{otherwise} \end{cases} . \tag{2.28}$$

Because the ghost regions estimated by the above equation are noisy, they are refined by an energy minimization approach using graph-cuts method. The minimize energy function is defined as:

$$E(f_k) = \sum_{p} D_p(f_k(p)) + \sum_{p} \sum_{q \in N(p)} V_{pq}(f_k(p), f_k(q)), \tag{2.29}$$

where the boolean label $f_k(p) \in 0, 1$ represents whether the pixel $p = (u, v)$ in the exposure L_k is affected by ghost or not. $N(u, v)$ represents the neighbourhood of the pixel p. More details about a data cost function D, a smoothness term V and more information generally about this method can be found in [32].

The optimized label map f_k is used to update the ghost map G and the process is repeated iteratively until convergence. The authors of this method found that two or three iterations are sufficient for convergence. This algorithm does not require accurate ghost detection and not suffer from the color artefact problem.

Optical-Flow Based Ghost Detection

Optical flow algorithms are recognized as one of the most successful algorithms in aligning differently exposed LDR images by motion compensation. There are already a lot of optical flow algorithms for HDR image acquisition such as Kang et al. [44], Mangiat and Gibson [56], Zimmer et al. [97]. Zimmer et al. use state-of-the-art optical flow approach to register LDR exposures before the merging process. They minimize their proposed energy function that uses a data term and smoothness term to reconstruct saturated and occluded areas. After alignment, the displacement fields obtained with subpixel precision are used to produce a super resolved HDR image [29]. Instead of energy-based function, there is possible to use gradient-based optical flow approach in this type of algorithms as well.

Markov Random Field Detection

Jinno and Okuda [41] use detection based on the Markov random field (MRF) model and estimate displacements, occlusion and saturated regions simultaneously by using Maximum a Posteriori (MAP) estimation instead of a ghost map creation. They do not estimate accurate motion vectors but compute displacement to the pixel with the closest irradiance.

Singular Value Decomposition Based Ghost Detection

This method [83] uses singular value decomposition (SVD) to resolve the ghosting problem. The method is based on extracting local spatio-temporal neighbourhoods and using the second biggest singular value of the matrix formed by values within the neighbourhoods as a measure for ghost detection.

2.2.2 Ghost Removal Methods

Ghost removal methods can be divided into two main categories. The first type of methods remove ghost artefacts while keeping a single occurrence of the moving object. It means the ghost will disappear and the moving object will be at the fixed location in the resultant HDR image. Other methods completely remove the moving object from the final image.

Keeping a single occurrence of the moving object

The simplest approach to keep a single occurrence of moving object in the final HDR image, is to apply the standard multiple exposure fusion method in ghost-free regions while selecting a single reference exposure in ghost affected areas. This approach requires a computed ghost map. The reference exposure is typically the image which is least saturated [76, 40] or the image whose ghost regions are best kept in range [26]. Another approach is to determine the correct number of exposures to use in different ghost affected areas [22]. However, using a single reference exposure introduces new artefacts in the resultant HDR image. These new artefacts are created at ghost regions boundaries. For the better result without new ghost artefacts in the final image, it is possible to use a Laplacian pyramid blending framework [69, 61], where the input images are decomposed into a Laplacian pyramid, which basically contains band-pass filtered versions at different scales and blending is performed for each level separately. Another possibility is to use a gradient domain approach [22] to avoid all ghost artefacts in the resulting image. This method is based on estimating an image whose gradient is closest, in the mean squared error sense, to the gradient of the estimated radiance map. Zhang and Cham [92, 93] use gradient information to generate ghost-free HDR images directly without a ghost detection.

Complete removing of the moving object

Some methods completely remove all moving objects from the final HDR image. This can be desirable in some cases where the photographer is interested in the scene or the background itself. The most simple approach to achieve this goal, is to discard exposures effected by ghost regions during the combination step of HDR acquisition process. This idea is used by Sidibe et al [80]. Two sets of exposures, A_{uv} and B_{uv}, are classify for each pixel location (u, v), where the exposures in the set A_{uv} contain ghosting at location (u, v), while the second set B_{uv} represents exposures without ghosts. Therefore, a ghost-free HDR image is created by combining only exposures from the set B_{uv}. A similar approach is used also by Gallo et al [22]. However, this algorithm is based on image patch processing rather than working with pixels individually. Methods proposed by Khan et al. [47] and by Pedone and Heikkilä [70] directly remove ghost artefacts without a ghost detection by adjusting the weighting function $w()$ when the combined radiance map is calculated. These methods do not need explicit ghost detection as they change pixels weights directly and iteratively to minimise the number of visible artefacts.

These algorithms assume that moving objects appear in a small number of images at each pixel location. Moreover, these methods require a sufficiently large number of images and can be computationally expensive since they require a number of iterations to produce good results [82].

2.3 HDRI Storing

High dynamic range images record a much wider gamut than standard 24-bit RGB (8 bits for each channel). HDR images are assembled from 32-bits floating point numbers. It means a big amount of data and a different format of data than a common 24-bit color image. Moreover, colors in LDR images are usually associated with some target output device while HDR image pixels have a direct relation to radiance in a scene [76]. Therefore, some special data formats are needed to store HDR images. The following Table 2.3 taken from [8] and supplemented from [76] provides a summary of existing HDRI data formats. Whole color gamut cover only TIFF, OpenEXR and JPEG-HDR. Besides the presented formats, there are few more so-called private formats which are not defined by a norm or they are not even documented. These formats are used only in specific software [8].

	Channels	**Total Bits per Pixel**	**Compression**	**Precision** (1-3)	**Dynamic Range (EV)**
TGA (8bit RGB)	RGB (+ Alfa)	24	RLE	1	8
Cineon, DPX	RGB	32	-	2	12
TIFF floating point	RGB (+ Alfa + ...)	96	ZIP, LZW	3	253
TIFF LogLuv 24/32	L+Index/Lu'v'	24/32	-/RLE	2	16/126
PFM	RGB	96	-	3	253
Radiance HDR	RGBE	24	RLE	2	253
OpenEXR	RGB (+ Alpha + Depth + ...)	48	PIZ, ZIP, RLE, Wavelet, PXR24, ...	3	30
JPEG-HDR	YCC	variable	JPEG	1	30
Windows WDP	RGBE	variable	Wavelet	1	

Table 2.3: HDRI data formats

The most important thing in HDRI storing is a data compression because captured real-world lighting results in very large data sizes. Uncompressed, a single HDR pixel requires 12 bytes of memory to store the three single precision floating-point numbers for the RGB values. This becomes a lot of data in images and video. Therefore, researchers have been working on more sophisticated compression schemes to store HDR content. The main strategy is to modify and adapt current compression standards such as JPEG, MPEG and block truncation coding. The main goal of the compression is to significantly reduce the amount of data that needs storing and transmitting with minimal loss of visual quality. This is a major challenge as the increased luminance means that many artefacts, which may not be noticed in LDR content, will be easily distinguishable in HDR image [6]. Table 2.4 summarizes what has been already achieved for compressing HDR images, textures and

videos. More details about techniques from the table can be found in [6].

Name	Description	BPP	Quality	Backward Compatibility
IMAGE COMPRESSION				
JPEG-HDR	backward compatible JPEG-HDR	0.6-3.75	MQ-HQ	Yes
HDR-JPEG2000	HDR-JPEG2000	0.48-4.8	HQ	Yes
TLCAHDR	Two-Layer Coding Algorithm for High Dynamic Range Images	1-8	HQ	Partial
TEXTURE COMPRESSION				
HDRTGS	HDR Textures Compression Using Geometry Shapes	8	HQ	No
HDRTBIO	HDR Textures Compression Using Bit and Integer Operations	8	HQ	No
HDRTSL	HDR Textures Compression Encoding LDR and HDR Parts	16	MQ	No
HDRTTMITM	HDR Textures Compression with Tone Mapping and Its Analytic Inverse	4-8	MQ-HQ	Yes
DHTC	An Effective DXTC-Based HDR Testure Compression Scheme	8	HQ	No
VIDEO COMPRESSION				
HDRV	Perception-Motivated High Dynamic Range Video Encoding	0.09-5	HQ	No
MPEG-HDR	backward compatible HDR-MPEG	0.2-6	HQ	Yes
H.264-HDR	Rate-Distortion Optimized Compression of High Dynamic Range Videos	0.26-4	HQ	Yes

Table 2.4: Summary of various HDR content compression techniques (taken from [6]). BPP means a range in the case of varying quality, MQ means medium quality and HQ means high quality.

2.4 Tone Mapping

Tone mapping is a huge and important part of HDRI. Tone mapping is an operation which converts the dynamic range of world luminance to lower dynamic range display luminance. An important requirement is that the perception of the real-world scene should match with the perception of the tone mapped image. Tone mapping is done by using tone mapping operator (TMO). There are a few groups of TMOs which differ in the function operator using for tone mapping or in the image processing techniques. The main groups are [6]:

- **Global operators** - The tone mapping is applied to whole image with the same operator.
- **Local operators** - Each pixel is mapped depending on its neighbours. These neighbouring pixels are given as an input to the function operator.

- **Segmentation operators** - A different mapping is applied to regions in which the image is segmented.
- **Frequency/Gradient operators** - A tone mapping operator is applied only to the low frequencies in the image.

Based on the design philosophy of the TMOs there is further classification [6]:

- **Perceptual operators** - The function operator models some aspects of the human vision system.
- **Empirical operators** - The function operator tries to create aesthetic-pleasing images without using the human vision system aspects.
- **Temporal operators** - These operators are also suitable for HDR video and animations.

The summary of TMOs is in the Table 2.5.

	Empirical	**Perceptual**
Global	Linear Mapping Exponential Logarithmic Mapping Quantization Technique	Perceptual Brightness Reproduction Contrast Based Scale Factor Visual Adaptation Model Histogram Adjustment Time Dependent Visual Adaptation Adaptive Logarithmic
Local	Spatially Variant Tone Reproduction Photographic Tone Reproduction	Multi-Scale Tone Mapping Operator for High Contrast Images Local Model of Eye Adaptation
Segmentation	Interactive Manipulation Exposure Fusion	Segmentation Approach Lightness Perception
Frequency/ Gradient	Low Curvature Image Simplifiers Bilateral Filtering Gradient Domain Compression	Trilateral Filtering Image Color Appearance Model Retinex Methods

Table 2.5: Taxonomy of tone mapping operators [6]

Only Perceptual Brightness Reproduction, Time Dependent Visual Adaptation and Local Model of Eye Adaptation are also temporal and suitable for HDR video content [6].

The detailed information about tone mapping can be found in the book *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting* written by Erik Reinhard et al. [76] which contains very detailed information about this field of study. Books [6, 8, 33] also include some chapters dealing with tone mapping.

Chapter 3

FPGA Programming Principles

Nowadays, FPGA (Field Programmable Gate Array) circuits are used in a wide range of applications mainly due to its programmability, efficiency and flexibility. An advantage is its steadily declining power consumption as well as the prize of the chip itself. They are typically used in the area with smaller series of devices where it does not worth to make a design of the customer integrated circuit and on the other hand, the solution with the universal processor is not enough. Their further use is in the prototyping of complex customer integrated circuits, where the designed circuits are implemented and tested before the actual manufacturing.

This chapter describes a theoretical basis about FPGA and its programming principles which are necessary to know for a correct solution design. The information in this chapter was mainly taken from books *Tree-based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization* written by U. Farooq et al. [19], *Introduction to Reconfigurable Computing: Architectures, algorithms and applications* written by C. Bobda [9], *The Designer's Guide to VHDL, Third Edition* written by Peter J. Ashenden [5] and from the Xilinx web page [39].

3.1 Field Programmable Gate Array

FPGAs are special programmable semiconductor devices based on a matrix of small programmable logic blocks called Configurable Logic Blocks (CLBs) which are connected through complex programmable interconnects (see Figure 3.1). FPGA can be programmed to the desired application or functionality requirements. FPGAs allow designers to change their designs very late in the design cycle, even after the end product has been manufactured and deployed in the field [39]. Programmability in FPGA is achieved through an underlying programming technology.

Some parts of FPGA can be reconfigured while the rest is still running. Any future updates in the final product can be easily update by downloading a new application bitstream. The programmable logic and routing interconnect of FPGAs makes them flexible and for a general purpose. Unfortunately, it makes them also larger, slower and more power consuming than standard Application Specific Integrated Circuits (ASICs). In spite of these weaknesses, FPGA-based products are basically very effective for low and medium volume productions because of their less cost and faster time to market in comparison with ASICs and because it is easy to program and debug them.

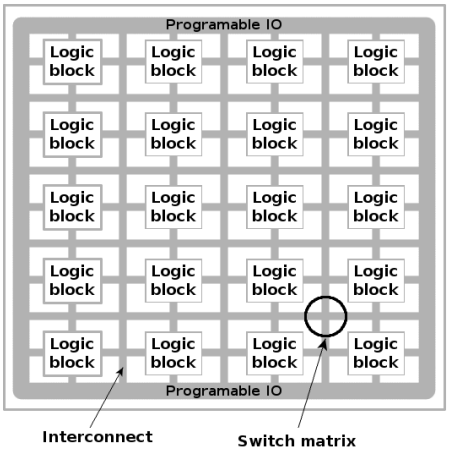


Figure 3.1: Core of FPGA architecture (taken from [9])

The progress in technology process greatly enhanced the logic capacity of FPGAs and made them successively an implementation alternative for larger and complex designs. Furthermore, programmable principles of their logic and routing parts has a considerable effect on the quality of final device's area, speed, and power consumption [19]. The base of FPGA architecture is a state machine and everything inside the architecture is parallel.

The following part describes basic components of a common FPGA.

Configurable Logic Block

Configurable Logic Block (CLB) is a basic logic unit in FPGA. These programmable logic blocks implement logic functions and storage functionality for a target application design. The basic component can be either a transistor or an entire processor but these are two extreme cases. In the case of transistors, the basic component is very fine-grained and requires large amount of programmable interconnect which could result an area-inefficiency. On the other hand, in the case of processor, the basic logic block is very coarse-grained which causes impossibility to implement small functions and it is a waste of resources. Between these two extremes there is a spectrum of basic logic blocks, such as NAND gates, an interconnection of multiplexors, lookup tables (LUTs) and programmable array logics (PAL) style wide input gates. These basic logic blocks are also called function generators because they serve as basic computing units to dynamically implement and reimplement new functions in a reconfigurable hardware devices. Commercial vendors usually use LUT-based CLBs which provide a good compromise between mentioned logic blocks [9].

Every CLB can comprise of a single Slice, or a cluster of locally interconnected Slices. Each slice consists of several basic logic elements (BLEs) (see Figure 3.2). BLE is a group name for LUTs, flip-flops and sometimes more complex units. This arrangement differs a lot in individual families of chips.

Look-up table is a group of memory cells which contains all possible results of a given function for a given set of input values. In FPGAs, LUT physically consists of a set of SRAM-cells to store values and a decoder to access a specific SRAM location and retrieve a correct result for a combination of inputs [9]. **Flip-flop** can serve as one bit register/memory [66].

CLB can be also consisted only from one BLE. However, modern FPGAs contain typically 4 to 10 BLEs divided into few Slices in a single cluster. The exact number of CLB and their features vary from device to device [39, 19].

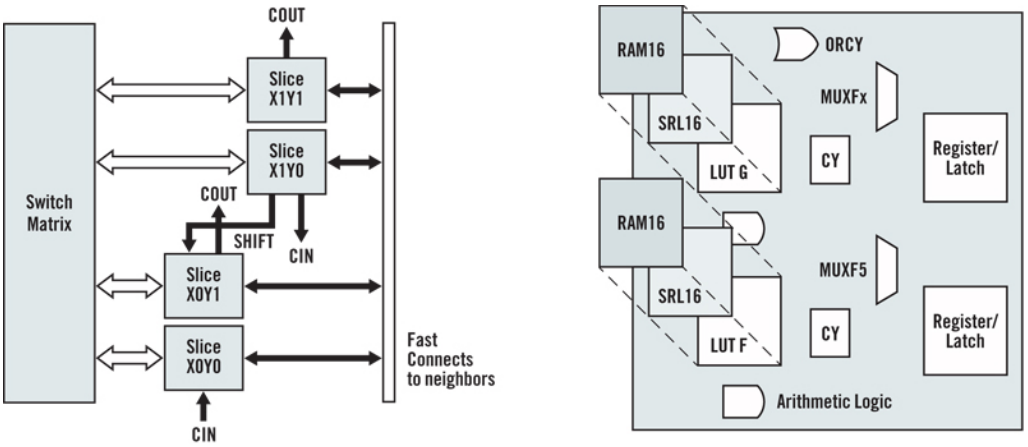


Figure 3.2: Basic Configurable Logic Block Structure (left) and Slice (right) (taken from [39])

Interconnection Network

Programmable routing interconnect of FPGAs comprises of almost 90% of total FPGA's area [9]. A flexible interconnect network routes signals between CLBs and to and from I/Os, while the CLB provides a logic capability. Routing comes in several ways. One is designed to interconnect between CLBs to fast horizontal and vertical long lines spanning the device. Other is global low-skew routing for clock and other global signals. The interconnect routing task is hidden to the user unless it is specified otherwise which importantly reducing design complexity [39]. The routing interconnect of FPGA are generally steered by small multiplexors [9].

Input/Output Blocks

Today's FPGAs provide support for a lot of I/O standards. This provides a perfect interface bridge in a final system. I/Os in FPGA is grouped in banks and each bank is able to support different I/O standards independently. The best FPGAs provide over a dozen I/O banks, thus allowing flexibility in I/O support [39].

IP Cores

Many modern FPGAs contain also some complex and frequently used circuits called Intellectual Property (IP) cores. There are two types of IP cores. Hard IP cores are heterogeneous mixture of blocks which include memory, multipliers, DSP blocks and even whole processor cores. They are fabricated directly on chip and very efficient at implementing specific functions because they are designed optimally for these functions. However, they represent a waste of big amount of logic and routing resources if unused. Second type, soft IP cores, takes a form of library. They are written in Hardware Description Language (HDL). Two major HDLs are VHDL and Verilog. Both of them describes hardware circuits on register transfer level [66, 19].

3.2 VHDL

VHDL is designed to fill a number of needs in a design process and is commonly used to write text models that describe logic circuits. VHDL allows a description of a structure of a system and tells us how it is decomposed into subsystems and how those subsystems are interconnected. VHDL allows a specification of a function of a system using familiar programming language forms. As a result, it allows a design of a system to be simulated before being manufactured. It gives designers an opportunity to quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping. VHDL also allows the detailed structure of a design to be synthesized from a more abstract specification which allows designers to concentrate on more strategic design decisions and reducing time to market [5].

VHDL has constructs to handle the parallelism in hardware designs. VHDL is strongly typed programming language and is not case sensitive. There are many features of VHDL which allow to directly represent operations commonly used in hardware, such as an extended set of boolean operators including NAND and NOR. VHDL allows arrays to be indexed in either ascending or descending direction as well.

VHDL has file input and output capabilities. These files are more commonly used by a simulation programs to stimulate or verify data than as a general-purpose language for text processing. It is relatively easy to produce code that simulates successfully but that can not be synthesized into a real device, or is too large to be practical.

VHDL allows to describe and verify the behavior of the required system which is the biggest advantage of VHDL when it is used for systems design. Another benefit is that VHDL is a dataflow language which allows a description of a concurrent system. A VHDL project is multipurpose which means that once created calculation block can be used in many other projects. However, parameters of this functional block can be tuned. A VHDL project is also portable which means that if the project is created for one element base, a computing device project can be ported on another element base.

A VHDL design consists at least of two components - entity, which describes the interface and an architecture, which contains the actual implementation. In addition, most designs import library modules and some designs contain multiple architectures and configurations.

3.2.1 Basic Architecture Blocks

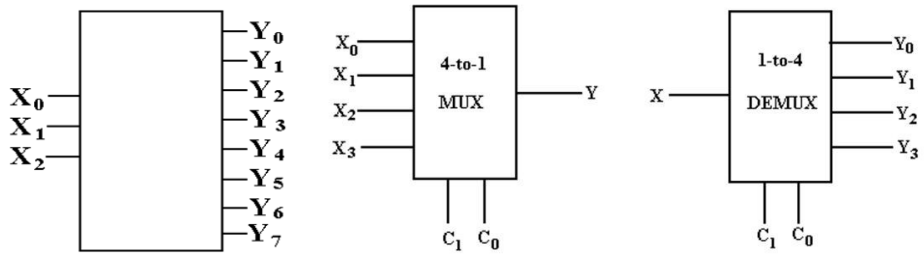
In a hardware design, basic functional blocks called combinational logic circuits, are used to design whole systems. Typical representatives of these functional blocks are [42]:

Decoder
Decoders generate some code based on a combination of input parameters. The most commonly used type of decoder is a binary decoder. It transfers binary code into code 1 to N, where only one output is active. The basic block of decoders is AND gate or NAND gate. A sample of decoder 3 to 8 is in Figure 3.3a.

Multiplexer
Multiplexer is a combinational logic circuit, which allows to transfer digital information from one selected input channel to output. Multiplexer consists of N inputs and one output. A sample of multiplexer is in Figure 3.3b.

Demultiplexer
Demultiplexers have an opposite function than multiplexors. They have one input

channel and many outputs. The output channel for input data is selected by address in a binary code while the other output channels have zero state (they are inactive). A sample of demultiplexer is in Figure 3.3c.



(a) Decoder 3x8 (b) Multiplexer (c) Demultiplexer

Figure 3.3: Sample of basic logic blocks[1]

Comparator
Comparator compares two binary numbers and generates an output signal based on their equality or diversity. Comparators are often called as XOR.

Adder
Adder calculates an arithmetic sum of two numbers. There are many types of adders, such as one-bit adder, half adder, full adder or parallel adder.

Subtractor
Number subtraction in logic circuits is treated as an addition of this number with a reverse sign.

Flip-flop
Flip-flops are the simplest sequential components. Flip-flop is a circuit that has two stable states and can be used to store state information. Flip-flops are asynchronous or synchronous. Asynchronous flip-flops immediately react to a change of input signals while synchronous flip-flops always wait for a synchronous clock impulse.

VHDL designs are based on a state machine. There are two common models of state machines - Moore and Mealy machine (see Figure 3.4). Both type of machine are triggered by a single clock and have an internal state that changes. The next state is determined by some combinational function of the inputs and the present state. The outputs are solely a function of the present state in the Moore machine. While in the Mealy machine, the outputs are a function of the present state and of the inputs [98].

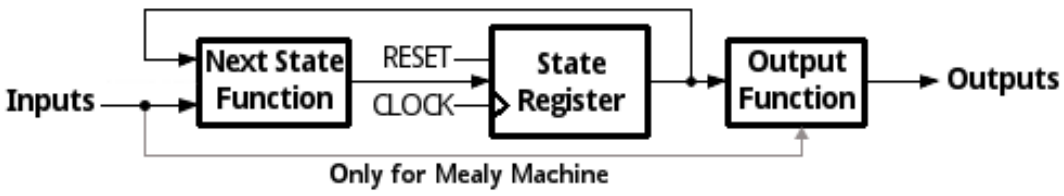


Figure 3.4: Moore and Mealy state machines[2]

[1]Source: http://www.edwardbosworth.com
[2]Source: http://www.csit-sun.pub.ro/

Chapter 4

Task Specification

This chapter defines the detailed specification of the thesis task. It includes main goals which should be achieved in this thesis. The chapter specifies de-ghosting algorithms described in Chapter 2.2 which are suitable for the given issue. A part of this chapter is also a description of the architecture and application specification.

4.1 Objectives of the Thesis

This thesis is a part of a research project dealing with the real-time HDR FPGA video camera. The FPGA video camera captures three different exposures and combines them together in a real-time processing to display a resulting HDR video on common LDR monitors. The video camera captures 60 frames per second and produces 20 frames of the resulting HDR video per second. However, there can still appear relatively big and visible artefacts in the final video. The goal of this thesis is to find or propose a de-ghosting algorithm to eliminate these ghosting artefacts. The proposed algorithm will be included into processing of the captured images on the FPGA video camera to prevent ghost artefacts in the final real-time HDR video. Fortunately, the camera using in the project stores raw data directly from the sensor into RAW format. Therefore, the output data are almost linear with a few exceptions - the level of black, quantization and saturation.

Because FPGA architecture has some specifications, the proposed algorithm has to respect few requirements. The implementation has to consume small amount of memory. Therefore, there is a problem to go back for some old data or intermediate calculation results. If the results are in the previous run of the program, the data have to be saved in an external memory. On the other hand, if the results are in the current run of the program, the access to them is easy. Another problem is a random access memory and randomness in general that causes very complex logic. The proposed algorithm has to use elements which can be easily implemented in FPGA, such as look-up tables. FPGA architecture and its programming principles were specified in Chapter 3. Due to the real-time application, the implementation has to be efficient and non-iterative. Moreover, the selected algorithm has to remove as many ghosting artefacts as possible and keep a single occurrence of moving objects in a video. These requirements prevent to use a lot of already developed de-ghosting methods. De-ghosting algorithms that come into consideration, are discussed in Section 4.3.

Development Procedure

In the beginning, this thesis will be about testing of existing de-ghosting algorithms. First, suitable algorithms for the given issue will be found. Then, they will be tested for the project requirements and quality of their results. The prototype in C++ programming language will be implemented for selected algorithms to be able to test their characteristics. After the testing of the most suitable algorithms, one of the implemented algorithms will be selected and its modification will be proposed for better use in FPGA architecture. As the last step of the thesis, the C++ implementation of the proposed algorithm will be created as a prototype and tested on images taken by the corresponding video camera. The review of the proposed algorithm and results from the final tests will be done at the end of this thesis.

4.2 Architecture and Application Specification

The implementation and programming on FPGA architecture is completely different compared to other reconfigurable platforms where the code is executed sequentially on multiple threads. On FPGA everything runs implicitly in parallel. The programming is not aimed at generating a set of sequentially going instructions for processor. Unlike this process, in FPGA, programmers seek to generate a logical circuit which implements the specified function. The logical circuit is then derived from a verbal input, such as VHDL program. This procedure is called logic synthesis. The mapping of the application to the FPGA resources is a step of the logic synthesis called technology mapping. The technology mapping targets in FPGA are look-up tables rather than NAND gate as it is in the case of many digital devices [9].

Based on the opportunity to create as many parallel computing units as needed, configurable hardware offers an excellent computing potential. There are no limits due to some existing instruction set compared to CPU or GPU. It can result in better performance and implementation of new methods which cannot be implemented on standard platforms. As a next benefit, there are not typically any unused logic functions as in fixed architectures where programmer usually use only small part of all device features.

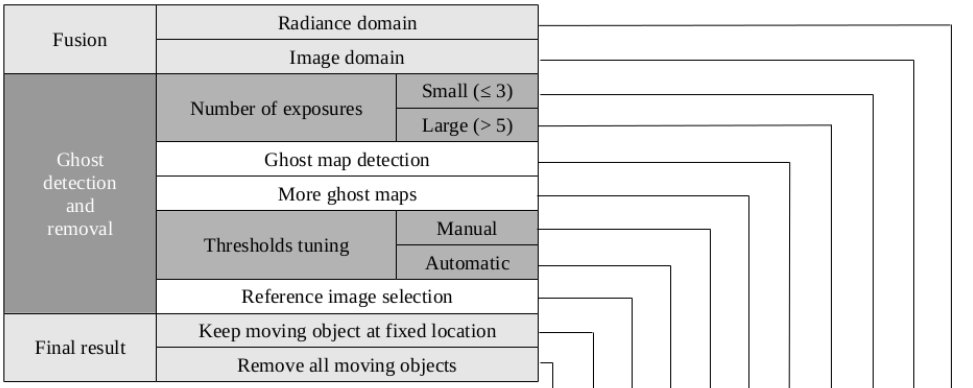
However, low level programming, such as programming for FPGA devices, has some disadvantages. Compared to standard programming on CPU, programmer has to make bigger effort to achieve the same functionality. The parallelism could complicate the design of the application in the case of bigger complexity [66].

FPGAs can be used in many applications such as pattern matching, signal processing, super computing, video streaming, image processing etc. In this case, our main interests are in image processing and video streaming. FPGA has limited resources, especially size of internal memory. Therefore the storage of whole image could be a problem. There are two solutions: a use of an external DDR (Double Data Rate) memory or a modification of the algorithm for stream processing. However, the problem with memory applies to all algorithms in general. It means that the memory space depending algorithms cannot be implemented on FPGA unless they are optimized or simplified [66].

4.3 Summary of Existing De-Ghosting Methods

The classification of the reviewed methods from Section 2.2 is shown in Figure 4.1. This classification is based on the following parameters:

- Fusion domain - radiance or image
- Number of exposures needed for good results of the algorithm
- Ghost map detection - if ghost map detection is first computed and number of these ghost maps - one or more using some exposure as a reference image
- Thresholds tuning - some input parameters such as a threshold value has to be set automatically or manually, respectively
- Reference image selection - if one of the input images is used as a reference
- Final result with an occurrence of moving object at fixed position or removal of all moving objects



Method											
Variance – *Reinhard et al. [76]*		X	X		X		X		X		X
*Entropy – *Jacobs et al. [40]*		X	X		X		X		X		X
*Prediction – *Grosch [26]*		X	X		X	X	X		X		X
Pixel Order Relation – *Sidibé et al. [80]*	X			X			X	X			X
*Bitmap – *Pece and Kautz [69]*		X			X		X		X	X	
*Histogram – *Min et al. [62]*		X	X	X		X	X		X		X
*Patch – *Gallo et al. [22], etc.*	X	X	X		X	X	X		X		X
*Graph-Cuts – *Heo et al. [32]*		X	X		X	X	X		X		X
Optical Flow – *Kang et al. [44], etc.*		X	X		X		X		X		X
Markov Random Field – *Jinno and Okuda [41]*		X	X		X				X		X
SVD – *Srikantha et al. [83]*		X		X		X	X		X		X
Density Estimation – *Khan et al. [47]*	X				X			X			X
Constraint Propagation – *Pedone and Heikkilä [70]*	X				X			X			X
Gradient – *Zhang and Cham [92, 93]*	X	X	X	X					X	X	

Figure 4.1: Classification of ghost detection methods (inspired by [82])

Tursun et al. [86] propose a more comprehensive and current summary which is shown in Table 4.1. The summary divides methods into five groups based on how approach the de-ghosting problem:

- **Global exposure registration methods** - Their aim is to align individual exposures globally.
- **Moving object removal methods** - These methods remove the motion by estimating a static background.
- **Moving object selection methods** - This type of methods detect motion in the scene by differences in the input pixel intensities and remove the ghosting artefacts by either locally using a single source image or combining a set of multiple sources which are consistent.
- **Moving object registration methods** - They recover or reconstruct the ghost pixels by searching for the best matching region in other exposures or in the affected image. The matching regions are used to transfer information to the problematic region. These algorithms may find pixel-based or patch-based dense correspondences.
- **Video de-ghosting methods** - They remove ghosting artefacts in HDR videos and make use of the temporal information of videos during processing for this purpose.

Although, the taxonomy in Table 4.1 is valid for most cases, there are some hybrid methods which are difficult to classify. These algorithms are classified based on their most dominant characteristics [86].

HDR De-Ghosting Methods						
Global Exposure Registration	Moving Object Removal	Moving Object Selection		Moving Object Registration		Video De-Ghosting
		Single Source	Multi Source	Optical-Flow Based	Patch Based	
[58]	[47]	[45]	[62] [65]	[10]	[60]	[44]
[12]	[70]	[26]	[74] [83]	[34]	[22]	[78]
[88]	[25]	[40]	[72] [93]	[97]	[68]	[56]
[99]	[80]	[51]	[50] [67]	[20]	[94]	[13]
[17]	[81]	[76]	[89] [84]	[41]	[75]	[14]
[23]	[91]	[69]	[32] [24]	[30]	[35]	[43]
[85]	[92]	[49]	[4] [87]		[79]	
[71]		[81]	[73] [48]		[95]	
[37]			[63]		[36]	
[3]					[96]	
[90]						

Table 4.1: Taxonomy of HDR de-ghosting methods (inspired by [86])

Based on Figure 4.1 and Table 4.1 the summary of methods which could be appropriate for the given issue can be done. This thesis does not deal with exposure registration because there is an expectation that input images are captured by a static camera on a tripod. Thus, the methods in the first column of Table 4.1 are unattractive for this thesis. The second column of the table is uninteresting for the thesis as well, because there is a requirement of keeping moving objects in the final HDR video.

Principles of methods mentioned in Figure 4.1 are presented in Section 2.2. This section shows that variance-based ghost detection method is not appropriate due to its weak results. On the other hand, entropy based method and prediction based ghost detection seem to be applicable for the given issue. Pixel order removal belongs to the group of methods that completely remove moving objects. Hence, this algorithm is not suitable for the purpose of this thesis. Bitmap movement detection and histogram-based ghost detection calculate histograms of input images and use median threshold bitmaps to remove ghost artifacts. These two algorithms have a simple computational complexity and give reasonable results. Therefore, they are selected for implementation and testing in this thesis which is described in the next Chapter 5.

Patch-based ghost detection is a big group of methods that could be suitable for the given issue. However, it depends a lot on circumstances of each method. It is impossible to use methods which have an iterative approach, methods which are computational challenging in the sense of FPGA architecture, such as RANSAC based ghost detection, or methods that completely remove ghosting objects from the final HDR image. As well as patch-based ghost detection, graph-cuts based ghost detection is possibly suitable for the given issue.

Unfortunately, the other methods from Section 2.2 are somehow inappropriate. Optical-flow methods are computationally challenging as well as the singular value decomposition based ghost detection and the gradient based ghost removal. MRF model and MAP estimation in the Markov random field detection are not easily implementable techniques for FPGA architecture. Density estimation and constrain propagation completely remove moving objects from final HDR images thereby unsatisfying desired requirements.

Based on above-mentioned reflections, the methods marked by an asterisk in Figure 4.1 seem to follow all desired requirements for the given issue.

Chapter 5

Realization of De-Ghosting Algorithms

This chapter focuses on the description of the implementation of algorithms which were selected in the previous Chapter 4. Bitmap movement detection is described in Section 5.1. This section also includes illustrations of results from this implemented method. Section 5.2 describes histogram-based ghost detection with its results as well. Based on the histogram-based de-ghosting method, Section 5.3 defines a modification applied on this method to simplify its implementation on FPGA architecture. As a part of this section, an evaluation of the proposed modification is presented.

Four different sequences were used to test implemented algorithms. All used images were captured by the FLARE 2KSDI video camera and saved into the RAW format. The images of Scene 1, which can be found in Figure A.1 in Appendix A, were used for this chapter. All the other input images are in Appendix A as well. The exposure times were set to 0.5ms, 2ms and 8.3ms.

The implementation is in C++ programming language and the OpenCV library is used to simplify image processing. Because the image sensor in the video camera has Bayer filter [7], the input images has to be converted by using *cvtColor* method before own image processing. Both implemented methods are tested on grayscale images.

The easy tone mapping operator $e^{(log_{10}(\mathrm{HDR}))}$ for all computed HDR images is used to display the final HDR image on common LDR monitors.

5.1 Bitmap Movement Detection

The theoretical basis of this method were described in Section 2.2. This method uses fusion in an image domain. However, it can be easily integrated into radiance domain fusion as well. For better comparison with the other method, the radiance domain fusion is used in this thesis. Moreover, the project dealing with the real-time HDR FPGA video camera uses also the fusion in the radiance domain.

First, histograms of input images are computed to find a median intensity pixels value in each exposure. The found median value serves to compute a median threshold bitmap. If the values in the input image are less than or equal to its median intensity pixels value, pixels of the binary bitmap are black. On the other hand white regions of the binary bitmap

indicate the pixels whose values are greater than the median intensity pixels value. The obtained binary bitmap reveals image features while removing intensity differences between different exposures. The binary bitmaps for input images of Scene 1 are shown in Figure 5.1. Note the similarity of computed bitmaps.

(a) Bitmap for under-exposed image

(b) Bitmap for mid-exposed image

(c) Bitmap for over-exposed image

Figure 5.1: Bitmaps for input images of Scene 1 generated by MTB technique

Next, the ghost map is calculated by summing up all computed bitmaps into image M. The pixels affected by movement are detected because each pixel preserves its bit value across all binary bitmaps in the static scene. Any pixel in the image M that is neither 0 nor N, where N is a number of input exposures, is classified as a movement. The morphological operations, erosion and dilation, with the erosion kernel size $s_e = 3$ and dilation kernel size $s_d = 17$, are applied on the ghost map to reduce noise. The s_e sets the sensibility of the algorithm to isolate and eliminate the outliers from the moving pixels and s_d is directly responsible for the enlarging of the moving clusters when moving pixels are missed. The kernel sizes were selected based on [69]. The final ghost map for input images of Scene 1 is shown in Figure 5.2.

Figure 5.2: Detected ghost map for Scene 1

The resulting HDR image is created by fusion in the radiance domain which means to apply Equations 2.7 and 2.8. To remove ghost artefacts from the final HDR image, the ghost map is used. The final HDR image has equal pixel values as the mid-exposed input image in the region where the ghost map has white pixels. In other words, the regions where the ghost artefacts were detected, are filled by pixels from the mid-exposed input image instead of the proper radiance domain image fusion. The final HDR image is shown in Figure 5.3b. To compare a quality of the de-ghosting method, the HDR image without the ghost removal is shown in Figure 5.3a.

(a) HDR image with ghost artefacts

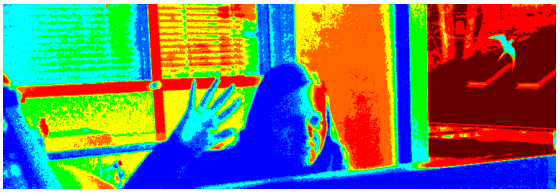
(b) HDR image with ghost removal

Figure 5.3: Final HDR image for Scene 1

The results of the other tested scenes are presented in Appendix B. Based on these results, we can see the quality of the bitmap movement detection depends a lot on the input data. The method produces excellent quality outputs in a scene where a motion appears on a dark background, how it is shown in Figures 5.3 and B.2. However, sometimes some additional artifacts can appear after the de-ghosting method is applied as it is in Figure B.1. These artifacts appear on boundaries of previously detected ghosting objects. It is caused by using only mid-exposed image in ghost regions instead of the proper fusion. This method has also problem with input images which contain too many dark pixels, too many bright pixels or too many pixels with same value. If the input image consists of dark or bright pixels only, the median threshold value is near zero or 255, respectively. This causes the computed median threshold bitmap is whole black and the rest of computations are affected by this fact. The same problem appears if most of the pixels in the input image are under or over the median threshold value as it occurs if most of the image pixels have same value. The results of the fusion where the over-exposed image produces a black bitmap, is presented in Figure B.3. Although a movement is removed, the available dynamic range is reduced.

5.2 Histogram-Based Ghost Detection

This method is based on the median threshold bitmap algorithm as well as the bitmap movement detection. However, this method improves the use of this algorithm. Instead of binary bitmaps, histogram-based ghost detection first calculates multi-level threshold maps for each input exposures. The multi-level threshold map is computed by classifying the intensity value of the input image into 8 levels based on thresholds. These thresholds are found by dividing the input image into 8 levels which gives a set of 8 threshold values, where each level has the same number of pixels. The multi-level threshold maps for the input images of Scene 1 are shown in Figure 5.4.



(a) Under-exposed image

(b) Mid-exposed image

(c) Over-exposed image

Figure 5.4: Multi-level threshold maps for input images of Scene 1

The calculated multi-level threshold maps are used to compute object motion maps. The mid-exposure LDR image is selected as the reference image. Then the object motion map is computed using the absolute difference between multi-level threshold maps of the reference image and some other image. If the difference is bigger than one, the movement is detected and the pixel value in the object motion map is set to one. Then, the morphological operations, erosion and dilation, with kernel size $s = 1$, are applied on the found object motion maps to reduce noise. Histogram-based ghost detection method produces $N - 1$ object motion maps where N is a number of input exposures. Two final ghost maps for Scene 1 are shown in Figure 5.5.

Figure 5.5: Ghost maps for Scene 1 using the mid-exposed image as a reference

Due to several causes, there are some false detections. Pixels having a similar intensity near the threshold values can be classified into different levels due to different exposures. To overcome this drawback, binary error maps are computed as:

$$E_{i,j} = \begin{cases} 1 & \text{for } T_{j,k} - 1 \leq I_{i,j} \leq T_{j,k} + 1, 1 \leq k < N \\ 0 & \text{otherwise} \end{cases}, \tag{5.1}$$

where $I_{i,j}$ is the intensity at ith pixel in the jth exposure and $T_{j,k}$ is the kth threshold value used for extracting the multi-level threshold map of jth exposure.

The error maps with the object motion maps are used to compute a weighting factor ME for the radiance fusion as follows:

$$ME_{i,j} = \begin{cases} 1 & \text{for } j = ref \\ 1 & \text{for } M_{i,j} = 0, j \neq ref \\ 0 & \text{for } M_{i,j} = 1, j \neq ref \\ 0.2 & \text{for } E_{i,j} | E_{i,ref} = 1, j \neq ref \end{cases}, \tag{5.2}$$

where $M_{i,j}$ is the object motion map and $E_{i,j}$ is the binary error map.

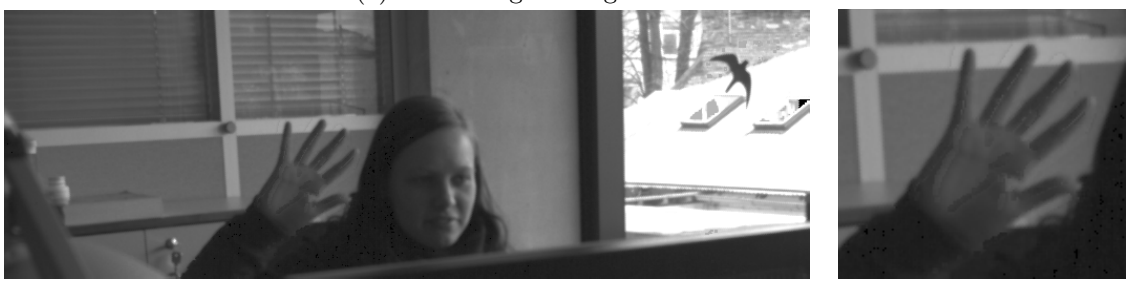
The final HDR image without ghost artefacts is obtained as:

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(ME_{i,j})(\ln Z_{ij} - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})(ME_{i,j})}, \tag{5.3}$$

where E_i is the irradiance, P is the number of input photographs, w is the weighting function from Equation 2.7, Z_{ij} is the input exposure pixel value and Δt_j is the known exposure time. Equation 5.3 does not include a camera response function because there is an assumption the camera response function is linear.

The resulting HDR image of Scene 1 using the histogram-based ghost detection is in Figure 5.6.

(a) HDR image with ghost artefacts



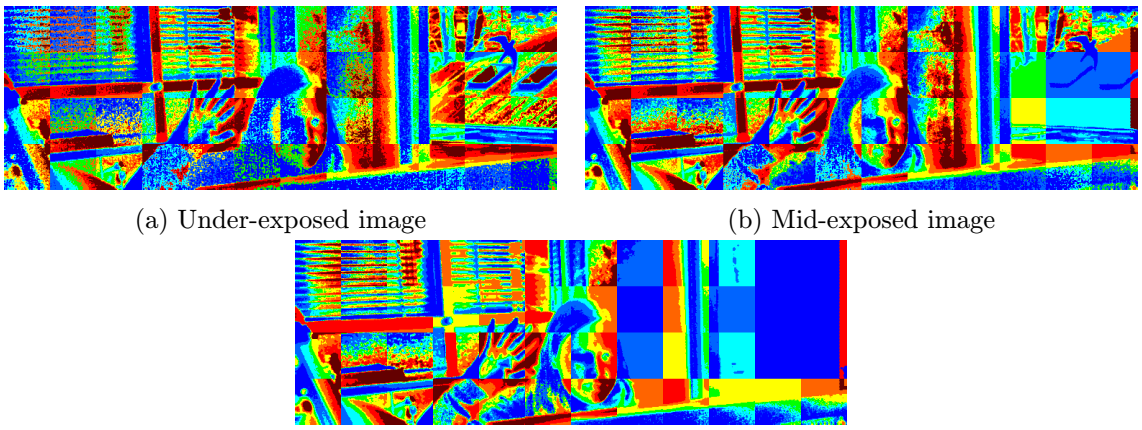
(b) HDR image with ghost removal

Figure 5.6: Final HDR image for Scene 1

The results of the other tested scenes are presented in Appendix C. Based on these results, we can see the histogram-based ghost detection has excellent quality outputs in a scene where a motion appears on a dark background as well as the previous bitmap movement detection. The good quality results of the histogram-based ghost detecting method are shown in Figures 5.6 and C.2. However, the method has weak results in scenes where the movement appears on a bright background. The ghost artifacts are eliminated but not all of them are completely removed in these cases. Examples of this situation are shown in Figures C.1 and C.3. The results of the histogram-based ghost detection are depended on the input exposures as well, but not so much in comparison with the bitmap movement detection.

5.3 Proposed Algorithm

Based on the complexity of implemented algorithms, the proposed algorithm is a modification of histogram-based ghost detection for easier implementation on FPGA architecture. Histogram-based de-ghosting method needs to calculate histograms of all input images as a first step. However, a difficulty of the histogram calculation depends on the input image size. The tested images have resolution 768x256 pixels which is quite big to calculate the histogram for whole image on FPGA architecture. It led to the decision to split input data into smaller blocks. More precisely, the proposed algorithm processes the input images by parts. Therefore, the proposed algorithm needs to calculate local histograms instead of global histograms. Because the number of pixels in the image block is smaller, smaller adders are needed for computation on FPGA architecture. Moreover, if a number of bins in the histogram will be decreased, a number of adders needed for implementation will be reduced as well. In this modification, there is a potential to eliminate more ghosting artifacts because the block processing introduces a local access into the global method. The proposed algorithm tries to transfer the global method into a local one. Size of processing blocks is defined by input parameters. This modification causes less memory consumption. The block size depends on the size of ghost artifacts and affects the quality of results. Figure 5.7 shows the multi-level threshold maps by using proposed modification with size of the processing block 64x64 pixels.



(a) Under-exposed image

(b) Mid-exposed image

(c) Over-exposed image

Figure 5.7: Multi-level threshold maps for input images of Scene 1 by using image processing on a block of 64x64 pixels

Next modification of histogram-based method is an omission of morphological operations. The difference between results with the morphological operations and without them is not visible by human eyes which has been found by testing and comparing both cases. Besides the mentioned modifications, the proposed algorithm is consistent with histogram-based ghost detection. The resultant HDR image by using proposed algorithm with size of the processing block 32x32 pixels is shown in Figure 5.8.

Another modification, an omission of binary error map computation, was tried. However, the results without application of binary error maps were poor in most tested cases.

Figure 5.8: Resultant HDR image by using proposed algorithm with 32x32 size of processing block

Evaluation of Proposed Algorithm

The quality of resulting HDR images by using proposed modifications is comparable with the original histogram-based de-ghosting method. The resultant HDR image for Scene 1 was presented in this section in Figure 5.8. More results for the other tested scenes are presented in Appendix D.

The proposed algorithm has excellent quality outputs in a scene where a motion appears on a dark background as well as the original histogram-based ghost detection. However, this algorithm has the same problems as the previous method as well. Weak resulting HDR images of Scene 2 and 4 are shown in Figures D.1 and D.3.

The resulting output depends a little bit on the block size which is set for image processing. As the tested sizes of blocks were selected 16x16, 32x32, 64x64, 128x128 and 256x256 pixels. The results from these tests are shown in Figure D.4 in Appendix D.

Chapter 6

Conclusion

The main aim of this master's thesis was to propose a modified de-ghosting algorithm based on already existing methods which will be suitable for implementation on FPGA architecture. Moreover, a prototype of the proposed algorithm had to be created by implementing it in C++ programming language. All the points set down in the task specification were accomplished in this thesis. The first point of the thesis was to study the relevant literature from the field of HDR image synthesis. The most important information about HDRI synthesis which were important to complete this thesis, were summarized in Chapter 2 as well as the information about ghost detection and removal in HDR images and existing de-ghosting algorithms that had to be studied to meet the second point of the task specification. The third point of the thesis was to propose a de-ghosting algorithm which will be suitable for implementation on FPGA architecture. To fulfil this point, the basic information about FPGA architecture and programming principles had to be studied. These information were presented in Chapter 3. The rest points of the task specification were processed in Chapters 4 and 5. Chapter 4 includes the architecture and application specification, and summary of existing de-ghosting algorithms and reflection on the suitability for FPGA implementation. The own implementation of selected algorithms and proposed method as well as their testing and evaluation are shown in Chapter 5.

The implementation of modified histogram-based algorithm gives the same results in comparison with the original histogram-based ghost removing method. Moreover, the implementation is faster, more efficient and more suitable for FPGA architecture. However, the original histogram-based ghost removal has worst results in some scene where the movement is on a bright background. The proposed algorithm has the same problem as well. Moreover, the set of testing data were quite small and only one setting of exposure times was used for capture input images.

Because there are already more than fifty existing de-ghosting algorithms and the research on this topic is very current, it was impossible to test each of existing algorithms during this thesis. Therefore, the possible continuation of this thesis could be to study, implement and test more existing de-ghosting algorithms. Based on the experiences acquired during the time of studying de-ghosting problem in HDR images, I can see the potential in some patch-based algorithms. However, it is necessary to find the best one for FPGA implementation which will fit all desired requirements.

Bibliography

[1] Ramratan Ahirwal, Yogesh Singh Rajput, and Dr. Yogendra Kumar Jain. Improved Histogram Based Ghost Removal in Exposure Fusion for High Dynamic Range Images. *International Journal of Computers & Technology*, 12(3):3329–3337, 2014.

[2] Nana Akahane, Shigetoshi Sugawa, Satoru Adachi, Kazuya Mori, Toshiyuki Ishiuchi, and Koichi Mizobuchi. A Sensitivity and Linearity Improvement of a 100 dB Dynamic Range CMOS Image Sensor Using a Lateral Overflow Integration Capacitor. *Solid-State Circuits, IEEE Journal of*, 41(4):851–858, April 2006.

[3] Ahmet Oguz Akyüz. Photographically Guided Alignment for HDR Images. *Proceedings of the Eurographics-Areas Papers*, pages 73–74, 2011.

[4] Jaehyun An, Sang Heon Lee, Jung Gap Kuk, and Nam Ik Cho. A Multi-Exposure Image Fusion Algorithm Without Ghost Effect. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1565–1568, May 2011.

[5] Peter J. Ashenden. *The Designer's Guide to VHDL, Third Edition.* Morgan Kaufmann Publishers Inc., 3 edition, 2008.

[6] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced High Dynamic Range Imaging.* A K Peters, 2011. ISBN 978-1-56881-719-4.

[7] Bryce E. Bayer. Color Imaging Array. `http://www.google.com/patents/US3971065`, 1976. US Patent 3,971,065.

[8] Christian Bloch. *The HDRI Handbook: High Dynamic Range Imaging for Photographers and CG Artists.* Rocky Nook, 2007. ISBN 978-1933952055.

[9] Christophe Bobda. *Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications.* Springer, 2007. ISBN 978-1-4020-6088-5.

[10] Luca Bogoni. Extending Dynamic Range of Monochrome and Color Images through Fusion. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 7–12, 2000.

[11] Yuri Y. Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, November 2001.

[12] Frank M. Candocia. Simultaneous Homographic and Comparametric Alignment of Multiple Exposure-Adjusted Pictures of the Same Scene. *Image Processing, IEEE Transactions on*, 12(12):1485–1494, Dec 2003.

[13] Tassio Castro, Alexandre Chapiro, Marcelo Cicconet, and Luiz Velho. Towards Mobile HDR Video. *Proceedings of the Eurographics-Areas Papers*, pages 75–76, 2011.

[14] Alexandre Chapiro, Marcelo Cicconet, and Luiz Velho. Filter Based Deghosting for Exposure Fusion Video. In *ACM SIGGRAPH 2011 Posters*, SIGGRAPH '11, pages 33–33. ACM, 2011.

[15] Paul E. Debevec and Jitendra Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378. ACM Press/Addison-Wesley Publishing Co., 1997.

[16] Luigi Di Stefano and Andrea Bulgarelli. A Simple and Efficient Connected Components Labeling Algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 322–327, 1999.

[17] Ashley Eden, Matthew T. Uyttendaele, and Richard S. Szeliski. Seamless Image Stitching of Scenes with Large Motions and Exposure Differences. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2498–2505, 2006.

[18] Erik Krause. Camera response curve [online]. `http://wiki.panotools.org/index.php`, 2007-11-23 [cit. 2015-01-14].

[19] Umer Farroq, Zied Marrakchi, and Habib Mehrez. *Tree-based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization.* Springer, 2012. ISBN-13 978-1461435938.

[20] Sira Ferradans, Marcelo Bertalmío, and Edoardo Provenzi. Generation of HDR Images in Non-Static Conditions Based on Gradient Fusion. *VISAPP*, 2012.

[21] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[22] Orazio Gallo, Natasha Gelfand, Wei-Chao Chen, Marius Tico, and Kari Pulli. Artifact-free High Dynamic Range Imaging. In *Proceeding of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–7, April 2009.

[23] Murat Gevrekci and Bahadir K. Gunturk. On Geometric and Photometric Registration of Images. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, April 2007.

[24] Miguel Granados, Kwang In Kim, James Tompkin, and Christian Theobalt. Automatic Noise Modeling for Ghost-free HDR Reconstruction. *ACM Trans. Graph.*, 32(6):201:1–201:10, November 2013.

[25] Miguel Granados, Hans-Peter Seidel, and Hendrik P. A. Lensch. Background Estimation from Non-time Sequence Images. In *Proceedings of Graphics Interface 2008*, GI '08, pages 33–40, 2008.

[26] Thorsten Grosch. Fast and Robust High Dynamic Range Image Generation with Camera and Object Movement. In *Proceedings of Vision, Modeling, and Visualization*, pages 277–284, November 2006.

[27] Michael D. Grossberg and Shree K. Nayar. Determining the Camera Response from Images: What is knowable? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11):1455–1467, Nov 2003.

[28] Kanita Karaduzovic Hadziabdic and Jasminka Hasic Telalovic. Report: State-of-the-Art DE-ghosting Algorithms for High Dynamic Range Imaging. *Southeast Europe Journal of Soft Computing*, 2(2):88–92, September 2013.

[29] Kanita Karaduzovic Hadziabdic, Jasminka Hasic Telalovic, and Rafal Mantiuk. Comparison of Deghosting Algorithms for Multi-exposure High Dynamic Range Imaging. In *Proceedings of the 29th Spring Conference on Computer Graphics*, SCCG '13, pages 021:21–021:28, New York, NY, USA, 2013. ACM.

[30] David Hafner, Oliver Demetz, and Joachim Weickert. Simultaneous HDR and Optic Flow Computation. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2065–2070, Aug 2014.

[31] Chis Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 17–33, 1988.

[32] Yong Seok Heo, Kyoung Mu Lee, Sang Uk Lee, Youngsu Moon, and Joonhyuk Cha. Ghost-free High Dynamic Range Imaging. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part IV*, ACCV'10, pages 486–500. Springer-Verlag, 2010.

[33] Bernd Hoefflinger. *High-dynamic-range (HDR) Vision: Microelectronics, Image Processing, Computer Graphics*. Springer, 2007. ISBN 978-3-540-44433-6.

[34] Imtiaz Hossain and Bahadir K. Gunturk. High Dynamic Range Imaging of Non-Static Scenes. *Proc. SPIE*, 7876, 2011.

[35] Jun Hu, Orazio Gallo, and Kari Pulli. Exposure Stacks of Live Scenes with Hand-Held Cameras. *Computer Vision ECCV*, pages 499–512, 2012.

[36] Jun Hu, Orazio Gallo, Kari Pulli, and Xiaobai Sun. HDR Deghosting: How to Deal with Saturation? *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1163–1170, 2013.

[37] Jaehyun Im, Seungwon Lee, and Joonki Paik. Improved Elastic Registration for Removing Ghost Artifacts in High Dynamic Imaging. *Consumer Electronics, IEEE Transactions on*, 57(2):932–935, May 2011.

[38] IMS CHIPS. HDRC Image Sensors (High-Dynamic-Range CMOS) [online]. `http://www.ims-chips.de/home.php?id=a3b15c1en`, cit. 2015-04-28.

[39] Xilinx Inc. Xilinx: What is an FPGA? [online]. `http://www.xilinx.com/fpga/`. Accessed: 2015-02-06.

[40] Katrien Jacobs, Celine Loscos, and Greg Ward. Automatic High-Dynamic Range Image Generation for Dynamic Scenes. *Computer Graphics and Applications, IEEE*, 28(2):84–93, March 2008.

[41] Takao Jinno and Masahiro Okuda. Multiple Exposure Fusion for High Dynamic Range Image Acquisition. *Image Processing, IEEE Transactions on*, 21(1):358–365, January 2012.

[42] Jiří Pinker and Martin Poupa. *Číslicové systémy a jazyk VHDL*. BEN - technická literatura, 2006. ISBN 80-7300-198-5.

[43] Nima Khademi Kalantari, Eli Shechtman, Connelly Barnes, Soheil Darabi, Dan B. Goldman, and Pradeep Sen. Patch-based High Dynamic Range Video. *ACM Trans. Graph.*, 32(6), November 2013.

[44] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High Dynamic Range Video. *ACM SIGGRAPH 2003 Papers*, 22(3):319–325, 2003.

[45] Wen-Chung Kao, Chien-Chih Hsu, Lien-Yang Chen, Chih-Chung Kao, and Shou-Hung Chen. Integrating Image Fusion and Motion Stabilization for Capturing Still Images in High Dynamic Range Scenes. In *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, pages 1–6, 2006.

[46] Spyros Kavadias, Bart Dierickx, Danny Scheffer, Andre Alaerts, Dirk Uwaerts, and Jan Bogaerts. A Logarithmic Response CMOS Image Sensor with On-Chip Calibration. *Solid-State Circuits, IEEE Journal of*, 35(8):1146–1152, August 2000.

[47] Erum Arif Khan, Ahmet Oguz Akyüz, and Erik Rainhard. Ghost Removal in High Dynamic Range Images. In *Proceeding of the IEEE International Conference on Image Processing*, pages 2005–2008, 2006.

[48] Chul Lee, Yuelong Li, and Vishal Monga. Ghost-Free High Dynamic Range Imaging via Rank Minimization. *Signal Processing Letters, IEEE*, 21(9):1045–1049, September 2014.

[49] Dong-Kyu Lee, Rae-Hong Park, and Soonkeun Chang. Improved Histogram Based Ghost Removal in Exposure Fusion for High Dynamic Range Images. In *Proceeding of the Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium*, pages 586–591, 2011.

[50] Zhengguo Li, S. Rahardja, Zijian Zhu, Shoulie Xie, and Shiqian Wu. Movement Detection for the Synthesis of High Dynamic Range Images. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3133–3136, September 2010.

[51] Huei-Yung Lin and Wei-Zhe Chang. High Dynamic Range Imaging for Stereoscopic Scene Representation. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 4305–4308, Nov 2009.

[52] Stephen Lin, Jinwei Gu, Shuntaro Yamazaki, and Heung-Yeung Shum. Radiometric Calibration from a Single Image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, June 2004.

[53] Yang Liu. The Design of a High Dynamic Range CMOS Image Sensor in 110nm Technology. Master's thesis, Delft University of Technology, August 2012.

[54] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[55] T. Lulé, H. Keller, M. Wagner, and M. Böhm. LARS II ? A High Dynamic Range Image Sensor with a-Si:H Photo Conversion Layer.

[56] Stephen Mangiat and Jerry Gibson. High Dynamic Range Video with Ghost Removal. *Proceedings of SPIE*, 7798, 2010.

[57] Steve Mann. Compositing multiple pictures of the same scene. In *Proceedings of the 46th Annual IS&T Conference*, volume 2, 1993.

[58] Steve Mann, Corey Manders, and James Fung. Painting with Looks: Photographic Images from Video Using Quantimetric Processing. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 117–126. ACM, 2002.

[59] Steve Mann and Rosalind W. Picard. On being 'undigital' with digital cameras: Extending Dynamic Range by Combining Differently Exposed Pictures. Technical Report 323, M.I.T. Media Lab Perceptual Computing Section, 1995. Also appears, IS&T's 48th annual conference, Cambridge, Massachusetts, May 1995.

[60] Nicolas Menzel and Michael Guthe. Freehand HDR photography with motion compensation. *Proceedings of VMV*, pages 127–134, 2007.

[61] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pages 382–390, October 2007.

[62] Tae-Hong Min, Rae-Hong Park, and Soonkeun Chang. Histogram Based Ghost Removal in High Dynamic Range Images. In *Proceeding of the Multimedia and Expo (ICME), 2009 IEEE International Conference*, pages 530–533, 2009.

[63] Tae-Hong Min, Rae-Hong Park, and SoonKeun Chang. Noise Reduction in High Dynamic Range Images. *Signal, Image and Video Processing*, 5(3):315–328, 2011.

[64] Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, 1999.

[65] Young-Su Moon, Yong-Min Tai, Joon Hyuk Cha, and Shi-Hwa Lee. A Simple Ghost-free Exposure Fusion for Embedded HDR Imaging. In *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, pages 9–10, January 2012.

[66] Martin Musil. Raster Image Data Transfers in FPGA. Master's thesis, FIT BUT in Brno, 2012.

[67] Tae-Hyun Oh, Joon-Young Lee, and In So Kweon. High Dynamic Range Imaging by a Rank-1 Constraint. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 790–794, September 2013.

[68] Sung-Chan Park, Hyun-Hwa Oh, Jae-Hyun Kwon, Wonhee Choe, and Seong-Deok Lee. Motion Artifact-free HDR Imaging under Dynamic Environments. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 353–356, September 2011.

[69] Fabrizio Pece and Jan Kautz. Bitmap Movement Detection: HDR for Dynamic Scenes. *JVRB - Journal of Virtual Reality and Broadcasting*, 10(2), 2013.

[70] Matteo Pedone and Janne Heikkilä. Constrain Propagation for Ghost Removal in High Dynamic Range Images. In *Proceeding of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 36–41, 2008.

[71] Ali Ajdari Rad, Laurence Meylan, Patrick Vandewalle, and Sabine Susstrunk. Multidimensional Image Enhancement from a Set of Unregistered and Differently Exposed Images. In *SPIE - Electronic Imaging, Int. Society for Optics and Photonics*, 2007.

[72] Shanmuganathan Raman and Subhasis Chaudhuri. Bottom-up Segmentation for Ghost-free Reconstruction of a Dynamic Scene from Multi-exposure Images. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '10, pages 56–63. ACM, 2010.

[73] Shanmuganathan Raman and Subhasis Chaudhuri. Reconstruction of High Contrast Images for Dynamic Scenes. *The Visual Computer*, 27(12):1099–1114, 2011.

[74] Shanmuganathan Raman, Vishal Kumar, and Subhasis Chaudhuri. Blind De-ghosting for Automatic Multi-exposure Compositing. In *ACM SIGGRAPH ASIA 2009 Posters*, SIGGRAPH ASIA '09, pages 44:1–44:1. ACM, 2009.

[75] Raissel Ramirez Orozco, Ignacio Martin, Celine Loscos, and P.-P. Vasquez. Full High-Dynamic Range Images for Dynamic Scenes. *Proc. SPIE*, 8436, 2012.

[76] Erik Reinhard, Greg Ward, Sumanta Pattanaik, Paul Debevec, Wolfgang Heidrich, and Karol Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2010. ISBN 978-0-12-374914-7.

[77] Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2):219–228, April 2003.

[78] Peter Sand and Seth Teller. Video Matching. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 592–599. ACM, 2004.

[79] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaesoubi, Soheil Darabi, Dan B. Goldman, and Eli Shechtman. Robust Patch-Based HDR Reconstruction of Dynamic Scenes. *ACM Transactions on Graphics*, 31(6):203:1–203:11, November 2012.

[80] Désiré Sidibé, William Peuch, and Olivier Strauss. Ghost Detection and Removal In High Dynamic Range Images. In *Proceedings of the 17th European Signal Processing Conference (EUSIPCO 2009)*, EUSIPCO '09, pages 2240–2244, 2009.

[81] Simon Silk and Jochen Lang. Fast High Dynamic Range Image Deghosting for Arbitrary Scene Motion. In *Proceedings of Graphics Interface 2012*, GI '12, pages 85–92, 2012.

[82] Abhilash Srikantha and Désiré Sidibé. Ghost Detection and Removal for High Dynamic Range Images: Recent Advances. *Signal Processing: Image Communication*, 27(6):650–662, 2012.

[83] Abhilash Srikantha, Désiré Sidibé, and Fabrice Mériaudeau. An SVD-based Approach for Ghost Detection and Removal in High Dynamic Range Images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 380–383, November 2012.

[84] Hwan-Soon Sung, Rae-Hong Park, Dong-Kyu Lee, and Soonkeun Chang. Feature Based Ghost Removal in High Dynamic Range Imaging. *International Journal of Computer Graphics & Animation*, 3(4), October 2013.

[85] Anna Tomaszewska and Radoslaw Mantiuk. Image Registration for Multi-exposure High Dynamic Range Image Acquisition. In *15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007.

[86] Okan Tarhan Tursun, Ahmet Oguz Akyüz, Aykut Erdem, and Erkut Erdem. The State of the Art in HDR Deghosting: A Survey and Evaluation. In *Proceedings of the 36th Annual Conference of the European Association for Computer Graphics*, EUROGRAPHICS'36, 2015.

[87] Chunmeng Wang and Changhe Tu. An Exposure Fusion Approach Without Ghost for Dynamic Scenes. In *Image and Signal Processing (CISP), 2013 6th International Congress on*, volume 2, pages 904–909, December 2013.

[88] Greg Ward. Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures. *Journal of Graphics Tools*, 8:17–33, 2003.

[89] Shiqian Wu, Shoulie Xie, S. Rahardja, and Zhengguo Li. A Robust and Fast Anti-ghosting Algorithm for High Dynamic Range Imaging. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 397–400, September 2010.

[90] Susu Yao. Robust Image Registration for Multiple Exposure High Dynamic Range Image Synthesis. *Proc. SPIE*, 7870, 2011.

[91] Wei Zhang and Wai-Kuen Cham. Gradient-Directed Composition of Multi-Exposure Images. *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 530–536, 2010.

[92] Wei Zhang and Wai-Kuen Cham. Gradient-Directed Multiexposure Composition. *Image Processing, IEEE Transactions on*, 21(4):2318–2323, April 2012.

[93] Wei Zhang and Wai-Kuen Cham. Reference-Guided Exposure Fusion in Dynamic Scenes. *Journal of Visual Communication and Image Representation*, 23(3):467–475, 2012.

[94] Jinghong Zheng, Zhengguo Li, Zijian Zhu, and Susanto Rahardja. A Hybrid Patching Scheme for High Dynamic Range Imaging. In *Asia Pacific Signal and Information Processing Association Annual Summit and Conf.*, 2011.

[95] Jinghong Zheng, Zhengguo Li, Zijian Zhu, Shiqian Wu, and Susanto Rahardja. Patching of Moving Objects for Ghosting-free HDR Synthesis. In *ACM SIGGRAPH 2012 Posters*, SIGGRAPH '12, pages 62:1–62:1. ACM, 2012.

[96] Jinghong Zheng, Zhengguo Li, Zijian Zhu, Shiqian Wu, and Susanto Rahardja. Hybrid Patching for a Sequence of Differently Exposed Images With Moving Objects. *Image Processing, IEEE Transactions on*, 22(12):5190–5201, Dec 2013.

[97] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Freehand HDR Imaging of Moving Scenes with Simultaneous Resolution Enhancement. In *Proceedings of Eurographics Computer Graphics Forum*, pages 405–414, 2011.

[98] Mark Zwolinski. *Digital System Design with VHDL (2nd Edition)*. Prentice-Hall, Inc., 2003.

[99] Lukáš Čermák and Václav Hlaváč. Exposure Time Estimation for High Dynamic Range Imaging with Hand Held Camera. *Proc. of Computer Vision Winter Workshop*, 2006.

Appendix A

Input Images

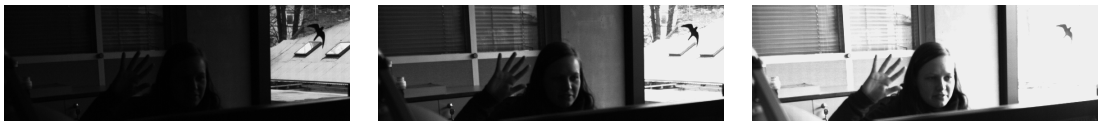


Figure A.1: Input LDR exposures of Scene 1 from under-exposed to over-exposed

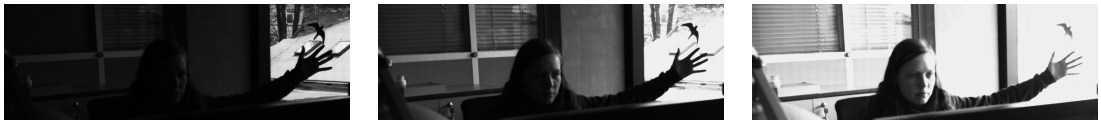


Figure A.2: Input LDR exposures of Scene 2

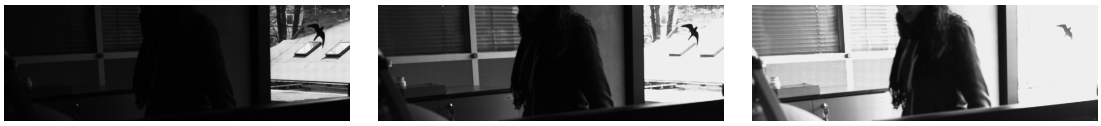


Figure A.3: Input LDR exposures of Scene 3

Figure A.4: Input LDR exposures of Scene 4

Appendix B

Results of Bitmap Movement Detection

(a) HDR image with ghost artefacts

(b) HDR image with ghost removal

Figure B.1: Final HDR image for Scene 2

(a) HDR image with ghost artefacts

(b) HDR image with ghost removal

Figure B.2: Final HDR image for Scene 3

(a) HDR image with ghost artefacts

(b) HDR image with ghost removal

Figure B.3: Final HDR image for Scene 4

Appendix C

Results of Histogram-Based Ghost Detection

(a) HDR image with ghost artefacts

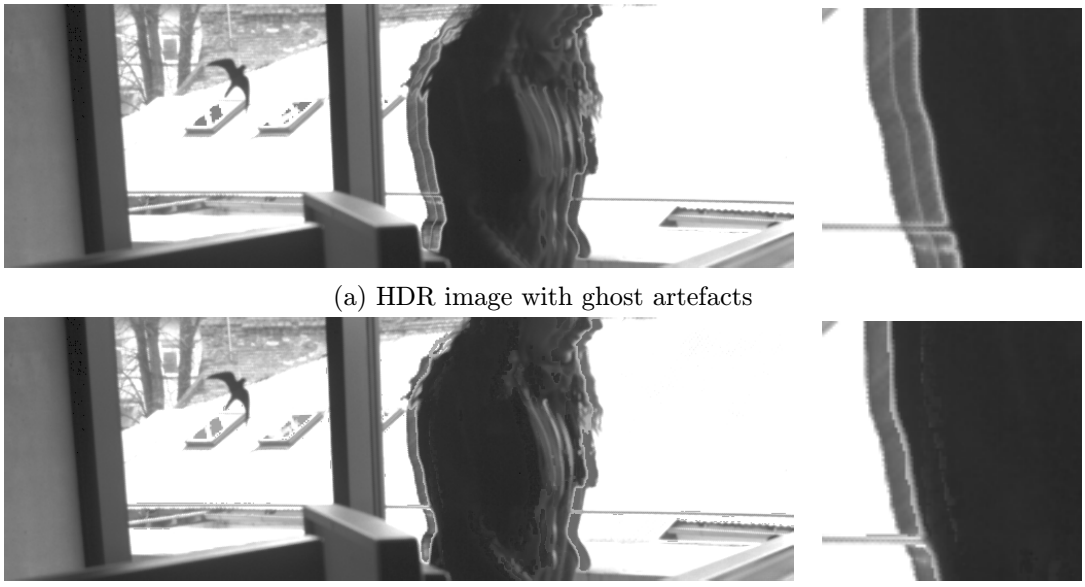
(b) HDR image with de-ghosting method

Figure C.1: Resulting HDR image for Scene 2 using histogram-based ghost detection

(a) HDR image with ghost artefacts

(b) HDR image with de-ghosting method

Figure C.2: Resulting HDR image for Scene 3 using histogram-based ghost detection



(a) HDR image with ghost artefacts

(b) HDR image with de-ghosting method

Figure C.3: Resulting HDR image for Scene 4 using histogram-based ghost detection

Appendix D

Results of Proposed Algorithm

Figure D.1: HDR image for Scene 2 using proposed algorithm with 32x32 size of block

Figure D.2: HDR image for Scene 3 using proposed algorithm with 32x32 size of block

Figure D.3: HDR image for Scene 4 using proposed algorithm with 32x32 size of block

(a) HDR image by using 16x16 block processing

(b) HDR image by using 32x32 block processing

(c) HDR image by using 64x64 block processing

(d) HDR image by using 128x128 block processing

(e) HDR image by using 256x256 block processing

Figure D.4: Resulting HDR images for Scene 1 by using modification of histogram-based ghost detection