TAMPERE UNIVERSITY OF TECHNOLOGY

TIINA KOSKIRANTA

IMPROVING AUTOMATIC IMAGING ALGORITHMS WITH DUAL CAMERA SYSTEM

Master of Science Thesis

Examiners:
Prof. Hannu-Matti Järvinen and
M.Sc. (Tech.) Petri Nenonen
Examiner and topic approved by the
Council of the Faculty of Computing
and Electrical Engineering on 9th of
April 2014

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
KOSKIRANTA, TIINA : Improving Automatic Imaging Algorithms with Dual Camera System
Master of Science Thesis, 51 pages
June 2014
Major: Software Engineering
Examiner: Prof. Hannu-Matti Järvinen and M.Sc. (Tech.) Petri Nenonen
Keywords: Dual Camera, Blur Detection, Automatic Exposure Control, Automatic White Balance, Mobile Camera, Digital Camera

Mobile camera users trust widely to the automatic camera settings, but the automatic settings do not perform equally well in all situations. All of the automatic settings have their own algorithms, which their operations are based on. Automatic white balance, automatic exposure control and automatic focus are known automatic camera settings.

Dual camera system can improve the automatic settings in mobile cameras. In this thesis dual camera algorithms were researched and dual camera algorithm development framework was created. Instead of creating the entire algorithm development framework, the dual camera functionality was added to the existing single camera algorithm development framework. The existing algorithm development framework was a PC application, which could perform the same image processing steps as mobile phone camera image signal processors. The dual camera development framework is able to collect image parameters and markers from one image and provide these to another image. The dual camera development framework manages the parameter mediation by writing and reading XML files.

Few dual camera algorithms were also researched in order to proof the dual camera framework concept. It turned out that the dual camera is able to improve image quality and the automatic camera settings. The automatic exposure control can be improved by collecting additional data from the histogram and by using this information when producing the final image. Motion blur can also be reduced by dual camera algorithm that compares edge data from two distinct images with each other and then provides valuable information about the motion to the exposure control algorithm. The automatic exposure control algorithm can thus reduce exposure time and increase the sensor sensitivity in order to produce sharp images. Automatic white balance can be improved with dual camera system by estimating illumination source with dual camera system.

Even though the dual camera seems to provide some advantages to the algorithms, and the concept that the dual camera development framework introduces, seems valid, addition of second camera to mobile phone should be considered carefully. Many of the dual camera algorithms could be transformed to work with only one camera. The dual camera algorithm development takes also a large amount of time and effort, given that the dual camera is not going to be applied to all future camera phones.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
KOSKIRANTA, TIINA: Automaattisten kuvantamisalgoritmien parantaminen kahden kameran järjestelmä avulla
Diplomityö, 51 sivua
Kesäkuu 2014
Pääaine: Ohjelmistotuotanto
Tarkastajat: Prof. Hannu-Matti Järvinen ja DI Petri Nenonen
Avainsanat: Kahden kameran järjestelmä, tärähdyksen tunnistus, automaattivaloitus, automaattinen valkotasapaino, mobiilikamera, digitaalikamera

Mobiilikameroiden käyttäjät luottavat laajalti mobiilikameroiden automaatiasetuksiin, mutta kameroiden automaattiasetukset eivät kuitenkaan toimi kaikissa tilanteissa täydellisesti. Mobiilikameroiden automaattiasetuksilla on omat algoritminsa, joiden avulla oikeat asetukset voidaan valita tilanteiden mukaan. Tyypillisiä automaattiasetuksia ovat automaattinen valotusajan säätö, automaattinen valkotasapainon säätö ja tarkennuksen automaattisäätö.

Kahden kameran järjestelmä voi parantaa mobiilikameroiden tarjoamia automaattiasetuksia. Tässä diplomityössä tutkittiin kahden kameran algoritmeja ja luotiin kahden kameran algoritmien kehitystyökalu. Sen sijaan, että koko työkalu olisi luotu alusta alkaen, olemassa olevaan yhden kameran kehitystyökaluun lisättiin kahden kameran tuki. Olemassa oleva kehitystyökalu oli PC-sovellus, joka matkii kameran kuvasignaalin käsittelyprosessorin toimintaa. Kahden kameran algoritmikehitystyökalu pystyy tallentamaan yhdestä kuvasta parametreja ja markkereita ja toimittamaan näitä toisen kuvan saataville. Kahden kameran algoritmikehitystyökalu hoitaa parametrien välittämisen XML-tiedostojen avulla.

Tässä työssä tutkittiin myös muutamia kahden kameran algoritmeja kahden kameran algoritmikehitystyökalun konseptin toimivuuden varmistamiseksi. Kävi ilmi, että kahden kameran järjestelmän avulla voidaan parantaa kuvan laatua ja kameran automaattiasetusten toimivuutta. Automaattista valotusajan säätöä voidaan parantaa keräämällä lisätietoa kahden kameran avulla kuvien histogrammeista. Liikkeen aiheuttamaa sumeutta voidaan pyrkiä vähentämään kahden kameran algoritmilla, joka kykenee havaitsemaan sumentuneita kuvia reunantunnistuksen avulla. Sumeuden havaitseminen perustuu kahden kuvan reunantunnistuksella saadun datan vertailemiseen. Sumentumista voidaan pyrkiä vähentämään lopullisesta kuvasta lyhentämällä valotusaikaa ja kasvattamalla sensorin herkkyyttä. Automaattista valkotasapainon säätöä voidaan parantaa myöskin kahden kameran avulla arvioimalla valonlähdettä.

Vaikka kahden kameran järjestelmä näyttäisi tuovan joitakin hyötyjä kamera järjestelmille, toisen kameran lisäämistä mobiililaitteisiin tulisi vielä pohtia. Monet kahden kameran algoritmeista voitaisiin saada toimimaan myös yhdellä kameralla, tai vastaavia algoritmeja voitaisiin kehittää myös yhdelle kameralle. Lisäksi kahden kameran algoritmien tutkiminen ja kehittäminen vie paljon aikaa ja vaivaa, vaikka kahden kameran järjestelmät eivät todennäköisesti jatkossa tule kaikkiin mobiililaitteisiin.

PREFACE

This thesis was carried out while working in Imaging team in Microsoft Devices Group (formerly Nokia Smart Devices Group) in Tampere, Finland.

I would like to thank all my colleagues in the Imaging Algorithms and Middleware group for introducing me to the subject and giving help when needed. Special thanks to M.Sc. (Tech.) Petri Nenonen who was my supervisor on behalf of Microsoft and Professor Hannu-Matti Järvinen from Tampere University of Technology, who was the examiner of this work.

Finally, I want to thank Aleksi, who listened to my problems and helped me to clarify my thoughts. I want to thank also all my friends, who gave me the balance between work and leisure.

Tampere June 13th, 2014

Tiina Koskiranta

TABLE OF CONTENTS

TERMS AND DEFINITIONS

AEC Automatic Exposure Control.

AF Automatic Focus.

AMBR Automatic Motion Blur Reduction.

AWB Automatic White Balance.

BSI Back Side Illuminated sensor.

CCD Charge Coupled Device sensor.

CIE Commission Internationale de l´Eclairage, engl. International Commission on Illumination.

CMOS Complimentary Metal-Oxide-Semiconductor sensor.

CMYK Cyan, Magenta, Yellow and Black color components.

CMY Cyan, Magenta and Yellow color components.

D65 CIE standard daylight illuminant.

F12 CIE standard fluorescent narrow band illuminant.

FSI Front Side Illuminated sensor.

HDR High Dynamic Range.

HVS Human Visual System.

ISP Image Signal Prosess(ing/or).

JPEG Processed digital image format.

RAW Unprocessed digital image format.

RGB Red, Green and Blue color components.

XML Extensible Markup Language.

YCrCb Luminance (Y), Red-difference chroma (Cr) and Blue-difference croma (Cb)color format, same meaning with YUV format.

YUV see YCrCb.

1. INTRODUCTION

In a field of mobile imaging, image capturing automation is a trend. As the users are not typically experts in the field of photography, mobile phone cameras are designed to capture images without the need of manual settings. The mobile phone cameras typically adjust the exposure time, focus and white balance automatically. In order to be able to do that, mobile phone cameras have to determine correct exposure time, focus and white balance algorithmically. As the algorithms are not perfect, new solutions are researched all the time.

Besides of imperfect algorithms, technical restrictions of the mobile phone camera systems limit the imaging possibilities. As the camera systems have limited capabilities to capture perfect images, new modern ways have been developed to solve these technical restrictions. The technical restrictions may partly be caused by the small size of mobile phone camera components, but most of the restrictions apply also to compact digital cameras and system cameras.

There are two possibilities to solve the technical restrictions: components of the cameras can be researched and the lenses, sensors and processors can be tried to make better, but the restrictions can be also tried to solve algorithmically. As the mobile phone camera components evolve slowly, the industry cannot relay only on component development.

One way to solve technical restrictions and algorithmic problems is a dual camera system. As the one camera can limit image capturing possibilities, the dual camera system can be used as a restriction remover. In the past, dual camera systems were added to the compact digital cameras and mobile phone cameras in order to produce 3D images. However, currently the dual camera systems are considered for different reasons. As the trend of combining images computationally has obtained attention, for example in a form of High Dynamic Range images, the dual camera systems have recently gathered more interest. Reason for this is the dual camera systems ability to capture multiple images simultaneously. As the dual cameras have also the capability to calculate depth map from the entire scene, dual camera system can be used to create differentially focused images, or the background can be altered, blurred, or totally changed with the help from depth map data. If the dual cameras are considered to be placed in a product, a question arises: what else the dual camera system can achieve?

In order to test the capabilities of dual camera system, the algorithm development framework for dual camera system must be created and prospective algorithms must be researched. This thesis focuses on adjusting the existing algorithm development framework for dual camera system and improving automatic imaging algorithms with dual camera system. Instead of researching heavy computational algorithms, that cannot be implemented easily to mobile phone camera systems, simple and efficient solutions were sought.

This thesis consist of seven chapters. Chapter 2 focuses on describing the digital camera systems in general level. Chapter 3 describes the mobile phone camera systems and dual camera advantages. Chapter 4 describes the dual camera algorithm development framework and the image data processing methods and presents their benefits and restrictions. Chapter 5 describes a method for blur detection with dual camera system, whereas Chapter 6 presents other prospective dual camera algorithms that were researched. Chapter 7 concludes the subject.

2. DIGITAL CAMERA SYSTEMS

A typical digital still camera is composed of lens system, image sensor, image signal processor, and display. Digital camera image formation can be divided into image capturing and image processing pipeline. Basics of the image capturing mechanism and image signal processing (ISP) pipeline are described in Sections 2.1-2.3. Aim of this chapter is to provide understanding of basic digital imaging steps. Even though this thesis focuses only on mobile phone cameras, the following descriptions apply generally for most of the digital still cameras.

2.1 Image Capturing Mechanism

Typically all displays and monitor use red, green and blue (RGB) components as primary color components. By combining these three color components in proper ratio, displays and monitors are able to produce all other colors, shades and tones within the range of their color production abilities. Red, green and blue as primary color components are common for all additive systems, whereas subtractive systems typically use cyan, magenta and yellow (CMY) as primary color components. [1; 2]

In additive devices, light is the source of different colors as the colors are produced by combining lights with different wavelengths. In turn, subtractive systems colors are produced by removing certain wavelengths from light source. Subtractive systems typically use colorants on transparencies or on reflective medium to absorb wavelengths from light. Paint in paper is good example of subtractive system. Figure 2.1 illustrates additive and subtractive systems.

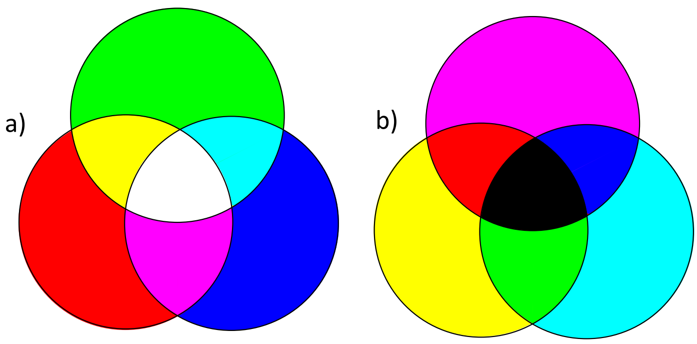

Figure 2.1: a) Additive color system, b) subtractive color system. [1]

As digital images should be displayed by monitors and displays, images must contain suitable RGB data for this purpose. Therefore, camera sensors are typically covered with RGB color filters and they collect red, green and blue data separately from the scene. For printing purposes, image data can be transformed to CMYK format (cyan, magenta, yellow and black). [1; 3]

Digital image capturing takes place in a sensor. Lens refracts light to sensor and the sensor gathers photons and transforms their energy to a raw digital image data. Schematic presentation of typical digital camera sensor can be seen in Figure 2.2.

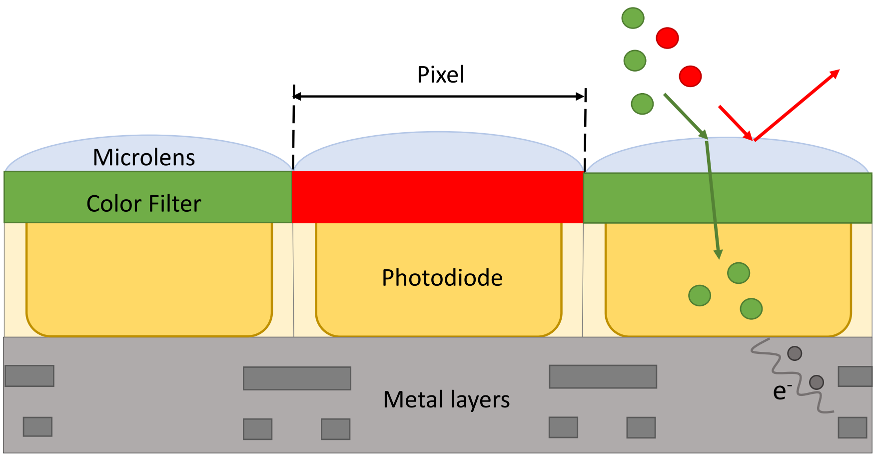

Figure 2.2: Back-illuminated (BSI) Complimentary Metal-Oxide-Semiconductor (CMOS) sensor. In BSI sensor the wiring of the sensor is placed behind the photodiodes and thus the wiring does not block the incoming light rays. [4]

Digital camera sensor is typically covered with a microlens array and a color filter array. The microlenses condense the incident light from the lens to every pixel, whereas the color filters let only light with certain wavelength through. In addition, sensors are usually covered with infrared (IR) radiation blocking filters in order to prevent sensors from detecting thermal radiation that is invisible to human eye. To gather all wavelengths visible to the human eye from all points of the scene, the color filter array is usually mosaic structure of red, green and blue color filters. One of the commonly used structures is the Bayer color filter invented by Bryce Bayer in 1976 [5]. The mosaic structure of Bayer filter can be seen in Figure 2.3. Because each pixel in Bayer array gathers only certain bandwidth of wavelengths (red, green or blue wavelengths), all the other color components must be interpolated from the neighbouring pixels. The interpolation can cause artifacts to the image, but it is an essential step in order to keep the resolution. To avoid artifacts, only wise interpolation algorithms should be used. [6; 7]

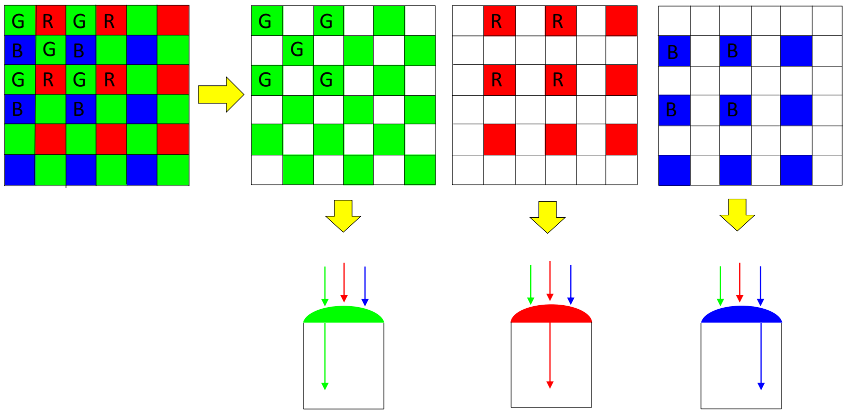

Figure 2.3: Bayer color mosaic: all color filters refracts only certain bandwidth of wavelengths. Bayer color mosaic contains 50% green sensitive pixels, and 25% red and blue sensitive pixels. [5]

Typically digital camera sensors are either Charge Coupled Device (CCD) or Complimentary Metal-Oxide-Semiconductor (CMOS) sensors. CMOS sensors can be Back-Illuminated Sensors (BSI) or Front-Illuminated Sensors (FSI). The newest sensor type BSI-CMOS sensors are widely used because of their better light capturing abilities. [4] More detailed sensor description is outside the context of this thesis, but can be further read for instance from book that handles sensors in detail [7].

2.2 Viewfinder

Modern compact digital cameras and system cameras can use display as a digital viewfinder (livepreview). Some expensive system cameras contain also optical viewfinders that are similar with traditional film camera viewfinders. Instead of an optical viewfinder, mobile phone cameras typically contain only digital viewfinders, which can display live image from the imaging situation, and thus reflect the image that the camera would take at that moment. [8]

In order to show pleasant and smooth live preview of the scene on the display, the frame rate needs to be fast. To achieve pleasant frame rate, the viewfinder must display multiple processed images per second. The image processing stages must take only fractions of seconds, and the exposure time of one image frame cannot be too long. Typically the live preview frames are taken with short exposure time and enhanced algorithmically. The live preview images may also go through fewer image processing stages to produce result image frames faster.

The viewfinder frames can be used to collect data from the environment before the final image capturing. But as the live preview needs to display all captured frames, the camera system cannot capture frames that would look unpleasant by the human eye even though they would contain valuable information that could be used to capture better still images.

2.3 Image Signal Processing Pipeline

Human Visual System (HVS) is unique mechanism, which can vastly adapt to almost every kind of situation it encounters. Currently manufacturing such a sensor, which reacts to changing situations in a similar manner as HVS, is not possible. Reasonably priced sensors and lenses contain also errors, which must be corrected before presenting the data on a display.

Image Signal Processing (ISP) pipeline is a signal-processing chain, which transforms the raw image data to visually appealing color image. ISP also affects to the image capturing mechanism by commanding lens system, and thus it can alter exposure time and focal length. Usually ISP pipeline can be divided into 6 steps: sensor and lens control, preprocessing, white balancing, demosaicking, color transform and post processing. Schematic presentation of ISP-pipeline can be seen in Figure 2.4. [6; 9]

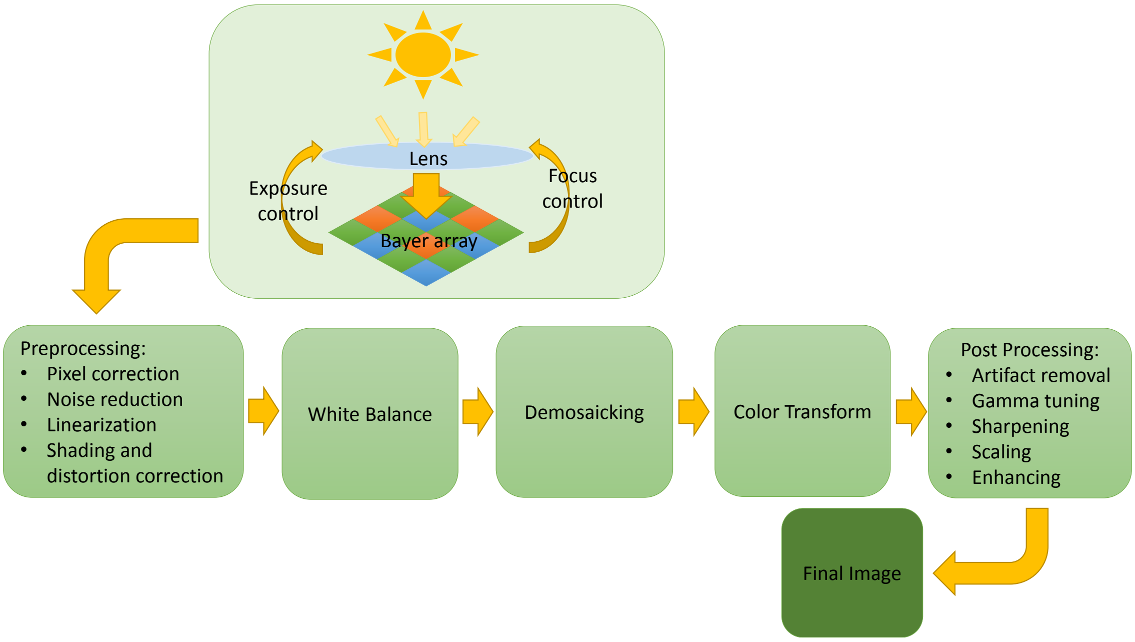

Figure 2.4: Digital image processing pipeline.

ISP pipelines differ between manufacturers and companies do not usually revel much detail about their ISP pipelines. However, the next sections describe the basics of ISP pipeline by generalizing what kind of steps the pipelines usually contain. Order of the steps may vary between manufacturers as well as number of steps.

2.3.1 Sensor and Lens Control

Image capturing mechanisms must interact with sensor in order to adapt to dynamically changing situations. To determine correct exposure time and focal length, sensor controls lens and shutter speed to meet the environments demands. By altering lens position, digital camera can change its focal length and thereby determine, which objects are in focus in the image. By controlling exposure time, digital cameras can adjust the images overall brightness. [7; 6]

Exposure Control

Usually the exposure control needs to determine and characterize brightness of the images in order to produce correctly exposed images that correspond to the experienced situations. Underexposed or overexposed images usually have lost part of their color information, and thus they do not look appealing. Figure 2.5 shows the effects of the exposure control.

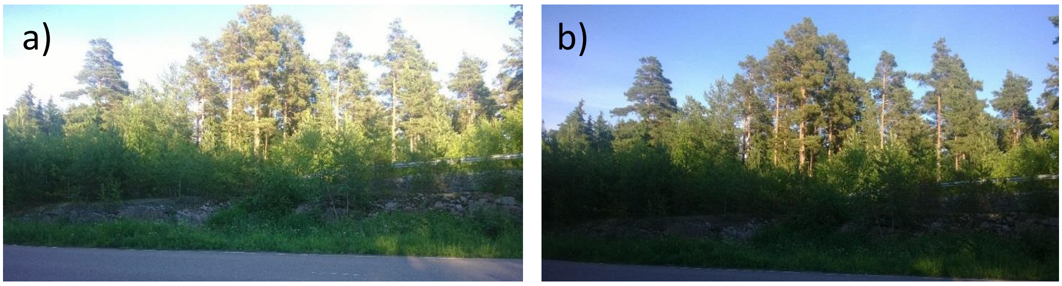

Figure 2.5: Effects of exposure control: a) overexposed image and b) correctly exposed image.

Because each sensor has a dynamic range, within it can measure the amount of light, the sensors ability to differentiate two varying colors is limited. All colors under the dynamic range are detected as black, and all colors above the range are detected as white. The amount of different tones within the dynamic range is also limited.

Pixels in a sensor can be imagined as photon wells: if the well overflows, the sensor cannot detect the overflown photons, and the sensor detects the largest amount of light no matter how much the well overflows. Up to a certain extent, exposure control can try to avoid these kind of situations by adjusting exposure time to prevent photon overflowing. [6; 10]

Correct exposure time can be determined automatically by the camera or manually by the user. For the automatic exposure control (AEC) digital cameras have their own algorithms. Basically, the AEC algorithm calculates light intensity of the image and determines the right exposure time based on these calculations. Usually the AEC algorithm calculates mean luminance of various sections of the image. Based on overall data and sectional data, AEC can select the best exposure time. AEC can also decide, weather the scene is backlit or frontlit, and thus modify the exposure time to suit to the situation. [6; 7]

Focus Control

Goal of the focus control is to adjust the lens position so that the lens can form a sharp image to the sensor. If the lens is placed too close to the sensor, the image seems blurred, as the image of the object is formed behind the sensor. On the contrary, when the lens is placed too far from the sensor, the image is formed in front of the sensor and and the image seems also blurred.

As can be seen from Figure 2.2: by changing lens position, it can be chosen which objects are sharp and which objects are blurred in the formed image. Digital camera lenses cannot show all objects from scene sharply, and even though this could be done, these images would not be visually pleasant to human eyes. [7]

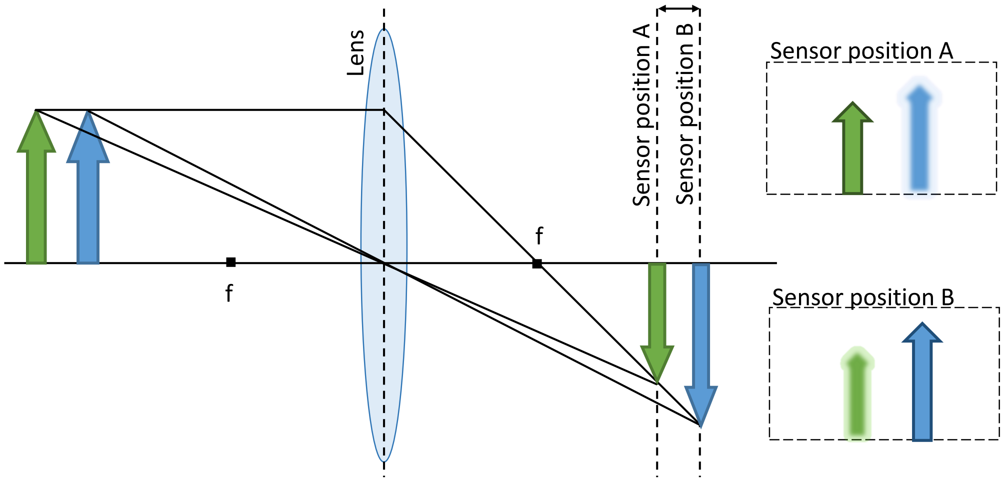

Figure 2.6: a) Green arrow can be seen sharp in a sensor, when lens is close the sensor, b) Blue arrow can be seen sharp in the sensor, when lens is moved farther the sensor.

Many digital cameras offer automatic focus (AF) option, and usually in cheaper cameras manual focus control is usually not even possible. Automatic focus can be performed algorithmically by determining image sharpness and amount of detail. This can be done by testing multiple options for lens position and by choosing the option that results the largest amount of sharpness and detail. Some distance measuring technologies, e.g. high frequency sound or infrared light, can also be used to determine correct focus. [11]

Even though, perfectly sharp images, from objects in varying places, can be formed only by changing lens position, some cheaper cameras have fixed-focus lens-sensor system. These fixed-focus cameras are designed so that they can form acceptably sharp images in usual imaging conditions. Fixed-focus cameras are typically designed so that they focus to the infinity and the aperture size is small. [12]

If the digital camera contains dynamic aperture, the aperture size can also affect to the focused area in the image. Small aperture size enables large depth of field, which means that objects can be in focus over a wider range of distance. Wide aperture size, on the contrary, reduces the depth of field. The aperture size has also affect to the exposure. Typically the mobile phone cameras do not contain dynamic aperture, and thus the depth of field and exposure cannot be controlled in this way. [13]

2.3.2 Preprocessing

As image capturing mechanism produces errors to image data, it is important that the camera can reduce these errors before displaying the image. Usually the ISP-chain contains many error correction mechanisms. Typically the images are improved in preprocessing stage through pixel correction, linearization, noise reduction, and shading and aberration correction.

Pixel correction is usually done to digital images, because camera sensors usually contain small amount of defective pixels. If these pixel errors would not be corrected in early stages, they could cause more errors (pattern errors) afterwards. Because e.g. demosaicking uses neighbouring pixel values when interpolating colors, the pixel errors can duplicate during the processing stages. [6]

Pixel defects can usually be noticed by comparing pixel values to neighbouring pixels. Long-term static pixel value and differentiation from neighbouring pixels are characteristic for pixel errors. These kind of symptoms can be monitored with different methods. Usually defective pixels values are corrected by interpolating values from neighbouring pixels. [6]

Linearization is usually done to captured images in order to ensure that the data is linear before other processing steps are performed. Even though CCD and CMOS sensors usually have linear response, involved electronics might transform captured image to non-linear space. [6]

Noise reduction is common ISP step. Environment causes different kind of disturbances to digital cameras and especially to sensors. Usually disturbance is experienced as noise in produced image. Image capturing system can try to filter the noise and thus enhance the image it produces. Filtering can be done before the light enters the sensor, or after the image data is collected by the sensor.

Noise can be divided to at least three categories: offset noise, pattern noise and aliasing noise. Offset noise is caused by so called dark noise, which is generated by the sensor even though it does not receive any light. For example, thermal noise is dark noise. Imbalance in pixel readout can cause pattern noise as well as other signal processing steps. Aliasing noise, in its turn, is caused by aliased signal and it can be reduced by adding optical low-pass filter between lens and sensor. [7]

Lens shading and color shading are caused by camera optics. Usually lens systems of digital cameras are characteristically less sensitive for light in the edges of field of view. This causes edges of the image to be darker, when compared to the center of the image. Lens shading can be reduced by correcting pixel values. Amount of correction can be made dependant on pixel distance from image center. Color shading is caused by different characteristics of colors. Lens shading and color shading are illustrated in Figure 2.7 [7]

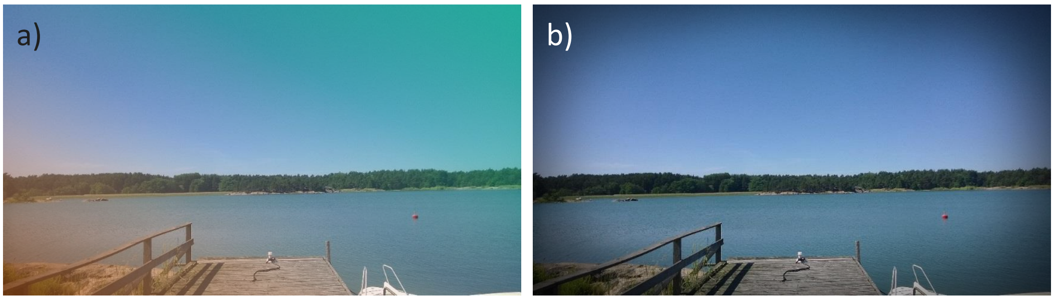

Figure 2.7: Affects of a) color shading and b) lens shading.

Lens system can also cause multiple **distortions and aberrations** to the image. One kind of distortion is lens distortion. There are two types of lens distortions: barrel distortion and pincushion distortion. These both distortions can be seen as edge curvature. In barrel distortion, edges bulge outwards, whereas in pincushion distortion, the edges curve inwards. Examples of lens distortion can be seen in Figure 2.8. [7]

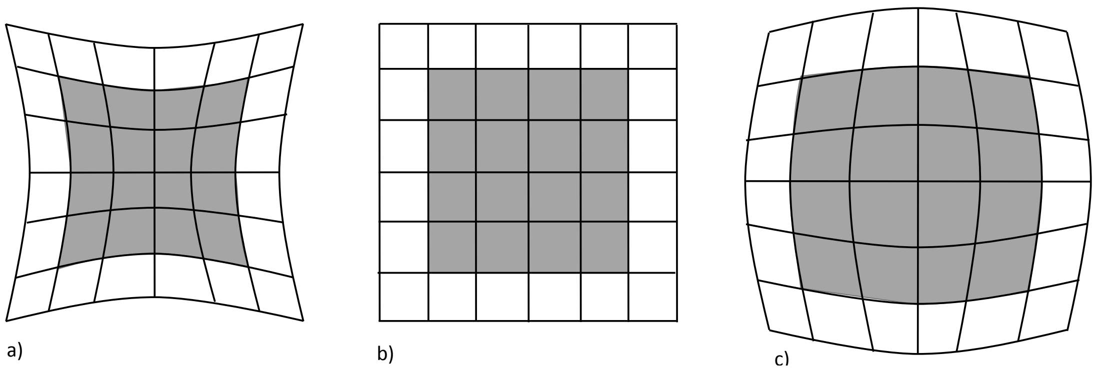

Figure 2.8: a) Pincushion distortion, b) normalized image, and c) barrel distortion.

2.3.3 White Balancing

One of the crucial processing steps in a ISP pipeline is white balancing. Because human visual system adapts to predominant situations, balancing and adjusting the colors captured by camera sensor is needed. HVS can distinguish white color (e.g. white paper) as white in every condition even though the actual reflected color of the object is something else in that situation. Figure 2.9 demonstrates, how the scene looks different by the camera under different lightning, even though human eyes would experience the scene very similar in all lightning conditions.

Figure 2.9: Unbalanced images from same scene: a) halogen lightning, b) fluorescent lightning, and c) daylight lighting.

To adapt in a similar manner with HVS, digital cameras need to determine right RGB signal levels for the images taken under different light sources. As the amount of blue and red light is usually in balance in daylight illumination, but red light has higher proportion in artificial lightning, the white balance must be adapted in order to prevent camera from capturing yellow toned images under artificial lightning. [7; 6; 14]

There are different methods to acheive correct white balance. One method is so called Gray world assumption, which assumes that all RGB components average out to middle gray. All the colors are thus scaled based on their difference from middle gray tone. The Grey world assumption is not perfect solution, as it produces large amount of error, when the image does not contain equal amount of all three color components. Another similar method with grey world assumption is Brightest white assumption, which assumes that the white color is always the lightest point of the image and determines the white balance based on that assumption. As the white balance highly depends on lightning source, the RGB signal levels can also be adjusted based on the knowledge of light source. The light source can be estimated from the calculated color distribution of the image (color gamut of the scene). There are many more methods for determining the correct white balance, and usually the white balance algorithm is combination of many of these methods. [7; 6; 14]

2.3.4 Demosaicking

Because a color image requires all three RGB color components at each pixel location, and Bayer color filter array has only one of the three RGB color components at each pixel location, interpolation is needed when producing final digital images. Interpolating color pixels is called demosaicking.

Some more expensive digital cameras contain three different sensors, one sensor for each RGB color, and thus these cameras do not need interpolation steps. These cameras split the entering light and project it to every color sensor. As typical compact digital cameras and mobile phone cameras contain only one sensor, estimating other two color samples to every pixel is needed. Image before and after demosaicking is represented in Figure 2.10. [15]

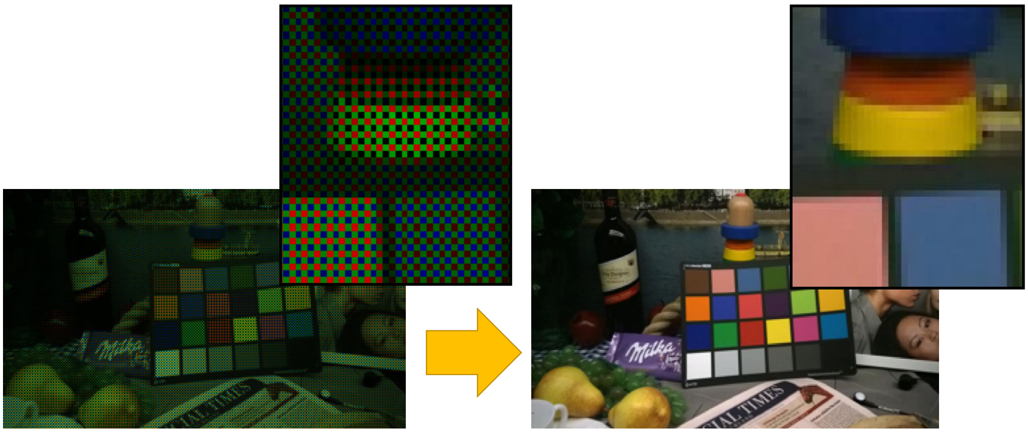

Figure 2.10: Demosaicking of Bayer color arrey image to RGB color image.

Bicubic interpolation and bilinear interpolation are simple demosaicking methods, but they produce artifacts to the resulting image, especially to the edges in image. Where bilinear interpolation calculates new pixel value from 4 neighbouring pixels, bicubic interpolation uses 16 neighbours. Bicubic interpolation can be improved by detecting edges and calculating the pixel value only by interpolating along the edges. [15] More specific demosaicking description is out of this thesis's context, but can be further read from [15].

2.3.5 Color Transform

Colour transformation is a process, where captured colors from the scene are transformed from camera color space to some generally used color space. As the captured colors are produced by the sum of e.g. optics, color filters and electronics involved, they do not usually follow up any known standards, and thus they cannot be displayed correctly by any device. Camera color space may also contain colors, which are not possible to display by today's devices. [6]

Captured images are usually transformed to follow up some standard output device color space. Vastly used displaying color spaces, that are suitable for most of the display devices, are sRGB and AdobeRGB. These color spaces make device independent color reproduction possible, but still the final color reproduction depends highly on the device and its color representation abilities. [6]

2.3.6 Post Processing

Post processing steps are usually important, because previous processing steps may cause artifacts to the images. Post processing usually includes artifact removal, gamma correction, sharpening, enhancing and scaling.

Gamma correction is done to captured images in order to preserve captured image better. Because human eyes do not have linear response to colors, they notice difference between two dark colors more easily than difference between two light colors. In order to preserve the captured image in as small space as possible, it is important that we preserve only the tone differences which human eye can notice. Gamma correction refers also to the correction done by monitors, which have nonlinear response to colors. [16]

Sharpening is usually one of the image processing steps, because human eyes prefer sharp edges, and images, which have even a small amount of blurriness in the edges, are easily experienced unpleasant. It is especially important to sharpen horizontal and vertical edges, because human eyes are most sensitive to them. [6]

Enhancing is done to captured images, in order to make them look more appealing. Typically this step can contain enhancement processes for colors, contrast, and edges. Some effects can also be added to the images. [6]

3. MOBILE PHONE CAMERA SYSTEMS

During last ten years mobile phone cameras have been trying reach the same level with compact digital cameras. The biggest single limitation for development has been size, as the size causes defects to the camera system. This chapter focuses on describing difficulties and weaknesses that the size requirements cause to the mobile phone cameras. This chapter also introduces the dual camera system and describes its applications in the area of digital photography.

3.1 Special Considerations for Mobile Phone Cameras

Camera phones alongside with web-cameras and compact digital cameras create markets for low cost camera sensors and optics. Nowadays sensor optics and lens systems are the most expensive parts to manufacture and thus they prevent the reduction of the overall camera manufacturing costs. In the meantime, CMOS sensor and other camera component manufacturing costs have been dropped down more rapidly. As CMOS sensors have capability for small size, low power and high integration, CMOS sensors have gained large interest and thus they have been vastly investigated during recent years. CMOS sensors, image processors and digital signal processors capabilities have been growing fast as their prices have been lowering rapidly. [17]

Because small size is more meaningful for mobile phone cameras than for compact digital cameras, mobile phone cameras typically demand more from camera component manufacturers. Smaller size causes more easily biases to the components, and mobile phone camera images suffer from blurriness and aberrations that small optical components cause to the captured images. Small distance between the lens and the pixel array is the main reason for peripheral blurriness and aberrations in mobile phone cameras. Also dead pixel count is much greater in mobile phone cameras, when compared to other digital cameras. Dead pixel count is usually result from low manufacturing testing done to the sensor. Most of the artifacts that the biases cause can be reduced by the ISP chain. [17]

Term computational photography has been created to represent the current trend in mobile photography. Computational photography refers to image processing and alteration, which is done in order to remove digital cameras technical restrictions. Computational photography refers especially to visual effects and enhancements that extends the capabilities of traditional digital photography and image signal processing. The line between traditional digital photography and computational photography is not always clear, but usually, for example, high dynamic range images, panorama images, and computationally added background blur (bokeh effect) are counted as different fields of computational photography. [18]

3.2 Dual Camera Systems

An additional second camera next to a main mobile phone camera can bring many advantages to imaging capabilities during and after image capturing. Even though the dual camera systems are currently rare in mobile phone cameras, as well as in compact digital cameras and system cameras, the dual camera concept seems to gather interest as a method for computational photography. The computational photography methods for dual camera system have been vastly investigated, but it is still unclear, how the dual camera system can improve image quality and help existing imaging algorithms.

3.2.1 Overview to To Dual Camera Imaging Mechanisms

Originally dual cameras were invented in order to produce 3D images. In the beginning of 21st century stereo imaging (3D imaging) started to gain attention in film industry, and around 2010 the stereo imaging was brought to digital cameras and mobile phone cameras with dual camera system. However, dual cameras in terms of 3D imaging did not become general in digital cameras and mobile phone cameras.

Recently dual cameras has gained interest for different reasons. Dual camera systems can be used to gather depth information, remove sensor restrictions, and improve imaging quality. As the mobile phone hardware becomes more powerful, it opens possibilities for computational photography, which can improve quality by combining images or through mathematical calculations.

Recently few dual camera related products that are meant for computational photography purposes, has been released: HTC One M8 smartphone, Toshiba's dual camera module and Corephotonics' dual camera module. HTC One M8 received critical reception, and its dual camera system did not gain much attention. HTC One M8 has one main camera that it uses mainly for image capturing and one assistant camera that is used only as a rangefinder and not for image capturing purposes. Therefore the camera system can create rendered effects like background blur (bokeh) and refocus. Also, the dual camera system can produce 3D-like images, that can create feeling of depth when the screen is tilted. [19; 20]

Another dual camera innovation is in January 2014 released dual camera module manufactured by Toshiba. The module is world's first dual camera module that is able to output both image data and depth data simultaneously. The module is intended for smartphones, tablets, and mobile devices. [19]

Corephotonics dual-lens module uses two lenses with two different focal lengths. This setup provides optical zoom, improved low light performance, and depth mapping possibilities. [21]

3.2.2 Dual Camera Advantages

Dual camera advantages are result from their ability to capture more information from the scene than one camera can by itself. But, at the same time dual camera system complicates image capturing mechanisms. The more advantages dual camera can offer and developers can research, the more reasonable it is to develop complicate dual camera capturing mechanisms.

Depth Map

Two cameras which are pointing to the same direction, can be used to produce depth map from the image. Distance between each object in a image can be calculated with quite a simple manner. Figure 3.1 represents trigonometry behind the distance calculation.

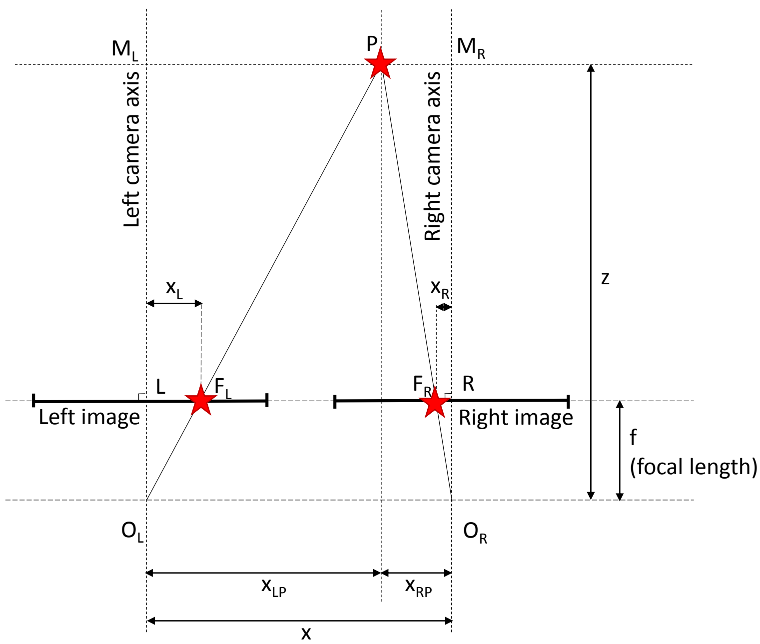

Figure 3.1: Distance from object P to left and right image can be calculated with help of simple trigonometry. [22]

Because rectangles M_LPO_L and L$F_L$$O_L$ are congruent, can be said that:

$$\frac{x_L}{f} = \frac{x_{LP}}{z} \tag{3.1}$$

and in a similar manner, because rectangles M_RPO_R and RF_RO_R are congruent, can be said that:

$$\frac{x_R}{f} = \frac{x_{RP}}{z} \tag{3.2}$$

These two equations 3.1 and 3.2 can be used to calculate distance z between the sensor and the object:

$$z = \frac{xf}{x_L + x_R} \tag{3.3}$$

By calculating the distance z with equation 3.3 for every pixel that the two images have in common, the depth map can be formed. [22]

Depth map data and dual camera can be used to produce images that are focused in every point of the image, or the images can be refocused later. Depth data can be also used for background alteration purposes. As the depth map enables the background distinction from foreground objects, the background can thus be removed, blurred or changed to something totally different.

High Dynamic Range -images

Photographers encounter challenges, when taking images from scenes that have wide dynamic range. As the cameras has limited dynamic range, typically photographers need to determine whether they want to save the dark tones or the light tones. High dynamic range (HDR) images are solution to this dilemma. HDR images can be taken with dual camera system: when the other camera produces image that focuses on saving the dark tones, the other camera can focus on saving the light tones. By combining these two images, a HDR image can be produced. The HDR image can contain more color intensities than one image can on its own. [23]

HDR images are useful in extreme imaging conditions, and especially when the scene contains high contrast differences. For example images that are taken against window contain lot of contrast and usually inside-parts are usually too dark and outside-objects too light. [23]

3D Images

3D images (stereoscopic images) are created to enhance the illusion of depth. Basic method for creating 3D images is capturing the scene from two slightly different angles and then displaying the separate images for both eyes individually. This can be achieved by special 3D classes, but also the display can be made such that it can display different image for both eyes. Only few 3D mobile phone cameras were made even though the 3D movies seem to be still in trend. [11]

Other Dual Camera Possibilities

A more recent idea is to use the dual camera system to improve the image quality. In the literature the following subjects, that could be implemented for dual camera systems, have been researched: Petschnigg et al. researched possibility of combining flash and no-flash images [24], Yuan et al. researched deblurring by combining two image pairs [25], Chen et al. researched dual motion deblurring [26], and Wo researched color constancy for stereo imaging [27]. Many of these methods are still in early investigation stages, and real product implementations are in far future.

4. DUAL CAMERA ALGORITHM DEVELOPMENT FRAMEWORK

In order to get deeper understanding about image processing algorithms and their development process, it is essential to get familiar with image data processing methods, memory management and algorithm development environment. While Section 4.1 focuses on data processing, Section 4.2 focuses on the framework that was used as an algorithm development environment in this thesis. The image processing pipeline described in this chapter reflects the image processing pipeline in the algorithm development framework, but the facts apply generally to most image processing pipelines.

4.1 Image Data Processing and Memory Management

In digital image processing, speed of the entire process is as significant as the quality of resulting image. In order to achieve acceptable speed and quality, image processing chain must have efficient methods for memory management, data processing, and data storage. Efficiency is extremely important for viewfinder frames, which should be handled in fractions of second. The following sections focus on describing image data processing and memory management methods.

4.1.1 Data Formats

Image data is formatted for three different purposes: modification, storage and display. Data formats for modification purposes are algorithm specific, as different algorithms prefer certain image formats. The algorithms are not always applied to whole image data, but only for certain components of the data. The reason behind component selection typically lies in human eye anatomy as the human eye is more sensitive to quality of certain components. Algorithms could be applied to all components, but usually that is not necessary, as the human eye would not notice difference. Good example of this are sharpening and noise filtering algorithms, which can sometimes be applied only for luminance component, because of the human eyes sensitivity to black-and-white information. [9; 11]

Storage formats attempt to save data efficiently. In storage format efficiency does not usually refer to speed, but instead to efficient use of storage space and compression. Similarly with modification formats, human eye also defines, which storage formats are efficient. YCrCb-formats are good example of storage format compressions. YCrCb-format consists of one luminance component (Y) and two chrominance components: blue-difference chroma (Cb) and red-difference chroma (Cr). Because human eyes tolerate less sharply bounded colors more easily than luminance values, YCrCb-format can be compressed to 4:2:0-format where four luminance components share one red-difference chroma component and one blue-difference chroma component. Different types of YCrCb-components are represented in Figure 4.1. [11; 28]

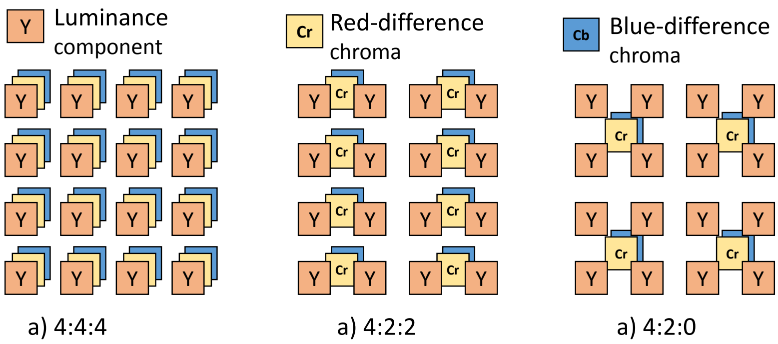

Figure 4.1: Representation of YCrCb formats: a) unpacked format, b) 4:2:2 packed format, and c) 4:2:0 packed format. [28]

Images in JPEG format are stored in YCrCb format with meta-data that is needed for reading the data correctly from the file. For displaying purposes image data is typically transformed into RGB format.

Data formats have great impact on memory consumption. Following Figure 4.2 represents the memory consumption of different image data formats in case each component has a size of 1 byte (i.e. 8 bits).

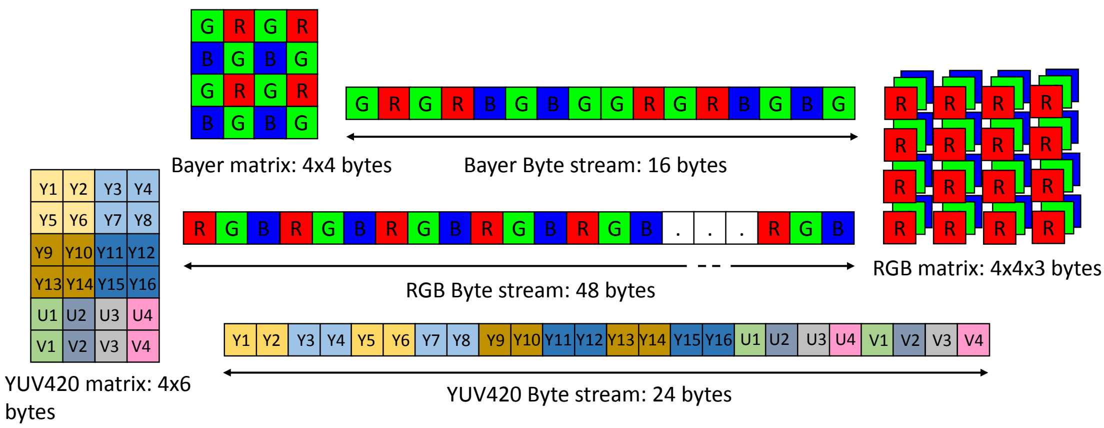

Figure 4.2: 4x4 pixel RGB-image in comparison with YUV-image and Bayer image. In this thesis YUV has the same meaning as YCrCb color format.

Another consideration of memory consumption is bit depth. As the bit depth determines how many different colors can be represented in images, the reduction of bit depth reduces accuracy. The bit depth has also influence on image processing possibilities, as the inaccuracy multiplies in the same ratio with the data values. It is important to keep the right balance between the memory consumption and accuracy. Figure 4.3 shows the alternative image formats and bit depths.

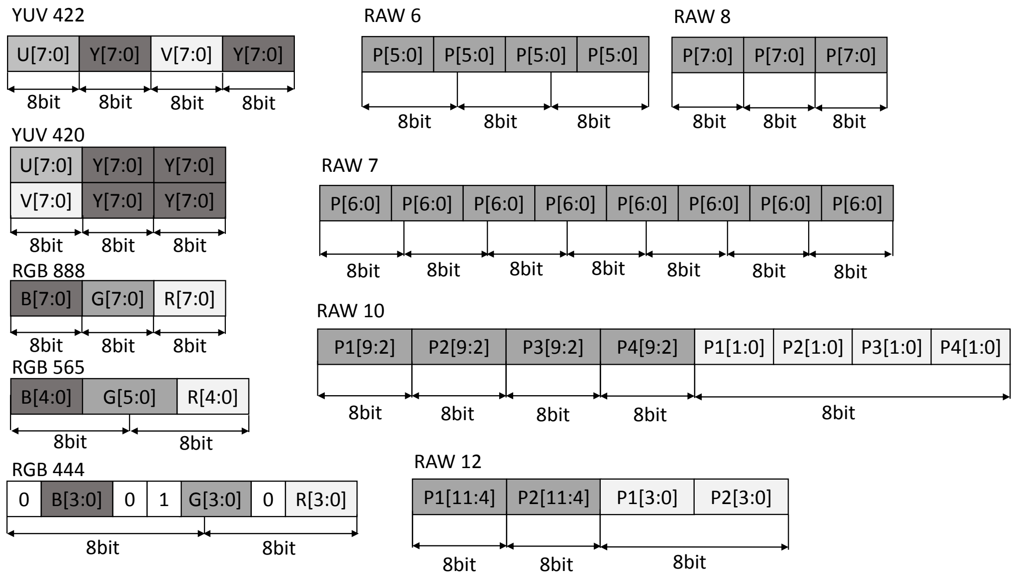

Figure 4.3: Image forms: all image forms are composed of these repetitive sections. Form YUV420 is an exception as its every second line is different. [28]

Only YUV422, YUV420, RGB888 and RAW8 formats follows straightly the byte boundaries, whereas the other formats are needed to be formatted in order to make them follow byte borders. Fitting is done typically by padding or packing. RGB565 format can be fitted into 2 bytes. RGB444 format can also be fitted to 2 bytes by adding 3 padding bits in order to make the format similar with RGB565 format. Four RAW6 pixels can be fitted to 3 bytes and eight RAW7 pixels can be fitted to 7 bytes. RAW10 and RAW12 can be made to follow bytes by dividing the pixels. Five RAW10 pixels can be fitted into five bytes, whereas four RAW12 pixels can be fitted into 3 bytes. [28]

4.1.2 Algorithms in Image Processing Pipeline

Basically all image modifying algorithms are executed in a pipeline so that each stage modifies the input data and sends modified image data as an output. The whole ISP pipeline takes a RAW formatted image data as an input and the resulting processed image is some general data format, usually in JPEG format. Basic idea of the ISP pipeline can be seen in Figure 4.4. The algorithms that collect data from the captured RAW image are typically executed before the processing algorithms in ISP pipeline. The collected data is usually saved to a data structure and algorithms have access to the data structure any time during the pipeline operations.

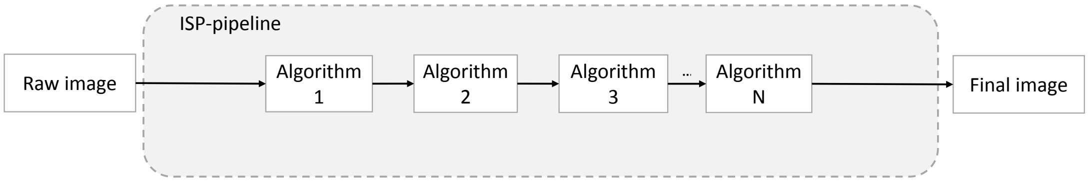

Figure 4.4: Basic structure of ISP-pipeline.

Most of the image manipulative algorithms are typically implemented in sections, and they execute the image processing operations though line buffer. Line buffer approach in image processing is efficient and low memory option. Typically line buffer contains only certain part of the image data at the time and the algorithms have access to only that part of the image. This means that the line algorithms have to have some method for telling that they need more data. Usually the line algorithms are executed as the Figure 2.2 suggests.

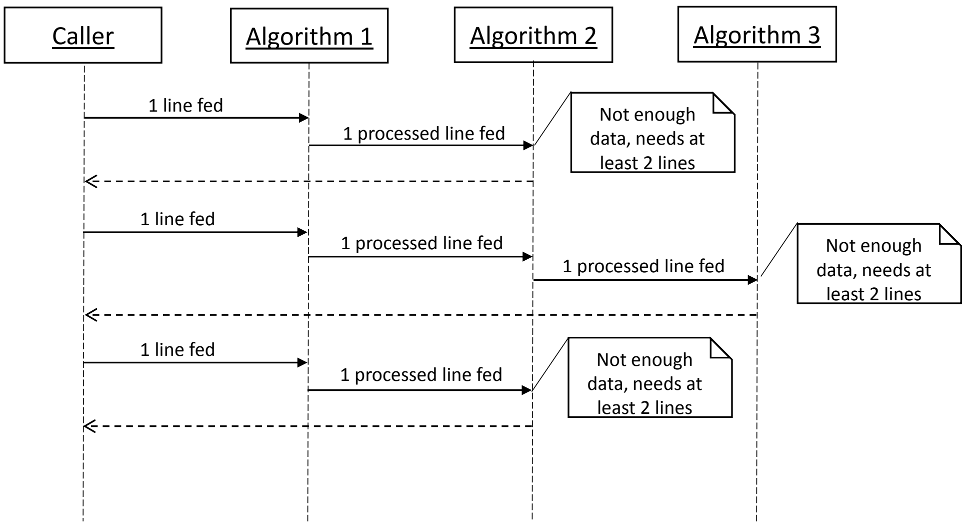

Figure 4.5: Sequence diagram of line buffer. [29]

This means that the algorithms can be executed only to certain level k, if the previous algorithm k-1, feeds enough lines for the algorithm k. [29]

4.2 Algorithm Development Framework

One of the first steps in the algorithm development is framework construction, as the final product or its prototypes are not usually the algorithm development environment. Algorithms are easier to test on a PC framework that can mimic the mobile phone image processing pipeline. Usually the PC framework is easier option, as the image processing algorithms can be tested with example images that are taken beforehand. Thus, the imaging situations are not needed to be repeated. The images can be taken a lot easier with already existing products than with prototypes that can be bulky, heavy or secret.

Algorithm development with PC framework is also faster and cheaper, as the compiled program is not needed to be installed to the device after every alteration or ultimately the used prototype is not needed to be constructed again. PC framework testing is not always the same as testing with the actual device, but usually it is the best option. The suitability of the PC framework also depends highly on the algorithm in question.

4.2.1 Dual Camera Concept in Development Framework

Dual camera system can be used in two different ways: it can produce two images or only one image that the user can see. In the first case, the dual camera system processes two images that are both visually appealing or the processed data from the first image is needed to improve the quality of the other image and their combination produces the final image. In the second case the secondary camera image is never shown to the user, but instead it is used as assistant image that can collect valuable information from the surroundings. This case differs from the first case since the assistant image does not try to look visually appealing and its pixel values are not used straight to form the final image. Figure 4.6 represents the dual camera concept in this thesis.

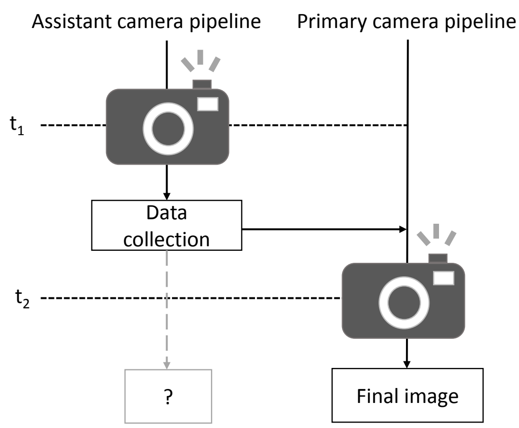

Figure 4.6: Dual camera imaging system.

This thesis focuses on only the cases where the second camera is used as an assistant camera and its image data is not used directly in image formation, but description of other dual camera methods can be read from Section 3.2.

The goal of this thesis was to build a pathway between two image processing pipelines and make data comparison between the images possible. Customization was the main theme when designing the dual camera development framework, as the idea was to create development framework, where it would be easy to implement more algorithms. Figure 4.7 illustrates the modularity of the development framework and the pathway between the pipelines.

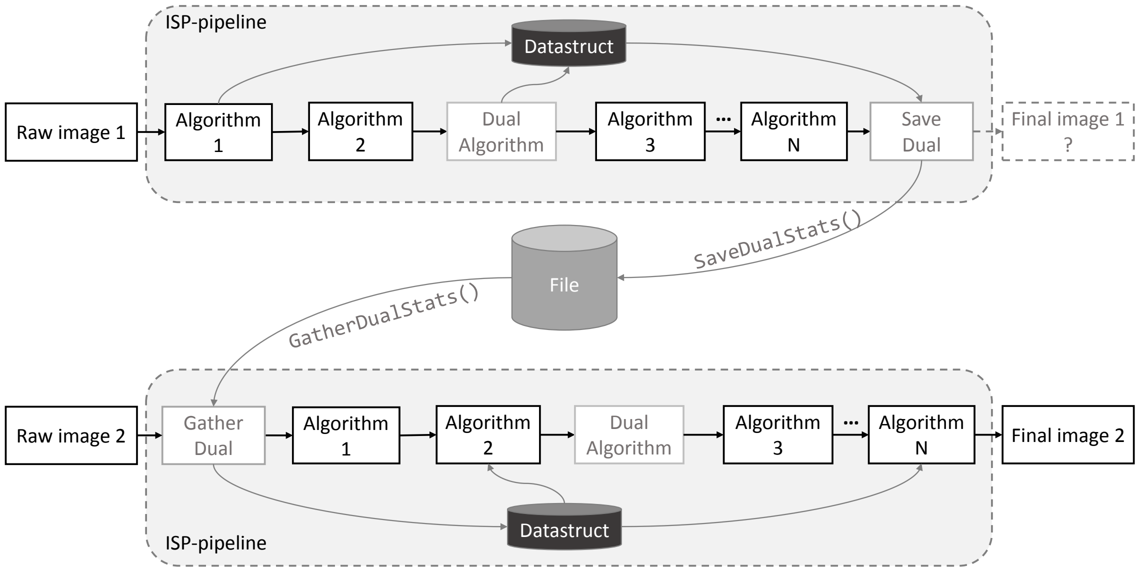

Figure 4.7: Basic structure of dual camera ISP-pipelines.

The dual camera is executed as additional modules that save the dual camera data to the file and gather the saved data of the previous image from the file. As the gather and save modules of the dual camera use the internal datastructure of the framework as a storage space, all the algorithms in the ISP-pipeline have access to the dual camera statistics. Additional dual camera algorithms can be included in the ISP chain.

4.2.2 Dual Camera Architecture in Development Framework

The dual camera system was included in algorithm development framework that previously operated only one camera image processing. The framework contains both user interface tool and command line tool for commanding the image processing. As the framework is focused on algorithm development, it contains multiple alternative image processing pipelines that can be selected for different purposes. The new pipeline for image processing was needed for dual camera algorithm development purposes. The partial class diagram of development framework can be seen in Figure 4.8.

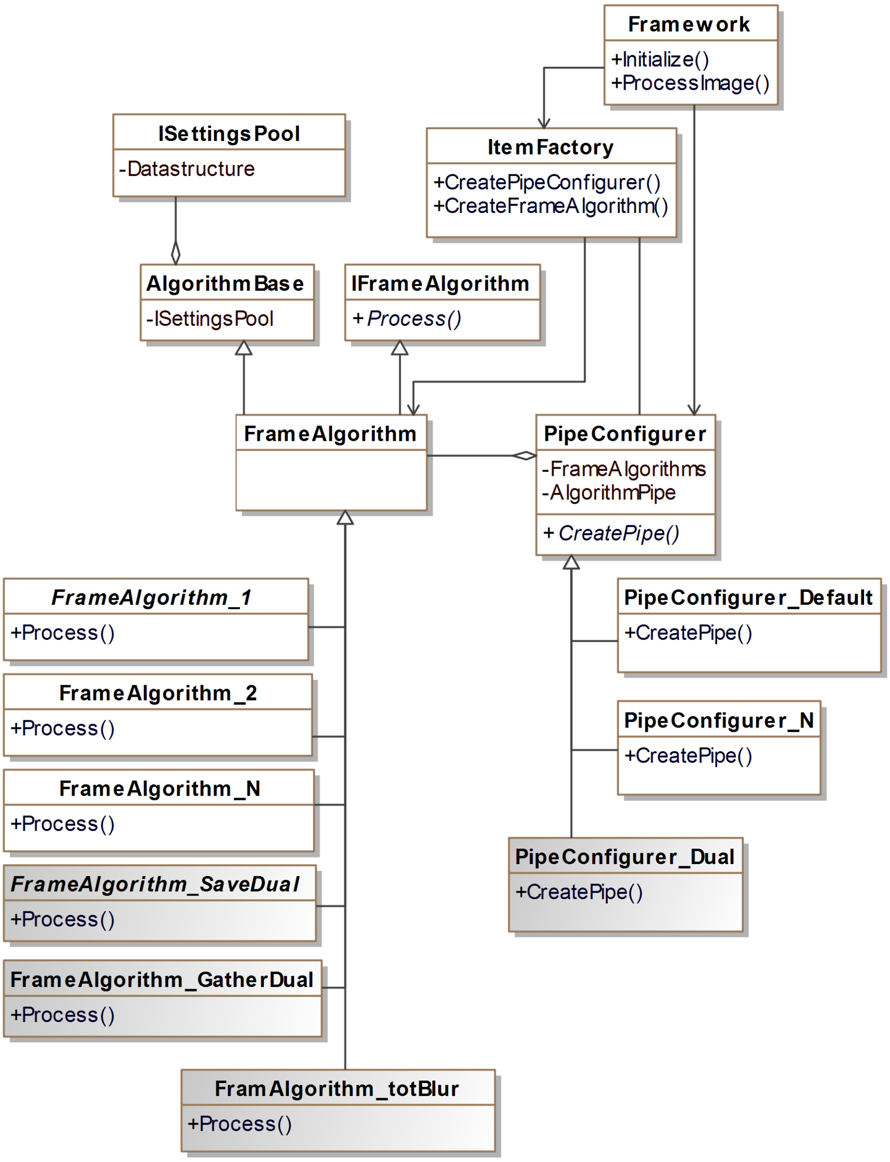

Figure 4.8: Class diagram. New dual camera classes are in gray.

The PipeConfigurer class has the key role in framework pipeline selection. As the user can tell the framework the name of the desired pipeline, the Framework class can initialize the correct pipeline by creating the correct configurer for the pipeline with the help of ItemFactory class. All specific PipeConfigurer classes are subclasses from the parent class PipeConfigurer and the correct PipeConfigurer becomes selected because of virtual pipe creation function. The PipeConfigurer has the knowledge of all algorithms that the selected pipeline contains but also their places in the pipeline. This way the framework is able to process the selected image with correct manner.

Data Format

The dual camera statistics can de saved in a file in XML (Extensible Markup Language) format [30]. The data can be read easily in development phase and the data can be processed by multiple other programs (for example MathWorks® Matlab) for examination and visualization purposes. Listing 4.1 shows an example of dual camera data format. The XML format was chosen because of its generality and readability.

Listing 4.1: Data format example of one image

```
<Image num=1>
        <exposureTime>200</exposureTime>
        <analogGain>200</analogGain>
        <digitalGain>200</digitalGain>
        <totBM>15</totBM>
        <brightnessValues camera=1>
                <bright997>231</bright997>
                <bright970>147</bright970>
                <exposureTime>200</exposureTime>
        </histogramValues>
        <histogramValues camera=2>
                <bright997>252</bright997>
                <bright970>172</bright970>
                <exposureTime>2000</exposureTime>
        </histogramValues>
        <relativeBrightness>10</relativeBrightness>
</Image>
```

Future Work

In the future, the dual camera statistic transfer method should be modified to operate through the shared memory between the images. Currently the XML file plays the role of shared memory, but the datastructure that contains all the image statistics, should be modified to share the statistics between two images. The modification could also made to share statistics from multiple images and not just the two successive ones.

5. AUTOMATIC BLUR DETECTION AND REDUCTION WITH DUAL CAMERA SYSTEM

To proof that the dual camera system can be used to improve imaging quality or speed up the image processing, some algorithms must be developed for the framework. There are many algorithms in literature, which use the dual camera for combinational image formation, but not so many researches where the assistant camera is used as an additional sensor that collects complement data from the surroundings. In this thesis methods for improving existing automatic exposure and white balance algorithms were sought. The aim was to find ways to help exposure time and white balance control to work automatically without users manual input. This chapter focuses on automatic blur detection and reduction, whereas Chapter 6 describes other possible dual camera algorithms and their findings.

5.1 Automatic Blur Detection and Reduction

Motion and hand-shake are common causes of blur in the images. Typically blur troubles photographers in dim and shadowy situations where exposure times are long. As exposure time increases, the movement typically causes trails to captured images. Blur can be reduced in multiple ways and the following sections introduce current blur detection trends and propose methods how to reduce and detect blur by using dual camera systems.

Principles of Blur Detection and Reduction

Usually the amount of blur can be controlled by altering exposure time, but shorter exposure time produces darker images. Short exposure time can be compensated by increasing sensitivity of the sensor (ISO value), but increase in sensitivity easily increases the amount of noise.

Basically blur detection and reduction can be divided into to two different categories: blur detection and reduction prior image capturing, and blur detection and reduction after image capturing. Commonly used term for blur detection and reduction after image capturing is deblurring. As said previously, blur can also be reduced algorithmically by detecting blur and lowering exposure time, but this can be done only up to a certain level, as cameras have limited image capturing abilities. The following sections describe commonly used methods and proposes one dual camera blur detection method.

Blur Reduction Methods

As said previously, blur can be reduced by at least two different methods: by deblurring or by controlling exposure time. Deblurring can make cameras better and remove technical restrictions, while automatic exposure time control can only be made to adapt to the situations better.

Many deblurring methods currently under development are based on detecting blur kernels (or point-spread functions, BSF). Blur kernels are representations of cameras spatial movements and can be estimated or measured during image capturing. Blur kernel methods can be further divided into two distinct categories: blind cases and non-blind cases. In blind cases blur kernel is estimated after image has been captured, but in non-blind cases blur kernel is measured during image capturing. [31]

Unblurred image can be reconstructed from blurred image by deconvolution with known blur kernel. Even though the blur kernel would have been measured, task is not easy, as deconvolution can cause artifacts to the resulting image. In blind deconvolution case, complexity of natural images and large diversity of possible blur kernels make reconstruction difficult. [31]

As interesting as the deblurring and image reconstruction sounds, the main focus of this thesis is the blur detection and reduction in more traditional ways, but with a contribution of dual camera system. The deblurring methods are usually very complex and thus heavy to perform with mobile phones. In turn, simple and easier deblurring methods easily cause artifacts to the images and are not therefore practical.

Blur Characterization

Motion blur can be divided into vertical, horizontal or rotational blur. Both camera object and camera can cause motion blur to the image. Typically motion blur simultaneously in both directions, vertical and horizontal, is caused by hand shake. A typical indicators of horizontal and vertical movement are long smeared vertical and horizontal lines or thick edges, which are result of changed edges of objects. Based on these facts cause of the blur can be estimated.

5.2 Dual Camera Usage in Blur Detection and Reduction

Dual camera system can be used to help blur detection and reduction. When using dual camera system as described in Chapter 4.2, dual camera system allows parameter comparison between the two cameras. One of the goals with dual camera system is to enhance image quality and keep algorithms simple and fast, but at the same time memory consumptions of the algorithms should be kept small. By keeping above criteria in mind, algorithms that suited for dual camera systems were sought.

Marziliano et al. investigated simple blur detection method, which based on calculating blur metric values from thicknesses of objects edges [32]. Because typically blur tends to cause spread in edges of objects, amount of blur can be estimated by calculating medium edge thickness for the image. Also, the blurred areas can be detected by dividing the image to smaller sections. Calculation of edge thickness for each section can provide an estimate of local blur.

Even though Mariziliano's tests with Gaussian blurred and JPEG 2000 compressed images seemed to produce promising results, the method does not seem to perform very well with motion blur that is more random. However, Mariziliano et al. did not test the method with a dual camera system, which can compare blur metric data of two images to each other. For this thesis, edge detection based method was chosen due to its simplicity and feasibility. Besides of better image quality, the goal of developing the algorithm for dual camera system was also to test the framework described in Chapter 4.2.

5.2.1 Edge Detection

Principle idea of the dual camera blur measure algorithm is similar with single camera edge data based blur detection method [32], but instead of evaluating whether one blur measure value is small or big, comparison between two blur measure values could be done. Basically the same scene is captured with two distinct cameras with two distinct settings: one camera takes image with automatic settings (automatic exposure time) and the other with much shorter exposure time. We are then able to estimate the blur of the image taken with automatic settings by comparing the blur values of the two distinct images. If the result is that the image contain blur, we can try avoid it by shortening exposure time and possibly by increasing sensor sensitivity. Principle idea of the algorithm is presented in Figure 5.1.

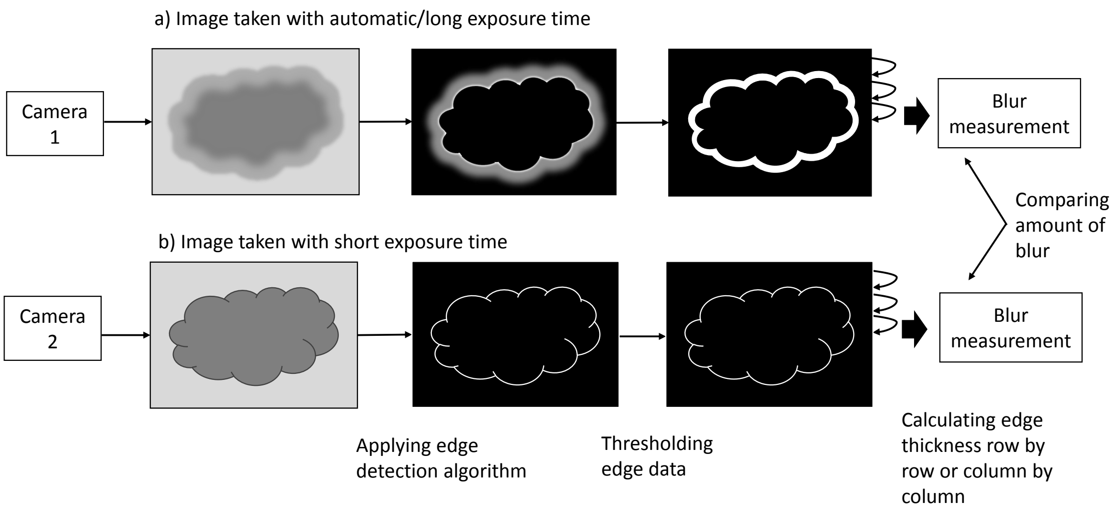

Figure 5.1: Algorithm calculates blur measure for both images. By comparing average edge thickness, algorithm can predict whether images contain blur. Edge detection can be performed only to the luminance (grey scale) image.

Edges can be detected from the image by calculating local gradients within the image. Mathematically, gradient is a vector which points in the direction of the greatest rate of increase. Magnitude of the gradient is the length of gradient vector. Large magnitude refers to large change between the pixels. Thereby large magnitude can be kept as a sign of edge areas and the edges can be distinguished from the image by calculating the magnitude of change between all neighbouring pixels. Gradient vector can be determined as follows [22]:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \tag{5.1}$$

and its magnitude can be calculated by taking square root of sum of its components squares:

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \tag{5.2}$$

Usually this can be simplified by the following approximation, which gives relatively good estimate of the magnitude:

$$|\nabla f| \approx |G_x| + |G_y|. \tag{5.3}$$

Sobel method is one way to estimate the gradients within the image. Sobel operators are quite simple, but at the same time good estimates of the gradient. Sobel operators can be implemented as masks s_x and s_y:

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{5.4}$$

which can be applied to the image by convolution. Mathematical expression to the convolution between mask and every pixel is:

$$G(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} s(m,n) f(x-m, y-n), \tag{5.5}$$

where (x, y) is pixel location in the image and m and n are locations in mask. Function s is the mask, and $f(x-m, y-n)$ is one pixel of the image. Image can be considered as function $f(x, y)$, because basically image is discrete function, which have two parameters x and y. Typically $s(0, 0)$ is the center of mask and thereby upper left corner is $s(-1, -1)$ and lower right corner us $s(1, 1)$. The mask can be applied to the image as in Figure 5.2.

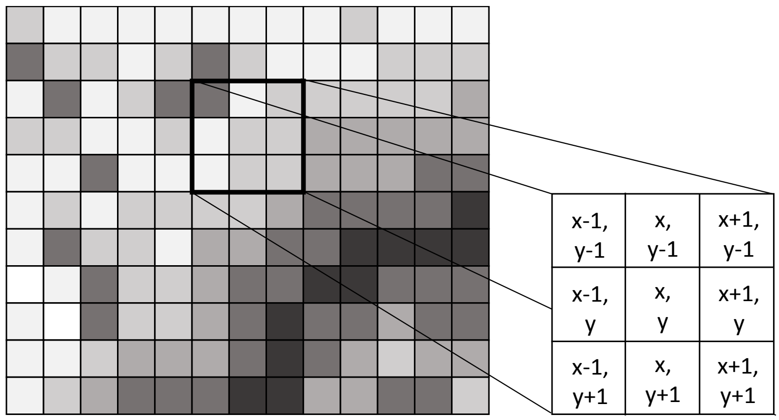

Figure 5.2: Representation of mask kernel.

Resulting matrix G contains information about the edges: when plotted as image it has light areas on edge locations in otherwise dark image. Convolution between mask and pixel can be expressed also as sum of pixel values:

$$\begin{aligned} G_x(x,y) = {} & f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) \\ & - [f(x-1, y-1) + f(x-1, y) f(x-1, y+1)], \end{aligned} \tag{5.6}$$

$$\begin{aligned} G_y(x,y) = {} & f(x-1, y-1) + 2f(x, y-1), f(x+1, y-1) \\ & - [f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)]. \end{aligned} \tag{5.7}$$

When convolutions with s_x and s_y are calculated separately, vertical edges can be separated from horizontal edges. Mask s_x detects only vertical changes, whereas mask s_y detects only horizontal changes. Thereby, G_x contains gradients in vertical direction, whereas G_y contains gradients in horizontal direction. Combined magnitude of the gradients can be calculated with equation 5.2. [22]

Thresholding can be done to the resulting edge image. By thresholding, it can be determined which values are taken account for edge detection. The lighter the pixel in edge-image is the greater the magnitude of difference between the neighbouring pixels is. Even small variation between the neighbouring pixels can cause non-zero values to the edge-image. By eliminating these nearly zero values, the edges can be differentiated better from surrounding artifacts.

By thresholding, image can be transformed into binary image, which contains only white and black pixels. Threshold value determines whether pixel is black or white: all pixel values that are below the threshold get the value of zero and all pixel values above the threshold get the maximum value. For 8 bit image the maximum value is 255.

Because Sobel method is sensitive to all distinct pixels, it usually detects noise in the image as well as the edges of objects. Usually this is not the optimal result. The easiest way to avoid this problem is to use smoothing filter [22]. In this thesis the following gaussian filter was used:

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \tag{5.8}$$

Filtering should be done to the image prior the Sobel edge detection method. The filter matrix 5.8 was used as convolution mask in a similar manner as in equation 5.5. Weights in Gaussian filter mask are typically chosen according to the shape of Gaussian function. Because sum of these weights is not 1, mask is typically divided by the sum of all weights. This is done in order to keep resulting values in same scale. In case of equation 5.8 matrix was divided by 159. [22]

5.2.2 Blur Measuring Method

The next step in blur measure calculation is determining edge thicknesses from the produced edge image. By calculating medium edge thicknesses, the final blur measure can be determined. As can be seen from Figure 5.3, edge image is processed stripe by stripe.

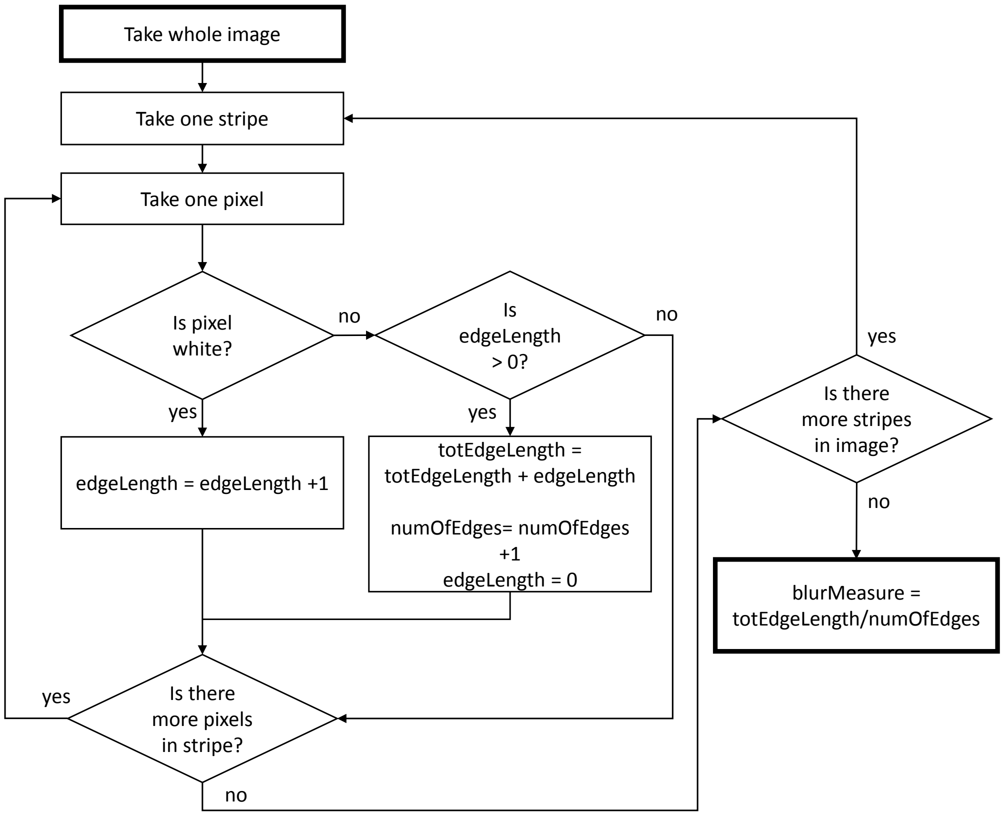

Figure 5.3: Graphic presentation of blur measure calculation.

The thickness of edges is estimated by calculating the amount of successive white pixels. Amount of the edges can be estimated by calculating how many white pixel chains the image contains. The final blur measure is the division between these two values. The amount of successive white pixels can be calculated whether stripe by stripe or column by column. The decision, in which direction the calculations are done, affects whether we calculate the length of vertical or horizontal edges. Large blur measure value in horizontal direction tells about vertical movement, whereas large blur measure value in vertical direction tells about horizontal movement.

Blur measure value cannot always tell the amount of blur reliably by itself, but when blur measures of the two images from the same object can be compared with each other, the results can be more reliable. Also in this way, the out-of-focus parts of the image do not influence on the blur estimation, as both images have same out-of-focus areas. If both images are equally blurred, the motion is not the cause of blur, and the reason for large blur measure value can be found for example from wrong focus.

Dual camera system enables simultaneous capturing of two images from the same object. By taking the other image with shorter exposure time, amount of blur in it can be minimized. The blur measure value of the short exposure time image can then be compared with the image taken with the automatic settings. The blur measure value of the longer exposure time image should be larger in case it contains blur.

Because this method can analyse the amount of blur only from already taken images, the method cannot make the image quality better without taking at least 3 images. The goal of this algorithm is not that, but rather to make image quality better during viewfinder. Adapting the algorithm for viewfinder is not in the context of this thesis, but instead the purpose of this thesis is to proof that the algorithm can work and it is, first and foremost, suitable for dual camera systems.

5.2.3 Algorithm Development with MathWorks® Matlab

Dual camera blur measure algorithm was first tested with MathWorks® Matlab. MathWorks develops mathematical computing software Matlab for scientific and engineering purposes. Matlab was chosen to first algorithm development tool because of its extensive library of mathematical tools. Visualization of edge data and all intermediate steps in form of graphs and images is much easier with Matlab than for example with C++. Figure 5.4 represents the blur detection results accomplished with Matlab. It is good to notice that all edge images are from now on presented as negative images (edges as black, background as white), as they look visually better.

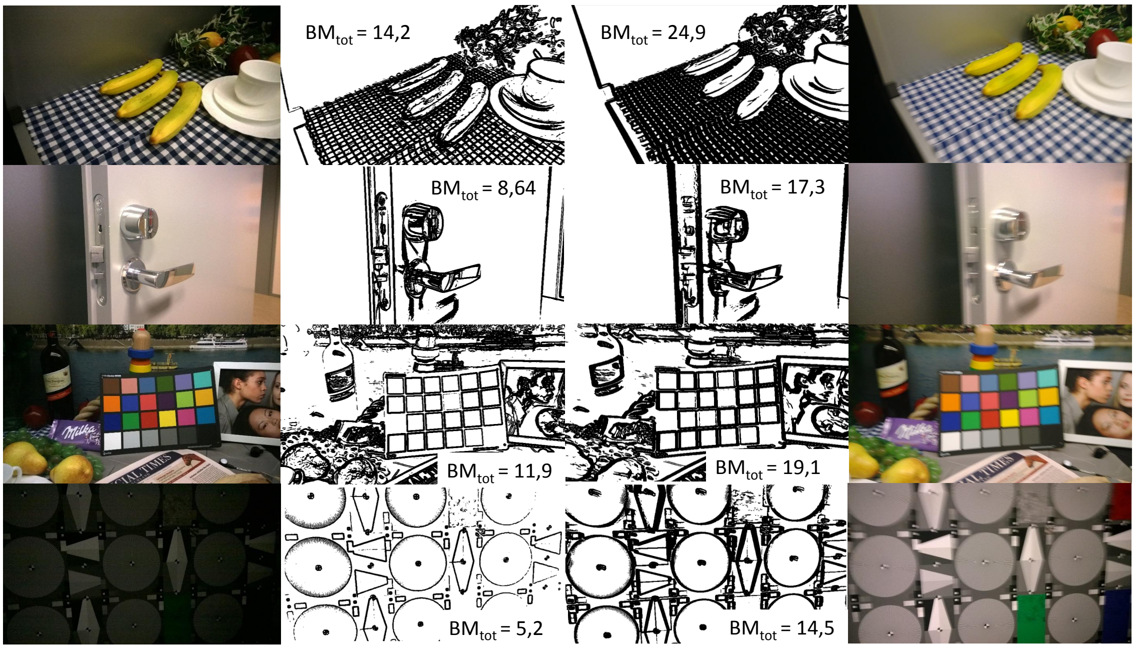

Figure 5.4: Comparison between no-blur and blur images, where BM_{tot} values are calculated with the method described in Figure 5.3. Edge-images are presented as negative images in order to achieve more pleasant look.

As the blur measure values (BM_{tot}) are calculated with the method described in Figure 5.3, BM_{tot} basically tells the average edge thickness as number of pixels.

Because the edge thickness depends on the image size, the BM_{tot} value can be divided by the pixel amount per stripe in order to make the value comparable between images that have variable amount of megapixels. Because that way the amount is typically between 0 and 0.03, the BM_{tot} value can be multiplied with 1000 in order to get the value between 0-30. All the BM_{tot} values in this thesis are calculated in this manner.

As can be seen from the Figure 5.4, blurred images tend to have larger BM_{tot} values than the sharp images. But as the highest and lowest image of Figure 5.4 suggest, the blurriness of the image cannot always be estimated from the BM_{tot} value itself. Instead, the comparison between blurred and sharp image can tell more reliably whether the image contain blur or not.

5.2.4 Framework Implementation

Even though Matlab can process images efficiently, it is essential to shift to develop algorithm in its real environment. In algorithm development framework, it is possible to test algorithm and its performance among other image processing algorithms. One important testable matter is the position of an algorithm in comparison to other algorithms.

The framework implementation of the blur detection algorithm uses RAW formatted image data. As the RAW image pixels do not have all RGB components, one RGBG pixel array was used to calculate one luminosity (grey scale) pixel value. Edge detection was then performed to the luminosity data. The luminosity (Y) values were calculated with the following equation:

$$Y = 0.299R + 0.587G + 0.114B, \tag{5.9}$$

where R and B are directly the values of red and blue pixels, whereas G is the average between two distinct green components. The coefficients for RGB color components are based on the proportion in which the human eyes are sensitive to them. [11]

The blur detection algorithm was placed in one C++ class, that is implemented as movable subclass of Framealgorithm and it thus inherits all functionality that the base class has in a similar manner with all other algorithms. Because of the similarity with the base class, the blur detection algorithm can easily be defined in one specific place on the pipeline and the framework can then use the services that the algorithm provides without knowing the type of algorithm. The algorithm can easily be moved by changing place of its definition. Structure of the framework can be further read from Section 4.2.

The framework implemented algorithm did not seem to produce as positive results as the Matlab version of the algorithm, so it was necessary to investigate what caused the difference. Because the Matlab version of the algorithm used JPEG images and the framework version uses RAW images, other algorithms must cause the difference between the results. As the JPEG images are gone through several processing phases, and the quality and sharpness is thus enhanced, they contain less artefacts than the original image and the sharpness can support edge calculations.

In order to detect how much the placement of the algorithm can affect to the edge detection, the blur detection algorithm was placed before and after noise reduction algorithms. The effects of algorithm placement can be demonstrated by two blur and non-blur image pairs. The best possible image that edge detection algorithm was able to produce in both places can be seen in Figure 5.5.

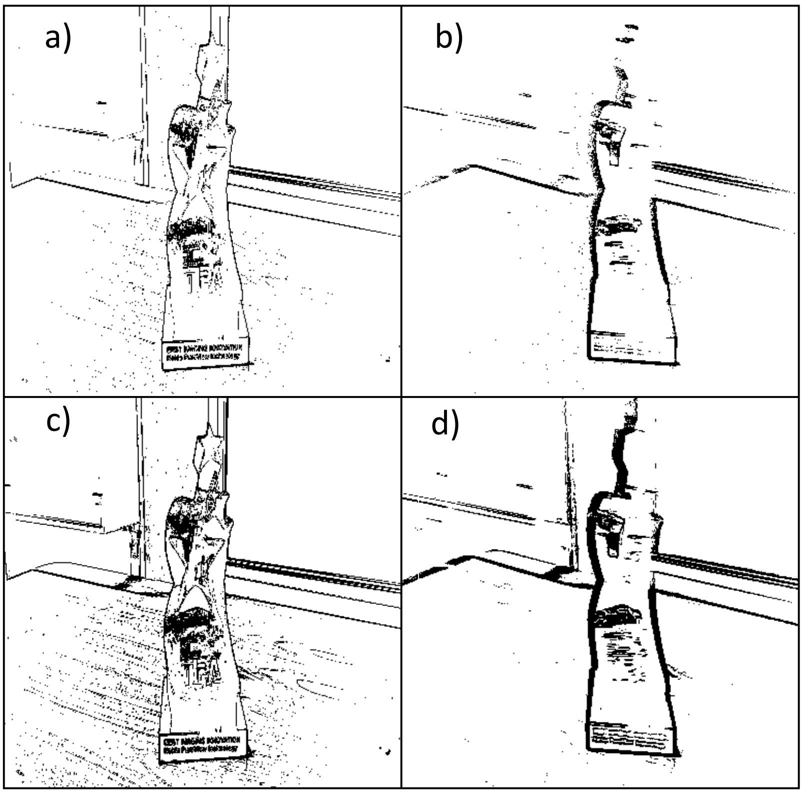

Figure 5.5: Edges visualized when blur detection algorithm at the beginning of pipeline a) no-blur image (BM_{tot} = 5,93) and b) blur image (BM_{tot} = 11,5). Edges visualized when blur detection algorithm after noise reduction c) no-blur image and (BM_{tot} = 6,69) d) blur image (BM_{tot} = 16,22).

As the Figure 5.5 shows, the placement of algorithm has effect on edge data. Because the edges of image 5.5b are more scattered than the edges of image 5.5d, the total blur measure (BM_{tot}) is lower in 5.5b than in 5.5d. It seems that the blur detection algorithm performs better at the end of the pipeline than in the beginning.

5.2.5 Results

Even though the algorithm is able to detect blur, it seems to be able to detect blur only up to a certain level. When amount of blur increases in images, edges become unnoticeable and average thickness of edges decreases as the algorithm detects less edges. At certain blur level, all details disappear and the algorithm cannot detect the edges any more. The algorithm also does not seem to suit well for images which contain only soft edges and curves or for images that does not contain noticeably details or the details are too small. But in cases like Figure 5.6 the algorithm might give valuable information.

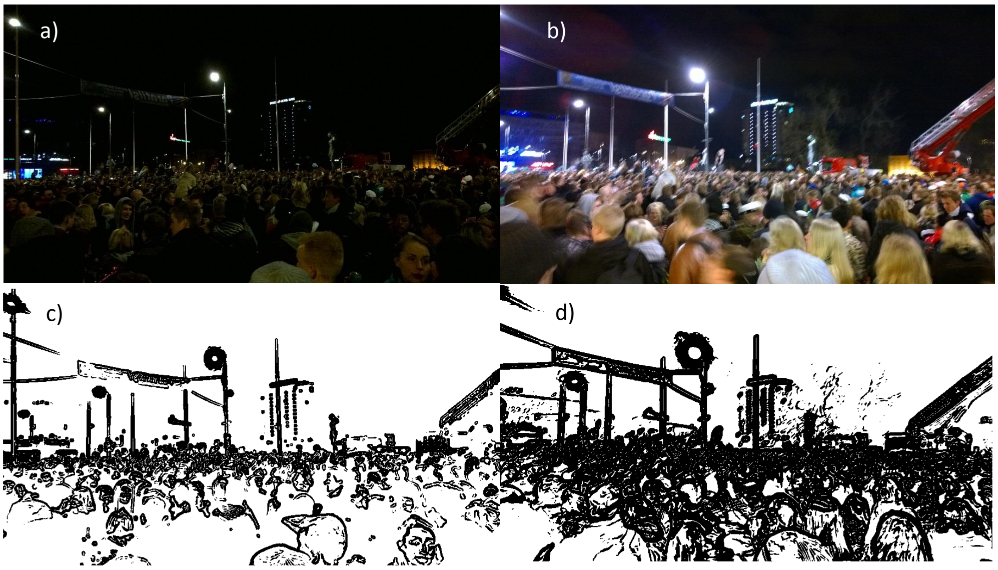

Figure 5.6: a) Image taken with short exposure time, b) image taken with automatic exposure time. Lower images contain only the edge data from upper images.

The left side image is not perfect image as it is too dark, but at least it is better than the blurred right image. In this case the blur detection algorithm would have noticed blurriness of the image that is taken with automatic settings and would thus have shortened the exposure time of the final image. On the contrary against general asumption, the left image does not contain blur even though it contains less details.

In summary: even though the algorithm can detect blur, it does not perform well in situations where:

- amount of blur is so large, that details disappear
- image does not contain edges or large enough variations
- blur difference is small between the images in comparison

- images contain noise, which can be detected as an edge
- short exposure time image is too dark.

In order to meet the previous challenges, the blur measure algorithm should be improved by combining it with another blur detection methods or by improving the algorithm itself. The algorithm can be enhanced for example by comparing the details in images as the blurred images contain less details. Currently the algorithm is not as efficient as it should be and it does not detect blur certainly in every situation. When finalized, it is a good addition to other blur detection methods.

5.2.6 Future Work

Broad investigation of the blur detection algorithms abilities should be done as a next step in order to determine whether the algorithm can function as part of the dual camera image processing pipeline. The most critical question about the algorithm is its limits.

As the blur reduces details in image, certain level of blurriness can make all details in the image disappear and thus the algorithm cannot detect the edges anymore. It needs to be tested, whether the algorithm can be combined with detail detection algorithm. If the image is so blurred that it does not contain any detail at all, the detail detection algorithm can notice those cases and tell that the image is so blurred that it does not contain any details at all.

Another matter to investigate is the best exposure time for the assistant camera. In order to detect blur in primary cameras image, the assistant camera cannot contain blur itself. Hence it is crucial to be able to take the assistant image with correct exposure time.

Also the benefits of dual camera implementation should be investigated. As the algorithm can predict amount of blur from just one image, it is wise to test how large benefit the assistant camera can provide. This is also interesting, because currently majority of mobile phones does contain only one camera. If it turns out that the dual camera benefits are marginal, the algorithm can also be used for a single camera systems.

6. OTHER DUAL CAMERA ALGORITHMS

Other methods for improving imaging algorithms were also researched. Automatic exposure time control can be improved by collecting additional histogram data with the assistant camera. Also, the scene illuminant can be determined by comparing sensor data from the cameras. The knowledge about the illuminant can give valuable information when determining the correct white balance. These methods are further described in Sections 6.1-6.2.

6.1 Dual Camera Automatic Exposure Control

Automatic exposure control can be one possible application area for the dual camera system. As the automatic exposure control has limited opportunities to determine the right exposure level, another camera could open possibilities to make the current algorithms better.

Automatic exposure control is closely linked to image histogram, as the image histogram reveals the overall luminosity of the image. A histogram is good photographic tool as it can tell whether the image is correctly exposed. Thus it can be used to estimate correct exposure time. Examples of typical histograms are shown in Figure 6.1.

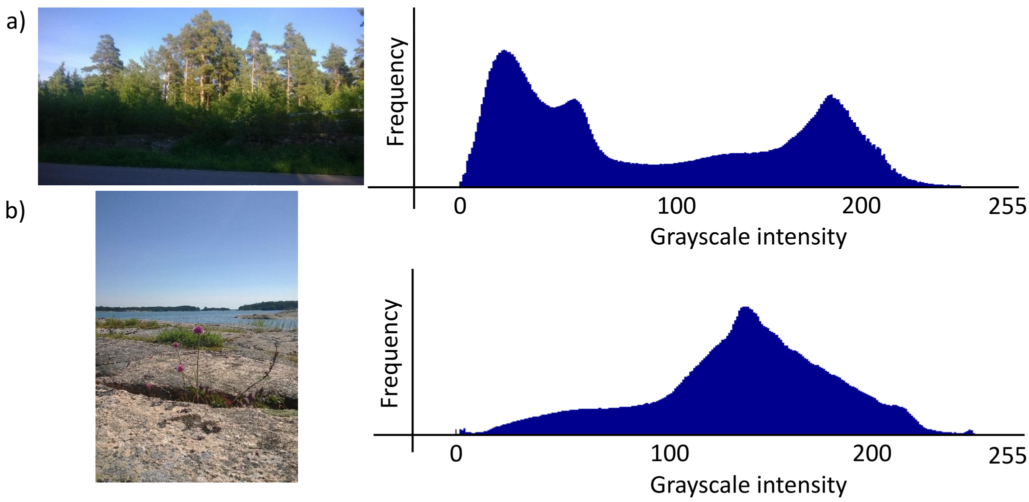

Figure 6.1: Image histograms: a) Wide intensity range image, where tones are divided into dark and light section, b) Wide intensity range image, where tones are concentrated to midsection.

Basically the histogram tells how the pixel luminosities are divided between the luminous range. Luminous range for 8-bit image data is 0-255. Dark tones in image can be seen in left side of the histogram, as the bright tones can be seen in the right side of the histogram. Images that contain extreme tones are typically two peaked histograms with wide range of different intensities. [33] Figure 6.1a represents situation where the scene is divided into light and dark areas, whereas Figure 6.1b represents situation where the image contain mostly midtones.

As the image can contain only certain number of different colors, exposure control must determine where it preforms the clipping of the histogram. After clipping, the tones brighter than the high clipping point are represented as the maximum value and tones darker than the low clipping point are represented as the minimum value. Situations where the histogram contains long tail in a bright end are typically difficult in an automatic exposure controls point of view, as the algorithm has to decide whether the bright tones are worth saving. If the algorithm decides to clip the bright tones, image may loose important information for example from tone variations of the sky. On the contrary, if the algorithm decides to save the bright tones, the overall look of the image can easily be too dark. The correct intensity distribution depends highly on situation, and sometimes it is just a matter of opinion.

Exposure Time with Dual Camera System

Because the main camera produces viewfinder frames to the screen constantly during image capturing, sometimes it is blind to sections that are clipped from the histogram. As the main camera has to show all viewfinder frames that it produces to the camera user, it cannot use trial-and-error method in order to select best possible histogram for the images. But, if we add another camera next to the main camera, more opportunities will open up. The additional assistant camera can take any kind of images we want it to take, as the images from assistant camera are not visible to the camera user. If we modify the assistant camera to take darker and underexposed images, we can have additional information from the scene.

There is one special situation, where the automatic exposure control algorithm encounters problems. In case the histogram contains long tail in bright tones side and there are certain amount of tones that are near the white point, the automatic exposure control algorithm needs to decide whether it saves the light tones, and thus makes the majority of the tones dark, or moves the average brightness towards the midtones. In former case the image looks easily darker than the eyes actually experienced it, and the light tones are preserved unnecessarily. Histograms of this kind of situation are presented in Figure 6.2.

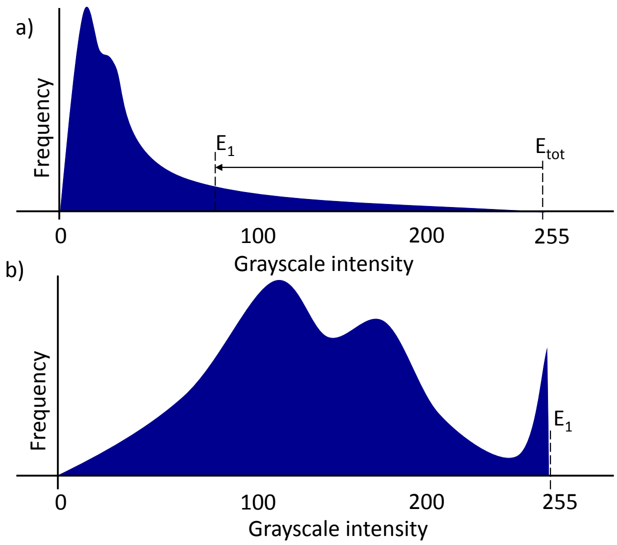

Figure 6.2: Image histograms with: a) long tail in right, b) right end clipped.

As the main camera cannot always predict how far the brightest tones would be, the algorithm can do the clipping from wrong point. In case the bright spot is not as long away from the midtones as it was in case of Figure 6.2, the clipping can be done too close from the midtones and important part of the bright tones can be lost. This case is represented in Figure 6.3.

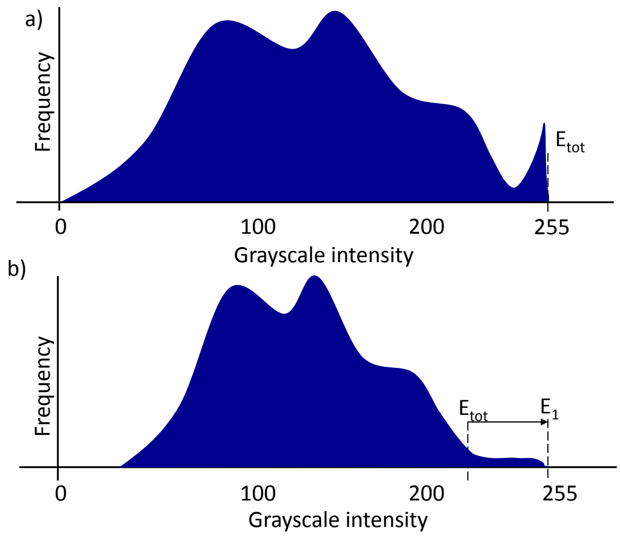

Figure 6.3: Image histograms with: a) right end clipped, b) tail in right.

With the help of assistant camera, it is possible to determine how far away the bright tail is from the average tones. If the assistant camera captures image, where the whole histogram is presented, and the main camera captures image, where the average tone is close the midtones, the length of the tail in primary camera image can be calculated with the knowledge of both images exposure times.

Practical Example

As previously said, the dual camera system is suitable for two types of situations concerning automatic exposure time control: firstly, in situations where the bright tone tail is long and histogram clipping is essential, and secondly, in situations where the bright tone tail is so short that it can be saved. Figure 6.4 represents the situation where the bright tone tail is so long that it is not wise to try save it.

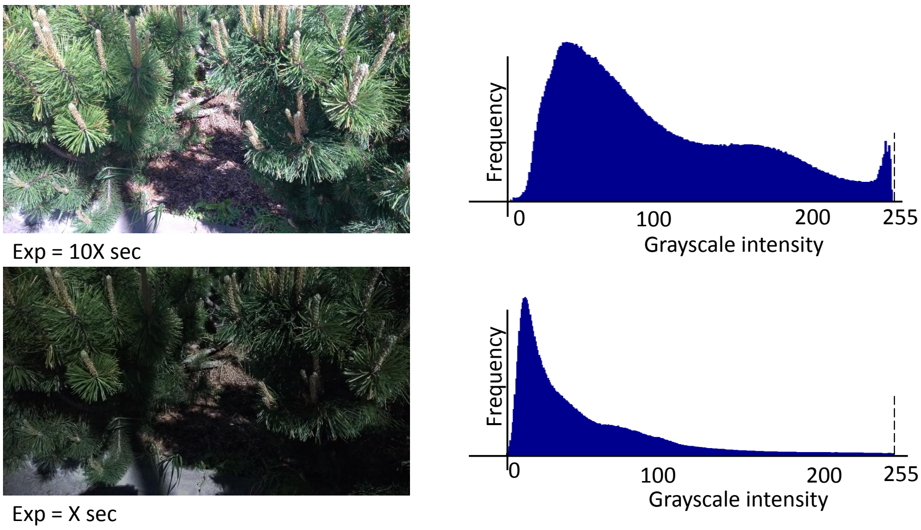

Figure 6.4: Saving the light tones is not always necessary, as it can make the whole image too dark.

As the exposure time of the brighter image is 10 times longer than the exposure time of the darker image, the bright tone tail of the brighter image would also extend 10 times farther if it was not clipped. If the maximum value of 255 would not limit the histogram, the maximum value in the brighter image would have value of 2550. As the maximum value would be so far and as the other algorithms have suggested that the brighter image is better, it is not wise to change the exposure time so dramatically.

Figure 6.5 represents the situation, where the bright tone tail is not long. In this case the exposure time of the brighter image is only 1.5 times longer than the exposure time of the darker image. Now the brightest spot would be at value 383, if it was not clipped. As the yellow flower is in major role on the image and the automatic exposure control seems to loose its contrast variations, shorter exposure time should be considered.

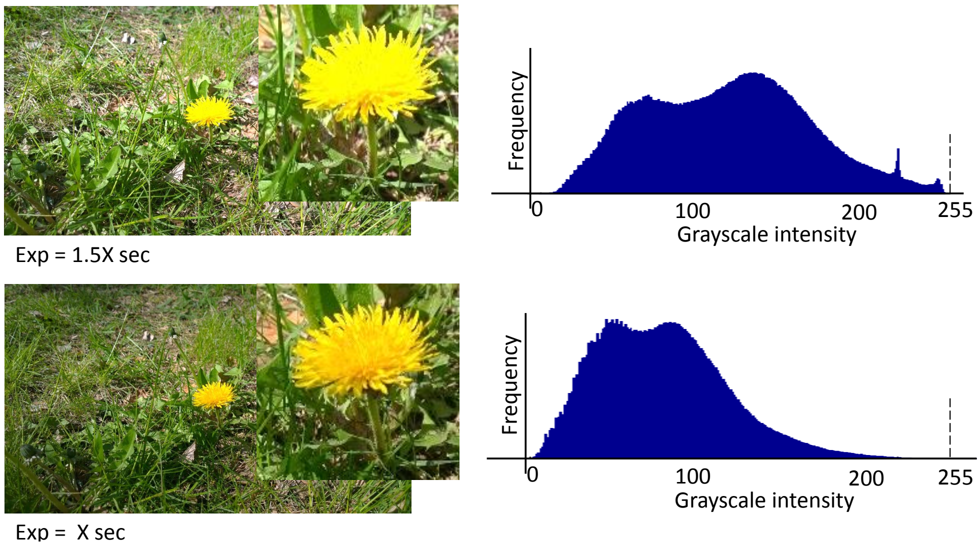

Figure 6.5: Saving of the light tones can be important, as the details can disappear if the histogram is clipped. In this case the yellow flower will loose details, if the histogram is clipped from the right tail.

The histogram parameters can be easily transferred from one image to other in Algorithm development framework. As some histogram markers are already collected from the images, only task in this case is the data transfer from the assistant camera to the primary camera. This can be done already with the Algorithm development framework by passing the 99,7% and 97% bright tone frequencies to the primary camera. Thus the automatic exposure time control can take the additional histogram data into consideration, when deciding the best exposure time.

Future Work

In the future the existing automatic exposure control algorithm should be modified to use the additional dual camera metrics. The limits, which define whether the bright tones should be saved or disregarded, must be determined in order to make the additional histogram data useful. Additional parameters, from which the saving or disregarding decision could depend, must be also investigated.

6.2 Determining the Illuminant with Dual Camera System

One of the problems in image processing is the identification of the illuminant. Determining the color of the illumination is crucial for correct white balance as the human eye adapts to different situations.

The idea behind determining the lightning condition with dual camera system lies in the two cameras distinct sensor response. The two cameras in dual camera can be manipulated to produce different kind of images in purpose or the two cameras can produce varying images automatically as the two camera sensors are never totally identical.

One way to determine the difference between the sensors is to take sample shots of colors and then compare the images together. The Machbeth Color Checker [34] contains good samples of natural colors and therefore its colors can be used in color tests. Example of The Machbeth Color Checker colors are represented in Figure 6.6.

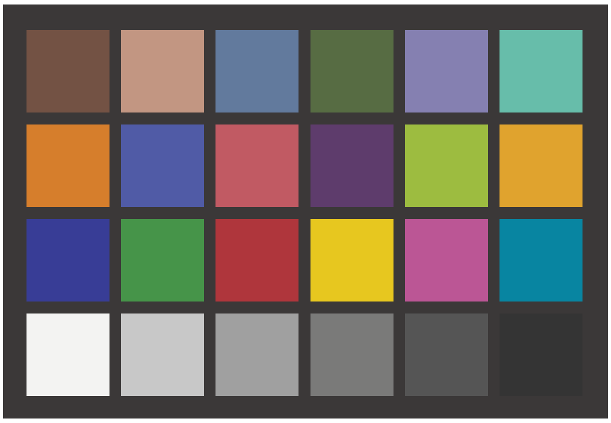

Figure 6.6: The Machbeth Color Checker colors. [34]

Instead of using the ordinary color checker, it is easier to stimulate the color checker colors and take individual images of every color. This way the shading is not needed to be taken account when comparing two colors together. Shading cannot be totally forgotten as all the images contain shading in corners and the shading can be different between distinct cameras. Good way to avoid the shading and retrieve correct data is to use only the pixels in the center of image and calculate average of them.

The tests were done in three different lightning conditions: Halogen, D65 (daylight) and F12 (fluorescent). All three illuminants are CIE (Commission Internationale de l´Eclairage) standard illuminants. D65 is the standard daylight illuminant, which average color temperature is approximately 6500 K. F12 is standard for narrow band fluorescent light. [35]

The concepts seems to be valid, as the lightning conditions seems to separate from each other when tested with right set of cameras. Figure 6.7 A represents the result of one camera pair, where ΔB is the detected difference for blue components between the two cameras and ΔR is the same difference but for red components. Figure 6.7 B represent the same parameters for another camera pair.

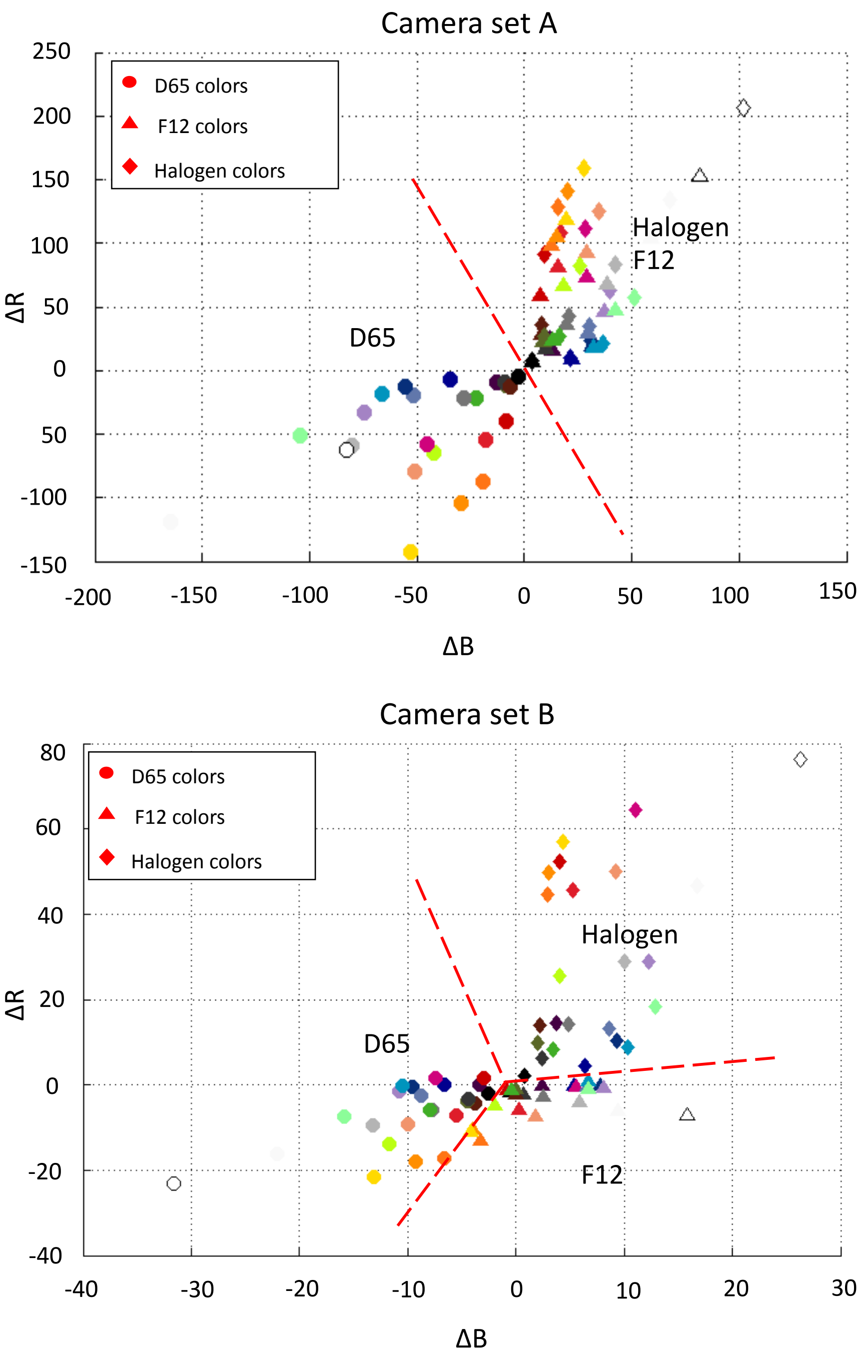

Figure 6.7: Two distinct cameras can detect color differently in different lightning conditions. Camera set A contained two distinct cameras, when compared to camera set B. The cameras differed also inside the pairs. All the markers represent the difference of two cameras in a set. The circles, triangles and diamonds represent the same 24 Machbeth colors, but in different lightnings.

As can be seen from the Figure 6.7, camera set A is not capable of separating halogen lightning from F12 lightning as well as camera set B, but still it can separate D65 from the other two light sources. Camera set B seems to be able to separate all three light sources from each other. This suggests that the cameras should be selected very carefully in order to make this algorithm work. It is good to notice that the sources separate only slightly. If the scene would have contained only few different colors, separation would not be so clear.

Calculations were done to the single color images, by averaging the RGB components. Even though the result seems promising, there are still many things that are needed to be taken in consideration:

- the two cameras need to be significantly different
- scene must contain large variation of colors
- other algorithms would have more limitations, as the two images would look very different
- there are also many other lightnings than D65, F12 and halogen
- lightning can be combination of many sources.

Before making more assumptions about the usefulness of the algorithm, it should be investigated more. Further investigation of the algorithm was left for the future, as the algorithm requires lot from the both cameras. Gain that the algorithm might bring to the dual camera system at its best seemed too minimal and hard to achieve. The decision of not to research the algorithm further seems wise, as the research at this point should focus on algorithms that might produce more gain for the image quality and do not cause restrictions to other dual camera algorithms and features, like HDR images.

7. CONCLUSIONS

In this thesis dual camera algorithms were researched with an aim to proof that dual camera concept could improve images quality. Also an existing algorithm development framework was modified for dual camera systems by adding a data transfer method to an existing PC application. This method can process a RAW formatted digital images in a similar manner as a mobile phone camera image signal processor.

After the framework was finalized, it was tested with few prospective dual camera algorithms. The researched algorithms focused on improving existing automatic exposure control and white balance algorithms. It turned out that the dual camera can be used to collect additional information about the histogram of the scene and this information can then be used as a help when deciding the best exposure time. The blur detection method is also possible to be accomplished with dual camera system. The blur detection method is based on calculating the edge thickness data that can then be compared between the two dual camera images. The white balance predictions can be enhanced with dual camera system by using the distinct sensors of the two cameras to predict the illuminant source.

As the goal of this thesis was to proof that the dual camera concept can improve image quality, the goal was achieved. The dual camera concept seems valid, but the dual camera algorithms must be researched further. Even though the dual camera algorithms provide useful data to other algorithms, e.g. exposure control and white balance algorithm, the dual camera concept cannot be said to be the optimal solution for improving the algorithmic defects. These same improvements could also be accomplished by single camera systems that are nowadays more general.

In the future, it should be decided, whether to continue the dual camera algorithm development or to try to adapt these researched algorithms to a single camera system. Especially the researched blur detection method could be implemented to a single camera system, although, without the blur measure comparison between two images. It should also be researched whether the image comparison can be accomplished with a single camera system.

As the researched dual camera algorithms need data from multiple distinct images, the single camera system cannot use the dual camera algorithms directly. One solution is to make the single camera system to take multiple images in a row, as if it would be dual camera system. This causes problems to the viewfinder, which should show live preview of the scene, but it is worth to research whether the problem could be solved.

If the dual camera algorithm research is decided to be continue, at least the blur detection algorithm needs a lot of improvement. Currently the algorithm is not efficient enough to be implemented in a mobile phone camera. As the algorithm has limited ability to detect blur, the algorithm should be developed to detect blur also from the images that are so blurred that all details have disappeared. The dual camera illuminant determination algorithm should also be tested further, as it is still unclear in which kind of situations the method can operate.

REFERENCES

[1] H. J. Trussell, E. Saber, and M. Vrhel, "Color Image Processing: Basics and Special Issue Overview," *IEEE Signal Processing Magazine*, vol. 22, no. 1, January 2005.

[2] G. Sharma, "Color Fundamentals for Digital Imaging," in *Digital Color Imaging Handbook*, G. Sharma and R. Bala, Eds. CRC Press, 2002.

[3] M. Vrhel, E. Saber, and H. J. Trussell, "Color Image Generation and Display Technologies," *IEEE Signal Processing Magazine*, vol. 22, no. 1, January 2005.

[4] Toshiba Corp., "BSI (Backside Illumination), [WWW]," http://www.semicon.toshiba.co.jp/eng/product/sensor/cmos/bsi/index.html, [Read 5.6.2014].

[5] B. E. Bayer, "Color Imaging Array," 1976, US Patent 3,971,065.

[6] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew, "Color image processing pipeline," *Signal Processing Magazine, IEEE*, vol. 22, no. 1, Jan 2005.

[7] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, 2005.

[8] K. J. Kabza, "Evolution of the Live Preview in Digital Photography, [WWW]," http://cameras.reviewed.com/news/evolution-of-the-live-preview-in-digital-photography, [Read 2.6.2014].

[9] W.-C. Kao, S.-H. Wang, L.-Y. Chen, and S.-Y. Lin, "Design Considerations of Color Image Processing Pipeline for Digital Cameras," *Consumer Electronics, IEEE Transactions on*, vol. 52, no. 4, pp. 1144–1152, 2006.

[10] Cambridge in Colour: A Learning Community for Photographers, "Dynamic Range in Digital Photography, [WWW]," http://www.cambridgeincolour.com/tutorials/dynamic-range.htm, [Read 11.6.2014].

[11] J. C. Russ, "Acquiring Images," in *The Image Processing Hanbook*. CRC Press, 2011.

[12] Gadgets to Use, "What are Auto Focus or Fixed Focus Smart Phone Camera - Which One is Better and Why, [WWW]," http://gadgetstouse.com/gadget-tech/auto-focus-vs-fixed-focus-camera-comparison/17149, [Read 12.6.2014].

[13] Cambridge in Colour: A Learning Community for Photographers, "Understanding Camera Lenses, [WWW]," http://www.cambridgeincolour.com/tutorials/camera-lenses.htm, [Read 10.6.2014].

[14] K. Parulski, "Color Image Processing for Digital Cameras," in *Digital Color Imaging Handbook*, G. Sharma and R. Bala, Eds. CRC Press, 2002.

[15] R. W. Schafer and R. M. Mersereau, "Demosaicking: Color Filter Array Interpolation," *Signal Processing Magazine, IEEE*, vol. 22, no. 1, 2005.

[16] Cambridge in Colour: A Learning Community for Photographers, "Understanding Gamma Correction, [WWW]," http://www.cambridgeincolour.com/tutorials/gamma-correction.htm, [Read 13.5.2014].

[17] N. Bellas and A. Yanof, "An Image Processing Pipeline with Digital Compensation of Low Cost Optics for Mobile Telephony," in *Multimedia and Expo, 2006 IEEE International Conference on.* IEEE, 2006, pp. 1249–1252.

[18] R. Raskar, "Computational Photography," in *Computational Optical Sensing and Imaging.* Optical Society of America, 2009.

[19] Toshiba Corp., "Toshiba Ships Industy's First Dual Camera Module Able to Simultaneously Output Rrecorded Images and Depth Data, [WWW]," http://www.toshiba.com/taec/news/press_releases/2014/assp_14_698.jsp, [Read 16.4.2014].

[20] GSMArena, "HTC One (M8) review: One and only, [WWW]," //http://www.gsmarena.com/htc_one_m8-review-1062p8.php, [Read 28.5.2014].

[21] CNET, "Corephotonics' Dual-camera Tech Will Change Smartphone imaging, [WWW]," http://www.cnet.com/news/corephotonics-dual-camera-tech-will-change-smartphone-imaging/, [Read 3.6.2014].

[22] R. Jain, R. Kasturi, and B. Schunck, "Machine Vision," *McGrawhill Inc., New York*, 2005.

[23] Cambridge in Colour: A Learning Community for Photographers , "High Dynamic Range Photography, [WWW]," http://www.cambridgeincolour.com/tutorials/high-dynamic-range.htm, [Read 2.6.2014].

[24] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital Photography with Flash and No-flash Image Pairs," *ACM Trans. Graph.*, vol. 23, no. 3, Aug. 2004.

[25] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image Deblurring with Blurred/Noisy Image Pairs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007.

[26] J. Chen, L. Yuan, C.-K. Tang, and L. Quan, "Robust Dual Motion Deblurring," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[27] B. Wo, "Color Constancy for Stereo Imaging," M.S. Thesis, University of Dayton, 2012.

[28] Nokia Corporation, ST Microelectronics NV, "SMIA 1.0 Part 2: CCP2 Specification, [WWW]," http://www.sunex.com/SIMA/SMIA_CCP2_specification_1.0.pdf, [Read 2.5.2014].

[29] J.-M. Maatta, J. Vanne, T. D. Hamalainen, and J. Nikkanen, "Generic Software Framework for a Line-buffer-based Image Processing Pipeline," *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 3, pp. 1442–1449, 2011.

[30] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML)," *W3C Recommendation*, 2006.

[31] Q. Shan, J. Jia, and A. Agarwala, "High-quality Motion Deblurring from a Single Image," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008.

[32] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A No-reference Perceptual Blur Metric," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, 2002.

[33] Cambridge in Colour: A Learning Community for Photographers , "Camera Histograms: Tones & Contrast, [WWW]," http://www.cambridgeincolour.com/tutorials/histograms1.htm, [Read 29.5.2014].

[34] X-Rite Inc., "ColorChecher Classic, [WWW]," http://xritephoto.com/ph_product_overview.aspx?ID=1192, [Read 1.6.2014].

[35] J. Schanda, "CIE Colorimetry," in *Colorimetry: Understanding the CIE system.* John Wiley & Sons, 2007, pp. 37–46.