

BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

MAGIC LANTERN VIDEO DECODER FOR CANON 5D CAMERA

MASTER'S THESIS

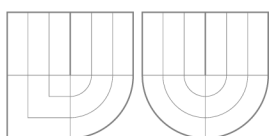
DIPLOMOVÁ PRÁCE

AUTHOR

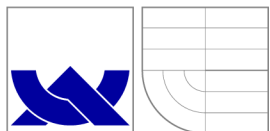
AUTOR PRÁCE

Bc. RADKA ŠKVAŘILOVÁ

BRNO 2015



BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF INFORMATION TECHNOLOGY
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA



MAGIC LANTERN VIDEO DECODER FOR CANON 5D CAMERA

„MAGIC LANTERN“ VIDEODEKODÉR PRO FOTOAPARÁT CANON 5D

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. RADKA ŠKVAŘILOVÁ

SUPERVISOR

VEDOUČÍ PRÁCE

Prof. Dr. Ing. PAVEL ZEMČÍK

CONSULTANT

ODBORNÝ KONZULTANT

Prof. ALAN CHALMERS

BRNO 2015

Abstract

This thesis introduces design for creating decoder for video captured by Magic Lantern software, which can be installed on Canon 5D camera. This video is special for its size of 14-bit and that's the reason of excellent quality of output. The thesis is about cutting video into single frames. The frames should be in a format, which can manage the high dynamic range images.

Abstrakt

Tato práce představuje návrh na vytvoření dekodéru pro video zaznamenané pomocí softwaru Magic Lantern, který může být nainstalován na Canon 5D. Toto video je význačné pro svoji velikost 14-bitů v raw formátu a proto může produkovat velmi kvalitní výstup. Práce má za cíl rozdělit video do jednotlivých snímků, ve vhodném formátu, který umí pracovat také s formáty obrazů s vysokým dynamickým rozsahem.

Keywords

Magic Lantern, Canon 5D, High Dynamic Range, video decoder.

Klíčová slova

Magic Lantern, Canon 5D, vysoký dynamický rozsah, video dekodér.

Citation

Radka Škvařilová: Magic Lantern Video Decoder for Canon 5D Camera, master's thesis, Brno, FIT VUT v Brně, 2015

Magic Lantern Video Decoder for Canon 5D Camera

Statement

Prohlašuji, že jsem tuto práci vypracovala samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka a odborného konzultanta pana prof. Alana Chalmerse

.....
Radka Škvařilová
July 31, 2015

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Dr. Ing Pavel Zemcik for expert guidance and useful comments. Furthermore, I would like to thank Prof. Alan Chalmers, Jonathan Hatchett and Josh McNamee for expert advice and consultations.

© Radka Škvařilová, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Contents

1	Introduction	2
2	State of the Art	3
2.1	Basics from graphics	3
2.2	Photo processing	11
2.3	High Dynamic Range Imaging	15
2.4	Software source	27
3	Analysis and Specification	28
3.1	Actual state	28
3.2	Design	28
4	Decoder design	30
4.1	Mlv data	30
4.2	Software needs	30
4.3	DNG file	31
4.4	EXR file	31
4.5	Debayering	31
4.6	Color correction	35
4.7	HDR	35
4.8	Future work	35
5	Conclusion	37
6	CD	38

Chapter 1

Introduction

This thesis is about processing video in a very good quality format, which is captured by the camera „Canon 5D Mark III“ using additionally installed software - „Magic Lantern“. The output of this capturing is an unprocessed video file, so it can not be just played in any routine video player. The main idea of this thesis is finding a solution for conversion between this video file format and single video frames, which can be later taken together into well-known video file format that ordinary player can play.

Finding a solution can be the first step for the decoder, which can convert high dynamic range video in one step, from camera file format to playable video file format. Also, the new cameras have a more amazing function, and if the main part of decoding will be done, it just need to focus on the technology news.

Magic Lantern project for cameras were first time introduced in 2009, since them it comes a long way, and it handles processing advantage of new technology on various kinds of cameras. Canon 5D Mark III camera has the advantages of getting good quality images in a low light scene and shooting images with high speed. This combination is useful for high dynamic range images capturing by multiple exposures. And of course, it has other option that makes this camera very popular for users.

The content of this work is divided into three main chapters.

The first chapter is about the introduction of concepts related to the topic. In the beginning, the basic notions from graphics such as color, the explanation of the human visual system and then an explanation of camera technology system are described. Afterward is explained the term of color space and other terms relating to colors. Furthermore, the first chapter contains information about raw images and how to process them. Another important passage is the explanation of the basic information about high dynamic range; this part is mainly focused on multiple exposures. Then there are described file formats like mlv, DNG, OpenEXR and the used software. The second chapter contains analysis and the more detailed work assignment. In the third chapter is specified the work on the project. There is described process of decoding and shows the images obtained during the decoder processing. And introduce the options of future work.

Chapter 2

State of the Art

2.1 Basics from graphics

In this section are introduced basic notion for better understanding in other sections. The section starts with explanations of colors, how the human visual system process colors, then is presented the camera from general view and the Canon 5D Mark III. Also in this section is described how are processed the raw images, and what is the high dynamic range. Then are introduced file formats and software .used in this thesis.

Colors

Study of the color itself touches several different fields like physics (for the propagation of light through space), chemistry (for its interaction with matter), neuroscience and psychology (for aspects relating to perception and cognitions of color), etc. But in computer science is usually studied how is light propagates through space. Light in the form of photons travel through space and in contact with any material it can interact by absorbing, reflecting or transmitting. In reflection, we can distinguish two types of material - specular material (ray reflects in a particular direction) and diffuse material (ray reflects in a random direction). For color, interpretation is important how much light reach the human eye.

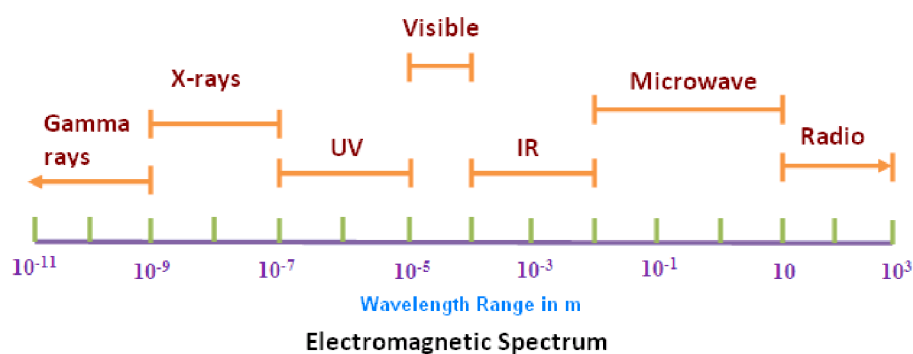


Figure 2.1: Wavelength spectrum

In computer graphics, light is traditionally not simulated by the photons, but in the spectrum. Spectrum is very large number of photons where each carrying a possibly different amount of energy - or equivalently say with a different wavelength (λ). Wavelength

spectrum 2.1 is much bigger than we can see, there are γ rays, X-rays, ultra-violet, infra radiant, microwave and other radio waves.

But for the color itself is important range between about 380 and 800 nanometers known as visible spectrum 2.2. In theory, the spectrum is sufficient model but from the mathematical sights the spectral rendering is computationally expensive. Therefore, the spectrum is decomposition in individual units. For different decompositions, there are define different color spaces and their operations. From [12] and [9].

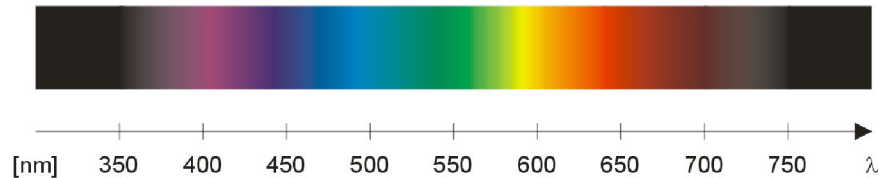


Figure 2.2: Visible spectrum

Human color vision

A human use eye for gathering light onto photoreceptors, which convert the light into the signals and the brain then process these signals in visual cortex area into a colorful visual scene.

At first light enters the eye through the cornea, which is the transparent membrane. Then it goes through the pupil, which is modified by the iris, a muscular diaphragm. After that is light refracted by the lens and hits the photoreceptor in the retina. In the eye, there are two types of photoreceptor - cones and rods.

Cones numbering around 6 million are more sensitive on daylight vision (luminance between $10^{-2}cd/m^2$ and $10^8cd/m^2$). They are responsible for the perception of the high-frequency pattern, fast motion, and colors. The rods, about 90million units, are sensitive at luminance levels between $10^{-6}cd/m^2$ and $10cd/m^2$, night vision. The rods are more sensitive than cons, but they do not provide color vision.

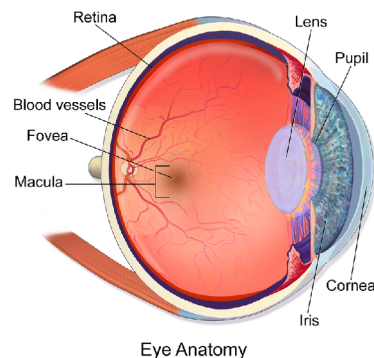


Figure 2.3: Descriptions of human eye

The main reason the human vision process was introduced is that from the technical point of view the technicians tries to design device similar to the human vision system. As

was already written human range of the visible light spectrum is between 380-800 nanometers, but it is still more than the typical device can capture or visualize. Therefore is so interesting developing better devices and software that can show more realistic view.

Lately, there is the High Dynamic Range approach (defined in chapter Hight Dynamic Range Imaging) evolving new software and hardware that can manage more light in one scene. The output looks more realistic because it can show more details in one scene. Source [9].

Camera

Technicians efforts to produce a machine with as similar quality as possible to the human visual system. The design of the camera is similar to the human eye for example in the lens, which is not randomly called the same. Then the similarity is in the aperture and pupil because they limit the amount of light entering the „device“. And as well as is in eye the scene projects on the retina in camera is this part called sensor.

In the world is many camera models, cheaper with less quality and on the other hand, the expensive with better quality. DSLR camera types are the one with the top image quality, and because the Canon 5D is one of them, so they will be introduced more.

DSLR

Digital Single-Lens Reflex camera is similar to classical Single-Lens camera besides, it uses a digital sensor. In image 2.4 is shown that the light goes through the lens (1).Then it bounce from the movable mirror (2) then going through the matte focusing screen(5) via the condenser lens (6) into the area of pentamirror (7) where the image is transfer into right position and an into viewfinder eyepiece (8). In exposition, the mirror (2) is downcast, and the light reach the sensor (4) for the time when the shutter (3) is open.

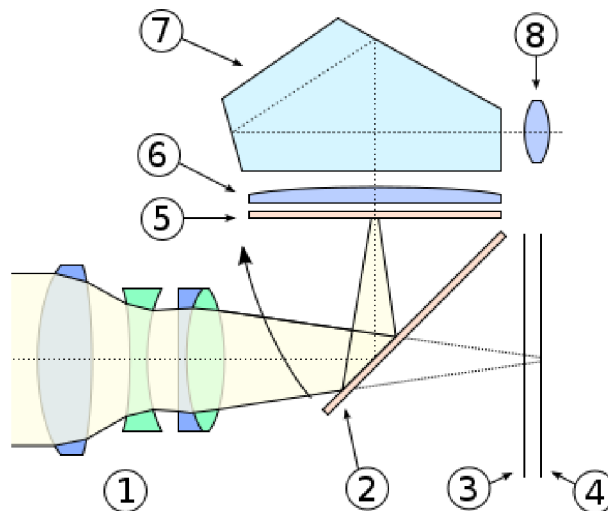


Figure 2.4: Cut trough SLR Camera system.

Lens

Lens are the one of the most important parts of the camera. If the camera were without the lens, the sensor should have a problem with capturing all illuminates. Because the light

beams coming from the object would be overlaps. Every point from the scene would have its image on every point on the sensor. Lens can focus the light from one point in the scene into one point in the sensor as in image 2.5. It depends on physics quality of lens how precisely it can work. The other issue is that light beams coming from the different distances focus in different distances behind the lens, so the lens have to move to set the right focus of the image on the sensor. The range of depths providing the sufficient focus is called the depth of field.



Figure 2.5: Lens in camera focus light from one point in scene into one point in sensor [4]

Aperture and Shutter

The aperture is typically part right behind the lens. This part reduces the amount of light that reach the sensor. It can help with reducing blurriness from wrong focus setting parts of an image. Practically setting aperture means changing the size of the hole that light is coming trough. The smaller the hole is, the more sharp image is. If the hole is wide opened, it lead to the result that only part of an image (with the right focal length) is sharp.

The shutter is the device that allow the access of light beams hit the sensor for the precisely defined time - exposure time. This device is placed right in front of the sensor. The slower shutter is, the more blur is in the moving and light objects.

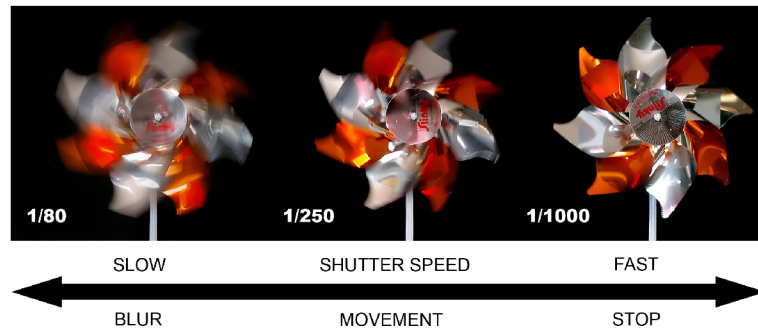


Figure 2.6: Shutter speed [6]

Sensor

The image sensor is part of the camera that transfer the incoming light into electrical signals. Size in professional camera is about 36x24 mm, for example in mobile phones is about 2.3x1.7 mm size sensor. The larger sensor can collect more light and produce less noise, but ass the quality rise the price rise too.

The most used technology in sensors are CMOS(Complementary Metal-Oxide-Semiconductors) or CCD(Charge-Coupled Devices). The sensor typically can capture in one cell(pixel) one

color (special wavelength of light). And then it is needed to apply some algorithm (demosaicing) that reconstruct an image in a full-color image. Usually, the sensor has red, green and blue cells in the grid.

ISO speed

The camera set ISO for control how sensitive is the sensor on light reaching it. This setting may affect the image quality. The higher speed makes the signal stronger but causes more noise, like the lots of tiny colored dots, which are especially visible in the dark areas.

Canon 5D Mark III

The first Canon EOS (Electro-Optical System) 5D camera was introduced in 2005 as 12.8 megapixel DSLR camera. The Canon 5D Mark III was introduced in March 2012 with 22.3 megapixel CMOS image sensor. This camera has many advanced functions, so it is determined for experienced users. Also, the weight about 860g without lens shows that this camera is not the type, the ordinary user takes everywhere.

From official pages of Canon, is Canon 5D Mark III is ideal for shooting weddings in the studio or out in the field, and great for still photography. Nevertheless, the most interesting for this thesis is the ability to shoot professional high definition video. Including many of industry-standard recording protocols and enhanced performance. The function shooting performance up to 6.0 fps with exceptional clarity and sharpness, the technology of shooting HDR images and Multiple Exposure mode makes the camera amazing multimedia tool.



Figure 2.7: Canon EOS 5D Mark III

Color spaces

For right representing of colors is needed to define a mathematical description, called color space. Color space is typically represented by tuples of components called primary colors. There are two types of color spaces - device dependent and independent.

Most color spaces are device dependent. Good example of device dependent color space is CMYK used in printers, where each color Cyan, Magenta, Yellow, and Key is part of printer device and other colors are defined by the combination of all these colors. So the results color depends on primary colors. Different printers can print the same image in little bit different colors. As an example of color independent space is known XYZ, which has a mathematical base.

In color spaces, there have to be defined a white point, which is the value interpret as the whitest point in the color space. Although this value do not have to be constant. White color association in the human visible system depends on the color illuminant. CIE defines most used standard D65 as daylight illuminant. Sometimes the black point is specified too.

Color spaces are based on several schemes. The well-known is an additive scheme. This color space work with three kinds of light - called primaries. Each light has specified own chromaticity. This color space gets other colors by mixing this three types of light together.

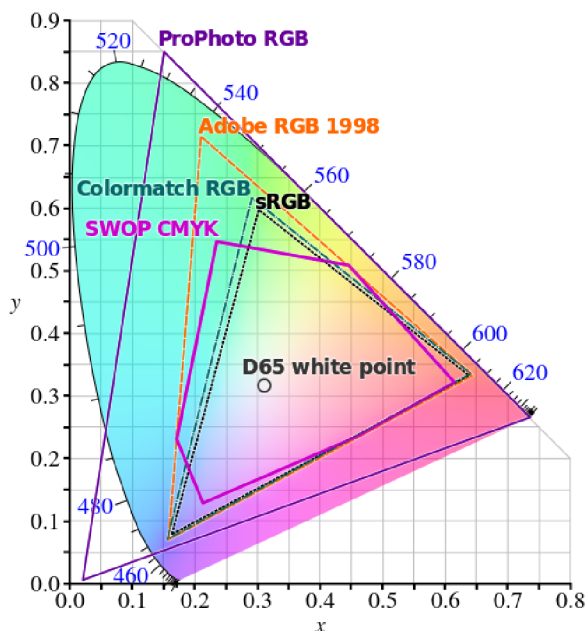


Figure 2.8: Color spaces cover area

RGB

One of the basic and well-known additive color space is RGB. It can be represented by Cartesian cube where the axis is Red, Green, and Blue. Generally in RGB spaces is besides, RGB color also defined the white point, which set minimal boundary from which is the color represent as white color. The primary color are specified in term of CIE 1931 color space chromaticity coordinates (x,y). It is known that human visual system can capture more color than RGB color space covers, so there were define more different RGB color spaces where each of them covers the different area of colors range - the gamut. This type of color space is device dependent. There is more specification of RGB color space, then is CIE - for example sRGB, Adobe RGB, etc.

XYZ

XYZ is one of the first color space dependent on mathematical model. It was created by CIE in 1931 as an alternative to RGB model. Color space XYZ is the most commonly used in technical colorimetric works. This model should simulate human vision with three types of stimuli (as three types of cones in the human eye). In this space, Y represent luminance and XZ contain all chromaticities at that luminance. The XYZ color space can be displayed in chromaticity diagram by evaluation of:

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z}$$

2.1.1 Other terms of graphics

This part contain the terms related to colors as its attribute.

Brightness & Colorfulness

The brightness is the amount of light coming from the stimulus. And the Colorfulness is the amount of hue in stimuli. More luminance in a scene is increasing the colorfulness of scene.

Chromaticity

The Chromaticity of color in its specification can be described as a combination of hue and saturation. In graphics, the chromaticity is associated with tristimulus values as chromaticity coordinates, which can be plotted as chromaticity diagram (shown in image 2.8). The chroma may be described like the colorfulness of stimuli according to the stimuli brightness that appear white under similar viewing condition.

Hue

The hue is one of the parameters of color which is more known as „names of the colors“ - like red, yellow, etc. But it also defines mixing of colors. The hue forms the full spectrum of colors, and sometimes it is divided into groups of primary, secondary and tertiary colors, or it is visually interpreted as a color wheel. The hue is a degree to which a stimulus can be described as similar to or different from stimuli that are described as red, green, blue and yellow.

Saturation

The color saturation is relative to own brightness. Sometimes is can be described as color „purity“. For example, the laser light is the pure color light because it has just one wavelength of light with high intensity.

Radiance & Irradiance

The Radiance is the amount of energy floating per unit of time from or to the one point in a specific direction. It is measured in $W * sr^{-1} * m^{-2}$. Whereas the Irradiance measures the amount of energy floating per unit of time from all direction at the point, is measured in $W * m^{-2}$

Luminance

The luminance is the weighted value of how much luminous power is arriving/leaving at a point in a particular angle of view (weighted Radiance). It is measured in $cd * m^{-2}$.

Color Temeperature

The color temperature or Chromaticity temperature characterize the spectrum of white light associated with blackbody radiator color. The human visual system can customize the color temperature on condition of light. For example, the clean white object under the different lighting may look colored by light. But even if the objects look different, a human

visual system is connected with the brain, so the rational person realizes what is the color of the object, even if its color by some light. In modern cameras is usually the function of automatic white balancing, but if the user want to capture image with specific lighting conditions, he can customize the color temperature setting. For example, the user may this way prevent the orange color in output image if he shooting in bulbs light conditions. Or for better result user may use some special filters.

Color Temperature is measured in K. And cameras have usually predefined the values of color temperature for different conditions, for the daylight it is value about 5000K or 6500K. The lower temperature, the redder is the scene. And the higher value of color temperature may cause a bluish appearance.

2.2 Photo processing

In this section is introduced the raw image file format, which is typically used by camera producers. Related to this, there is described debayering process and white correction.

raw image format

raw image format is one of the camera's (or another capturing device) image format where the data from the sensor are just minimally processed. The name „raw“ comes from idea that these image data have to be processed before we can use them as images as we know. Sometimes raw is called digital negative - it is similar to camera sensor negative, but there are no negative colors, it just mean that there are saved color direct from a sensor.

Usually, accurate format definitions depend on device maker because different devices have different hardware sensor structure, so the output is different. Even one device producer can have on each device different raw format. Sometimes is raw cyphered so as unauthorized programs can not read them.

Not every camera device has a raw output; they just save a photo in jpeg or other format, where the colors are processed. But using these not raw formats usually mean a loss of some information, because of data compression. Whereas raw format has all sensor information and saves each pixel color in defined bit size, it can have bigger color range than typical 8-bit image format although it means that raw file is typically bigger, and we can save fewer images on our device.

Choosing raw file format give us better quality for the price of bigger memory requirements and need of later post-processing. For right interpretation of color in raw image is important to do demosaicing, apply white balancing. Do color correction for final format and device. Eventually, we can apply some customer adjustment.

Structure

In raw file are usually saved some format metadata like byte ordering, identifier, etc., then there are the pixel data and other camera metadata.

Pixel data are stored in a form of device sensor data. Ordinarily camera sensor has a structure of Bayer pattern where each pixel has only one color. Within a stored values can be used some compressing algorithm, but data are not stored with any white or color balancing, or no gamma, etc. Data are typically in the size of 12 or 14 bits per pixel.

Metadata carries the circumstances in which was an image captured. It is needed to save metadata because without them is almost impossible to interpret the pixel data in the right way. Few example of metadata:

collor filter array pattern describes hardware composition of a sensor.

size of sensor describes how many pixel can be saved in raw file

data area sometimes is set that image data are smaller than whole sensor's size

color profile characterize how interpret colors

camera parametrs save values of lens model, shutter, ISO, aperture etc.

customer parametrs author, date etc

video parameters if the video is saved as raw frames there are values like frame per second, frame ordering, etc

As was already written each producer can save different metadata in raw file.

Demosaicing

Most of camera sensors have mosaic structure, although there exist other types of sensor, for this thesis is sufficient just introduce RGB mosaic structure on square grid, which is called „Bayer matrix“ after Dr. Bryce Bayer.

Bayer filter

In monochrome cameras is over each pixel in the mosaic pattern placed a single color filter [2.9](#). There is exactly no color information, but each pixel records how many photons struck the sensor in the pixel. Easy way how to get a color image from the sensor is combined red, green and blue images (in case RGB), but we need these images from the same time. So in RGB model, each pixel has only one of these colors. Order of these colors depends on camera producer. However, it is defined as a color filter array(CFA)[2.10](#). In most RGB CFA is used grid where on 50% of pixels is green color and 25% red and 25% blue.

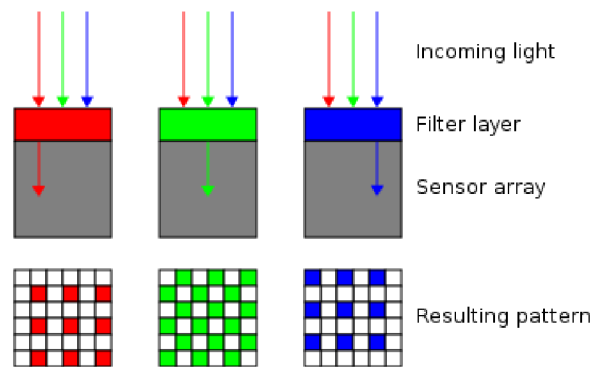


Figure 2.9: Camera sensor filter

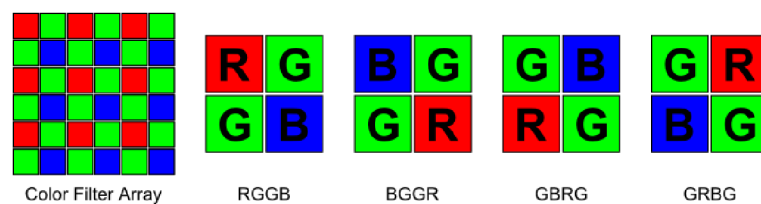


Figure 2.10: Bayer filter array

Debayering process

Debayering or generally called demosaicing is a process, by which is image decoded from mosaic pattern or color matrix into a full-resolution color image. In practice, this means that we have to reconstruct all red, green and blue colors for each pixel. There are many

algorithms for reconstructions of colors; they can be very simple or on the other hand very complex. But the difference in quality can easily recognize an even inexperienced person. And it must be taken into account that there is a trade-off between speed and accuracy.

How was written, each pixel has before debayering only one color and we need to reconstruct for each pixel all colors. This process usually means to use algorithm, which calculate the missing color from the color in neighborhood of this pixel.

Example: Simple interpolation

For better imagination how debayering work it is introduced in this thesis one of the easiest method, called superpixel method. It just takes four CFA source pixels in 2x2 matrix and uses them for obtained values for one destination pixel. Algorithm takes red and blue values and simply copy them into destination pixel, and because we have two green pixels in the matrix, algorithm use average of these green values and save result value in destination pixel.

The new debayered image has at the end one-quarter size of the original CFA image. There are half width and half height of original image size. But a method is very fast and has visually no artifacts, so it is really good method when we have sufficient sensor resolution, and we need result very fast.

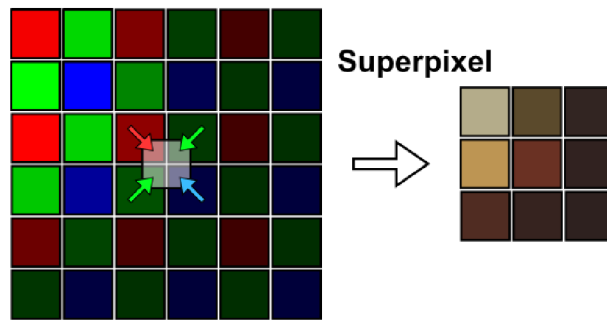


Figure 2.11: Example of debayering by superpixel method

White balance

The white balancing is the part of processing color, where the values from metadata are applied to pixel values for the right representation of colors in the defined color space. This use the knowledge of the white point (or the black point). Figuratively it means that the lightning in the scene may create different color that the user want, so user may use some gain on the red, green, and blue part of the sensor to changing these lightning conditions. Also, if the white balance chosen in the camera shooting raw image was set incorrect, it might change in other process steps as user wish. The process of white balancing is using the value of illuminant, or this value may be measured as „color correlated temperature“ or is measured in term of chromaticity coordinates. The „correlated color temperature“ is the applicable on blackbody radiators that are within $0.5u'/v'$ units of the Planckian locus (also called black body locus, which is the path that the color of black body would take in a particular chromaticity space as the blackbody temperature change). The list is of common scenes is shown in table 2.1, from [?]. The most used illuminant in color spaces is the CIE standard illuminant D65, known as natural or average daylight. And in photography is often used D55 illuminant.

Scene	Kelvin	x	y
Candle flame	1850	0.543	0.410
Sunrise/sunset	2000	0.527	0.413
Tungsten (TV/film)	3200	0.427	0.398
Summer sunlight at noon	5400	0.326	0.343
CIE A (incandescent)	2854	0.448	0.408
CIE B (direct sunlight)	4874	0.384	0.352
CIE C (indirect sunlight)	6774	0.310	0.316
CIE D50 (noon skylight)	5000	0.346	0.359
CIE D65 (average daylight)	6504	0.313	0.329
CIE E (normalized reference)	5500	0.333	0.333
CIE F2 (office fluorescent)	4150	0.372	0.375

Table 2.1: Correlated color temperature and chromaticity coordinates (x y)

In case of changing color appearance is defined chromatic adaptation transform matrix. In human visual system is the chromatic adaptation linked with three types of the retina as the von Kries defined already in the year 1902 [11]. But in photography may this chromatic adaptation be used directly on conversion to different color space or just left the values in the same color space. For example, chromatically adapting directly in the XYZ tristimulus value is transformed by predefined matrix M.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The difference between chromaticity appearance shows the image 2.12.

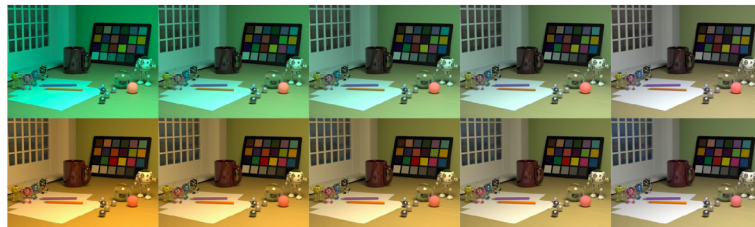


Figure 2.12: The different levels of chromatic adaptation under two different illuminants

2.3 High Dynamic Range Imaging

[9]. High dynamic range (HDR) imaging is quite a new developing field of imaging that does have the ability to use and manipulate physically-real light values. Nevertheless, the first idea was found about the 1850s and has been developing all time since that date. But in few last years the hardware technology made a big progress in HDR field. HDR displays and cameras were designed, and it lead to an expansion of developing in HDR software area.

High dynamic range is a technological approach, which can capture more range of luminance between the brightest and darkest point in the image than usual cameras do. HDR technology is trying to simulate human eye color and light range, but there is still hardware and software deficit, which is tried to eliminate by new technology.

On the other hand, Low dynamic range (LDR) image means that the ratio between of darkest and brightest part of the scene is smaller. For example, dynamic range of CRT monitor is 100:1 or JPEG don't exceed 255:1, while in HDR the ratio may be up to 100000:1(as human eye do).

Capture HDR content usually need special hardware or software. And the size of the files is by its nature larger than LDR files. These files just have to save more information for each pixel. For better managing of these big files, there are special types of files or special compression methods. After storing data, they can be processed somehow with other software that may enhance the total impression. Then this image can be presented on HDR monitor or LDR Monitor. In case of LDR monitor, it has to be used tone-mapping, which convert values of the image for a range of LDR device. All this process is shown in image 2.13.

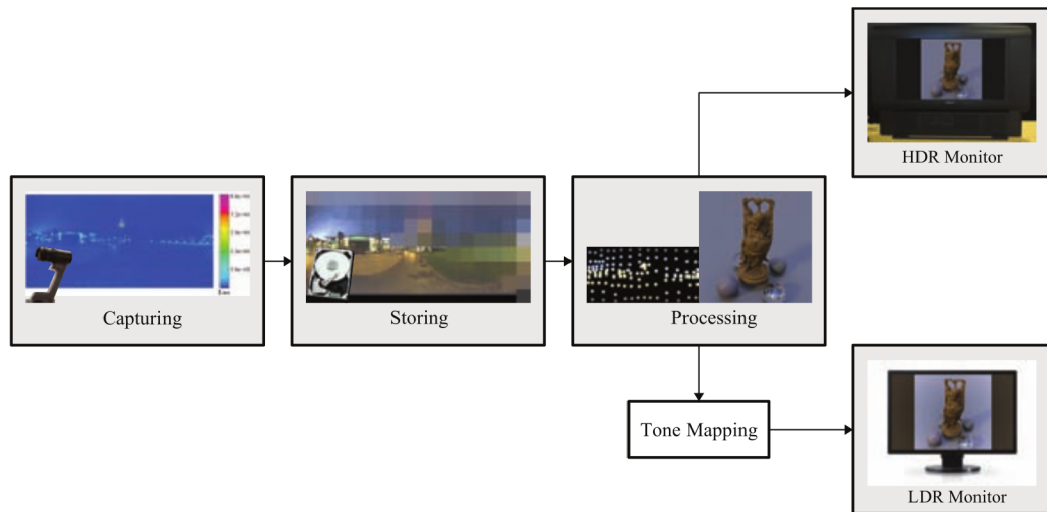


Figure 2.13: HDR pipeline from [9]

Capturing HDR Content

There are more capturing methods that generate HDR content. Method using special hardware or method using the combination of LDR images with different exposures, then there is the creation of HDR directly from virtual software environments and the method using software technology to expanding the dynamic range of LDR captures etc.

Today still the HDR cameras are not so widespread because still their price is too expensive (more than \$35,000). The reason of big price is that cameras have to have so quick or exact mechanism for capturing.

For the ordinary users, there exist other methods, which work with the output of the ordinary device if a form of 8 to 14b its. These captures do not cover all range of irradiance in the scene. The solution to this problem is to take multiple single-exposure images of the same scene. Each image capture different detail in the dark and bright area. HDR image creates from multiple exposures assume, that these single images are perfectly aligned, not moving objects. It can cause problems like ghosting, noise or problem with classic alignment. In this area, there are new algorithms developed for minimizing this problem.

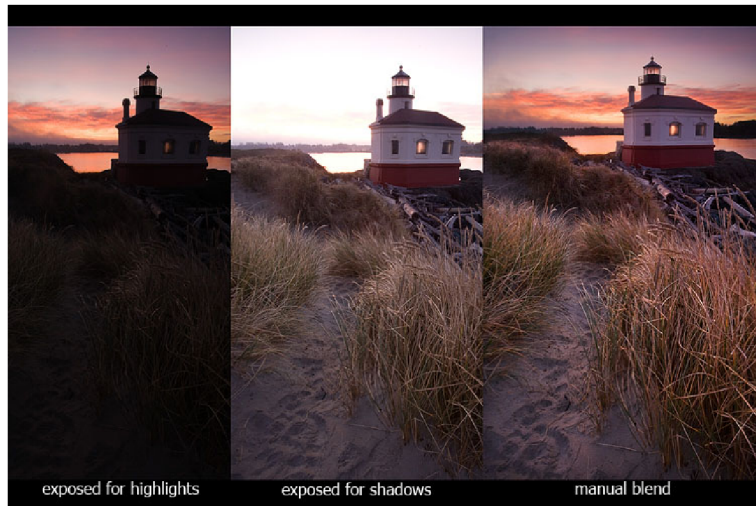


Figure 2.14: Multiple exposure LDR images to HDR

Example: how to merge LDR images into HDR

In the multiple exposure method, the pixels in each image properly exposed, underexposed or overexposed. At least in one or more images is each pixel properly exposed so the merging algorithm may ignore very bright or very dark pixels. Assume that the camera exposure time is perfectly linear. Then can be by applying the equation, which each pixel in the image divide by its exposure time so that pixels are in the same domain for all images. If the images have the same domain, their values for each pixel can be averaged, except the underexposed or overexposed pixel. The result of this process is HDR image. In equations:

$$L_{ij} = \sum_{k=1}^n \frac{Z_{ij} w(Z_{ij})}{\Delta t_k} / \sum_{k=1}^n w(Z_{ij})$$

where $Z_{ij} = L_e(i, j) = E_e(i, j) \Delta t_k$ is the recount value of raddiance (L_e) or irradiance (E_e) for pixel, $w(Z(i, j))$ is weight factor, i, j set location in image and Δt_k is exposure time for

exposure k . But in practice the camera usually haven't the linear exposure time device. So it is needed to account response time of the camera $f()$. The new equation is:

$$L_{ij} = \sum_{k=1}^n \frac{f^{-1}(Z_{ij})w(Z_{ij})}{\Delta t_k} / \sum_{k=1}^n w(Z_{ij})$$

where $Z_{ij} = f(L_e(i, j)) = f(E_e(i, j)\Delta t_k)$.

The hardware for direct capturing HDR content with multiple exposures is now part of some new kinds of camera. The main idea is to control the amount of light reaching the sensor. This can be control by setting the aperture, exposure time, and the ISO value. Change of hardware parts as filters during capturing is not exactly the fastest option, and for capturing the same image multiple time is needed to capture them as fast as it can. So the best way is to changing exposure time and thereby control the amount of light. The new cameras with HDR function have this as a feature, because it take some time to the user to change setting for exposure time. Each camera may have the different type of a maximal number of different exposures. From [?]

HDR videos captured by the method of multiple exposures at this time is one of the main developing fields because in a video more than in single image is a problem with moving objects. For now the ideal scene for creating HDR image is the completely static scene. There is also a new method called Dual ISO, this method needs HW support and it combine different ISO on each line of the image.

Video ghosting

This effect was an earlier problem of analog television signal problem. But with new methods of capturing HDR videos like using multiple exposures this problem appears again. If the camera capture video of moving objects (shown in image 2.15), or the camera itself may slightly move. It means that it capture serie of images where the object is on each image in a little bit different position. And with HDR images create by merging the corresponding pixels from different images (with different exposures), the moving object's pixels are not at the same place on both images. So merging cause in the HDR image ghosts effects from nonoverlapping pixels of moving object. This problem is more intensive in natural scenes like the wind, the sea, the clouds, the flowers, etc. From [10]

Tone mapping

The HDR content is already available, but the most of users still have a display with LDR range. And each display may have a different technology (LCD, LCoS, PDP,DLP,OLED, etc) even the same technology with a different product may have different capabilities. And in these days the display technology is not only about computer display but the mobile displays, electronic books, and other built-in technology. All these display may have different contrast, black level, and each technology can be used in characteristic viewing conditions. That's the reason the tone mapping algorithm should be specialized on specific display. In [2] was introduced tone mapping that minimize visible distortions. Their method is able to adapt to different conditions, like the setting of mobile backlight.etc.

The typical part for compression are the values of luminance, they make the image with dynamic range, so the simple scaling of values is the part of tone-mapping. But the simple scaling causes the loss of details, interpret as contrast. As in human visual system do (e.g.



Figure 2.15: Example of ghosting effect in HDR

in dark), the tone-mapping have to handle a change of the light range. In human eye is the pupil the part, that manage the amount of light going into the eye, if the environs is darker, the pupil let more light into the eye. And then the rods and cones photoreceptors absorbed the light in their range - both together cover all human visual range and the brain do the rest. The basic HDR tone mapping model process the scene luminance values and extract parameters from the scene that is after that used in the appropriate algorithm of tone mapping.

2.3.1 Formats

mlv video format

MLV is the product of Magic Lantern software, which is suitable for Canon cameras. In this format is saved raw pixel data with metadata from the camera and Magic Lantern software setting.

14-bit format

Magic Lantern video want to use the potential of Canon 5D sensor, which is able to capture color values in the 14-bit range. But in the binary world the 14-bit is not typical value format. So Magic Lantern use structure of 16-bit size. It is known that raw files itself have quite a big size so Magic Lantern design that on seven 16-bit values will be saved eight values as in table 2.2 .

Since the Magic Lantern video capture raw format, the table 2.3 defines Bayern pattern for storing pixels in the mlv file.

Row/Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	a	a	a	a	a	a	a	a	a	a	a	a	a	a	b	b
2	b	b	b	b	b	b	b	b	b	b	b	b	b	c	c	c
3	c	c	c	c	c	c	c	c	c	c	c	d	d	d	d	d
4	d	d	d	d	d	d	d	d	e	e	e	e	e	e	e	e
5	e	e	e	e	e	e	f	f	f	f	f	f	f	f	f	f
6	f	f	f	f	g	g	g	g	g	g	g	g	g	g	g	g
7	g	g	h	h	h	h	h	h	h	h	h	h	h	h	h	h

Table 2.2: 14-bit encoding

collum/row	01	23	45	67	89	AB	...
0	RG	RG	RG	RG	RG	RG	...
1	GB	GB	GB	GB	GB	GB	...
2	RG	RG	RG	RG	RG	RG	...
...

Table 2.3: raw pixel color structure

DNG

Digital negative, or DNG, is a non-proprietary file format using for storing camera raw images. It was developed by Adobe in 2003 as an open format. There is license published, which allows anyone to exploit DNG, with no intellectual property from Adobe.

This file format support wide variety of cameras and sensor types and save them in the same file layout. The end user may also use DNG as an intermediate format for storing images, which were originally captured in a proprietary camera raw format.

Because there is so much camera raw formats and not every of them has proper public documentation, DNG is trying to offer one compatible format, which has open documenta-

tion of data as well as metadata. So it is easier to work with that format data, and many software programs can add support for this file format.

Dng file format is an extension of the TIFF6.0 format and is compatible with TIFF-EP standard.

TIFF specification 6.0

Informatin from [8].

Image File Header A TIFF file begins with an 8-byte image file header, containing information as byte order, number that identifies the file as TIFF file, an offset of the first IFD, defines a location of the beginning of the TIFF file.

Image File Directory An Image File Directory (IFD) consist of a 2-byte count of the number of directory entries, followed by the sequence for field entries, followed by the offset of the next IFD. Image File Directory is essentially a flexible set of tags that an image writer software wishes to specify.

Multiple Images per TIFF File There may be more than one IFD because DNG can save in one file more than one image, so each IFD defines a subfile.

Document Storage and Retrieval There are few fields that may be use for describing document such as Document name, Page name, Page number, X position, Y position.

Compression TIFF specification defines more type of compression like LZW compression, Huffman compression, etc.

Tiled Images Typical organize of images is in strips, but there is another option in tiled. For high-resolution images can be accessed more efficiently, if the image is divided into square tiles instead of horizontally-wide but vertically narrow strips

Color spaces- In TIFF files, there can be used different types of color space. Supported types are CMYK, RGB (also as CIE XYZ), YCbCr

TIFF specification image types

Bilevel image type A Bilevel Image contains two colors -black and white. Tiff allows an application to write out bilevel data in either a white is zero or black is zero format. This information is saved in field Photometric Interpretation. Data can be stored as compressed or simply uncompressed. Image is organized as a rectangular array of pixels defined by ImageLength and ImageWidth. Tiff support breaking the image into separate strips for increased editing flexibility and efficient I/O buffering.

Grayscale image type The grayscale image is a generalization of a bilevel image. Grayscale images can store shades of gray. To describe such images must be defined BitsPer-Sample, which specifies the quantity of gray color number (16 or 256).

Palette-color image type These images are similar to grayscale images, they still have one component per pixel, but this component's value is used as an index into a full RGB-lookup table. To describe such images, the Photometric Interpretation should be set to 3, as Palette Color. And Color Map has to be set.

Tag name	Decimal	Type	Value
ImageWidht	256	SHORT or LONG	
ImageLength	257	SHORT or LONG	
Compression	259	SHORT	1,2 32773
PhotometricInterpretation	262	SHORT	0 or 1
StripOffsets	273	SHORT or LONG	
RowPerStrip	278	SHORT or LONG	
StripByteCounts	279	SHORT or LONG	
XResolution	282	RATIONAL	
YResolution	283	RATIONAL	
ResolutionUnit	296	SHORT	1 or 2 or 3

Table 2.4: Bilevel image requirement

Tag name	Decimal	Type	Value
ImageWidht	256	SHORT or LONG	
ImageLength	257	SHORT or LONG	
BitsPerSample	258	SHORT	4 or 8
Compression	259	SHORT	1 or 32773
PhotometricInterpretation	262	SHORT	0 or 1
StripOffsets	273	SHORT or LONG	
RowPerStrip	278	SHORT or LONG	
StripByteCounts	279	SHORT or LONG	
XResolution	282	RATIONAL	
YResolution	283	RATIONAL	
ResolutionUnit	296	SHORT	1 or 2 or 3

Table 2.5: Grayscale image requirement

Tag name	Decimal	Type	Value
ImageWidht	256	SHORT or LONG	
ImageLength	257	SHORT or LONG	
BitsPerSample	258	SHORT	4 or 8
Compression	259	SHORT	1 or 32773
PhotometricInterpretation	262	SHORT	3
StripOffsets	273	SHORT or LONG	
RowPerStrip	278	SHORT or LONG	
StripByteCounts	279	SHORT or LONG	
XResolution	282	RATIONAL	
YResolution	283	RATIONAL	
ResolutionUnit	296	SHORT	1 or 2 or 3
ColorMap	320	SHORT	

Table 2.6: Pallete color image requirement

Full-color image type For example in a full-color RGB image, each pixel is made up of three components: red, green and blue. There is no Color Map. To describe an RGB image has to be defined BitsPerSample for each channel (for example 8,8,8). Photometric Interpretation is set 2 as RGB. And SamplePerPixel defines how many components describes pixel. For RGB is 3.

Tag name	Decimal	Type	Value
ImageWidth	256	SHORT or LONG	
ImageLength	257	SHORT or LONG	
BitsPerSample	258	SHORT	8,8,8
Compression	259	SHORT	1 or 32773
PhotometricInterpretation	262	SHORT	2
StripOffsets	273	SHORT or LONG	
SamplesPerPixel	277	SHORT	3 or more
RowPerStrip	278	SHORT or LONG	
StripByteCounts	279	SHORT or LONG	
XResolution	282	RATIONAL	
YResolution	283	RATIONAL	
ResolutionUnit	296	SHORT	1 or 2 or 3

Table 2.7: RGB full color image requirement

DNG format extension overview

Information from specification [1].

Byte Order DNG allow either byte order for easier manipulation and versatility.

Masked Pixels Most cameras measure the black encoding level using fully-masked pixel at the edges of the sensor. They can be included in stored image, but then must be specified ActiveArea tag, which specifies the image borders. Or the black encoding level can be saved in special DNG tags. Black encoding level is required for DNG readers to process the image for computing the black level for different camera's models.

Defective Pixels There are two ways how to deal with defective pixels in DNG. Map out before storing the raw data, or include a bad pixel fixing opcode.

Metadata Metadata can be saved by using TIFF-EP, EXIF, IPTC or XMP metadata tags.

Proprietary Data Camera manufacturers may want to include their own proprietary data, for their programs. DNG allows proprietary data to be stored using private tags, private IFDs, and private MakerNote.

Camera Profiles DNG allow multiple camera profiles in single DNG file. The primary camera profile is stored in IFD 0, as it was in the earlier version, for backward compatibility. A Camera profile contain set of tags like CalibrationIlluminant, ColorMatrix, ForwardMatrix, ProfileHueSatMapData, ProfileLookTableData, ProfileName, ReductionMatrix, etc.

Opcode Lists For additional processing steps can be specified Opcode Lists. This allows moved off the complex processing from the camera hardware, which often has limited processing power. This processing is moved into the DNG reader, which is often running on more powerful hardware. For example, lens corrections that ideally should be performed on the image data after it has been demosaiced. And still retaining the advantages of the raw mosaic data format.

Floating Point Image Data DNG from version 1.4.0.0 support floating point image data in bit depths 16,24 and 32 bits per sample.

Transparent Pixels DNG 1.4.0.0 allow set the level of transparency of pixels for the image data. This is useful if not all of the pixels in the image bounding rectangle are defined. Transparency is supported by using masks that are stored in IFDs with the same width and length as the image IFDs. Fully transparent pixels are stored with a mask value of zero, and fully opaque pixels in the mask are stored with a value of 1.0 for floating point, and the maximum possible unsigned integer for integer masks. Intermediate values are allowed for semi-transparent pixels.

Proxy DNG Files Proxy DNGs are files with a function of placeholders for original DNGs. They are used in an application that have limited bandwidth or available storage space. Proxy DNGs store information about the pixel size of the original DNGs file. That allow rendering parameters to be transferred back to the original DNGs with almost similar visual results.

Bits Per Sample DNG support values from 8 to 32 bits/sample.

DNG has many other options that can be set or use; it only depends on the user how good he can manage the whole potential of this file format.

OpenEXR photo format

OpenEXR is an open-source high-dynamic-range-floating-point image file format for high-quality image processing and storage[7]. OpenEXR file format was found out in 1999 by Industrial Light & Magic, after few years it was redefined and ILM released OpenEXR as an open-source C++ library. The last big revision was in 2013 when OpenEXR version 2.0 was released by WetaDigital and ILM. OpenEXR is one of the formats which is often used for HDR imaging because it has many features like 16-bit or 32-bit floating-point numbers, good color resolution, lossless and lossy data compression, arbitrary image channels, scan-line or tiled and multi-resolution images, multiple views, deep images, multi-part files, additional data, C/C++ programming interface, fast multi-threaded file reading and writing, portability etc.

OpenEXR notions

Introduction for notions used in OpenEXR definitions.

Flat Image Define image where the pixel has at most one stored value or sample per channel.

Deep Image This image can store an unlimited number of samples per pixel. Each of those samples is associated with distance from the viewer or depth. All channels in

a single pixel have to have the same number of samples, but the number of samples may vary from pixel to pixel.

Parts OpenEXR file may store a single part file or multi-part file, where each part has an own header and corresponding pixels.

Pixel space Pixels are stored in the 2D coordinate system, where x is increasing from left to right and y is increasing from top to bottom. The pixel may have the form of data sample or list of data samples.

Scan Lines/ Tiles When pixel are stored in horizontal rows, it's called scan lines. If an image is subdivided in smaller rectangles, it's called tiles.

OpenEXR File Structure

An OpenEXR file contains one or more file parts. Each file part has a header and a corresponding array of pixels[7].

Header

An OpenEXR header is part of the file where metadata is saved. It ensures that file can be easy read or interpret by programs using OpenEXR. The header is basically a list of attributes that describes pixel data. Each attribute is a pair of name and value. Name is string, and the value is an item of an arbitrary data type.

displayWindow Define the boundaries of an image by an axis-parallel rectangular region in pixel space.

dataWindow If there is not define all pixels in the display window or the data are beyond the boundaries; the data window rectangle may be set. Simply it characterizes in which rectangle are the data saved.

pixelAspectRatio This value (dx/dy) is distance between pixel space location (x,y) and (x+1,y) as dx, and between (x,y) and (x,y+1) as dy.

channels Defines channels stored in an image. Channels is define by name („R“, „Z“, etc.) which help program with interpretation of this channel.

compression Specifies the methods of compressions, which is applied to the pixel data for all channels in the image

lineOrder Specified if the lines of Y are increasing, decreasing, random or its tiled image.

screenWindowWidth/screenWindowCenter These values are used in perspective projection in image

tiles Only if the image is organized as tiled, which means that image is subdivided into an array of smaller rectangles (tiles).

Header adds for multipart or deep files

name Specifies the unique name of the part.

type Each part can store different types - scan line image, tiled image, deep scan line or deep tiled.

maxSamplesPerPixel This parameter is for deep files, it indicates the maximum number of samples in any single pixel within the image.

Pixel buffer

Data values are stored in pixel buffer in scan lines or tiles.

data types Each channel specified in pixel buffer can have a different type. There are three types defines. HALF is 16-bit binary floating-point used for regular image data. FLOAT is 32-bit binary floating-point. This is used where the range of 16-bit numbers is not sufficient, for example, depth channels. UINT is 32-bit unsigned integers used for discrete per-pixel data such as object identifiers.

RGB color space In OpenEXR pixel data can be used RGB color space, which is specified by red, green and blue channels.

Channel	CIE X	CIE Y
Red	0.6400	0.3300
Green	0.3000	0.6000
Blue	0.1500	0.0600
White	0.3127	0.3290

Table 2.8: Default chromaticities attribute for RGB to CIE X, Y

CIE XYZ space If pixel values are represented as CIE XYZ tristimulus values, X, Y, and Z values should be stored in the file's R, G and B channels in following values of chromaticities

Channel	CIE X	CIE Y
Red	1	0
Green	0	1
Blue	0	0
White	1/3	1/3

Table 2.9: Default chromaticities attribute for XYZ to CIE X, Y

Data Compression

OpenEXR offers several different data compression methods. Optionally, there is possibility store the pixels in a uncompressed form, but that means the bigger files. Nevertheless with fast file systems, uncompressed files can be written and read significantly faster, than compressed files.

Lossless methods These methods left the pixel data not altered. OpenEXR supports PIZ, ZIP, and RLE

Lossy methods During these compressions can be the pixel data slightly changed. OpenEXR supports PXR24, B44, and B44A.

2.4 Software source

This section describes the programs and libraries used in the program section of this work.

mlv_dump

mlv_dump is one of the modules, which are part of Magic Lantern source code. This program helps with basic post processing operations of mlv video format. For us is most important function that dump mlv file into dng and wav, although it has few other function. It can also convert mlv into raw and wav files, can compress and decompress frames using LZMA, can convert bit depth in a range from 1 to 16 bits, or it can just help with debugging.

As we already know, mlv is video file format and dng is image format. Which means that after using mlv_dump we have in one dng file saved one video frame. In these frames is easier to do some image processing that will lead to the final format. Since dng can hold some metadata it is very useful to save some. First we have to save some values of camera profile because the image color values are now saved in raw format. Meaning that each pixel has only one of the red, green or blue color values. And camera profile data is important for reconstructing the best image from these values.

metadata example

For example, there are some samples of important metadata stored in dng which can be seen in verbose mode output:

raw.info This block exactly specifies how to parse the raw data. It contains size of the image, number of bits per pixel, active area coordinates, CFA pattern for debayering, etc.

rtci Contains the current time of day information that can be gathered from the camera.

expo Exposure attributes as ISO values and shutter

lens Available information about lens

idnt Contains camera identification data, like serial number and model identifier

wbal Information about the current white balance status as kelvin value and colors gain.

program

For command line export mlv into dng frames just run mlv_dump as: „mlv_dump.exe -dng <in>.mlv -o <prefix>“

Libraries

DNG Software Development Kit (SDK) is programming code published by Adobe for support reading and writing DNG files. This software has a version for Windows or Macintosh. DNG SDK is written in C++ code, easily use in Microsoft Visual Studio.

OpenEXR library OpenEXR library is C++ code published by Industrial Light & Magic for Easier manipulation with OpenEXR files. On official pages, OpenEXR are located documents with examples, showing how to use this library.

Chapter 3

Analysis and Specification

In this section are introduced actual state and the details of the specification.

3.1 Actual state

As was already introduced Canon 5D can also capture RAW images in 14-bit size on each color channel. It is more than usual cameras do. That give us more quality. It can be real advantage in connection with HDR content. HDR images are not only about high dynamic range, but it is also an issue of high-quality images. The new HDR displays give reason to develop new types of better and more quality images.

Once the Canon 5D can capture images in 14 bits, open source project Magic Lantern develop the software, which can capture video in this quality. Magic Lantern has own video format, which do not have a player already. So they need to be post processed by some software to get playable video output.

Magic Lantern already developed software that can make the frames from the mlv video, but only to format DNG, which is still saved as RAW image. The problem is if the user wants to use mlv video for storing HDR videos captured as two LDR images with different exposure times. Because if the video is transferred into single frames, they are still not HDR frames. It needs another algorithm to mix the two frames into one HDR. And if the images are in RAW format, it is hard to mixed them. So the first step is to have full-color images, and then they can be mixed into one HDR image and after that they can be again joined into a video file.

Although in these days HDR display are still too expensive. It does not mean that these HDR images we can get from cameras are undisplayable. There are good algorithms that can do tone mapping. So they can remap values for displaying device and images still look really good.

3.2 Design

The first step is to have mlv video file format. This mean to install Magic Lantern software onto Canon 5D Mark III camera. And shoot some video.

After that, the next step is to process video. This video should be cut into frames, that can be joined as HDR images. In cooperation with the University of Warwick and their software, we decide to cut the video in OpenEXR file format frames because this file format has good support for HDR images.

The first idea of design was to use the source code of Magic Lantern module program (mlv_dump), which already can cut the video into DNG frames. But because the modules are closely linked together and because working with OpenEXR files is easier with its library, there was the problem to get this two parts together. The mlv_dump is written in C not compatible with Visual Studio, and the openEXR library has documentation primarily describing the build in Visual Studio.

So after discussing this problem, we decide to do the intermediate step to cut the video into DNG frames by an already existing program. And then do conversion between DNG RAW file to OpenEXR files. After this step, the frames can be joined into the video or there can be done the transformation to HDR frames and after that join them.

Chapter 4

Decoder design

The decoder takes data from Magic Lantern software, which is installed in the Canon 5D Camera Mark III. So the first step is to capture this video. Then as was already explained, is this video divide into DNG frames in raw format by program `mlv_dump`. These frames have to be processed by debayering algorithm and by some color corrections. Then this frames may be used for other HDR developing.

4.1 Mlv data

The first step for the decoder is getting the proper data for decoding. For that purpose was used Canon 5D camera with Magic Lantern software, which save RAW video data in `mlv` format

Magic Lantern video format was made as format, with good compression, it is not quite user-friendly for software reading values. And the documentation for this format has no proper documentation, it is part of developer forums. It is easier to convert video to another well-specified format.

In one module of Magic Lantern is the program called `mlv_dump`, which already can transfer `mlv` into individual DNG frames. It copies data as well as save metadata in DNG file. It also converts 14-bit values into typical 16-bit structures, so it is easier to work with that, in C++ programming language. But still the data stay in raw format, which is not suitable for other data processing.

4.2 Software needs

The big issue in first few month was finding the right software working together. For future development in connection with graphics department at the University of Warwick was chosen the Windows operating system. For Windows already exist build `mlv_dump.exe` on official pages. The next step was compiling DNG_SDK and OpenEXR library project in VisualStudio. Despite the OpenEXR projects have a good quality of documentation, compiling DLL libraries in Visual Studio hampered many errors. When I found page with instruction how to compile library on Visual Studio 2010.

In short, the first step is download OpenEXR and ILMBase projects. Download and install `zlib`. In ILMBase project correct project properties like set macros `PLATFORM_WINDOWS` and `OPENEXR_DLL`, correct paths to configuration files, etc. And after all this corrections are done the project should be built successfully. Then in project OpenEXR copy few files

in right places. And in IlmImf project add missing files from source code and add macros ZLIB_DLL, ZLIB_WINAPI and then it should build successfully. And finally we have dll libraries.

From the legal issues, the Adobe DNG SDK have to be download in the form of source code. And of course, it has so many dependencies to install and on the official pages there is no proper manual what to do (the well-described instructions can be found fo Linux). So for installation of DNG SDK 1.4 have to be also installed the XMP SDK. The need is to install the expat and zlib third-party software and use right version of cmake.

If it builds successful, it just need to include right the dll libraries to DNG_SDK properties setting

4.3 DNG file

For reading DNG file the first step is to create `dng_file_stream` which is the class defined for reading or writing file. Then the `dng_host` is defined, this class is the main communication interface between user setting and DNG SDK, it customize the behavior, e.g. memory allocation, etc. Now the program parse metadata from header of DNG file into `dng_info` class, it use the values from `dng_file_stream` and `dng_host`. From this class, the program extracts the required values for other processing. And then finally the program parse `dng_host` with `dng_stream` and `dng_info` to save DNG data into `dng_negative` class. Through this class the program has access to buffers storing the data of pixels.

To access buffer from the `dng_negative` is used `dng_image`. This value has an option to process saved data as `Stage[number]Image`. In the program the data from `textttStage1Image` are processed by linearization to `Stage2Image`.

Now we have pixel data in form of linearized pixel buffer that we can acces trough value `dng_const_tile_buffer`.

4.4 EXR file

For storing values in EXR file, are defined three types HALF, FLOAT, UINT32. For this part of the project, the type HALF is sufficient. It has the 16-bit size for each value in the channel.

For creating the file, is the frist step formulating header parameters. There is set the size of the image. Also in this case is it the size of the matrix for each channel. Then is in defined how many channels and what types will be in the file. In need, the header may contain other user parameters.

If the header is defined, the file may be create according to its definition. Then the values for each channel are inserted in structure `FrameBuffer` by rows and after that is the structure with all its rows stored in the file.

This function is in the program called `writethalf`. It takes as parameter name of the final file, three arrays(example in image 4.1) of HALF type with values for red, green and blue image colors and width and height of the final image.

4.5 Debayering

In this phase, the image still has only one color on each pixel. And it is needed to apply debayering algorithm to reconstruct the other colors from the raw image shown in image

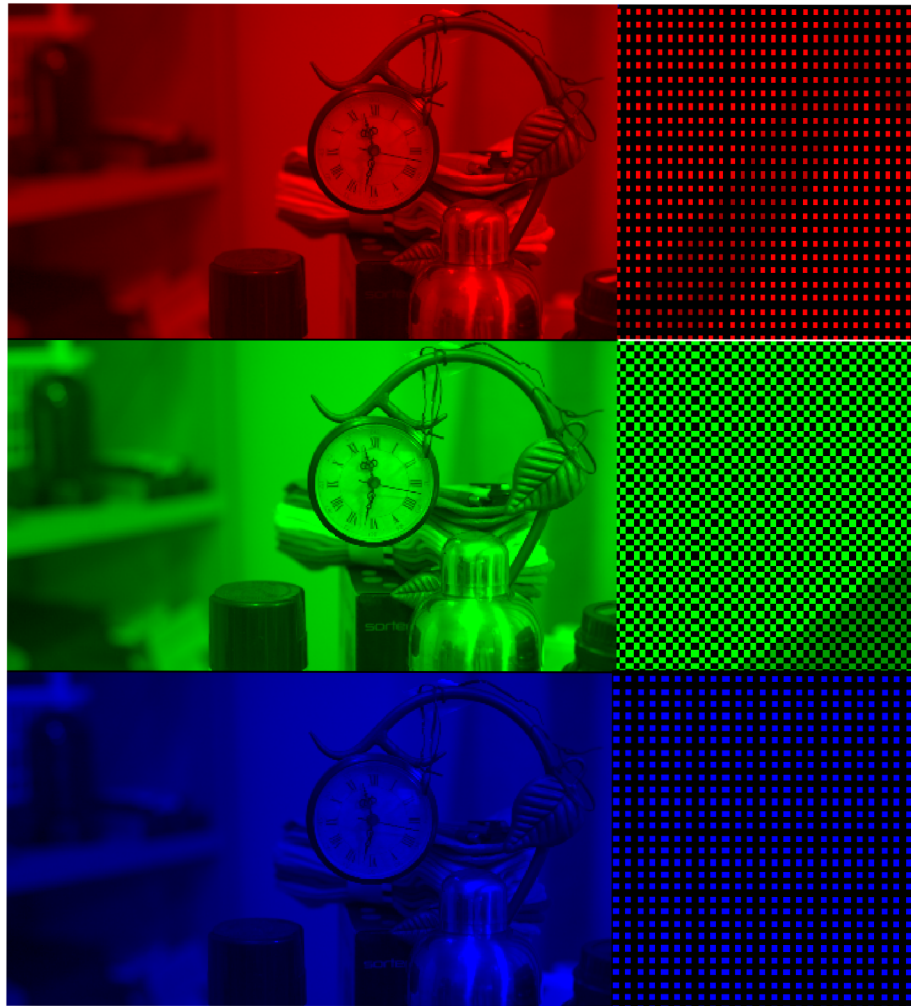


Figure 4.1: The separate channels in OpenEXR

4.2. For data testing was implement algorithm introduced in section 2.2 as the Simple interpolation method. This method is computationally simple, but the resulting image have one-quarter of the original size, which is why this method was used only for testing. But this thesis is not only about the correction of one image, it's about video processing. So I decide to choose an algorithm with small computational demands, interpolation algorithm. But of course in need of better output in the price of needs more time, it just about add function for a different demosaicing algorithm.

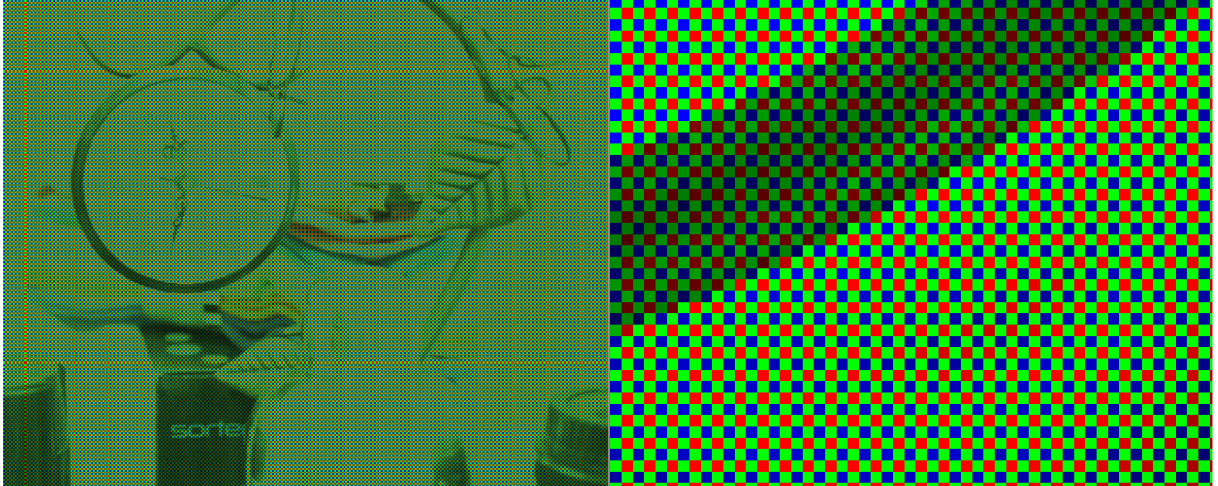


Figure 4.2: Raw data in EXR file

4.5.1 Bilinear

One of the simple interpolation methods is a bilinear method. It just take the 3x3 neighborhood of each pixel and do the average of these values.

As in image 4.3 If the value of color is already saved it does not change. For missing colors:

Red missing

```

red row:
    new=(buffer[row][column+1]+buffer[row][column-1])/2
blue row, red column:
    new=(buffer[row+1][column]+buffer[row-1][column])/2
blue row, green column:
    new=(buffer[row+1][collumn+1]+buffer[row+1][column-1]+buffer[row-1][column+1]
    +buffer[row-1][column-1])/4

```

Green missing

```

for each:
    new=(buffer[row][collumn+1]+buffer[row][column-1]+buffer[row+1][column]
    +buffer[row-1][column])/4

```

Blue missing

```

blue row:
    new=(buffer[row][column+1]+buffer[row][column-1])/2
red row, blue column:
    new=(buffer[row+1][column]+buffer[row-1][column])/2
red row, green column:
    new=(buffer[row+1][column+1]+buffer[row+1][column-1]+buffer[row-1][column+1]
    +buffer[row-1][column-1])/4

```

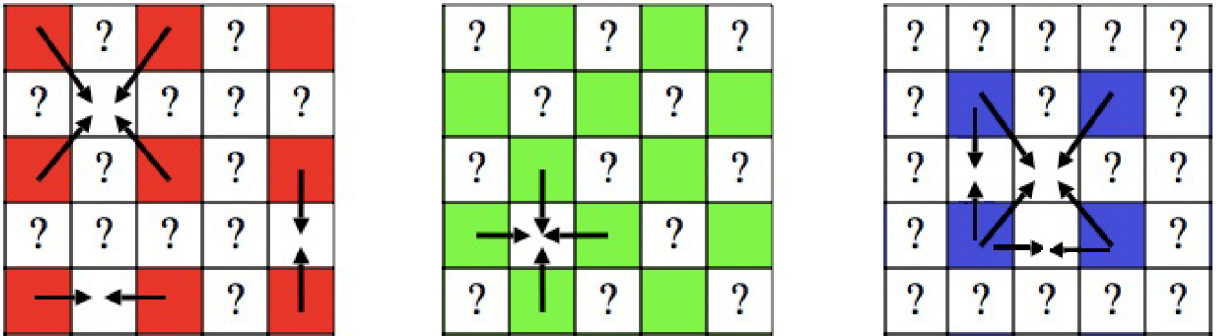


Figure 4.3: Bilinear interpolation [5]

4.5.2 Freeman

Despite the fact that in this thesis the emphasis is placed on fast algorithm, the bilinear algorithm still left too much artifact. The Freeman algorithm is just expanding the result of the bilinear algorithm. The first step is the same as in bilinear algorithm but then is applied the median filter on color differences. As we can see in image 4.4, it covers the color mixing on edges.

Applying median filter means that for each color are recount all values as the median value of 3x3 matrix around the calculate pixel, the value of color in the calculate pixel counts too.



Figure 4.4: Difference between Freeman(left) and Bilinear(right) debayering

As was written now we have data in OpenEXR, but still in raw format. The next step is to create an image with full color in each pixel. We use a demosaicing algorithm. We can

implement more algorithm that the user can choose if he wants more speed, or more quality focused algorithm. After that, we have to apply other color correction from metadata save in a file. Like, set a black level, white level, color correction by camera environment setting

4.6 Color correction

Before the data are dabyered, they used the algorithm from DNG SKD, which for building `Stage2Image` from `dng_negative`. This operation work as the linearization of data, or it can be said that it recount values according to the white and the black point in the image. The result with or without this linearization is shown in image 4.5. After the linearization and



Figure 4.5: Image on the left side is processed without linearization (too much red color) and on the right side is with linearization

debayering is used the value stored by `mlv_dump` called `AsShotNeutral`. After recounting this value, the image changes the original appearance from the major green to the right color image. That we can store in OpenExr file. The result image 4.6.

4.7 HDR

In the end, we have individual frames, which contain 14-bit size channels. This should ensure great quality image, suitable for use with HDR imaging. Also, OpenEXR has good HDR support. In cooperation with WMG there is a possibility to use two or more OpenEXR images for reconstruction one HDR OpenEXR image and after that convert individual frames into video file, which is easy interpret video on a computer or another device.

4.8 Future work

The next step after processing HDR images is to try work of this decoder with dual ISO. This method were described in [citace]. This method work with the uncompressed 14-bit raw format on some Canon DSLRs cameras with CMOS technology.

They are saving values in way as in table 4.1, where the upper-case RG/GB is part of an image with different ISO than the lower-case rg/gb. Although the CMOS technology allows to set different ISO for each line, is better to set the difference by 2 lines, because then is easier to reconstruct image from all color in Bayer pattern than have this line separated. The isolated lines should lost some color information.

If the Canon 5D Mark II capture image with ISO 100 and then with ISO 1600, it cut the four stops of highlight and return three stops of shadow detail. It is not exactly the type of



Figure 4.6: Examples of result images

HDR from multiple exposures like was introduced in this thesis, but it give the advantage of shooting the two types of images at the same time. So maybe its the way how to solve the video moving objects problem for the price of other needed processing. From [3]

collum/row	01	23	45	67	89	AB	...
0	RG	RG	RG	RG	RG	RG	...
1	gb	gb	gb	gb	gb	gb	...
2	rg	rg	rg	rg	rg	rg	...
3	GB	GB	GB	GB	GB	GB	...
0	RG	RG	RG	RG	RG	RG	...
	gb	gb	gb	gb	gb	gb	...
2	rg	rg	rg	rg	rg	rg	...
...

Table 4.1: dual ISO raw structure

Chapter 5

Conclusion

The goal of this thesis was to design and implement the decoder for video captured by Magic Lantern software on Canon 5D camera. This video in a form of the 14-bit raw video should be split into individual frames. Considering the cooperation with the University of Warwick, the final format should support high dynamic range image format. This goal was achieved on the level of theory as well as on the level of the program.

From the specification of this thesis milestones, the Magic Lantern video format and Canon 5 camera were introduced in State of the Art. The proposal of implementation of the decoder was described partially in chapter Analysis and Specification and partly in Decoder design. The implementation itself is part of the project and is demonstrated in Decoder design chapter, as well as the results and propose of future work.

This thesis was partially processed during the Erasmus+ project in England at the University of Warwick. This cooperation was for me beneficial in case of new contacts with important people in the academic area. The work on this thesis also includes the visitation of the „Training School on HDR Imaging“ where I had opportunity to meet young people studying the same subject as this thesis involve.

It is hard to admit, but at the beginning of this work I heard the words high definition range image the first time in my life. But I am glad I had the opportunity to study and work in this fast developing area. Because this HDR technology maybe in a few year will be the matter of course in everyday life.

In the future, this project could be part of the bigger software, that may do HDR video file format right from the mlv video and do not need the step between - the DNG and OpenEXR images. But the more interesting area is trying to use the dual ISO in HDR videos, which may solved the ghosting problem on moving objects in the videos.

Chapter 6

CD

Programy

Elektronická forma tohoto dokumentu

Bibliography

- [1] Digital Negative Specification. http://www.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng_spec_1.4.0.0.pdf, version 1.4.0.0, Published: June 2012, Adobe Systems Incorporated, Accessed 20.6.2015,.
- [2] Display Adaptive Tone Mapping. <http://resources.mpi-inf.mpg.de/hdr/datmo/mantiuk08datm.pdf>, author: Rafal MANTIUK, Scott, DALY, Luis KEROFISKY.
- [3] Dynamic range improvement for some * Canon DSLRs by alternating ISO during sensor readout. http://linuxdingsda.de/~wintix/dual_iso.pdf, published July 2013.
- [4] Image lens. http://www.mff.cuni.cz/verejnost/zpravicky/07_fotoaparar.jpg.
- [5] Image of bilinear demosaic. <http://www.cs.unc.edu/~lazebnik/spring09/interpolation.jpg>.
- [6] Image of shutter speed. http://azphotojournal.com/wp-content/uploads/2013/10/SHUTTER_SPEED_SAMPLE_GRAPHIC.jpg.
- [7] Technical Introduction to OpenEXR. <http://www.openexr.com/TechnicalIntroduction.pdf>, published:11 November 2006, updated November 2013, Accessed 25.5 2015, Author: Lucas Digital.
- [8] TIFF revision 6.0. <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>.
- [9] BANTERLE, F.; ARTUSI, A.; DEBATTISTA, K.; et al.: *Advanced High Dynamic Range Imaging*. A. K. Peters, Ltd. Natick, MA, 2011, ISBN 978-1-56881-719-4.
- [10] Khan, E.; Orlando; Akyiiz, A.; et al.: Ghost Removal in High Dynamic Range Images. *Image Processing, 2006 IEEE International Conference, 2006*: p. 2005 – 2008, ISSN 1522-4880.
- [11] von KRIES, J.: Chromatic Adaptation. *MIT Press*, 1902: page 120–126.
- [12] SHIRLEY, P.; ASHIKHMIN, M.; GLEICHER, M.; et al.: *Fundamentals of Computer Graphics Third Ed.* A. K. Peters, Ltd. Natick, MA, 2011, ISBN 978-1-56881-469-8.