TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MARKUS LAINE

EVALUATION OF FOCUS CURVES BASED ON GOODNESS CRITERIA

Master's thesis

Examiner: Associate University Lecturer Heikki Huttunen
Examiner and topic approved by the Faculty Council of the Faculty of Computing and Electrical Engineering on 6 March 2013.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
LAINE, MARKUS: Evaluation of focus curves based on goodness criteria
Master of Science Thesis, 72 pages
October 2013
Major: Signal Processing
Examiner: Professor Heikki Huttunen
Keywords: Autofocus, focus curve, goodness criteria, image processing, pipeline, smartphone, camera

In smartphones there are restrictions for imaging systems like computation capabilities, power and physical size, which have caused usage of relatively low quality camera sensors and modules. To achieve acceptable image quality, low quality images are enhanced and processed with many different algorithms. These algorithms can be executed in different order in the imaging pipeline. Poor order may cause processing blocks executed later to create something undesired to images while in optimal order each processing block should enhance image quality. One very important block is autofocus (AF) statistics calculation block. Poor AF statistics may cause AF algorithm to choose incorrect focus point, which may cause image to become blurred. In addition of producing low quality images, blurry images may cause big problems for later processing blocks in imaging pipeline.

This thesis is done for Intel Finland Oy. The thesis is about studying how much different execution orders of processing blocks affect to accuracy of AF algorithm. To study the subject images were captured from same scene with different focus lens positions and evaluated how easily some AF algorithm could find the best focus point. For that task single statistic was calculated for each differently focused image, which allowed plotting of focus curve. As statistic average amount of edge content in image was used. To calculate it images were filtered with high pass filter. This kind of filtering discards low frequency information and takes to account higher frequency content, which contains mostly information of edges. For evaluating focus curves goodness criteria were developed. Goodness criteria represent the capability of recognizing spike, where image is correctly focused, from focus curve.

In this study it was noticed that decreasing noise made task of AF algorithm significantly easier. Also reasonable downscaling improved situation for AF algorithm, but it also caused time to time something unexpected behavior. On the other hand color correction is something that should be done after AF statistics calculation, because it emphasizes noise.

PREFACE

This Master of Science thesis is done for Intel Finland Oy at Tampere site. The supervisor of this work is Dr. Jarno Nikkanen from Intel. The examiner of this work is Professor Heikki Huttunen from Tampere University of Technology.

I would like to thank Jarno Nikkanen for guidance, advice and motivation in this interesting topic. I would also like to thank my examiner Heikki Huttunen for hints and tips. I thank also my co-workers in Intel, especially Evgeny Krestyannikov, Tomi Heinonen and Perttu Kuronen.

20.12.2013, Tampere

Markus Laine

CONTENTS

ABBREVATIONS AND NOTATIONS

λ	Wavelength
θ	Angle
c	The speed of light in air, $2.998*10^8$ m/s
C	Color component
d	Distance
E	Energy
f	Frequency or focal length
$\underline{F}$	Vector of focus values
h	Planck constant, $6.626*10^{-34}$ J*s, $4.136*10^{15}$ eV*s
I	Raw image
n	Refraction index of material, neighborhood or integer value
P	Power
R	Radius
s, t	Coordinates of neighborhood
v	Speed of light in material
w	Weight
x, y	Coordinates of image

AF	Autofocus
AE	Auto exposure
AD	Analog to digital
AWB	Auto white balance
B	Blue color component of image
BL	Black level
CCD	Charge coupled device
CCM	Color correction matrix
CFA	Color filter array
CIE	Commission Internationale de l'Eclairage, The International Commission on Illumination
CMOS	Complementary metal-oxide semiconductor
FM	Frequency modulation
G	Green color component of image
Gb	Green color component next to blue color component of camera sensor
Gr	Green color component next to red color component of camera sensor
GrRBGb	Image consisting of two green, red and blue components
GW	Gray World AWB algorithm
HVS	Human visual system

IEC	International Electrotechnical Commission
IR filter	Infrared filter
ISO	International Organization for Standardization
ISO speed	Amount of amplification used to amplify the current from sensor in digital cameras
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LSC	Lens shading correction
R	Red color component of image
RAW	Minimally processed raw sensor data consisting of components Gr, R, B and Gb
RGB	Image consisting of red, green and blue color components
SNR	Signal to noise ratio
sRGB	Standard RGB colors pace
VF	View finder

1 INTRODUCTION

Consumers are becoming more and more familiar with modern digital devices. This means that they have become also pickier with products. Proof of this can be seen from numbers of sold tablets and smartphones. Smartphone is common name used in marketing mobile phones with higher capability of computing and connectivity. Consumers insist more and better features from smartphones. Digital camera is one very good example of that. Probably every new smartphone model already includes pretty decent digital camera. Increasing processing power of processors and cameras enables capturing higher and higher quality images. The quality gap between lower-end compact digital cameras designed just for capturing images and high-end smartphones is diminishing all the time. Competition in the area of smartphones has led to situation where many companies are trying to stand out with better cameras.

Smartphones, as many other mobile devices, have limitations like price, physical size of device, battery life and computational power. This has led to the situation, in which the quality of camera module can be relatively low. However the quality of image is digitally improved in several ways by means of digital image processing before saving and storing the final image. These improvements can be divided to different processing blocks. Noise reduction, color correction and sharpening are examples of such processing blocks. In addition to those blocks, there are other processing blocks focusing on other areas than image quality improving. One example is focus value calculation block. Altogether those blocks create an image processing pipeline. Quality of the final image can be very different depending on in which order processing blocks are executed in the pipeline. However these processing blocks can't be always executed in optimal order because of some limitations. Physical design of device and cost of implementation are just some examples of possible limitations.

In older mobile phones focus of the camera system was fixed and it led to blurry images if distance between object and camera was too small. In modern smartphones this problem has been overcome with movable focus lenses. Also some studies of adaptive lenses have been done [1]. For better and easier user experience, autofocus (AF) algorithms are developed to take care of focusing instead of user. AF statistics calculation can be modelled as processing block and it may be set to different places in the pipeline. There are some limitations, which may limit at which point AF statistics calculations can be done. For example for capturing sharp images first camera have to calculate AF statistics from couple of frames and when knowledge of right focus point is found actual image can be captured. On the other hand acquiring right focus point shouldn't slow down the image capturing process too much. There might be some limitations, which forces some processing blocks to be executed before AF statistics calculations. On the other hand some blocks are preferred to be executed before AF statistics calculations. For these reasons in this thesis it is studied how these preceding processing blocks affect to accuracy of AF algorithm. This thesis is done for Intel Finland Oy. It should be also mentioned that besides producing bad quality images, poorly focused blurry images may cause problems for the following processing blocks in the imaging pipeline.

For evaluating the amount of focus, very simple algorithm is used in this study. There exist also many other algorithms, but those are mainly based on same ideas [2]. The algorithm calculates the amount of edge content in image and uses it as focus criterion. To fully study the effects of altered position of AF statistics processing block, images of a certain test scenes are captured in supervised conditions in dark room laboratory. From each scene raw images are captured with every possible focus lens positions, which meant with used camera 166 images per scene. These images are then processed in different ways in MATLAB and focus values are calculated. For each studied case 166 focus values are achieved. When these values are plotted to graph, usually it’s easy to find clear spike in values. At that spike images are in focus. However noise and other factors may cause this spike to become in wrong place and make AF algorithm to choose wrong focus lens position, which might cause blurry images.

For comparing AF performance, certain criteria are developed. These criteria describe how well the focus spike can be distinguished from the focus curve. To get a final goodness criterion, those criteria are weighted and Euclidian norm is calculated from weighted criteria. Criteria proposed by Lian and Qu in their study [3] were chosen to be used as basis for used criteria.

This thesis is divided to 5 chapters. First one introduces the whole study. Second chapter is about basics of light, human visual system (HVS), digital camera and image processing. Chapter 3 describes the image capturing pipeline and introduces used test scenes and designed focus criteria. In Chapter 4 results of criteria with different combinations of processing blocks are presented. Some conclusions are also presented based on performance of algorithm. Finally chapter 5 concludes the topic.

2 THEORETICAL BACKGROUND

In this chapter is presented fundamentals of light, human visual system, digital camera and signal processing. This knowledge is needed to fully understand the basics behind the studies.

2.1 Sensing light

Vision is one of the most important senses that humans use in their everyday life. Vision of humans is restricted to just visible colors, while some animals can see also some other areas of electromagnetic spectrum. However human visual system (HVS), which is responsible for light sensing experiences, is pretty complicated and there may be some differences between humans. For example some people suffer from red-green color blindness, which means that they don't see difference between green and red. In this chapter it is presented what light is and how HVS works. [4]

2.1.1 Electromagnetic radiation

To understand how digital cameras work, it's important to have basic knowledge about light and HVS. Term light usually refers to electromagnetic radiation, which is visible to human eye. However visible light is just small part of the vast electromagnetic spectrum. The electromagnetic radiation can be presented as propagating waves. These waves consist of stream of massless particles, which move at speed of light and contain certain amount of energy. These particles are called photons and their energy depends of the frequency they are oscillating. The electromagnetic radiation can be expressed with wavelength, frequency or energy. [5, pp. 42 – 45]

Wavelength is proportional to frequency according to formula

$$\lambda = \frac{c}{f} \tag{1}$$

where λ is wavelength, f is frequency and c is the speed of light in air, which is approximately $3*10^8$ m/s. Spectrum may be presented differently depending on the context. Whole spectrum may be divided to differently named regions. In figure 1 is example of dividing spectrum to certain regions. [5, pp. 42 – 45]

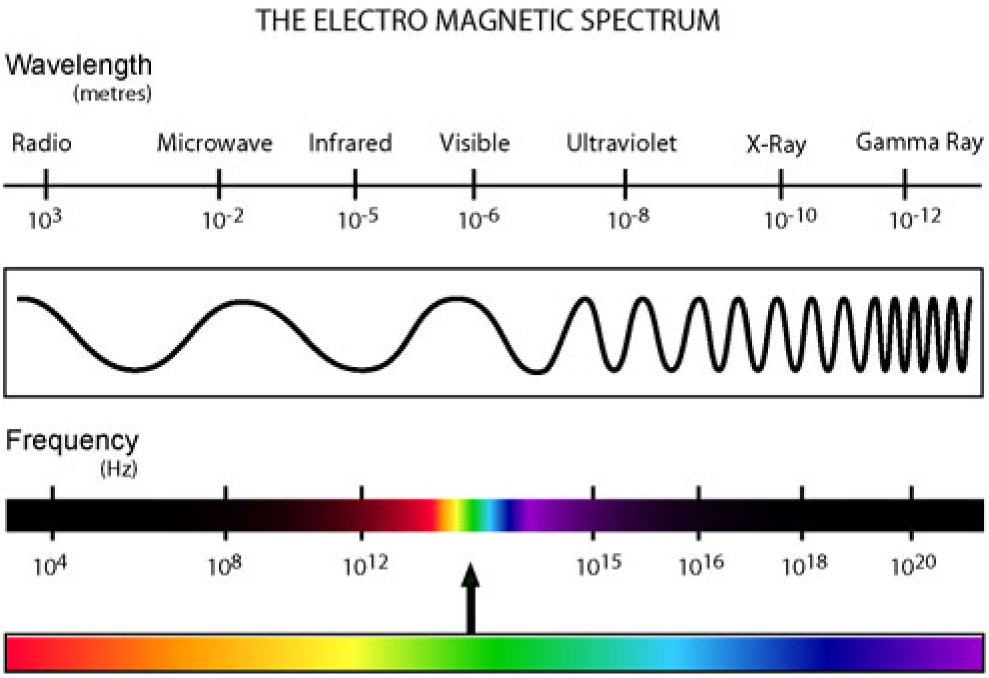

***Figure 1** Electromagnetic spectrum expressed with energies frequencies and wavelengths. Region of visible light is emphasized.*

Usually in case of normal photography main interested lies in visible light region. There is also slight interest in infrared and ultraviolet regions. Wavelengths are mostly used units, when presenting visible light. Usually visible light region is defined to have wavelengths from 380 nm to 750 nm. It's possible to further divide that region to some loosely defined colors. For example violet color can be defined to have wavelengths 380 nm – 450 nm. There are also precise standardized values for blue green and red wavelengths. The International Commission on Illumination (CIE) determined in 1931 that blue color is exactly 435.8 nm, green color is exactly 546.1 nm and red color correspondingly 700 nm. [5, pp. 283 – 284]

On the other hand usually in communications technology most interesting region is radio waves. For that reason radio waves are represented more accurately and radiation is usually expressed with frequencies. For example very high frequency (VHF) band includes frequencies 30 MHz – 300 MHz, which approximately corresponds to wavelengths from 1 meter to 10 meters. FM (frequency modulated) radio works in that frequency band [6].

It's also possible to express electromagnetic radiation as energy according to formula

$$E = \mathrm{h}f \tag{2}$$

where E is energy, h is Planck constant and f is frequency. Unit of energy depends of used unit of Planck constant. If energy is presented in commonly used units electron-volts, Planck constant of approximately $4.136*10^{15}$ eV*s should be used. Because of such a high energy amounts, X- and Gamma rays are harmful to human. [5, pp. 42 – 45]

2.1.2 Visible light

To observe light, some light source is needed. Usually light coming from that source is combination of many wavelengths. To perceive color, light of light source needs to hit some object. Perceived color depends of illumination level of light source and reflectivity of object. Object may absorb some wavelengths and reflect others. If light source emits white light, light that has approximately equal amount of all visible wavelengths, all visible wavelengths are fully reflected from object. If object absorbs all wavelengths, object looks black. Sun is one good example of light source emitting white light with very high illumination levels. Other commonly used light sources are fluorescent and tungsten lamps, which aren't usually emitting perfectly white light. [4]

Besides of reflection light has three other properties called refraction, dispersion and diffraction. Refraction means deflecting of light rays while they travel from one material to other. This happens at the border of two materials. Amount of refraction depends of the angle light is coming to border of materials and difference of refraction multipliers of adjacent materials according to formula

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{\lambda_1}{\lambda_2} = \frac{v_1}{v_2} = \frac{n2}{n1} \tag{3}$$

where θ_1 and θ_2 are incoming and leaving angles of light. λ_1 and λ_2 are wavelengths of light in different materials. v_1 and v_2 are speeds of light in different materials. n_1 and n_2 are refraction indexes accordingly. Dispersion means that light travels in material with different speed depending of wavelength. When refraction and dispersion is combined it means that light travelling through some material bend different wavelengths differently. In camera this means that sun light travelling through lens scatters the light without careful lens design. Diffraction means that light bends around obstacles in its path. If light travels through small opening, like aperture of camera, it radiates to every direction after the opening. Radiation becomes weaker when the angle increases. [4]

2.1.3 Human visual system

The human eye and a digital camera create images out of the surrounding lighting differently. However the main idea behind both of them is pretty similar. There exist elements that determine the amount of incoming light, elements that take care of focusing and elements that are able to sense light. Next is presented some basics of the HVS needed for creating images.

In the human eye light strikes first the transparent cornea. The cornea refracts light and is responsible for most of the focusing. Curvature of the cornea can't be altered heavily. However it can be altered slightly, by altering pressure inside cornea. This isn't really a problem with distant objects, because even very small changes are enough to alter the focus correctly. After the cornea, light encounters iris. Iris is uniquely colored origin, which is responsible for controlling the amount of incoming light. In the center of the iris is a pupil. The pupil is an aperture, whose size is controlled by muscles in the iris. Those muscles can increase or decrease the size of the pupil. Right behind the iris

can be found a lens. The lens is transparent and flexible. The lens is another part in the human eye, which is responsible for focusing. It helps to focus objects at smaller distance. When a human is focusing to close range, the lens is round shaped. While focusing further away, muscles around the lens stretches the lens. When focusing further than about 5 meters, the lens becomes flat and doesn't refract light. [4]
After the lens, light strikes back of the eye. There is the sensing element of the eye called the retina. In the retina there are photoreceptor cells. These cells can be divided to two basic types: rods and cones. In the retina there are approximately 75 to 150 million rods. Rods are sensitive in low illumination. They are also sensitive to motion and responsible for peripheral vision. However the rods are not sensitive to color. Number of cones is much smaller, roughly 6 to 7 million. The cones are highly sensitive to color. They are also responsible for the highest visual acuity. The receptor density is highest in the fovea (central of the retina). Most cones can be found from the fovea. However when number of the cones is decreasing, number of the rods is increasing accordingly. Density is pretty constant to around 25 degree of the fovea. From that point on number of receptors is decreasing as can be seen in figure 2. There is also blind spot in the retina, where aren't any receptors. This is because nerves and blood vessels exit the eye from that point. Blind spot can be found from different side of fovea in left and right eye. Finally information from rods and cones are delivered to brains, where all the heavy processing happens and image is created. [5, pp. 34 – 37, 284 – 285; 4]

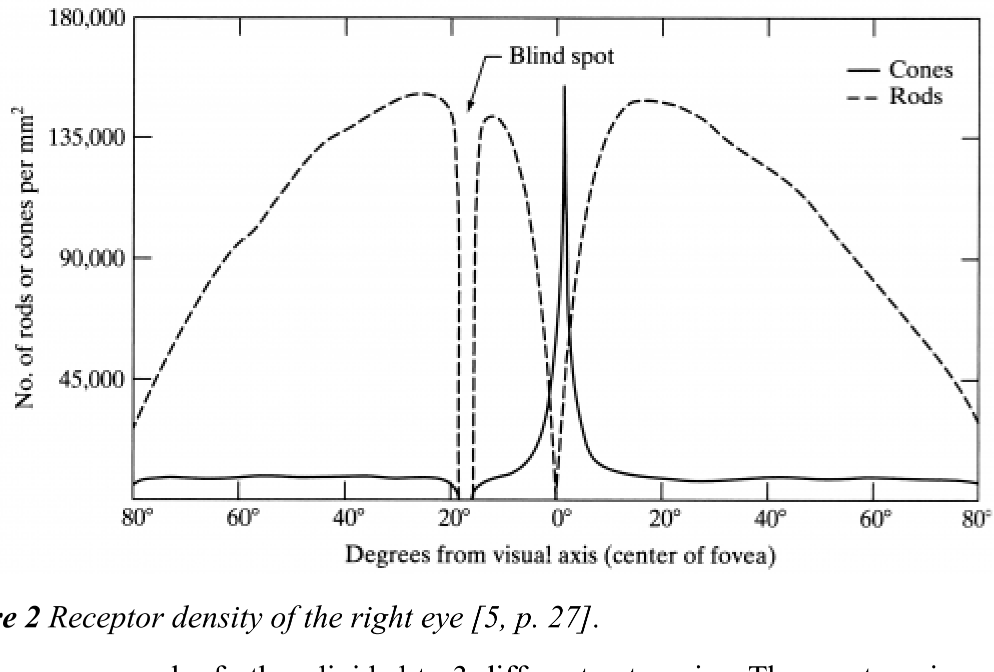

Figure 2 *Receptor density of the right eye [5, p. 27].*

The cones can be further divided to 3 different categories. These categories can be called to R (red), G (green) and B (blue). Confusingly compared to other cones, the R cones are most sensitive to yellow or little greenish color. However the R cones are also most sensitive to primary color red. Approximately 65% of the cones belong to R, 33% belong to G and only 2% belong to B. The cones belonging to B are however most sensitive and the G cones are slightly more sensitive than the R cones. Sometimes categories are called S (short wavelengths), M (medium wavelengths) and L (long wavelengths); after all they are describing the same categories. In figure 3 is presented sensitiveness of different cone categories weighted with their amounts.

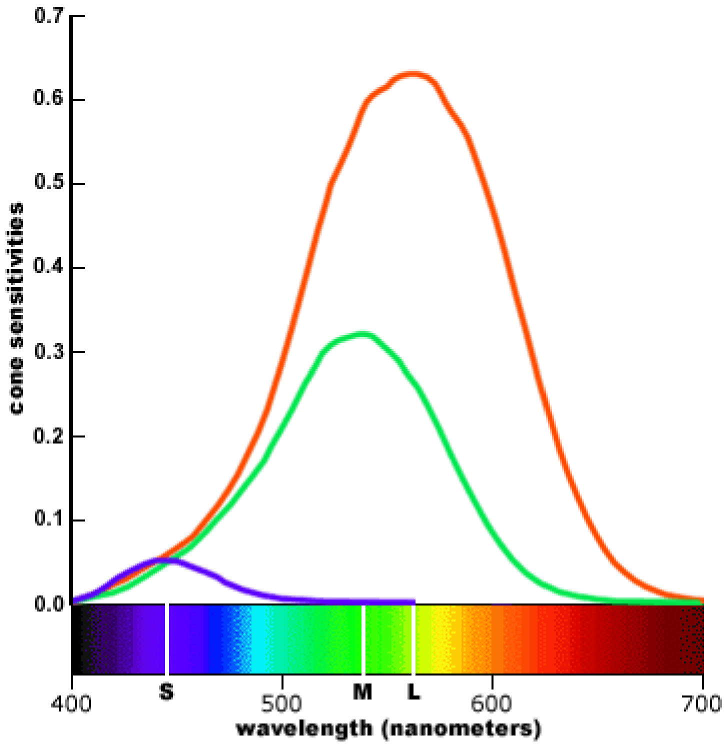

***Figure 3** Population weighted cone sensitivity functions in linear scale [4].*

It's also interesting that the B cones can be found mostly outside the fovea, while the most of the R and G cones can be found from the fovea. If red and green colors are focused to fovea, blue color refracts so much that blue light won't hit the fovea. This may be one reason for distribution of the B cones. The rods are much more sensitive to light than the cones. Even though the rods are blind to color they still are more sensitive to smaller wavelengths. With this knowledge one could think that humans won't sense blue color very well. However it's suggested that HVS in human's brain amplifies the blue color. [5, pp. 34 – 37, 284 – 285; 4]

2.2 Camera

In this chapter basics of digital camera sensing are explained. Also some functionalities of camera are presented.

2.2.1 From light to image

Camera sensor correspond retina of the human eye. There are two dominating technologies used as camera sensors: CCD (charge-coupled device) and CMOS (complementary metal-oxide semiconductor). For this research it's enough to know that both have advantages and disadvantages, but the basic idea of technologies doesn't differ much. For simplicity those can be thought as two different ways of creating sensing circuits. More information can be found for example from [7, pp. 4 – 11].

Here is described typical process of capturing image with digital CMOS camera. To capture image camera needs to collect some light. That light has to travel through camera system all the way to the sensor. During that trip light travels at least through one lens. In modern camera systems, even in mobile phone camera systems, there usually exist many lenses. Each lens has its own function. For example some lens or lenses can be responsible for focusing while another lens or lenses may be responsible for zooming and rest of lenses concentrate in correcting optical aberrations and other non-idealities. [8]

In this part one biconvex lens is used to simplify optics. To make images look sharp, lens must be at certain distance from sensor. This distance is dependent on distance from the lens to object and shape of the lens. By altering position of the lens it can focus light rays properly to get sharp images. Example of this can be seen in figure 4.

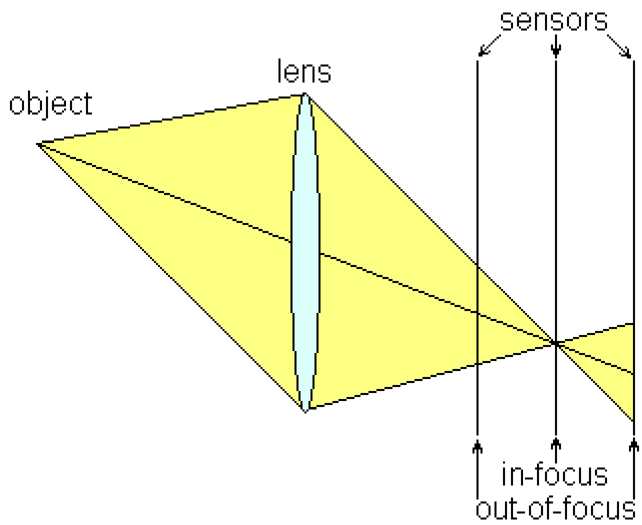

Figure 4 *Example of focusing. Different distances between lens and sensor makes light beams from object to converge more or less sharply.*

Images become properly focused, when the distances between object and lens and lens and sensor correlate to focal length of lens. Relation of these 3 parameters are presented in formula

$$\frac{1}{f} = \frac{1}{d_1} + \frac{1}{d_2} \tag{4}$$

where f is focal length, d_1 distance between object and lens and d_2 distance between lens and sensor [9, p. 13]. To calculate parameters for system with multiple lenses, calculations of individual lenses must be combined. In those calculations “object” is always previous lens and “sensor” next lens. Shape of a lens is determining focal length. Focal length is distance between focal point and center of the lens. Focal point can be found by shooting a lens with a collimated beam of light straight in front of lens. Depending on the lens light beams are converged or diverged. In converging lens light beams are converged to travel through focal point. In diverging lens light beams are diverged in a way, that one could find a crossing point for beams by imaging fictional extensions of diverging beams to the direction of incoming light. Examples of finding focal point can be seen in figure 5.

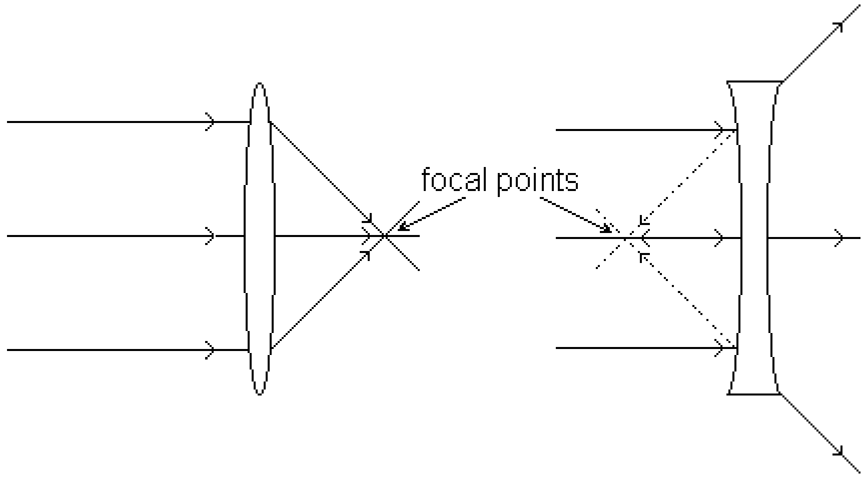

Figure 5 *Example of finding focal point for converging and diverging lenses.*

Focal length of a thin lens can be also calculated according formula

$$\frac{1}{f} = (n-1) * \left(\frac{1}{R_1} - \frac{1}{R_2}\right) \qquad (5)$$

where f is focal length, n is refractive index of the lens material, R_1 and R_2 radius values that describes curvature of lens. $R1$ is radius of imaginary circle in light sources side and $R2$ is radius of imaginary circle in other side. [8; 9 pp. 13 – 17, 31]

After light has travelled through lens system it reaches filtering layer. First light usually meets ultraviolet and infrared pandbass filter, which allows just visible light to pass to color filter array (CFA) [10]. CFA consists of filters placed over the sensor. These filters respond to certain colors and are arranged in certain order. The sensor consists of pixels, where each pixel consists of one or more photo sensing elements. For each pixel there is one color filter. The studied camera utilizes very commonly used CFA called Bayer filter. One 2x2 block of Bayer filters consists of one red, two green and one blue color components. Those components are aligned from left to right and top to bottom in order of Gr, R, B and Gb like in figure 6.

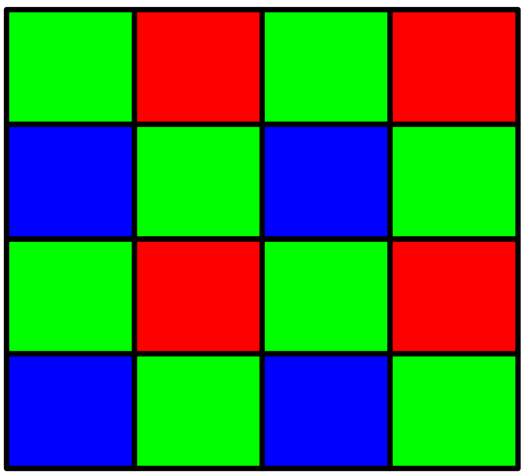

Figure 6 *Example of Bayer filter.*

After CFA light reaches the sensor. At sensor photons excite electrons, which create charge on each pixel. These charges are then collected. However charges are still very small and should be amplified. In CMOS cameras every pixel has its own amplifier. International Organization for Standardization (ISO) has developed ISO speed standard to describe the sensitivity of film. In era of digital cameras ISO speed is used to describe the amount of used amplification. After certain exposure time amplified voltage of capacitors are measured. For this far everything has been in analog form. After voltage has been measured it's transformed to digital form (analog to digital conversion, AD). Depending of number of bits per pixel, image can have certain amount of possible levels. The studied camera uses 10 bits per pixel, which means that each pixel may have 1023 different values. Actually that's not the whole truth; usually some black level is also set. Black level is set to overcome noise caused by dark current, which is always existent in camera sensors. Thermal noise is more visible on low values, which means that black color would look lighter than should and it would have bigger variations. In used camera black level is set to be 42. This means that used camera should get values between 42 and 1023. However because of noise, it's also possible, even though more uncommon, to achieve values smaller than 42. [10]

At this point incoming light is transformed to vector of digital values. This data is called raw data, because it's minimally processed. From now on many cameras have two non-exclusive options. Cameras can add some header data in front of actual image data and save the whole vector as raw image. Another possibility is to convert raw data to GrRBGb image, which consists of GR, R, B and Gb components, or to RGB image, which consists of R, G and B components. In RGB image, two green components of GrRBGb image are averaged. These images can then be further processed. Further processing may contain several different processing steps. Many of those steps are done to improve quality of the image. Finally after processing the image is stored in some format. Most common format is JPEG [11], which compresses image to much smaller size by using lossy compression.

There is clear advantage in raw images compared to JPEGs. No data is lost and it can be chosen how to process the image. In JPEG image is compressed and also some processing is done. This processing may be something non-optimal and, at least after JPEG conversion, impossible to revert. Of course raw image has some drawbacks too. To show raw image properly it needs to be converted to some other format, like RGB. Also more space is needed to store those images, even if lossless compression is used. Of course images could be stored in smaller space by using lossy compression, but that ruins the whole idea of raw image. Another problem is that even though standardized raw formats exist, those aren't widely used among big camera manufacturing companies. Almost every manufacturer uses different header data. This makes it little bit tricky to user to use raw images. Luckily camera manufacturers usually offer some raw image convertor tool. With enough knowledge it's also possible to convert images oneself. Also 3rd party image processing tools may offer tools to convert most used raw formats. [10]

2.2.2 Exposure

Exposure is combination of aperture size, shutter speed and ISO speed. In digital cameras the device can set up these settings automatically (auto exposure, AE) or user can set up those manually. With poor exposure values image may appear to be very imbalanced. It may be for example too dark or too saturated. Usually choosing values for these settings is some kind of balancing between the options. If one setting is changed, two others should be chosen to correlate with that setting in current conditions. [12]

Aperture size controls the size of hole, where light travels inside the device. Simultaneously it affects to the amount of light getting to device. Aperture corresponds to the pupil in HVS. Aperture size is affecting to depth of field. Bigger aperture gives more narrow depth of field. It blurs more distant areas of scene, which can be sometimes desired feature. This way object in foreground can be highlighted. Smaller aperture size can give very detailed photos even with objects in very different distances. The smartphone used in this study, like smartphones commonly, uses fixed aperture size. [12]

Shutter speed controls the duration of light getting in device. In other words it determines exposure time. Longer exposure time means less noise but higher blur if camera moves or there is motion in scene. Sometimes this is a desired feature, because it can make images look more living. Noise is reduced because desired signal, which is the scene, becomes bigger compared to noise signal. This means better signal to noise ratio (SNR). Shorter exposure time on the other hand grants sharper pictures of fast movement and makes it easier to take shot with free hand without “motion blur” caused by unsteady hand. Traditionally shutters have been mechanical but with digital cameras it’s also possible to use electrical shutters. Electrical shutter resets pixels and starts to react to incoming photons. The studied camera uses electronic rolling shutter. [12]

ISO speed controls the sensitivity of sensors. Bigger ISO speed means that sensors are more sensitive to incoming light. In practice actual sensitivity of sensor isn’t altered, but the amplification of sensor readings is altered. Increasing ISO speed too much also increases drastically the amount of noise, because noise power becomes bigger in relation to the signal power. In low light situations it’s more preferred to use higher ISO speed. Especially in situations with flash disabled or for background areas while flash is enabled. Lower ISO speed is much better in properly illuminated scenes because of lower noise. [12]

2.2.3 Focus and autofocus algorithm

In modern cameras there is usually option to let camera alter focus automatically (AF) or manually by rotating focus ring. Anyhow focus is altered by moving focus lens back and forth. In mobile phones manual option isn’t valid method, because that kind of big optical system is impractical in thin mobile phones. Sometimes focus lens is even made fixed, which means that focus cannot be altered. Modern smartphone cameras usually have AF system.

In camera modules AF is implemented by moving focus lens and calculating focus value alongside. Every time focus lens is moved to somewhere focus values are recalculated from that focus point and compared to previous ones. From that information AF algorithm decides whether the best focus point is found or should it be still searched. Algorithm also decides the position where focus lens is next moved. Lens should be moved forward and backward until right focus point is found. Usually lens is first moved in bigger steps and adjusted with smaller steps when roughly searched focus value is achieved. [3]

Usually when capturing image, in scene there is something interesting (meaning shapes and differences in color). In region of interest (ROI) there are normally some noticeable contours and edges. One way of calculating focus value is to detect edges from scene. Detecting edges can be done in many different ways. Normally it's some kind of transformation done to ROI, which return just one value. This value describes edge information in ROI. Transformation can be for example summing gradients of image. One simple approximation of gradient is the Sobel operator which can be calculated by using 3x3 mask of formula 6 [5]. ROI is filtered with that mask and outcome is summed up. It returns higher values for areas where is bigger vertical differences in intensity. If sobel operator is transposed, information of horizontal edges is acquired. By summing up vertically and horizontally filtered image, image with strongest edges highlighted can be plotted. Example of using Sobel operator can be found from figure 7. Typically in smartphone cameras transformation is done with high pass filter. High pass filter is used, because it's fast to implement and gets rid of lower frequency components which correspond smaller changes ROI. In this thesis high pass filter was used.

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{6}$$

This way calculated focus values are very dependent on scene. Scene with a lot of objects in ROI returns much bigger values than ROI without objects. However if there exist edge content in ROI in-focus images should get bigger focus values than out-of-focus images. [5, pp. 134 – 137]

This kind of algorithm creates higher spikes to focus curve at the focus lens positions (focus points), where objects are well focused. In real world cases global maximum focus value isn't necessarily the best focus point, because there may be objects at different distances. However for simplicity and practicality in this thesis global maximum value is assumed to be the right focus point. This said it's very important for algorithm, that the found maximum really is the global, not any local one. [13]

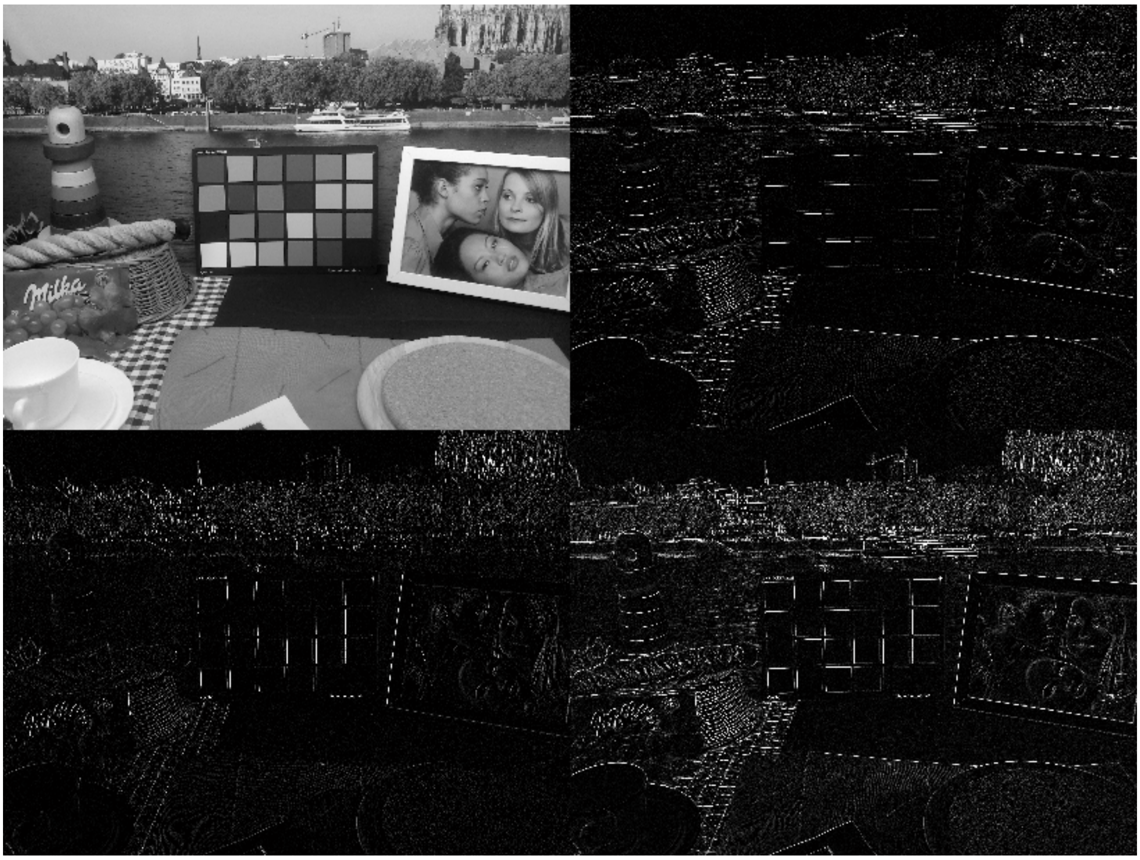

***Figure** 7 Top left image is filtered with Sobel operator presented in formula 6 to detect vertical edges and transpose of the same Sobel operator to detect horizontal edges. In bottom right vertical and horizontal edges are summed together.*

2.3 Image quality

In this chapter is presented methods for evaluating image quality, some things affecting image quality and some ways to improve image quality.

2.3.1 Evaluating image quality

Measuring image quality may be tricky task. After all human is looking images and evaluating the quality of them. If objective measurement is used, it should correspond to HVS. Still there is differences between individuals how they evaluate quality of images with different aberrations. This behavior is hard to implement to some formula.

There are many ways to do subjective image quality assessment. Some of them work better in certain cases and are more or less standardized. For example International Telecommunication Union (ITU) recommends using standards presented in [14] to evaluate subjective quality of images and videos. Because arranging subjective quality measurement event is time consuming and expensive, many times some formulas are used to evaluate quality of image. Best methods model HVS well, but are very complex. In [15] some pretty simple methods are presented. However subjective quality is usually very important and should be taken into account.

One objective image quality measurement is signal to noise ratio (SNR). SNR compares power ratio of useful signal and noise. It can be calculated according to formula

$$SNR = 10 * \log_{10} \frac{P_{signal}}{P_{noise}} \quad (7)$$

where *SNR* is in decibels, P_{signal} average power of useful signal and P_{noise} average power of noise in signal. As power measurement one can use for example variance. In this quality measurement method either reference image and noisy one or noisy image and knowledge of additive noise needs to be known. [16, p. 105]

2.3.2 Noise and noise sources

Traditionally noise in digital cameras is considered to be additive. That noise is typically impulse- or Gaussian-like noise. Additive noise is something, which can be fully removed from image if pattern of noise is known. Additive impulse noise, also known as salt & pepper noise, creates randomly minimum and maximum values to the image. It's more probable that in real world impulse noise causes outliers to data more than minimum or maximum values. However in this thesis it was chosen to use minimum and maximum values as impulse noise. Gaussian noise, also known as white noise, on the other hand creates Gaussian distributed noise randomly to each pixel. More about different noise types can be found from [5, pp. 220 – 230]. Amount of noise isn't necessarily as important as SNR, because image may look even better with higher amount of noise if only useful signal is strong enough to mask the noise. That can be seen in images where are areas with a lot of details (stronger signal) and areas with very few details (weaker signal). If noise with same variation is added to whole image, areas with less details look much noisier compared to areas with a lot of details. [10]

There exist many sources of noise when image is captured. Some of the noise is caused by user and some is caused by device itself. User caused noise sources can be for example unsteady hand while taking picture or low light situation with poor flash. Next are presented some noise sources from device and physics.

Photon noise, also known as shot noise, is caused by fluctuations in numbers of photons that source emits. This noise exists always in digital cameras. It's related to physics of light. Photon noise is more existent in lower light, than in higher light. This means that photon noise effect can be reduced by using longer exposure times, because more light is getting to the sensor and SNR is increased. This source of noise follows a statistical Poisson distribution, which is pretty similar to Gaussian distribution. It's more like biased Gaussian distribution. [10; 7, p. 31]

Dark current noise, also known as thermal noise, is caused by heat of the sensor. In sensor photons excite electrons. Electrons are also excited by heat. These electrons create a charge on a capacitor, which is finally measured. All this means that even from shots without any light source, some non-zero voltage values are normally measured. By cooling sensor this effect can be reduced. Also shorter exposure times reduce this effect, because longer exposure time creates more heat and decreases SNR. Dark current noise resembles mostly Poisson distribution. However it's typically modeled with Gaussian distribution, because Gaussian distribution approximates pretty well Poisson distribution when photon arrival rate is high. [10; 7, pp. 4 – 5, 30]

Amplifier noise, also known as readout noise, is caused by imperfect amplifier. Measured voltage on a capacitor is amplified. Amount of amplification depends of ISO speed. Bigger ISO speed means also bigger amplification. Amplifier may perform differently at different times, despite of original level of voltage being same. With lower ISO speed this problem can be reduced, because it increases SNR. Amplifier noise follows Gaussian distribution. [10]

Quantization noise also plays a role in overall noise of image. Even though its impact is usually very small compared to many other noise sources, it still exists. Information in digital cameras is converted to digital, when analog amplified voltage is transformed (rounded) to discrete voltage level. At this AD conversion slight amount of information is lost. This error can be reduced by increasing amount of bits, which would decrease distance between adjacent voltage levels. Also suitable non-uniform discretization could help subjectively, because of nature of the HVS. Quantization error follows Gaussian distribution.

Fixed-pattern noise, also known as spatial noise, in CMOS sensors is mostly caused by differences between amplifiers. It also takes account other noise sources, which create similar noise pattern to every image. It notices for example dead pixels, which gives always certain constant value. Longer exposure and higher temperature increase fixed-pattern noise. It can be reduced by subtracting dark frame or average of several dark frames. Latter one is preferred way, because effect of temporal noise is diminishing after averaging. Fixed-pattern noise can be seen as impulse-like noise, even though values aren't necessarily minimum or maximum, but they are randomly distributed. [10, p. 31]

2.3.3 Noise reduction

Noise is almost always existent in electronic devices. Sometimes amount of noise is so small that it doesn't disturb at all. When dealing with images, video or audio it's usually necessary to pay attention to noise. Sometimes noise can be even desired feature, but most of the time it's not. For that reason noise reduction, also known as denoising, techniques can be very important. For noise reduction, numbers of techniques are designed. Some of them can be very complex while others are much simpler. Usually more complex algorithms need more processing and may result in better noise reduction. Some noise reduction algorithms can be found from [5 pp. 230 – 253].

Noise reduction algorithms are usually based on assumption that image consists of the useful information signal and additive noise, which doesn't correlate with the useful signal. Anyhow problem in noise reduction is to keep the useful information signal detached while decreasing the amount of noise. This means that reduction algorithm will probably work better if it can separate noise and the useful signal from each other. Usually performance of noise reduction algorithm can be improved if something about nature and behavior of noise is known.

In smartphones it's important that image capturing is fast enough and battery isn't wasted too much. This leads usually to simpler noise reduction algorithms. Requirements in AF calculations are even stricter. For these reasons next are presented two very simple noise reduction algorithms and one much more complex. Algorithms used in this study are mean filtering, median filtering and BM3D. [17]

Mean filtering is very simple way of reducing noise, especially Gaussian noise. However by smoothing local variations, this algorithm also blurs edges and small details. Mean filtering can be done in very simply way for each color component by just calculating average of certain size neighborhood with certain weights according to formula

$$C(x,y) = \sum\left(n(s,t) * \frac{w(s,t)}{\sum w}\right) \qquad (8)$$

where x and y are coordinates of image and s and t are coordinates of neighborhood around the point (x, y). Neighborhood means the nearest pixels around some point. Usually neighborhood is square. This means that $C(x, y)$ is certain pixel of color component, $n(s, t)$ is certain pixel in $C(x, y)$ neighborhood and $w(s, t)$ is weight in corresponding position. Idea of using weights is to make some pixels more significant in calculations. Usually bigger weights are used to values, which are closer to the center of the mask. Masks can be used to calculate mean values. In figure 8 are examples of 3x3 matrices that can be used for average filtering.

$$\frac{1}{9} * \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad \frac{1}{16} * \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Figure 8 *Averaging 3x3 masks. In right mask weighting is used.*

Values of mean filtered image is achieved by sliding for example presented masks over the color components and summing outcome of pixel-wise multiplication of mask and neighborhood. Example of image filtered with figure 8 filters can be found from figure 9. [5 p. 231]

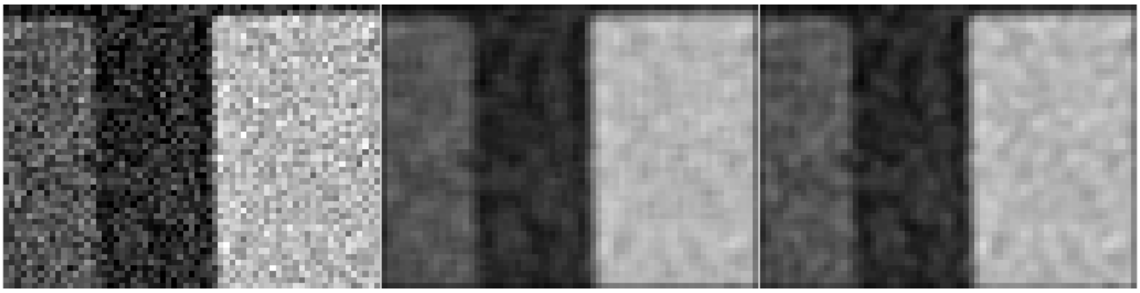

Figure 9 *Left image is filtered with filters presented in figure 8 in same order.*

Median filtering is very effective to impulsive noise and it usually reduces also Gaussian noise. It has less blurring effect compared to linear mean filtering. Value of median filtered image is achieved by calculating median value of neighborhood around the pixel. This can be expressed as formula

$$C(x,y) = median(n) \tag{9}$$

where x and y are coordinates of image, $C(x, y)$ is certain pixel of color component and n is neighborhood around coordinates x and y. Example of image filtered with median filter can be found from figure 10. [5 s. 234]

BM3D is computationally very demanding algorithm, which can reduce added Gaussian white noise very effectively while still preserving details. Performance of algorithm is based on finding similar blocks in image. Correlation of those blocks is used as advantage when noise is reduced. The algorithm filters noise out of the image by calculating weighted average of overlapping similar blocks. If there are some unique features, algorithm preserves those. Example of image filtered with BM3D can be found from figure 10. [18]

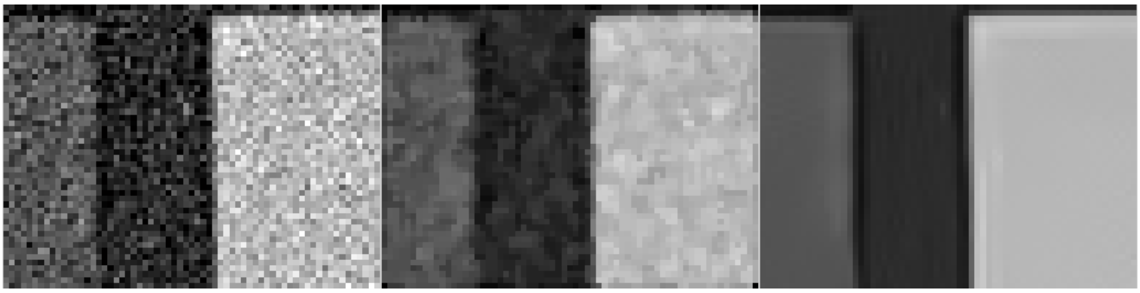

Figure 10 *Left image is filtered with median and BM3D filters.*

2.3.4 Vignetting

Vignetting is artifact that is usually well visible in raw images taken with smartphones and it's caused by camera system. Vignetting is typically divided to 3 categories: natural, optical, mechanical vignetting. In addition there exists color shading, which is usually listed under vignetting, because its nature is very similar. Natural and optical vignetting can be seen as gradual illumination falloff from the center of the image. [19]

Natural vignetting consists of three elements. First and most affecting element is difference in distance that light has to travel from aperture to the sensor. Electromagnetic radiation contracts according to distance it travels. Second affecting element is the area wherefrom light travels to different parts of the sensor. From center of the sensor aperture is round, but when looking the aperture from the edge of sensor it's elliptic and covers smaller area. The last effecting element is based on the difference of area that light covers when it's reflected to sensor plane. Light beam coming in an angle makes the covered area bigger and distributes illumination of that beam to whole area, which decreases the light intensity at single point. These elements of natural vignetting are presented in figure 11. [19]

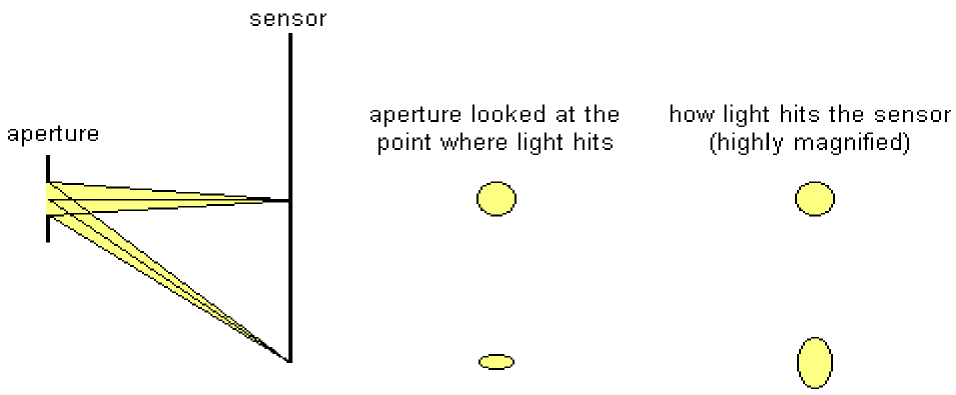

Figure 11 *Image of elements that effect natural vignetting. Image is drawn without lenses even though light beams are drawn in a way like lens is focusing them correctly. There may be lenses before and after aperture. Lenses direct light in to aperture and from it in a certain angle.*

Optical vignetting is caused by different amount of light travelling to camera system from different angles. With aperture size and position this effect can be increased or reduced. If aperture is very close to the opening or aperture size is small enough, it's possible to collect light equally from pretty wide area. If aperture is too far away of opening or size of aperture is too big, light from wider angle gets to system from smaller area. Area becomes elliptic, which means that less light is getting to the edges of the sensor. This effect is presented in figures 12 and 13 . [19]

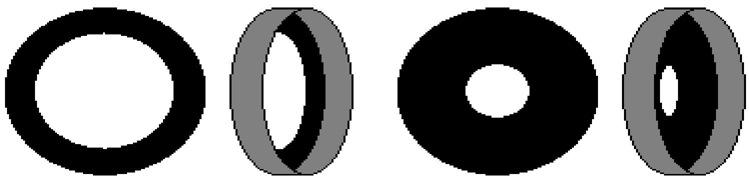

Figure 12 *Here is presented very simplified version of camera lens system. It has only aperture and the opening where light comes into the system. Black color represents aperture and gray color inner borders of camera lens system. This figure consists of 4 images. From left to right 1st and 3rd image are seen in front of camera, whereas 2nd and 4th image are seen from certain angle. In 1st and 3rd image only aperture is visible. In 2nd and 4th image the rightmost ellipse is the opening and inner white area is aperture corresponding to 1st and 3rd image.*

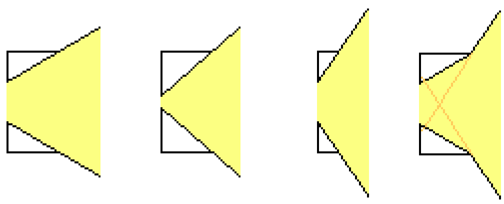

Figure 13 *Here are cross-section images of figure 12 cases. Also effect of distance between aperture and the opening is presented.*

If camera system is designed well, mechanical vignetting shouldn't exist. However it's caused something that is blocking light. In figure 13 the rightmost case is describing that phenomenon. This means that too big extensions to the lens system or too thick filters may block the light entering to the edge of sensor and cause vignetting. Mechanical vignetting is usually more sudden than natural or optical vignetting. It may make edges completely dark. In smartphones this problem doesn't usually exist. [19]

Probably the most troublesome vignetting problem is color shading. It is caused by infrared filter (IR filter), which is physical filter layer. IR filter is bandpass filter which should filter out infrared and ultraviolet regions. However thin smartphones causes IR filter to fail. Because of thinness light beams hitting peripherals of sensor become in such a high angle, that frequency response of IR filter changes and filters more desired wavelengths. This can be seen as heavy color errors at the peripherals of raw image. Color error is different depending on the frequency response of the light source. [20]

Vignetting is easy to correct if photo is taken from uniform flat field with certain aperture size in certain illumination. Counter filter, which makes the image uniform, needs to be developed. Idea is that corner values should be as bright as the lightest value. With this simple correction unfortunately noise is also multiplied and SNR degraded. Vignetting correction is demonstrated in figure 14.

Figure 14 *Here is presented vignetting correction. From left to right first is presented unprocessed image. Next are show 4 smaller gray scale images, which are corresponding vignetting correction for each color component (Gr, R, B and Gb). Next is shown overall combined effect of these color components. In last image vignetting has been corrected.*

When correcting vignetting first every color component should be filtered with big averaging filter to reduce noise. Then maximum of each color component is divided with each pixel value of corresponding color component according to formula

$$C_v = \frac{\max(C)}{C} \tag{10}$$

where C means certain color component and C_v matrix, which is used to correct vignetting. Vignetting of image taken in same illumination can be removed by pixel-wise multiplying color component of that image with corresponding C_v. If enough matrices are attained in different illumination, it's also possible to form model between them and calculate values for matrices in other illuminations. As can be seen from figure 14, different color components create slightly different patterns. Lens is causing this, because it refracts different wavelengths (different colors) differently.

2.3.5 Scaling

Spatial resolution determines the amount of pixels belonging to image. This amount can be increased or reduced by scaling. This thesis mostly focuses on downscaling. Scaling is important especially when images are shown in different medias. Also some requirements can be set for stored images, which may affect to needed spatial resolution of images. Scaling can be done with many different algorithms and results of scaling can vary a lot.

Usually upscaling can be done by upsampling image. This means that zeros are added evenly to image. It's also known as zero padding. Amount of added zeros depends of amount of upscaling. Idea is to predict value for zeros so that they blend to image. Downscaling on the other hand means downsampling image. Image values are extracted evenly depending of the scaling factor and values. Neighboring pixels may be made to better correspond discarded values. This means smoothing the gap between adjacent pixels. Commonly unintended gap between two adjacent parts of image is called aliasing. Smoothing the gap is known as anti-aliasing. Anti-aliasing also slightly blurs other areas of image. [5 pp. 62 – 66]

The most basic and computationally very efficient scaling method is known as nearest neighbor scaling. When upscaling, values for added zeros are copied from nearest neighbors. This causes aliasing to image. Deciding which neighbor is used to be the nearest in image processing depends of implementation of the algorithm. There aren't any global rules. When downscaling is done, the gap between new neighbors isn't compensated anyhow. Pixels are just discarded evenly, which causes aliasing. [5 pp. 62 – 66]

Subsampling is special case of downsampling with nearest neighbor method. There every n^{th} pixel is extracted from original image. The difference to normal nearest neighbor method is that *n* is integer. This kind of subsampling is also known as decimation. Subsampling can be expressed with formula

$$C_{scaled} = C(1{:}n{:}width, 1{:}n{:}height) \tag{11}$$

where *C* is color component, *n* is scaling factor while *height* and *width* are dimensions of image. Example of subsampling can be found in figure 16.

Pixel binning means averaging of non-overlapping blocks as in formula

$$C_{scaled}(x, y) = \frac{\sum C(x*n-n+1{:}x*n, y*n-n+1{:}y*n)}{n^2} \tag{12}$$

where *x* and *y* are coordinates and integer value *n* is size of used binning factor. Binning factor *n* means that new pixel value for each color component is calculated as average of *n*x*n* block of certain color component. This creates aliasing between the blocks. For example in figure 15 new scaled $\widehat{\mathrm{R}}1$ would be output of value of (R1+R2+R3+R4)/4 calculation, if 2x2 binning would be used. Example of pixel binning can be found in figure 16.

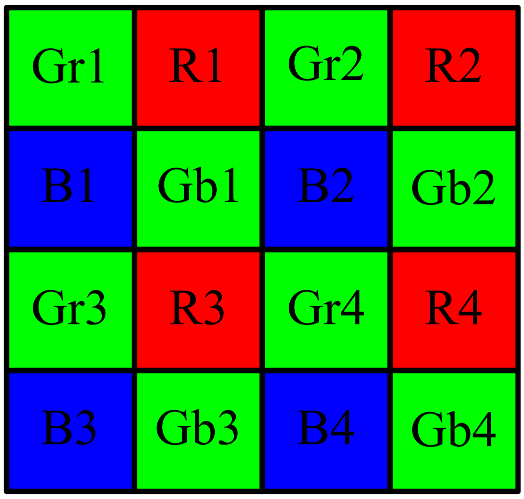

Figure 15 *Example of Bayer mosaic.*

Bicubic scaling was done by using MATLAB built-in function *imresize* with scaling multiplier as parameter. Instead of using 4 surrounding pixel values in prediction, bicubic interpolation uses 16 surrounding pixel values to predict value in certain point. Example of bicubic scaling can be found in figure 16.

Figure 16 *Example images of downscaling. Downscaled images are 1/24th of the original one. All images are however presented in same size. From left to right and top to bottom are presented original image, subsampled image, image downscaled with pixel binning and bicubicly downscaled image.*

First third degree polynomial is calculated from 4 vertical or horizontal values and then from horizontal or vertical values. Based on these results the final value is calculated. By default *imresize* uses anti-aliasing filter, which smoothes the differences between blocks [21]. Bicubic scaling provides continuous transition over pixel values and provides slightly smoother images compared to bilinear interpolation.

2.3.6 Color correction

Human eye is pretty sensitive to abnormalities in color images, especially if original image can be compared with image with slightly distorted color content. This means that if image with slight distortion in color is shown, one won't necessarily notice immediately anything strange. If more time can be spent on investigating image, some abnormalities may be noticed. However if same image with proper color content is showed at the same time, it's easy to notice immediately that original image looks more natural and usually subjective quality is better.

Usually raw images aren't looking natural without white balancing and color correction. Problem is camera module, which can't compensate differences in illumination. Auto white balancing (AWB) tries to compensate this problem. For example in studies [22; 23] performances of a couple of common AWB algorithms are evaluated. In first article context is a little bit different, but still it gives good idea of different AWB algorithms. Used algorithm is very simple and basic one. Algorithm is called Gray World (GW) and it was introduced also in both articles. Example of GW algorithm can be found from figure 17. More about used AWB algorithm can be found from chapter 3.1.6.

Figure 17 *Example of white balancing. From left to right are presented original image, white balanced image with GW algorithm and image, where gains of color components are manually altered.*

Color filters and lens of camera module are also causing color errors in captured images. Usually digital camera consists of sensor, whose single pixels are sensing filtered light. Filter type determines what sensor really senses. Because spectral characteristics of these filters differ from HVS characteristics, some color error occurs. One way to reduce this error is to multiply image with suitable color correction matrix (CCM). Usually this matrix is 3x3 matrix with multipliers summing to one within each row. Multipliers of color correction matrix should be recalculated for every differently illuminated image.

There already exist studies about color correction algorithms. One of those is presented in [24]. There also exist some studies of noise amplification caused by color correction. For example in articles [25] and [26] this is studied. Because this study isn't about the way of calculating CCM, CCM fitting to camera sensor was used. That matrix was used with tunable parameter to change multiplier values. Used CCM is presented in greater detail later in chapter 3.1.6.

Because raw images aren't gamma encoded, gamma correction needs to be done for images to show them correctly on modern displays. This means better utilization of color space. Camera sensor is responding linearly to increasing level of light. On the other hand eye perceives light levels logarithmically. This means that with linear presentation needs more bits to present light levels correctly, because eye is more sensitive to changes in low light levels and less sensitive to high light levels. For this reason images are normally gamma encoded to better utilize the number of bits per pixel. Minimally processed raw images are presented in linear scale. Modern displays on the other hand are tuned to correct gamma encoded images back to linear form. This means that in displays images are multiplied with exponential function. Hence raw images needs to be gamma encoded before showing on display. [27]

3 SIMULATION AND EVALUATION

Above was described some basic knowledge about image capturing and processing. This knowledge is needed to be able to properly evaluate focus curves. More specifically to evaluate how easily some AF algorithm could find the best focus point from focus curves achieved with differently ordered image processing pipes. Here are studied impacts of executing different processing blocks of the pipeline before AF statistics calculations. The work is divided to 6 parts. First part is about effects of different size AF blocks, of which focus values are calculated. In second part effects of adding noise to images are studied. This is done to study how low quality images affect to focus curves. Next is studied how noise reduction affects to images with and without added noise. Next is concentrated on scaling part, where scaling effects of images with and without added noise are studied. 5th part is about effects of color correction done to images with and without noise. In final part is studied how size of used AF filter affects to focus value calculations. These parts can be also seen from the pipeline in figure 18. The pipeline describes all the processing steps, which eventually converts images to focus curves.

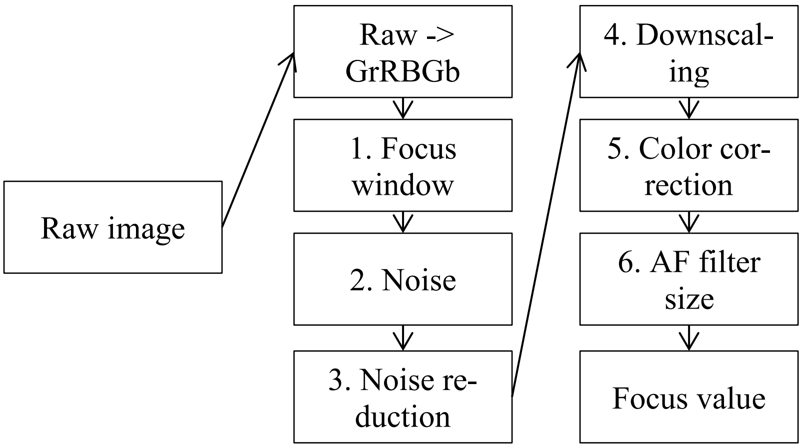

***Figure 18** Processing pipe studied in this thesis.*

To achieve these goals smartphone was used to capture raw images in laboratory. It's worth mentioning, that used smartphone wasn't final consumer product. It was prototype, which means that some functions may not work as should. However any major problems weren't encountered. In laboratory it was possible to control the illumination level. To eliminate effects of shaking hands and keep the scene as constant as possible, camera was attached to tripod. Images of different scenes were captured in high and low illumination with every possible focus lens position. Images were captured with command line script to minimize differences between captured images. However when illumination level was changed sometimes phone needed to be woken up, which may have caused little alterations to captured scene between low and high light situations. After all it shouldn't cause much interference to evaluation of focus curves.

Capturing images with all possible lens positions resulted to 166 images per scene in certain illumination. In camera this meant that focus lens position was altered with values from 50 to 215. Those same values are used, when focus curves are later presented in this thesis. With used values it was possible to capture sharp images of scenes, whose distance was from 10cm to 2m. Camera is able to take sharp images from objects even further. With such a small camera module light beams coming from objects located two meters and further away from camera are already parallel, when focus lens is set to maximum. To see how real noise affects to AF performance, focus curves of scenes captured in low illumination were also evaluated. After raw images were captured, images were processed with MATLAB in a certain way and focus values were calculated.

Scenes are called AF box, Siemens star, light studio and barcode. These scenes are presented in figure 19 from top to bottom in same order.

Figure 19 *From top to bottom downscaled images of scenes AF box, Siemens star, light studio and barcode. From left to right are presented in-focus image in high illumination, in-focus image in low illumination and out-of-focus image in high illumination. Later in this thesis images in first column are called noiseless.*

All other scenes than barcode are provided by German company Image Engineering [28]. Even though AF box is actually name of the light box, same name was used as the name of the scene. Real name of the scene in Image Engineering's database is TE261 Slanted Edges. Siemens star can be found from Image Engineering's database with name TE253 Modulated Sinusoidal Siemens Star. Images of Siemens star were captured with two different amounts of 55W fluorescent lamps. Those lamps were physically detached to each other. Color temperature of lamps was 4500 K. Light source was positioned to room in a way to get desired illumination level at the point of camera, when camera is pointing to scene. Light studio is light box called lightSTUDIO, which consists of different physical three dimensional objects and a background. Every object has its own purpose and those objects help in determining image quality. Barcode is simple piece of cardboard. On that cardboard is glued a sticker, where barcode is printed on. Barcode mainly has straight vertical lines. Idea was to study if this kind of image with horizontally high frequency content has some special effects in studies. Especially scaling was expected to give some interesting results.

In table 1 is presented measured illumination readings of scenes. In AF box used illuminant is CIE's standard illuminant D50. Images of barcode were taken inside light box lightSTUDIO and illumination of light box was utilized. For this reason both light studio and barcode scenes were captured in CIE's standard illuminant F12. Barcode's difference in measured illumination was because light of the light box didn't hit the barcode so well. Illuminant doesn't affect much to focus curves if there is enough light. However it might be useful information for someone who is trying to repeat the study.

Table 1 *Illumination readings of captured scenes measured with Konica Minolta's chroma meter CL-200A. Measures were taken from the location of camera and chroma meter was pointed to the objects direction. Ev represents the level of luminance, T the color temperature, Δuv the distance from the Plancian locus and x and y the coordinates in the CIE xy chromaticity diagram [29].*

	Ev [Lux]	T [K]	Δuv	x	y
AF box	14.5	5027	0.0014	0.3446	0.3541
	1035	4908	-0.0008	0.3476	0.3954
light studio	17.7	2963	0.0016	0.4418	0.4093
	1016	2876	-0.0039	0.4393	0.3943
Siemens star	8.1	4852	0.0044	0.3506	0.3648
	1226	4937	0.0038	0.3478	0.3614
barcode	12.2	2801	-0.0041	0.4445	0.3952
	355.4	2772	-0.0019	0.4501	0.4020

In figure 19 all used scenes are presented in scaled size to fit the page. Three images of each used scene are presented. For each scene first is presented in-focus image in high illumination, then in-focus image in low illumination and finally out-of-focus image in high illumination. Later in this thesis these in-focus images in high illumination are called noiseless, even though those images aren't in reality noise free. Images were captured in raw format with flash disabled, while all other settings set to automatic. After that normalization was done to images according to formula 14 in page 30. Next images were divided to four color components. Finally to visualize images, images with four color components (GrRBGb) were converted from to images with three color components (RGB). Two green components Gr and Gb were combined to G by calculating pixel-wise average of the two green channels. RGB images were still gamma corrected by raising every image value to the power of 0.45, to approximate standard RGB (sRGB) color space gamma. sRGB is widely used colors pace, which is standardized by International Electrotechnical Commission (IEC) [30].

Because mosaic images are too big to fully fit on screen, images are downscaled by using nearest neighbor interpolation, which basically chooses nearest pixel value. Single images are however presented in a full size. This is done to present lots of images in small space. Images are still big enough to present the changes and effects. Word may still further downscale images, which are too big to fit the page. Later in this thesis images processed in above presented way are called unprocessed, because these steps are pretty much mandatory to show RGB images correctly. Because images are unprocessed, they look little distorted. As can be seen from figure 19 these images aren't still processed properly. For example their green color component is still too strong. This is usually corrected with proper white balance and color correction algorithms. It's also very visible that AE compensates differences in lighting conditions by altering sensitiveness and the amount of light getting to sensor. If AE values were fixed in a way that they provide good images in high illumination, images taken in low illumination would look much dimmer.

Later on in this thesis captured images are visualized in a way presented above. However actual focus values are calculated before any visualization. Focus values of images are calculated from images for which only desired image processing steps are done.

After capturing all images of scenes, a pipeline was designed. The pipeline consists of different image processing blocks. It was important to build a pipeline, where it's possible to choose just desired processing blocks at the time. MATLAB was used to model the pipeline, because it's handy tool in this kind of studies. In final block focus values for each differently focused image are calculated. These focus values can be plotted to achieve focus curve. From focus curve AF algorithm tries to find the focus point, point where images are as sharp as possible. However real AF algorithm wasn't implemented, because it's studied how well generally some AF algorithm could find the best focus point from focus curves.

In scenes there is usually just one object in the ROI. For that reason curves usually have one bigger spike in in-focus areas and curves are more flat in out-of-focus areas around that spike. In figure 20 two examples of possible focus curves are presented. As can be seen the left focus curve is way better compared to right one. Left focus curve can be said to be pretty much ideal focus curve of practical scenario. To achieve such focus curves, curves need to be normalized by dividing focus values of curves with maximum focus value according to formula

$$\hat{F} = \frac{F}{\max(F)} \tag{13}$$

where F is vector of focus values. Meaning that after normalization maximum amplitude is 1 and minimum something between 0 and 1.

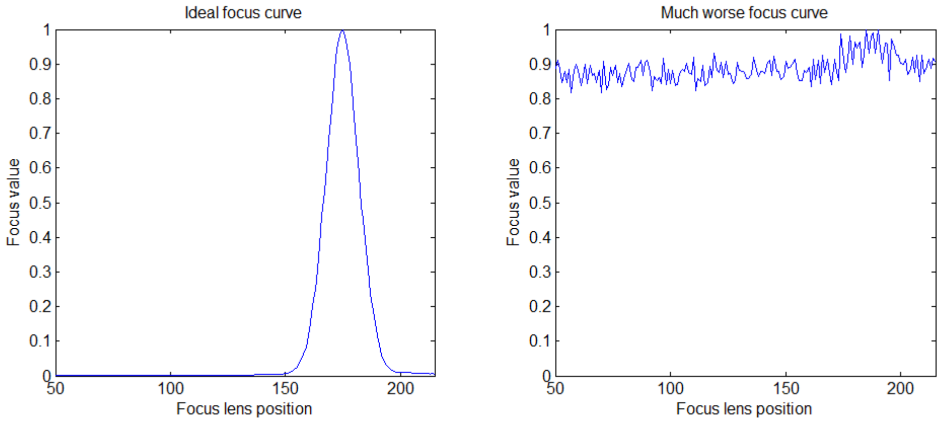

Figure 20 *Two examples of possible focus curves. Left one is ideal for real scene and second one much worse but still possibly recognizable for AF algorithm.*

In chapter 3.1 the designed pipeline is presented. There different processing steps are introduced in more detail. In chapter 3.2 is presented designed goodness criteria for evaluating the impact of different processing steps before focus calculations. In next figures outputs of different processing steps are visualized with image of light studio captured in high illumination (see figure 19). That image is chosen to be used in visualizations, because it has much color content and a lot of small details.

3.1 Pipeline

Focus curves are studied with a pipeline presented in figure 21. The pipeline consists of 7 different image processing blocks. First block is about converting raw image to GrRBGb image, which is always necessary before any further processing can be made. Second block is about choosing the size of focus window. This means that certain amount of pixels from the center of the image is used in further processing. Other pixels are discarded. In third block noise is added to image to study how AF algorithm performs with low quality images. In fourth block is studied how much easier noise reduction makes the process for the AF algorithm. Fifth block is about downscaling of the spatial resolution, meaning decreasing the amount of pixels in image. In sixth block color correction is done to image. In seventh block filter size of high pass filter is determined. High pass filter is used to calculate the edge content of image. As output of the processing block edge content is combined to single focus value. It’s important to notice that some of these processing blocks may alter image values outside the normalized values from 0 to 1. In these cases values under and over are truncated to 0 and 1. In figure 21 these processing blocks are presented in visual form.

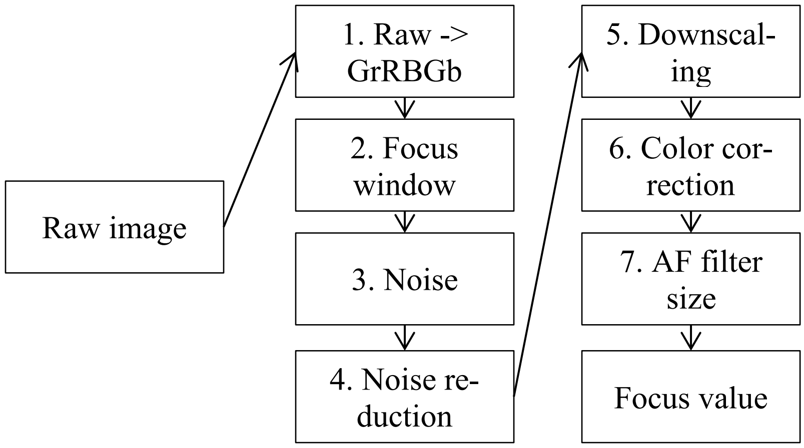

Figure 21 *Processing pipe studied in this thesis.*

How image is processed inside each processing block can be altered. However as mentioned earlier in this thesis, idea is to choose some of the blocks to be executed before focus value calculation. When studying certain effect on focus value calculations, all differently focused images should be run through the pipeline by executing same blocks. Next different stages of the pipe are explained in more detail. It's also explained how these processing blocks are used in studies.

3.1.1 Raw to GrRBGb conversion

First images are transformed from one-dimensional raw data to two-dimensional raw images. From those images 4 color components are extracted. These images are called GrRBGb images, where letters tell the order of color components in Bayer filter. Because photo sensors of camera can just sense the amount of incoming light, not the color, incoming light needs to be filtered to bypass just desired wavelengths. Bayer order tells, in which order these color components are presented in raw image. Used camera provides raw data, which consists of header and data of raw image. All this data are in one vector, which is converted to GrRBGb image. Raw images are two-dimensional matrices with minimally processed data from sensor. In the matrix values are coded with certain number of bits in certain layout. These matrices are often called Bayer mosaics. Used camera forms pixels from 2x2 blocks with 2 green values, 1 red and blue value as in figure 22.

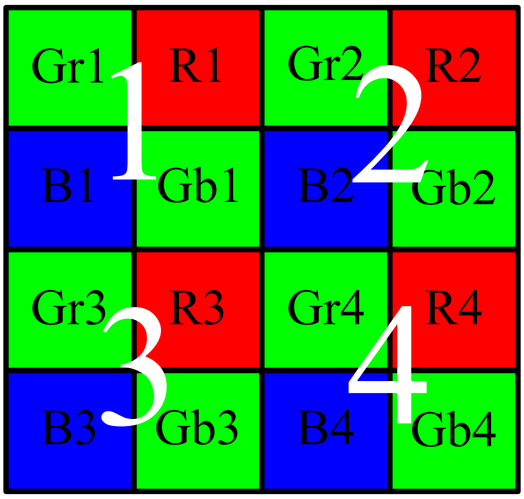

Figure 22 *Example of Bayer mosaic.*

The studied camera codes raw images with 10 bits per pixel. To prevent noise in dark areas of image, digital cameras have certain black level. Black level depends on the sensor design. Below black level single pixel of sensor can't give accurate results. With such a low levels noise amount becomes relatively too high. For example thermal noise could cause pixels to give very different values below black level, even though all incoming light would be blocked. In used camera black level is set to be 42. This means that raw images should have values between 42 and 1023.

To better utilize the bits and possible light levels when raw images are processed, black level is removed from raw images and image is scaled between certain values. In this study images are normalized between values 0 and 1 according to formula

$$I(x,y) = \frac{(I(x,y)-BL)}{maximum-BL} \qquad (14)$$

where x and y are coordinates of image, I is image, BL is black level and $maximum$ is maximum value pixel can get. If black level would be subtracted from image without any scaling, saturation point would change. Improper values are truncated to maximum or minimum. These values are mostly values below black level because of noise. After normalization color components Gr, R, B and Gb are constructed according to figure 22. After transformation image is ready for further processing. This processing block is necessary to execute for every processed image.

3.1.2 Focus window size

Focus window is certain size window at the center of image. Only pixel values belonging to that window are kept, all others are discarded. In figures 23 and 24 are examples of those windows. In this study used focus window sizes are 10%, 20%, 30%, 40%, 50% and 100% of the full image.

Figure 23 *Image where different size of focus window sizes are presented. Window sizes are 10%, 20%, 30%, 40%, 50% and 100% of the full noiseless image.*

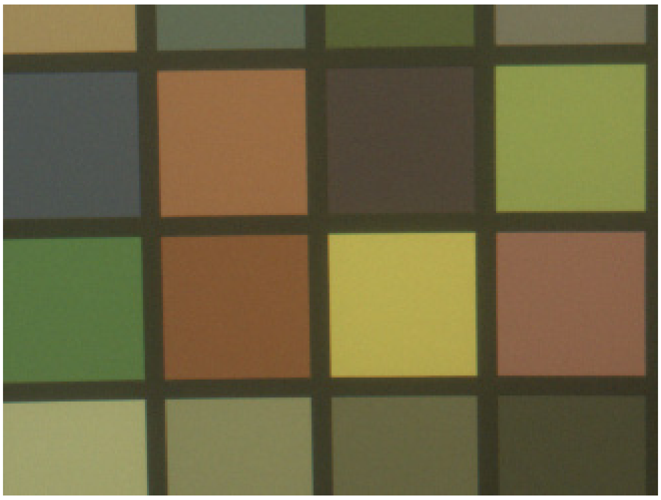

Figure 24 *20% focus window of figure 23. This image is used in next steps in visualizations to show smaller changes in bigger size. Later in this thesis this image is called noiseless block.*

When focus window size effects to AF performance are studied, from every image focus values are calculated with all presented focus window sizes. This way 6 focus curves from each well illuminated scene are achieved. As default in other studies 20% focus window is used.

3.1.3 Adding noise

At this processing block it's possible to add noise to images to test how AF algorithm performs with low quality images. However this processing block can also be skipped. Gaussian and impulse noises are added to images with different volumes. There are two different options. In first one is studied the impact of adding randomly generated noise pattern to every image. Gaussian noise effects are studied by adding Gaussian noise with three different noise variances. Variances are 0.0001, 0.0005 and 0.001. Impulse noise effects are studied by adding impulse noise with three different noise densities. Densities are 0.0001, 0.001 and 0.01. Noise patterns are added to images by using MATLAB built-in function *imnoise*.

In second option the impact of constant noise patterns are studied, meaning that same noise pattern is added to every image. These patterns were created with same noise variances and densities as in first option. Noise patterns are created manually in the same way imnoise creates them. These patterns are created only once. After patterns are created, patterns are added to every image.

Figure 25 consists of 12 downscaled images. In first two rows are examples of Gaussian noise. In last two rows are examples of impulse noise. When only noise pattern is presented, it's visualized by taking absolute values to achieve just positive values.

As default noise isn't added to images. However while studying effects of other processing blocks, usually effects of adding noise is also studied. Noise effects of constant noise patterns were studied only in noise section. Low illumination images of scenes were studied only in noise and noise reduction studying sections. Low illumination images have already real noise. For this reason no artificial noise is added to those images.

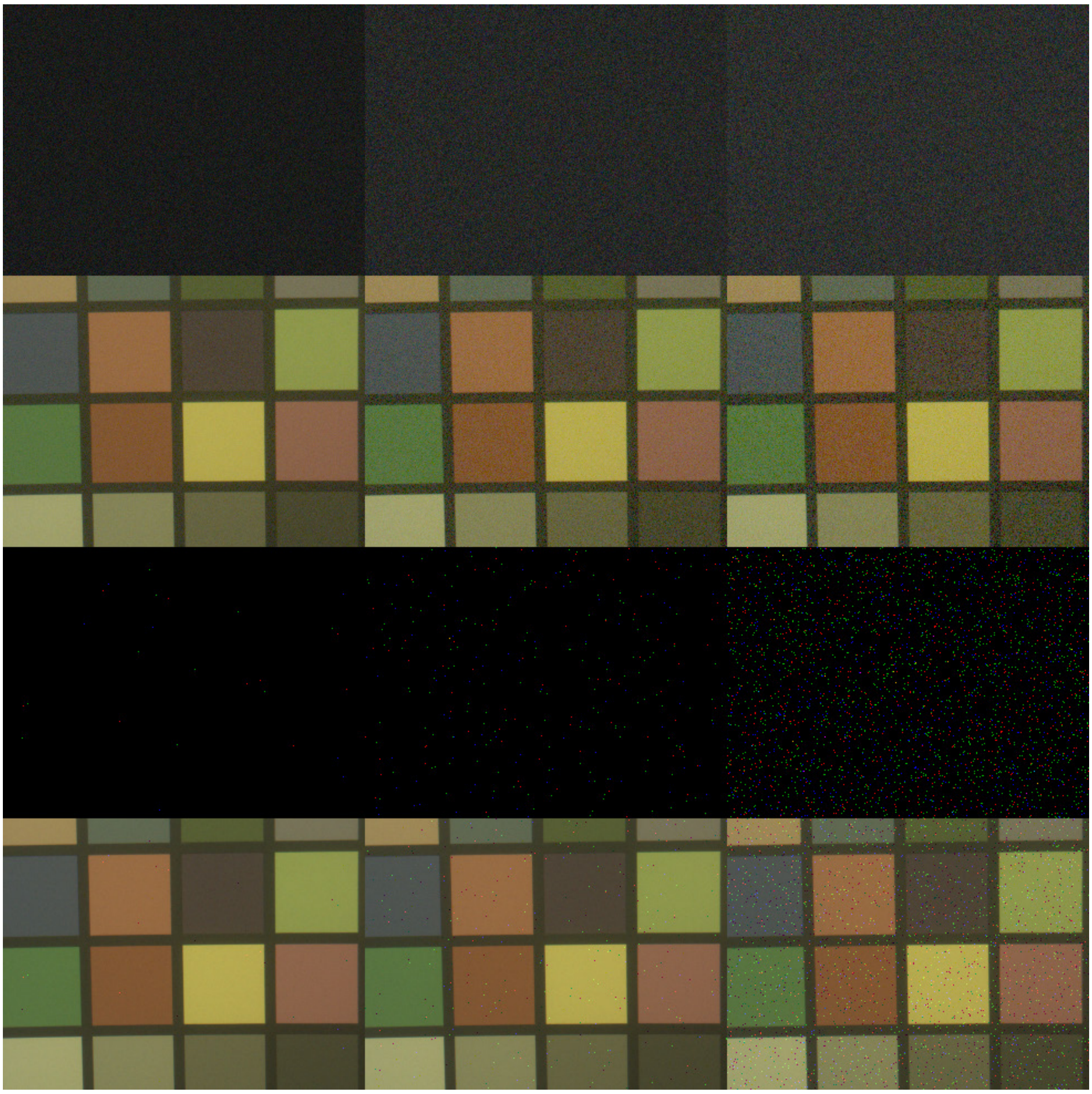

***Figure 25** Samples of noise patterns and images with noise patterns added. When just noise patterns are presented, they are visualized by taking absolute values of real values. From left to right top two rows present Gaussian noise with 0.0001, 0.0005 and 0.001 variances. Lower two rows present accordingly impulse noise with 0.0001, 0.001 and 0.01 noise densities.*

3.1.4 Noise Reduction

In fourth step effects of reducing noise can be studied. 4 different noise reduction algorithms are used: median filter, mean filter, median & mean filter and BM3D. Idea is to study how very simple filters like median and mean affect to focus curves and how much it differs from much more complex algorithm. Also effects of combining median and mean filters are studied. There mean filtering is executed after median filtering. BM3D is complex and very powerful algorithm for Gaussian noise reduction [18]. It was used, because of its effectiveness and already implemented MATLAB code, which is publicly available for non-commercial use. Median filter was implemented by using

MATLAB built-in function *medfilt2* with default parameters. By default *medfilt2* calculates median values from 3x3 blocks. Mean filter was implemented by using MATLAB built-in function *imfilter* with 3x3 averaging matrix as parameter. BM3D was executed with default parameters. By default noise reduction algorithms aren't used.

Performance of these algorithms can be seen in figures 26 and 27. In figure 26 are presented noise reductions for noiseless image. From figure it's easy to say that for noiseless images, noise reduction algorithms don't have big impact on the quality of image. The impact is still visible, images are smoother. Mean filtering smoothes also the edge content, which causes some blur to image.

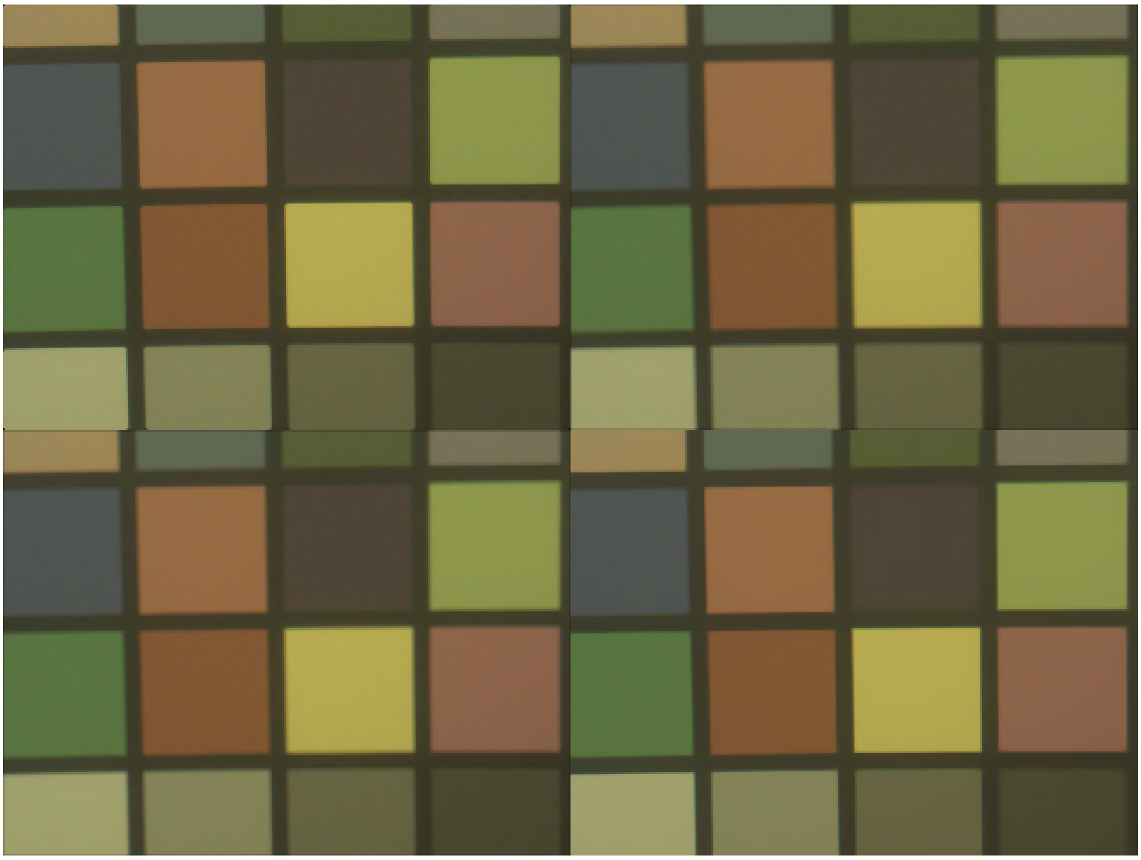

Figure 26 *Output of noise reduction algorithms for noiseless block. From left to right and top to bottom algorithms are median, mean, mean & median and BM3D.*

In figure 27 are presented images with Gaussian noise of variance 0.001 and impulse noise with noise density of 0.01.

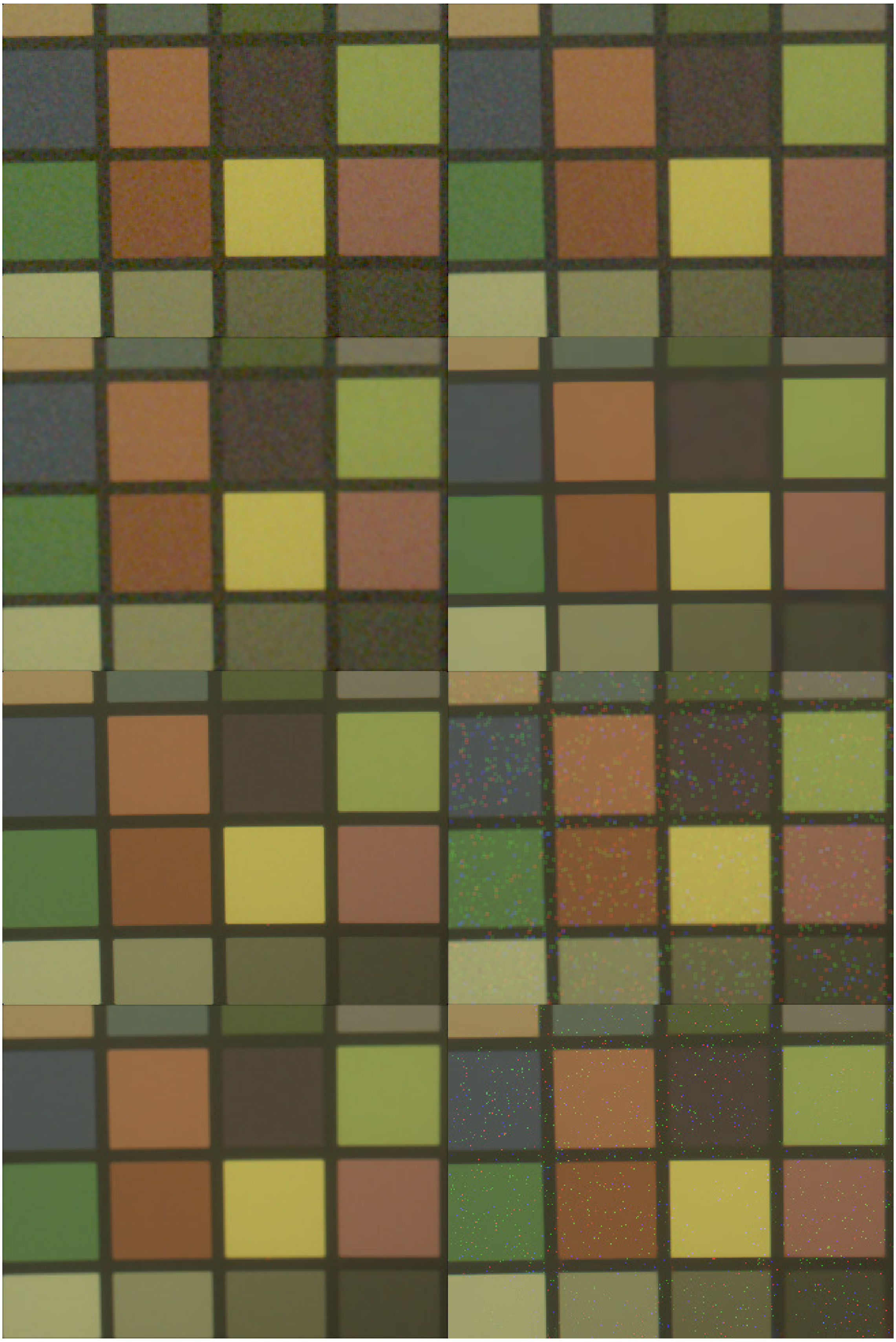

Figure 27 *Images with different noise reduction done. Images in two first rows are presented as in figure 26 with added Gaussian noise of variance 0.001 and images in two last rows as in figure 26 with added impulse noise of density 0.01.*

In case of Gaussian noise BM3D's superiority is clear. It reduces very well noise without losing important edge information. In impulse noise case median filter is simply the best studied algorithm. Mean filter decreases intensity of the noise, but spreads the noise to wider area. BM3D is capable of reducing little noise, but it won't smooth the edges of image. With lower noise densities BM3D starts to perform relatively much better.

3.1.5 Downscaling

In downscaling stage effects of three different scaling methods were studied: subsampling, pixel binning and bicubic scaling. Effects of downscaling noisy images were also studied. However by default no downscaling is done.

Effects of scaling can be seen from figures 28 and 29. To better illustrate differences caused by scaling, downscaled images are upscaled to correspond original size of images. Upscaling is done with nearest neighbor interpolation. As can be seen from figures, bicubic scaling gives the most pleasant images, however it blurs images slightly. Pixel binning spreads edges and blurs images even more. Subsampling on the other hand doesn't make images so blur, but instead of blurring it causes aliasing and some deviations.

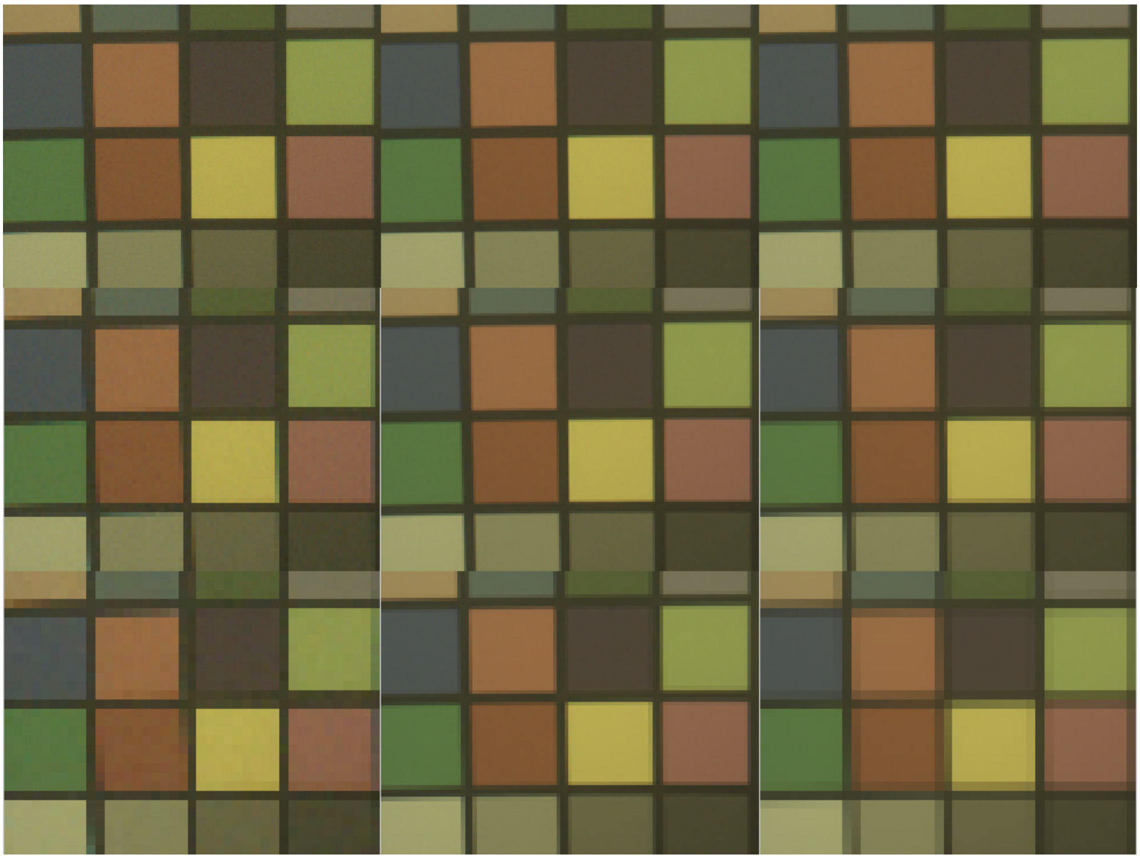

Figure 28 *Output of downscaling algorithms for noiseless block. From left to right scaling methods are subsampling, bicubic scaling and pixel binning. From top to bottom subsampling factors are 2, 4 and 8, scaling multipliers are 0.75, 0.5 and 0.3 and binning factors are 2, 4 and 8.*

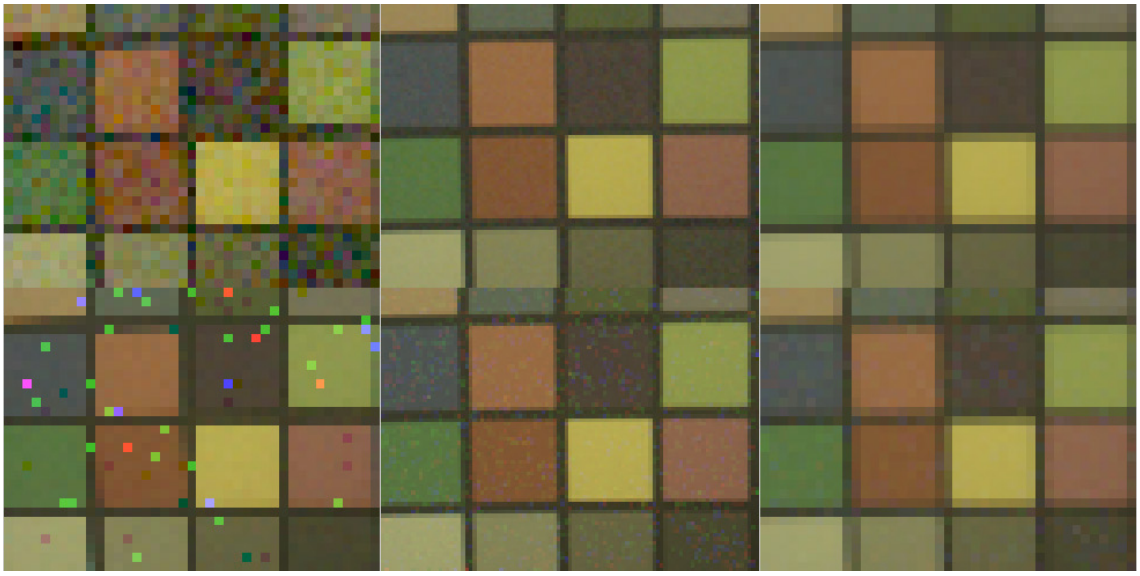

Figure 29 Downscaling methods executed for noisy images. In first row Gaussian noise is added with variance 0.001. In second row impulse noise is added with density 0.01. From left to right used downscaling methods are as in figure 28.

Subsampling was studied with subsampling factors 2, 4 and 8, bicubic scaling with scaling multipliers 0.75, 0.5 and 0.3 and pixel binning with binning factors 2, 4 and 8. It's also important to notice that subsampling factors and binning factors correspond to each other, but scaling multipliers are different. Scaling multipliers were chosen differently to test for example if there are differences when using non uniform scaling. Reciprocal values of scaling factors are corresponding better subsampling and binning factors. However these values aren't necessarily integers. Scaling multiplier 0.5 is corresponding subsampling and binning factor 2.

By looking figures blurring that averaging of bicubic scaling and pixel binning can be easily noticed. It's also easy to see that subsampling makes noisy image subjectively pretty poor in quality.

3.1.6 Color correction stage

Used color correction algorithm assumes that images are in RGB format. If color correction is done GrRBGb images need to be converted to RGB images. This is done by averaging two green components.

After conversion white balancing is done. White balance gain coefficients are calculated. GW algorithm, which assumes, that scene has equal amount of R, G and B on average, is used [22; 23]. This algorithm can give pretty weak results if in reality some color component is very dominant in scene. However in this thesis it is studied how much AWB affects to focus curve, hence interest is not in the performance of AWB algorithm. For calculating white balance coefficients, formula

$$gainR = \frac{average(G)}{average(R)}, gainB = \frac{average(G)}{average(B)}, gainG = 1 \tag{15}$$

$$gainR_c = \frac{gainR}{\min(gainR,gainB,gainG)} \tag{16}$$

$$gainB_c = \frac{gainB}{\min(gainR,gainB,gainG)} \tag{17}$$

$$gainG_c = \frac{gainG}{min(gainR,gainB,gainG)} \tag{18}$$

is used, where *R*, *G* and *B* means red, green and blue color components. Gains are multipliers of original image and gains with lower c are multipliers used for color correction. Each color component is multiplied with corrected multiplier. After white balancing color are corrected with color correction matrix. For color correction formula

$$CCM = \left(\begin{pmatrix} 2.06 & -0.77 & -0.29 \\ -0.40 & 1.56 & -0.16 \\ -0.07 & -0.52 & 1.59 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right) * m + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{19}$$

is used, where m is color correction multiplier used to scale values and CCM is color correction matrix. These values are chosen for color correction matrix, because those values work well in used smartphone for color correction of images taken in natural day light. However it doesn't matter much which values are used, because idea is to test if some color correction makes any differences in an AF algorithms performance. As CCM multiplier m it was decided to study values 0, 1 and 2. Bigger CCM multiplier alters more colors. Color components are multiplied from right according to matrix multiplication rules. Multiplication is broken down in formula

$$\hat{R}(x,y) = CCM(1,1) * R(x,y) + CCM(1,2) * G(x,y) + CCM(1,3) * B(x,y) \tag{20}$$

$$\hat{G}(x,y) = CCM(2,1) * R(x,y) + CCM(2,2) * G(x,y) + CCM(2,3) * B(x,y) \tag{21}$$

$$\hat{B}(x,y) = CCM(3,1) * R(x,y) + CCM(3,2) * G(x,y) + CCM(3,3) * B(x,y) \tag{22}$$

where *x* and *y* are row and column indexes of color component. In *CCM*(i,*j*) *i* and *j* are row and column indexes. $\hat{R}$, $\hat{G}$ and $\hat{B}$ are corrected color components. Every time when color correction is executed, white balancing is done before multiplying with CCM. When CCM multiplier is 0, only white balancing is executed. Anyhow by default no white balancing or color corrections are done.

From this point on whole white balancing and multiplication with color correction matrix is called color correction. In figures 30 and 31 can be found effects of white balance and color correction.

Figure 30 *Output of white balance and color correction algorithms for noiseless and noisy images. In top row are noiseless images. In bottom row are images where Gaussian noise is added with noise variance 0.001. From left to right color correction multipliers are 0, 1 and 2.*

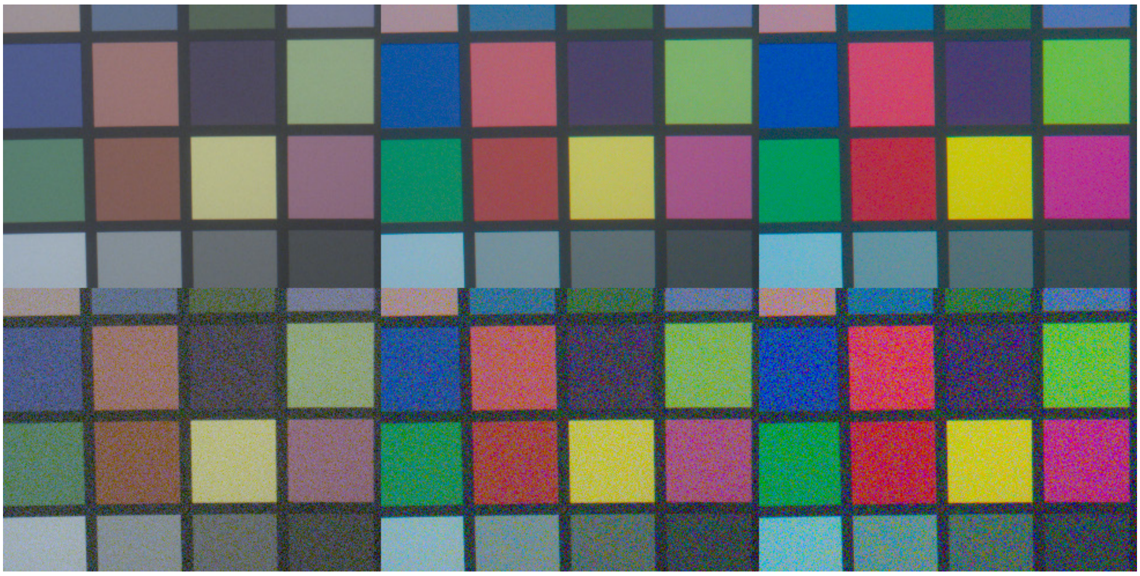

Figure 31 *20% window images of figure 30 images.*

By looking images it can be seen that already white balancing makes images look much better, but colors are still pretty dull. Using color correction multiplier 1 makes images look already more natural and with multiplier 2 in some color batches of color chart some color components are saturated. Usually humans like more saturated images, because colors are warmer and stand out better. This phenomenon happens even though images aren't representing the actual scene accurately anymore. In figure 30 lens vignetting is visible and color correction even emphasizes it. When bigger color correction multiplier is used also noise is emphasized. Artificially added impulse noise isn't emphasized because values under and over 0 and 1 are truncated. However real impulse noise isn't necessarily 0 or 1 and that noise may also be emphasized. However this means that Gaussian noise is mostly emphasized.

Used CCM isn't best one for used illumination, however it doesn't matter. Main interest is in evaluating the effects of some CCM for focus calculations. This CCM is making blue color slightly too dominant, which can be seen by looking the lower row of color chart, all those batches should be achromatic, meaning those should be only different shades of gray. Cup and plates also give a good hint too of color error.

3.1.7 AF filter size

Filter size affects to the amount of pixels used in calculating the amount of focus in the center point of the filter. High pass FIR filter can be one or two-dimensional with certain filter order. Filters were designed to have stopband at normalized frequency 0, transition band from 0 to 0.8 and passband from 0.8 to 1. Values were chosen, because it is known that those perform well. However other values could have been used as well. Filter size needs to be always decided to calculate focus values. By default horizontal 7x1 filter is used. One-dimensional filters are traditionally more commonly used, because those are cost effective and simpler to implement than two-dimensional filters. Two-dimensional filters were chosen to this study to see how much those would affect to focus curves.

Used one-dimensional FIR filters are shown in figure 32 and their amplitude responses in figure 33. Filter length is obtained by increasing filter order by one. Filters are designed by using MATLAB built-in function *firpm*, which designs filter by Parks-McClellan optimal equirriple FIR filter design [31]. Only difference in two-dimensional filters is the dimension. Two-dimensional filters could be created by rotating 360 degrees one-dimensional filters along y-axis. Because two-dimensional filters are square (for example 7x7) instead of circular shape, some approximation and rounding need to be done. Used filter sizes are 7x1, 9x1, 11x1, 7x7, 9x9 and 11x11. From figures it can be seen that longer filter multiplies current pixel value with larger multiplier while neighboring pixels are multiplied with smaller negative multipliers. However more neighbors are used. In other words bigger filter uses more frequencies from transition band.

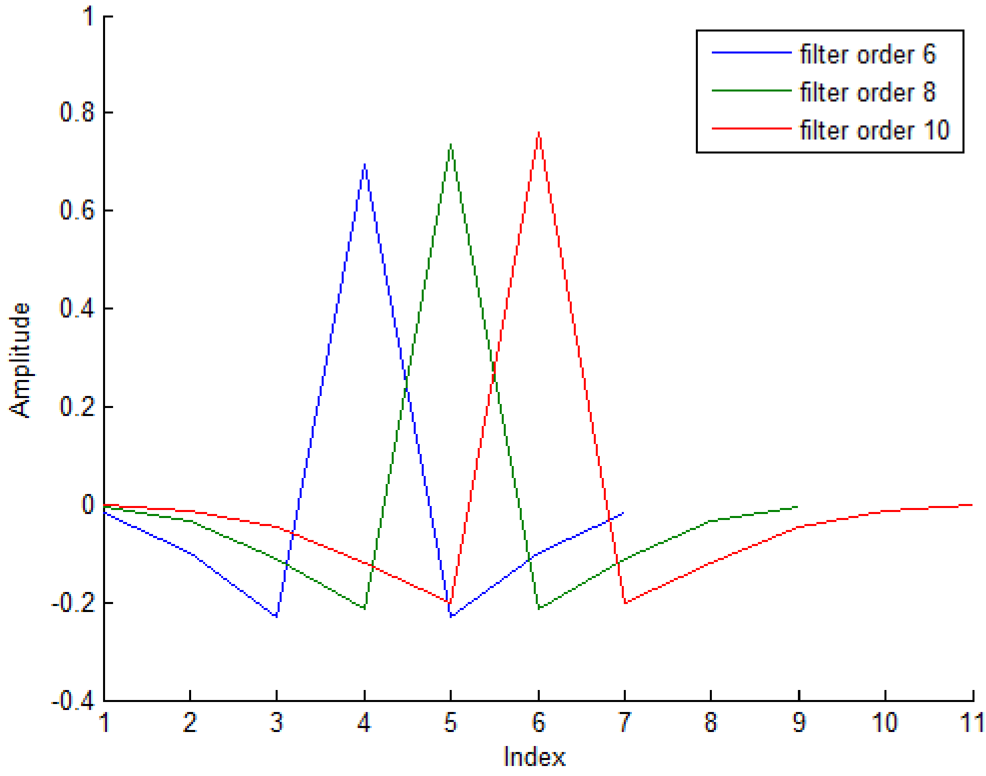

Figure 32 *One-dimensional FIR filters. Filter length is filter order + 1.*

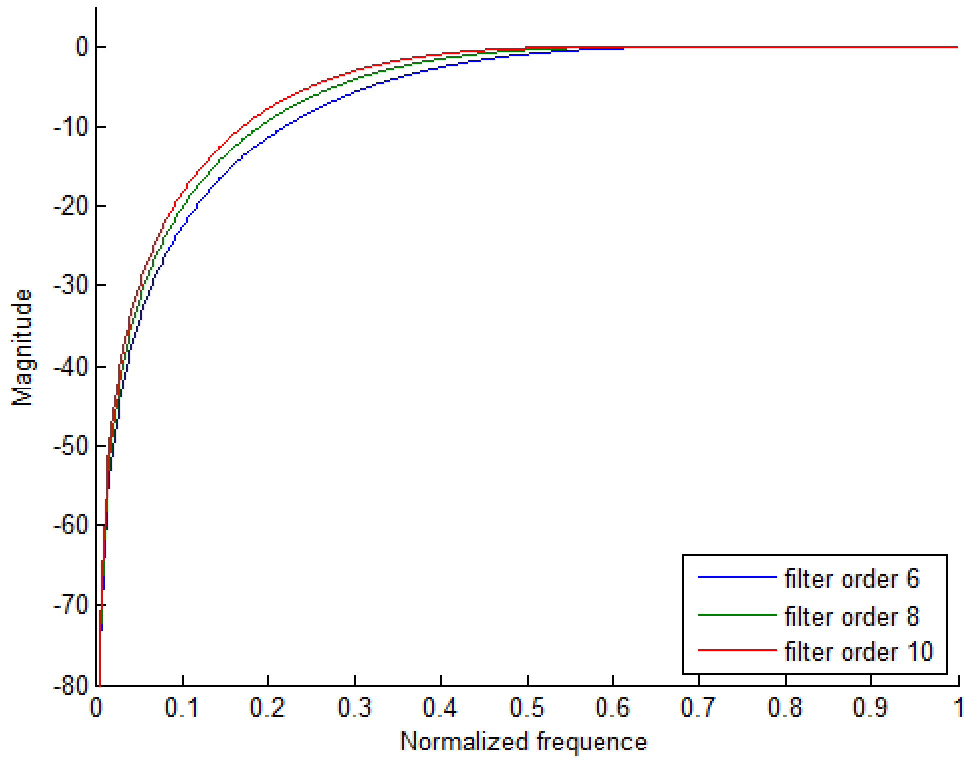

Figure 33 *Amplitude responses of one-dimensional FIR filters.*

Finally focus values are calculated by filtering pixels of focus window with designed high pass filter. These values are averaged to get single value, which describes the focus of the image. Example of this can be seen in figure 34. AF algorithm uses these focus values from differently focused images to decide the best focus lens position. Actual AF algorithm wasn't implemented. This study was about finding out how easily some AF algorithm could find the best focus point from focus curves.

Figure 34 *Example of filtering image with high pass filter of order 6 presented in figure 32.*

3.2 Focus curve goodness evaluation

To evaluate focus curve statistics, certain measures are done. In [3] some measures were combined from other researches. In the paper they modified slightly some of these methods and they also proposed some new measurements to better fit their needs. However the paper used these measurements to compare different AF algorithms, while target for this thesis is to compare how easy it's for some AF algorithm to make the decision of the best focus point.

Because approach in this thesis is slightly different, not all suggested measures could be used. However some of the suggested measures could be used as they were or with slight modifications. In addition in this thesis some other measures are proposed for evaluating goodness of focus curve. These measures are called goodness criteria. Used criteria are accuracy, peaks inside 90, noise level, peaks mean and level. For calculating final goodness value, some weighting is done to all criteria. Those weighted values are combined to one goodness value by calculating Euclidian norm out of them.

Because of nature of the used focus calculation algorithm, noise impact becomes relatively much smaller when there are much sharp edges in image. When image isn't sharp, it's blurry and noise creates relatively bigger differences between neighboring pixels. This means that usually AF algorithm should perform well even in cases where there is more noise, if there is big enough difference between in-focus images and out-of-focus images. If difference is small, even pretty small amounts of noise may distract AF algorithm. For that reason exponential weighting is used in final goodness score calculation. Values used in weighting are based on experimental testing. Weights were determined in such a way that they describe how probably AF algorithm will fail. Bigger value means more probable failure of AF algorithm. In next chapters rejected, approved and modified measures from [3] are presented. Also some new criteria are proposed for evaluating focus curves.

3.2.1 Rejected measurements

Rejected measurements from [3] are width at 50% maximum, weighted width at 50% maximum, width at 80% maximum, range, peak slope, number of false maxima outside range 50% and computation time. Idea of all width measures is to find out the width of the area where focus values first time decline below certain percentage of maximum or height (maximum-minimum).

All width measures were rejected. Lower values would mean sharper focus spike, which should be better for AF algorithm. However focus curves with high amount of noise, can give very good results to these measures, even though usually noisy focus curves should be worse. Idea of range measurement is also based on measuring some kind of width, which leads to rejection in this study. Range measures the width of two neighboring local minima around the local maximum.

Peak slope measure is based on sharpness of the very top of focus spike. Absolute difference of global maximum and both neighboring focus values are calculated. Final peak value is gotten by averaging those two values. This kind of measure isn't necessarily describing all the times very well how good the focus curve is. Also noise may have pretty big impact to this measurement.

Number of false maxima outside range 50% is counting the number of local maxima from values, which are lower than 50% of focus curve height. Idea is that AF algorithm may choose one of these local maxima as the right focus. However this measure isn't taken to account. It's presumed that that AF algorithm detects these as false maxima.

Because any specific AF algorithm isn't actually used, computation time measure can be rejected. It can be still agreed that it's very important that execution time of AF algorithm isn't too long and it should be taken to consideration when AF algorithms are compared against each other.

Normally AF statistics aren't calculated for every lens position, which probably reduces the amount of noise in focus curves. In this thesis some smoothing to the focus curves could have been done. It could have made some of these rejected measures much better for the scope of this study. However AF algorithm should take care of possible smoothing and it was left out of this study. Focus curves are studied as they come out of the smartphone after some basic processing.

3.2.2 Used criteria

In figure 35 is graphically presented how some of the used criteria can be measured.

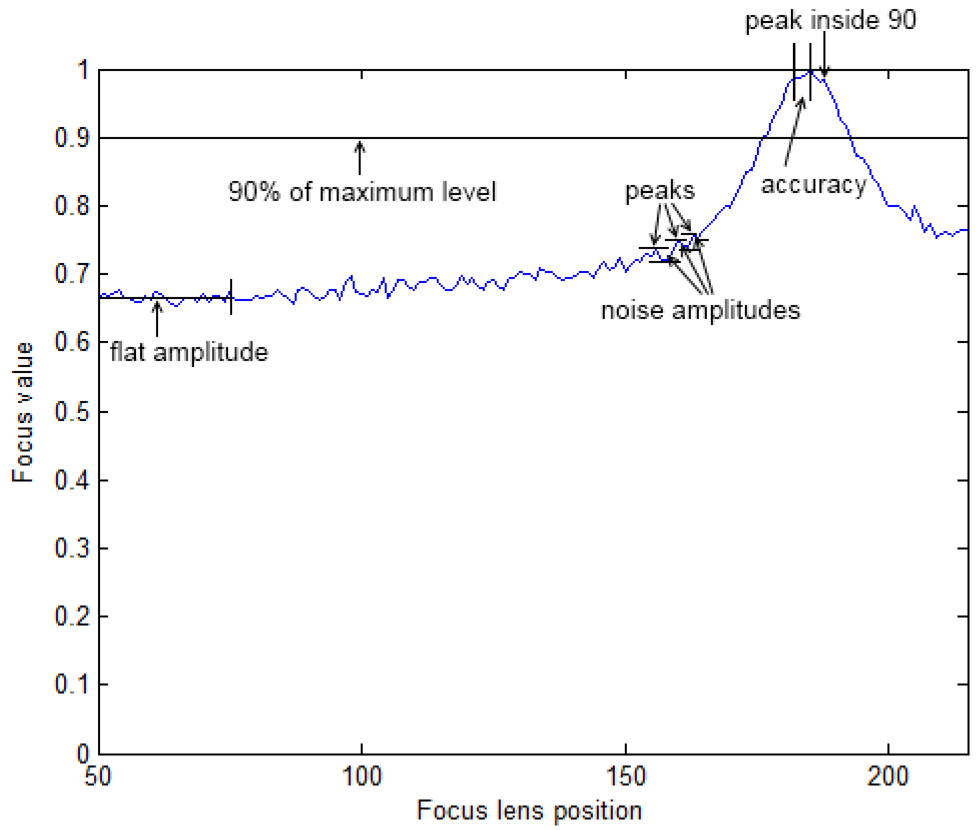

Figure 35 *Examples how to calculate some of the goodness criteria. Just a couple of peaks and noise amplitudes are drawn to image.*

Accuracy illustrates how well an AF algorithm can find the best focus point after certain image processing. It was also presented in [3]. To calculate accuracy, maximum focus point of unprocessed scene needs to be found. This focus point is called real focus and it tells at which focus lens position the sharpest image is achieved. For processed scene maximum focus point is searched and accuracy is absolute difference between this focus point and real focus.

Peaks inside 90 presents possibility that wrong focus lens position, that is still probably rather close to the real focus, is accidentally chosen as best focus point. This is almost the same measurement than in [3] proposed number of false maximum within range 20% measurement. Only the percentage level is modified. For this measurement, first needs to be found out focus points at both sides of maximum, where values are first time below 90% of the maximum. Peaks inside 90 is number of local maxima between these two focus points. 90% of maximum was chosen to be the limit, because higher peaks are more probable to cause problems for an AF algorithm.

Noise level presents how much noise there is in the focus curves and how much it affects in determining right focus lens position. Noise level measurement is also presented in [3], but it was further improved to better correspond needs of this study. Noise level is combination of two different calculations: noise amplitude and flat amplitude. Noise amplitude is almost corresponding noise level measurement in [3] and it's calculated by taking median of differences of focus values of adjacent local maxima and minima. Median is chosen instead of mean, because sometimes there are just few local maxima and minima. In these cases one local maximum in in-focus area may cause very big noise level even though there weren't actually much noise. Flat amplitude is average of 25 first focus values of focus curve, which presents magnitude of out-of-focus areas. 25 samples were chosen to be used, because in this study there isn't usually much increase in average amplitude within that number of samples. It's also enough samples to reduce biggest effects of noise. Noise level is multiplication of these two calculations with proper weights. It can be calculated with formula

$$noise\ level = ((w_{n1} * noise\ amplitude)^{w_{n2}}) * ((1 + flat\ amplitude)^{w_{n3}} - 1) \tag{23}$$

where w_{n1} , w_{n2} and w_{n3} are weighting parameters.

Peaks mean is criterion, which describes how probably AF algorithm accidentally chooses some local maximum as the focus point. It is simply calculated by taking average amplitude of all local maxima. It also tells something of overall magnitude of curve.

Level is criterion which resembles how easy it's for AF algorithm to find right focus spike. Level is calculated by subtracting flat amplitude from maximum focus value and then taking reciprocal of the outcome according to formula

$$level = \frac{1}{\max(FC) - flat\ amplitude} \tag{24}$$

where FC is focus curve.

3.2.3 Calculation of goodness

Finally goodness is calculated by calculating Euclidian norm of criteria with certain weights. Goodness describes the probably some AF algorithm finds the best focus point from focus curve. Slightly confusingly bigger goodness values mean that AF algorithm performs worse. More describing expression would be measuring weakness values. Nevertheless original expression goodness was decided to be kept. Weights were achieved through experimental testing. Experimental testing resulted to using weights $w_{n1}=80$, $w_{n2}=1.25$ and $w_{n3}=1.45$ would perform well in formula 23. Goodness is then calculated according to formula

$$goodness = \mathrm{round}\left(\sqrt{(A^{w_a})^2 + ((P90)^{w_{pi}})^2 + (N)^2 + \left((w_{pm1} * PM)^{w_{pm2}}\right)^2 + ((L)^{w_l})^2}\right) \tag{25}$$

where w are weights. After experimental testing weights, where w_a is 1.6, w_{pi} is 0.7, w_{pm1} is 5, w_{pm2} is 2 and w_l is 2 were chosen to be used. In formula A is *accuracy*, $P90$ is *peaks inside 90*, N is *noise level*, PM is *peaks mean* and L is *level*.

Based on this study it's possible to make assumption that it's pretty hard for an AF algorithm to perform reliably, when goodness value of focus curve is 40 or more. Usually quality of focus curves is good enough when goodness value is under 40. That assumption was made to classify good and bad focus curves. It's mostly based on careful investigation of focus curves getting values around 40. Usually curves over 40 have pretty low global maximum. Of course good AF algorithm could still find the right focus point, but this assumption has been made to evaluate the goodness of focus curves more generally. There isn't maximum value for goodness, because criteria level isn't restricted anyhow. With weighting included theoretical maximum values for other criteria are approximately following: accuracy is 3532, peaks inside 90 is 22, noise level is 414 and peaks mean is 25. This may give some idea what importance is given to certain criterion even though it doesn't tell anything about occurrence probabilities. It also means that without impact of level criterion, maximum goodness value is 3556. Minimum goodness value is 1, because level can't have lower value than 1. It's also worth mentioning that goodness values are rounded to integers. This was done, because goodness values are exponential and changes in decimals are meaningless with good goodness values.

4 RESULTS

In this section results are presented and evaluated. Results are mainly shown for Gr component. In color correction studies G component is chosen instead of Gr component. Because green is the most common color in nature and there wasn't big differences in performance between color components, green component was chosen to be used in calculations. However in very biased conditions some color component can perform superbly compared to others. If it can be afforded performance of each color component should be calculated and the best one should be chosen for calculations. Here some generalizations based on the acquired results are presented if possible.

Results of studies are presented in following order: AF block size, noise, noise reduction, scaling, color correction and filter size. For some reason focus curves of light studio images have big fluctuations around focus lens position 150. By carefully investigating images around focus lens position 150, it was possible to see some bigger alterations in noise levels. However no reason has been found for this behavior. Because phone is prototype there might be some problems in camera hardware or software, which causes this kind of action. Other possible cause for the action can be inconsistent conditions in laboratory, because some of the image capturing process happened without constant supervision. For this reason it's also possible that someone has altered illumination in laboratory. However even smaller alterations in illumination should be better visible in captured images. Because this wasn't critical for evaluating AF performance, it was decided not to recapture those images.

To make comparison of goodness values fair between different studies random noise patterns were generated with certain parameters for every possible lens position. This meant that 166 noise patterns were created for each different noise type with different amounts of noise. These patterns were stored and used in all studies, where noise effects were tested. It should be kept in mind that goodness criteria aren't perfect but more like good approximate. Those contain also some margin of error. Also when comparing good focus curves against each other goodness criteria doesn't make such a big difference in value even though actual curves might have noticeable difference. When results are reviewed also some curves are presented to show the real differences and behavior of curves.

In next chapter are presented the most interesting results of this study. In tables are presented little bit more results. In figures are shown some focus curves. In the focus curve figures goodness values are also presented. G is goodness, A is accuracy, PI is peaks inside 90, N is noise level, PM is peaks mean and L is level.

4.1 Focus window size

After raw images are converted to GrRBGb images, only size of focus window is altered. With bigger focus window sizes (mostly with 100% focus window size) vignetting affects to images, because lens shading correction (LSC) isn't done. However it's affecting to every differently focused image almost equally and it's not necessary to execute LSC. In the test pipe it's taken care of that there is enough room to use the biggest studied filter without border effects when different focus windows are used. This means that actually slightly bigger focus windows are used, except in case of 100% focus window, in processing steps. Extra pixels are discarded after performing final high pass filtering before averaging of all values.

By comparing goodness values, curves and images behind those, it becomes clear that content of the image affects heavily on the performance of differently sized focus windows. At certain distance, more there is high frequency content inside focus window, easier it's to find out the best focus value. That happens because when image is defocused, it's so blur that there aren't really any edges. On the other hand, when reaching best focus point edges come more and more visible. These effects can be seen in figure 36.

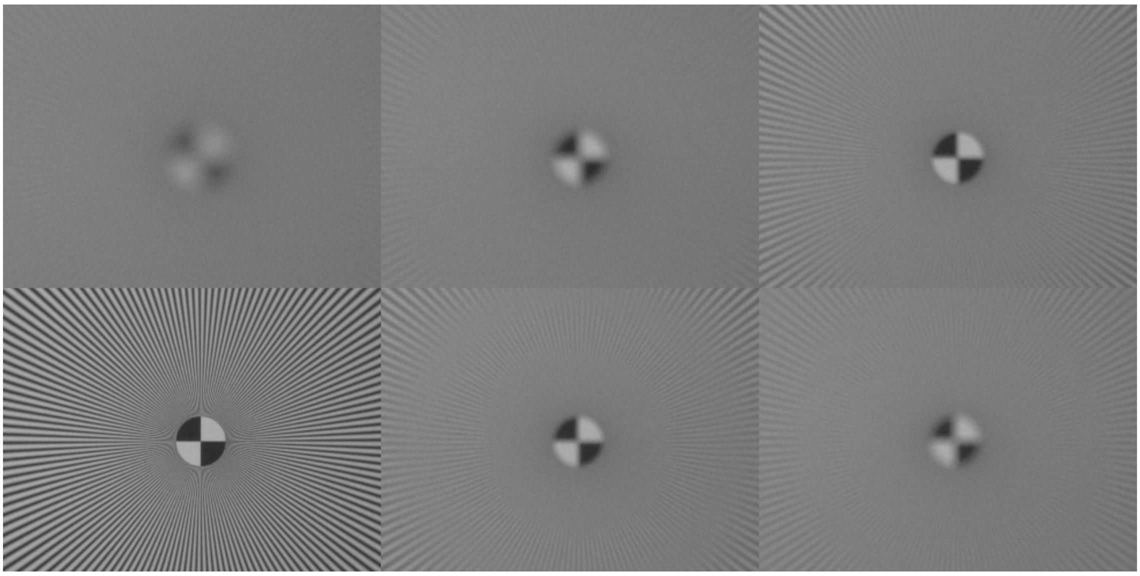

Figure 36 *Images of 20% focus window of Siemens star in high illumination. From left to right and top to bottom focus lens positions are 50, 135, 155, 175, 195 and 215. Best focus is achieved at focus lens position 175.*

In the test scenes biggest difference in goodness value were between 10% and 100% focus windows with barcode as can be seen from figure 37.

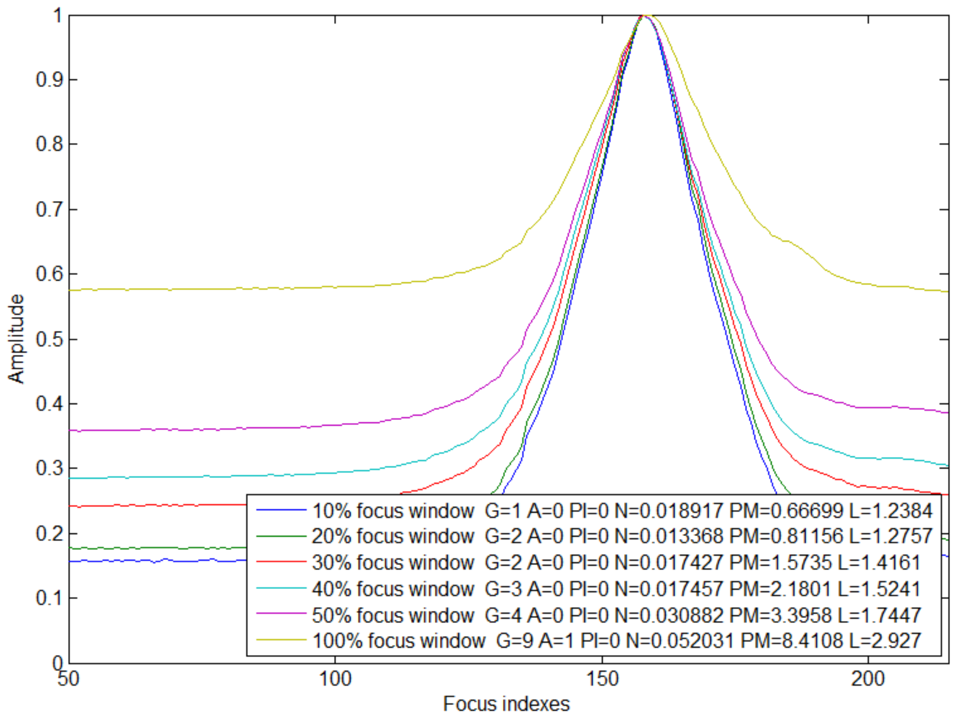

Figure 37 *Focus curves of barcode with different focus windows.*

Also Siemens star had little bit higher difference between the worst and the best focus window sizes. That's mainly because both scenes have highest frequency content at the center of the image, which comes more dominant with smaller focus windows. Nevertheless all focus curves of both scenes are still very good.

In AF box and light studio scenes differences between worst and best block sizes weren't very big. In those two scenes both the best and the worst focus windows weren't the two studied extremities 10% and 100% as can be seen from table 2 and figure 38. This also refers to the fact that content inside the focus window has big effect on focus calculations. In figure 38 the earlier mentioned fluctuation in light studio's focus curve can be easily seen. In table 3 is presented how well scenes perform with 20% focus window size. As can be seen all scenes perform well in good illumination with 20 % focus window size.

Table 2 *Goodness values of light studio with different focus window sizes.*

Variable	Goodness
10% focus window	13
20% focus window	14
30% focus window	13
40% focus window	12
50% focus window	12
100% focus window	11

Table 3 *Goodness values of different scenes with 20% focus window.*

Scene	Goodness
AF box	15
barcode	2
light studio	14
Siemens star	1

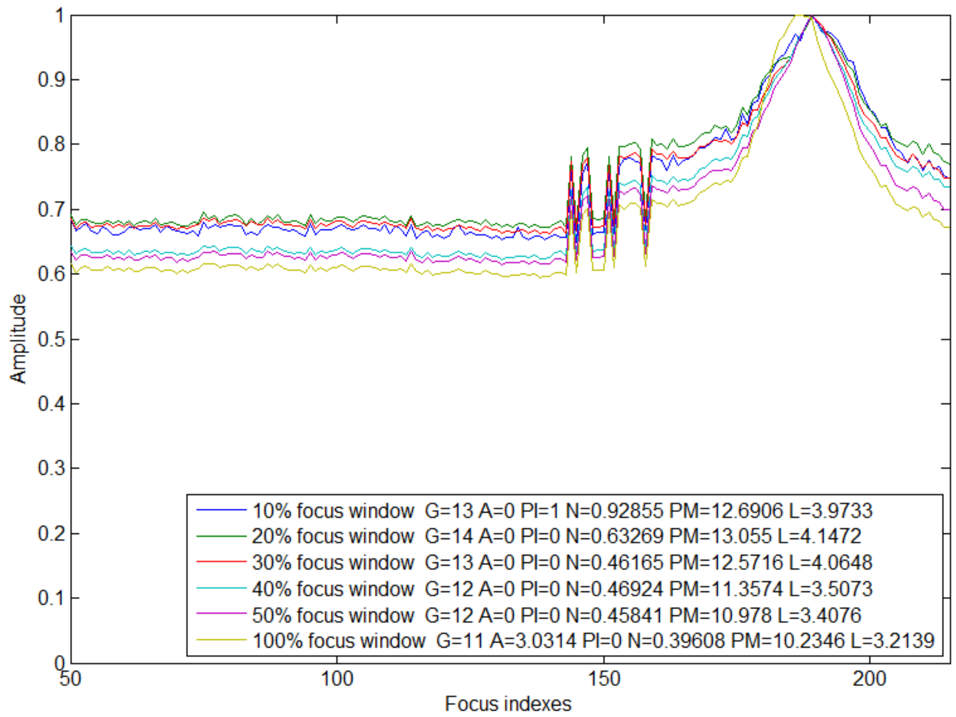

Figure 38 *Focus curves of light studio with different focus windows.*

In real world scenes there is usually something interesting (object) in focusing area. Still that object can be pretty big with flat smooth surfaces without any textures. In these cases choosing too small focus window may cause AF algorithm to choose some very poor focus point, because of temporal noise. On the other hand if the object is very small, choosing too big focus window may cause AF algorithm to focus on something else than desired object. In conclusion it can be stated that in real scenes size of focus window doesn't affect too much if it's still big enough without being too big and noise level isn't too high. It's also hard to predict which focus window size would be the best for certain scene without knowing the content of the scene.

In light studio scene 20% focus window performed worst. Anyhow all studied focus windows perform well enough as they do in AF box too. It should be easy for AF algorithm to find the focus point correctly. It could help camera system to attain the sharpest images if user could easily alter size and position of focus window. Also some segmentation or object recognition could be useful tool to automatically alter the size and shape of used focus window. This can be little problematic because it means longer processing time and heavier process overall. However best focus window would adapt to the shape of the desired object.

4.2 Noise

From focus curves can be noticed that adding Gaussian noise narrows the difference in focus values between in-focus and out-of focus images. This means that height of focus curve is decreasing, which makes task of AF algorithm harder. Reason for this is that adding noise may create more high frequency content in out-of-focus images. On the other hand Gaussian noise also interfere sharp in-focus images by blurring sharp edges. Impulse noise has much bigger impact on out-of-focus images than in-focus images. Out-of-focus images usually have very little edge content, while impulse noise creates there some. At the same time in-focus images usually have already pretty much edge content and adding impulses there doesn't blur edges. This way high frequency edge content isn't lost, but even gained more. Impulse noise causes much more fluctuation to focus curves than Gaussian noise. Impulse noises fluctuation is mainly caused by random positions where impulses occur, while Gaussian noise adds small values to almost every pixel.

Without further knowledge one could make assumption that same noise pattern added to each image affects similarly as randomly formed noise. With little bit deeper brainstorming and investigation it is obvious that adding noise pattern increases noise as much as random noise in average, but it doesn't cause fluctuations. This can be explained by thinking that scene doesn't have any added noise, content of the image has only changed. Content has changed in a way, that it has little more edge content on every image. In other words noise is part of scene.

For AF algorithm there comes a point when image has too much noise and algorithm can't find right focus point anymore. For AF box and light studio it's possible to say, that this point is met when Gaussian noise is added with variances 0.0005 and 0.001 as can be seen from figure 39 and table 4. There is still clearly slight peak at in-focus images, but it might be already too small for proper recognition. It's also interesting to notice that strange fluctuations in light studio are covered by Gaussian noise with higher noise variances as can be seen in figure 40. Interestingly impulse noise added with density 0.01 to AF box causes inaccuracy of 3 focus points, even though goodness value is still good enough as can be seen from table 4. In figure 41 is demonstrated how big impact this kind of small inaccuracies has to image quality. In table 5 can be seen how different content of the noisy image affects to AF statistics calculations.

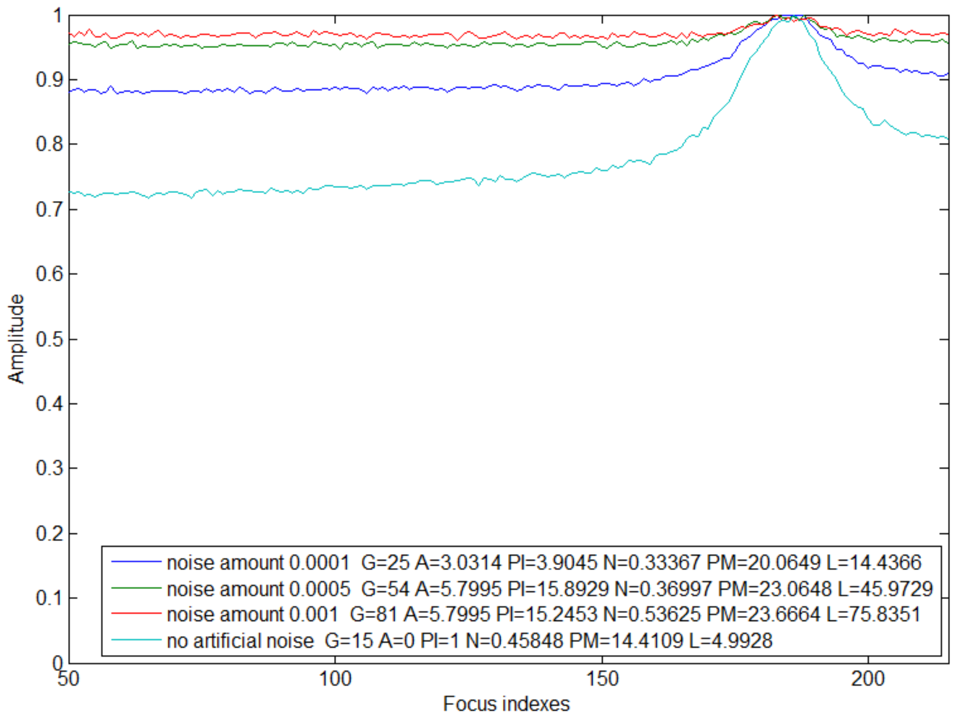

Figure 39 *Focus curves of AF box with different amounts of Gaussian noise*

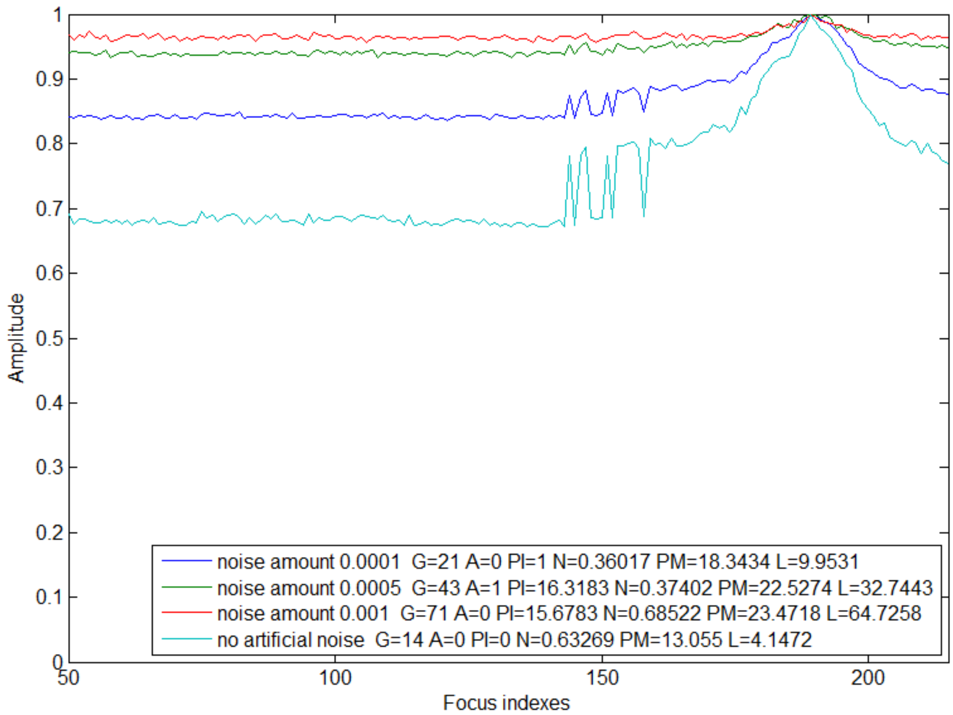

Figure 40 *Focus curves of light studio with different amounts of Gaussian noise*

Table 4 *Goodness values of light studio with different amounts of noise.*

Variable	Goodness
no added noise, no artificial noise	14
no added noise, real noise	41
Gaussian, noise amount 0.0001	21
Gaussian pattern, noise amount 0.0001	21
Gaussian, noise amount 0.0005	43
Gaussian pattern, noise amount 0.0005	41
Gaussian, noise amount 0.001	71
Gaussian pattern, noise amount 0.001	87
impulse, noise amount 0.0001	15
impulse pattern, noise amount 0.0001	14
impulse, noise amount 0.001	16
impulse pattern, noise amount 0.001	15
impulse, noise amount 0.01	24
impulse pattern, noise amount 0.01	23

Table 5 *Goodness values of different scenes with Gaussian noise of variance 0.001*

Scene	Goodness
AF box	81
light studio	71
Siemens star	7
barcode	17

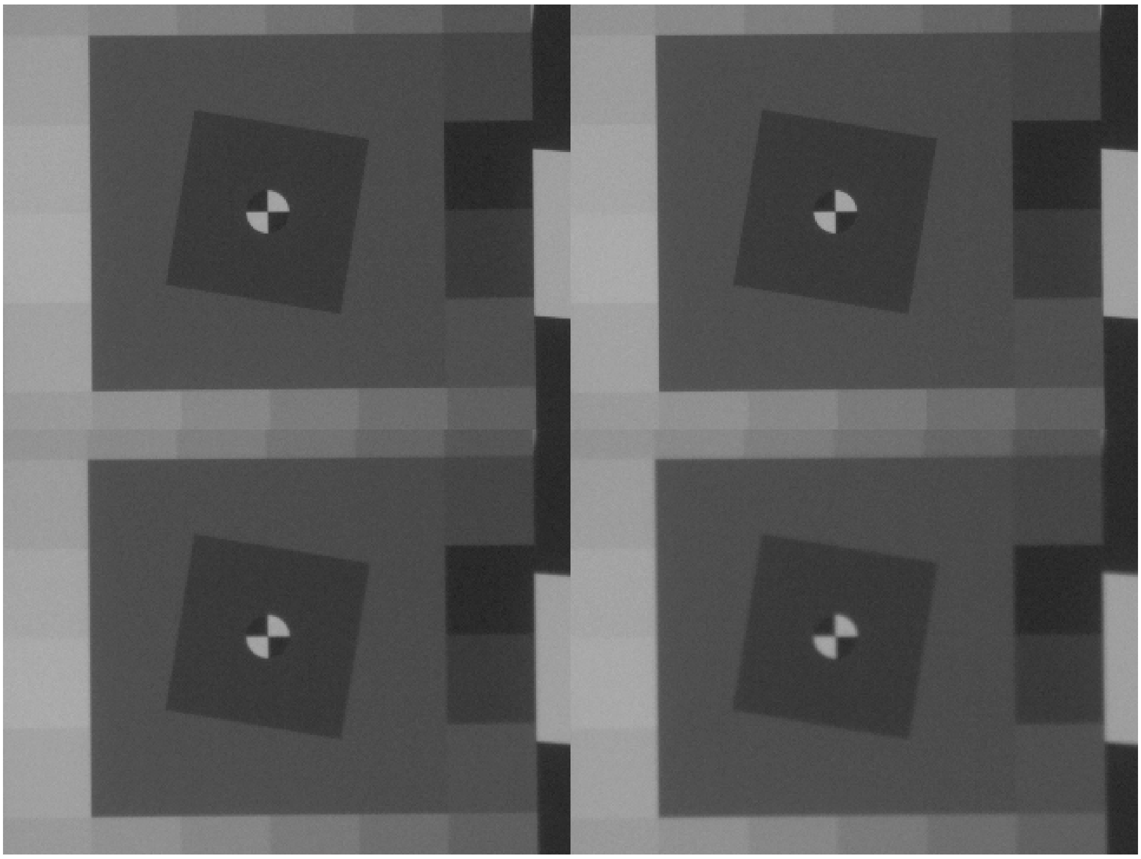

Figure 41 *Image of different inaccuracies in focus point. From left to right and top to bottom focus points are 186, 182, 178 and 174. 186 is the best focus point for AF box.*

In AF box difference of 4 focus points between accurate and inaccurate image is still pretty small. It’s still noticeable if looked very carefully, especially if images are looked at original size. With bigger inaccuracies difference starts to be more and more visible. Already inaccuracy of 8 focus points has pretty visible impact. Inaccuracy of 12 focus points makes image clearly too blur. By converting these inaccuracy values to accuracy criterion approximately values 9.2, 27.9 and 53.3 are achieved. The last value is already more than the roughly agreed boundary of goodness value 40, which harshly separates good and bad focus curves. This means that inaccuracy of 12 focus lens positions makes image quality already pretty poor.

From focus curves it can be noticed that even though impulse noise creates much more fluctuation than Gaussian noise, the average level of curve with impulse noise is still lower and consequently goodness values are better as can be seen in table 4. When comparing Gaussian noise against Gaussian noise pattern and impulse noise against impulse noise pattern, there is a trend that noise patterns cause less fluctuation, but flat amplitude values are higher. This is especially visible when comparing impulse noise, even though differences are still pretty small. However because of big fluctuation, curves of impulse noise pattern have usually worse values. It’s unclear why flat amplitude values are worse for noise patterns. There shouldn’t be big difference and lower fluctuations should make goodness values of noise patterns better. This problem could be probably fixed by altering multipliers in goodness calculations, but it doesn’t affect to the out-of-focus average amplitude, which is the reason for this behavior. It may be result of bad luck when noise patterns are created, but more probably there is something else behind the behavior. However it should be further studied in some other research.

Siemens star and barcode weren’t as sensitive to noise as AF box and Light studio were as can be seen from table 5. It could be said that AF algorithm can’t perform very well with big amounts of noise with more realistic scenes. However scenes with a lot of high frequency content aren’t so sensitive to noise and AF algorithm would probably find right focus points, even if scenes have huge amounts of noise. Still have to keep in mind that it’s highly unlikely that camera module would capture images with such a big amount of noise. When looking image without added noise and same image with added Gaussian noise 0.0001 in full size, it’s easy to say that already there exists pretty much noise. That noise makes clear impact on focus curve. Still even the best focus point for AF box can be found easily, even though 20% focus window of AF box contains pretty little edge content.

By analyzing effects of real noise it can be noticed that focus curves of differently illuminated scenes usually had focus spikes in slightly different place. This inaccuracy isn’t too big, but still focus spike is obviously slightly misplaced. Camera may have moved slightly between capturing images of certain scene in different illumination. It may have caused this problem. However it’s not proofed to be root cause for this problem, but at least it may have some effect. In figures 42 and 43 are examples of the effects of real noise for AF box and Siemens star.

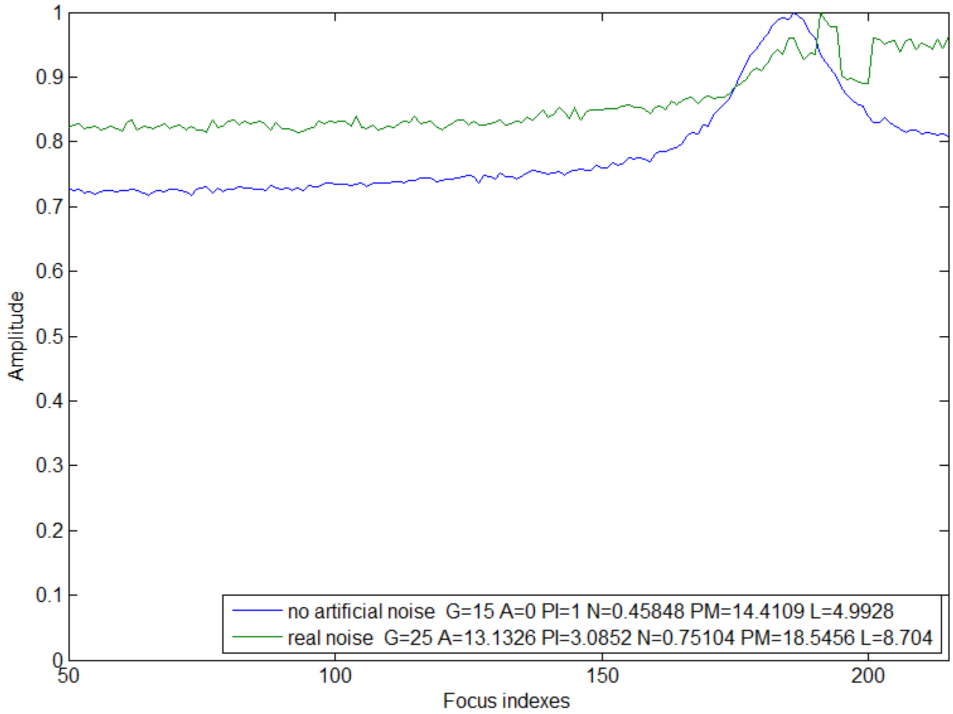

Figure 42 *Focus curves of AF box without added artificial noise.*

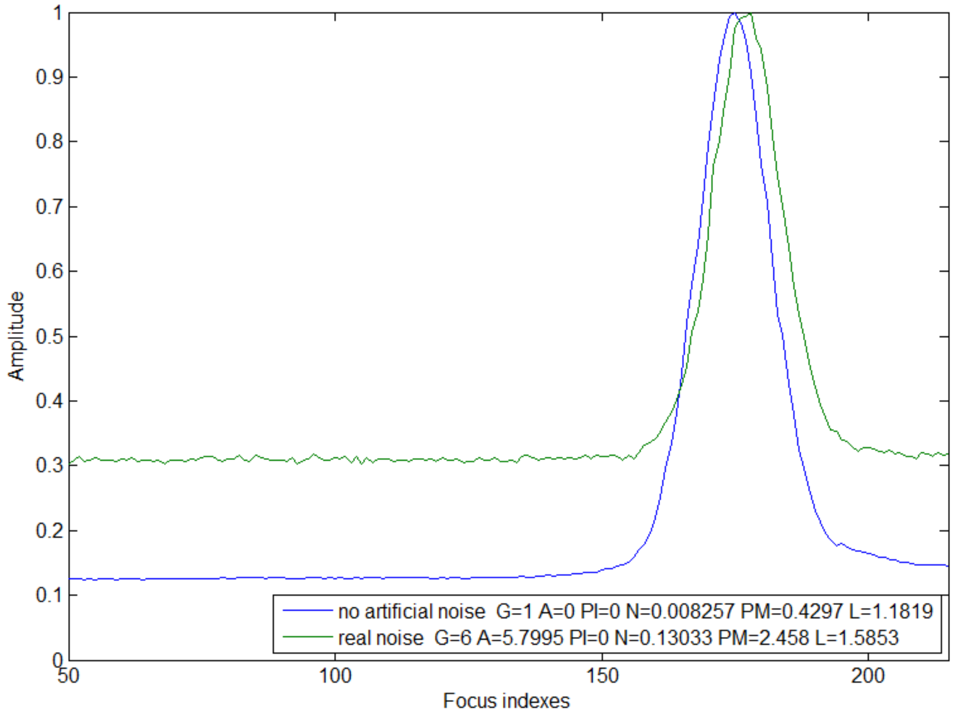

Figure 43 *Focus curves of Siemens star without added artificial noise.*

As can be seen, real noise focus curve of AF box acts very weirdly. Camera software or hardware is the most probable reason for this phenomenon, but it’s not proven to be the root cause. In barcode the spike is slightly distorted, but still the best focus point is found correctly. In other scenes accuracy of real noise curves isn't 0. In light studio low illumination has much bigger impact to focus curve. Goodness value changed from 14 to 41, when illumination was lowered drastically. In AF box same change is from 15 to 25, which is much smaller.

Real noise in scenes AF box and barcode has smaller effect on focus curves, than smallest studied amount of Gaussian noise. Low illumination of light studio is affecting surprisingly much when compared to other scenes. Focus curves of light studio's images in low illumination and with added Gaussian noise of variance 0.0005 are pretty similar. Their goodness values were 41 and 43. In Siemens star scene focus curve of real noise was something between Gaussian noises 0.0001 and 0.0005. Still it needs to be kept in mind that these real noise images are captured in really low light without flash. It's something that is done very seldom. As conclusion it can be clearly stated that noise has an impact for focus value calculations. Images should be captured as noise free as possible after all to ensure reliable performance of AF algorithm. However slight increase in the amount of noise can be justified if something important is achieved.

4.3 Noise reduction

Of course in every good imaging system some noise reduction is done to create better images, but when calculating focus values noise reduction isn't always necessary and sometimes it may even destroy very small details. By looking focus curves and goodness values, it's obvious that noise reduction algorithms improve curves and goodness values alongside. All algorithms improve goodness values of Gaussian noise, even simple mean and median filters have clear impact to focus curves as can be seen from table 6. These noise reduction algorithms make it much easier for AF algorithm to find out the best focus point. Of course BM3D, which is designed for Gaussian noise reduction, is superior compared to simpler algorithms, but it also needs much heavier computation and consumes much more time. From table 7 can be seen that noise reduction algorithms also improve so called noiseless images by removing still existing noise.

Noise reduction of impulse noise gives some interesting results. If noise amount is small enough BM3D is best tested noise reduction algorithm, but if noise amount is increased enough executing both median and mean filtering becomes superior in normal scenes. In scenes with high frequency content effect is similar with one exception, using both simple filters is as effective as using just median filtering. Median filtering may be even better, because it won't blur so much the edge content. This can be seen from table 8. In this study impulse noise density 0.001 is small enough for BM3D to be the best algorithm and noise density 0.01 makes median filtering based methods better. Still for the worst impulse noise all the algorithms lower overall average amplitude of focus curve, but amplitude of fluctuations is increased with mean and BM3D algorithms. This increase causes sometimes worse accuracy values as can be noticed from figure 44.

Table 6 *Goodness values for AF box with different noise reduction algorithms and noises.*

Variable	Goodness	Variable	Goodness
BM3D, Gaussian 0.0001	1	BM3D, impulse 0.0001	2
BM3D, Gaussian 0.0005	2	BM3D, impulse 0.001	7
BM3D, Gaussian 0.001	2	BM3D, impulse 0.01	25
both, Gaussian 0.0001	9	both, impulse 0.0001	6
both, Gaussian 0.0005	17	both, impulse 0.001	6
both, Gaussian 0.001	19	both, impulse 0.01	6
mean, Gaussian 0.0001	13	mean, impulse 0.0001	9
mean, Gaussian 0.0005	21	mean, impulse 0.001	13
mean, Gaussian 0.001	23	mean impulse 0.01	23
median, Gaussian 0.0001	12	median, impulse 0.0001	8
median, Gaussian 0.0005	20	median, impulse 0.001	8
median, Gaussian 0.001	23	median impulse 0.01	8
no reduction, Gaussian 0.0001	25	no reduction, impulse 0.0001	16
no reduction, Gaussian 0.0005	54	no reduction, impulse 0.001	18
no reduction, Gaussian 0.001	81	no reduction impulse 0.01	26

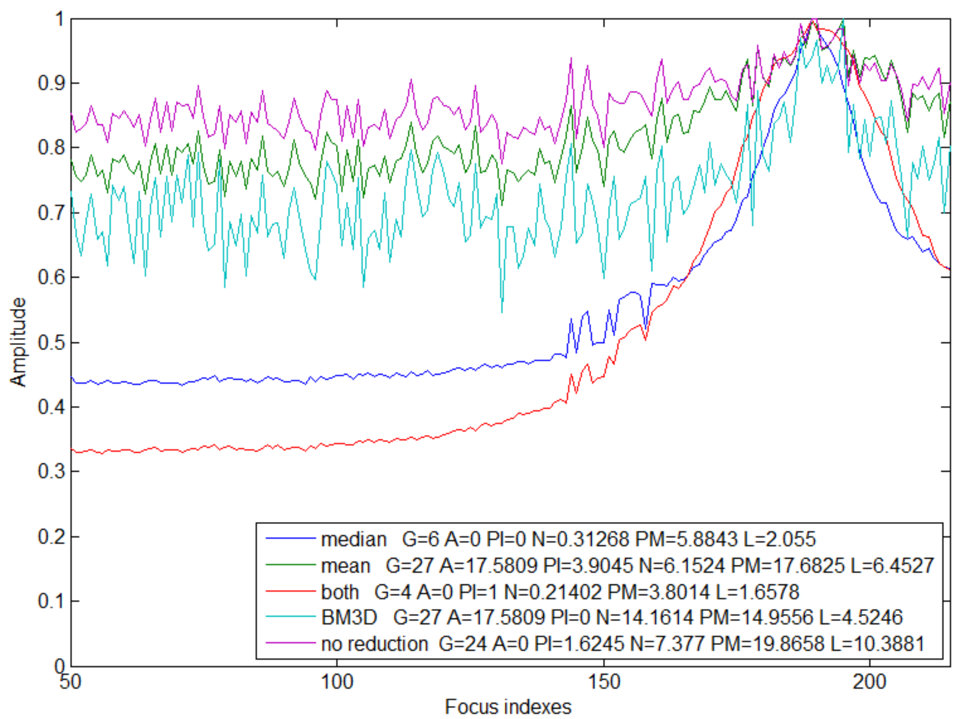

Figure 44 *Focus curves of light studio. Noise reduction is done to images where impulse noise is added with density 0.01.*

Table 7 *Goodness values for AF box with noise reduction algorithms to noiseless images.*

Reduction	Goodness
BM3D	1
both	6
mean	7
median	8
no reduction	15

Table 8 *Goodness values for barcode with noise reduction algorithms to impulse noise of density 0.01.*

Reduction	Goodness
BM3D	3
both	2
mean	4
median	1
no reduction	6

Using both median and mean filtering usually further increases the performance compared to just using one of those algorithms. Using both mean and median filtering is second best method or best method, in studied cases where BM3D isn't the best one. Biggest effect of BM3D was in AF box with Gaussian noise of variance 0.001. It reduced goodness value from 81 to 2 as can be seen from table 6. In that scene using both median and mean filtering and both of them separately decreased the goodness values to adequate level. In impulse noise cases, median based algorithms reduce noise very well and make focus curves pretty good. Also BM3D algorithm gives good focus curves if amount of impulse noise isn't too high.

With all that said it can be seen that noise reduction is important when image is very noisy. Reducing noise may also weaken or completely overcome abnormalities in focus curves as can be seen from figure 45.

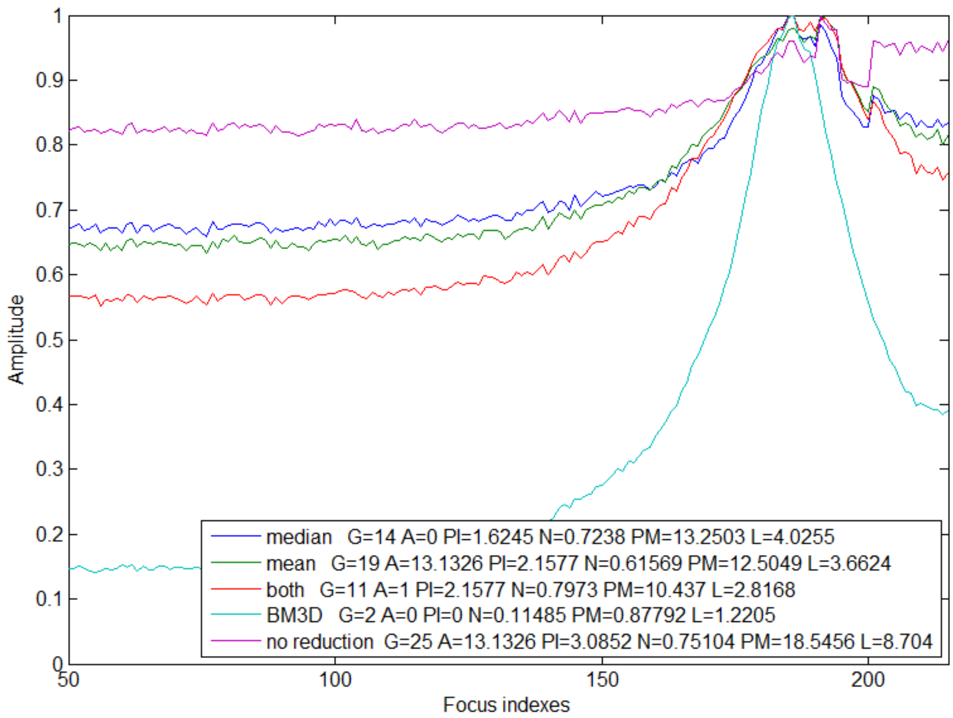

Figure 45 *Focus curves of AF box. Noise reduction is done to images with real noise.*

With weakening abnormalities, noise reduction can also make accuracy of focus curves better. Also something interesting about performance of noise reduction algorithms is found by looking focus curves. When image has less high frequency content in focus window, noise reduction algorithms improve focus curves more when image has less noise. On the other hand when image has more high frequency content, noise reduction algorithms improve focus curves more when image has more noise.

By comparing results of noise reduction algorithms, it can be said that something as easy as median and mean filtering works well if noise type is known. Usually camera produces more something like Gaussian noise, which can be seen from real noise images. It should be kept in mind that AF statistics should be calculated very fast to react on changing scene. Processing time of all processing blocks executed before AF statistics calculation should be added to total AF statistics calculation time. This in mind it can be stated that even though BM3D is very good algorithm it's probably pretty heavy to use in AF statistics calculations, even though hardware and software would have been optimized for it. Of course after finding the focus point, it may be worth reducing noise with BM3D, at least BM3D could be useful algorithm for improving image quality in some post processing block or software. When comparing median and mean filtering against each other, they perform almost equally with Gaussian noise. However in impulse noise median is way better as assumed. Still in real images there isn't much impulse noise, which kind of makes the comparison even. If certainty of reducing impulse noise is needed, it's easy to say that best choice is median filtering. If it can be afforded it could be useful to use either both median and mean algorithms together or some more complex algorithms. It's also possible to use median or mean algorithms with different parameters, which may affect very differently to focus curves. Altogether reducing noise means that finding the best focus point becomes easier. This results in simpler, computationally lighter and faster AF algorithms, while still retaining high accuracy.

4.4 Scaling

By studying focus curves, it is obvious that out-of-focus amplitude of moderately downscaled images is lower, than focus curve of original image. At the same time downscaling widens the in-focus spike as is presented in figure 46. Still moderate downscaling generally results to better focus curves. Too harsh downscaling on the other hand may lose too much important information and create bigger differences between adjacent pixels, if no anti-aliasing is done. This leads to worse focus curves. In subsampling and bicubic scaling this behavior is easy to explain by the averaging between certain numbers of pixels. It naturally reduces amount of noise, but also blurs sharp edges. Explaining behavior of subsampling is little bit trickier. It's based on the reason that used scenes have edges which are still so big that small downscaling keeps most of the important information in image and reduces amount of edgeless content. The ratio between edge content and noise become this way better, which can be seen as lower out-of-focus areas in focus curves. Bigger subsampling causes bigger fluctuations to focus curves. Especially this is visible in noisy images, because no averaging is done.

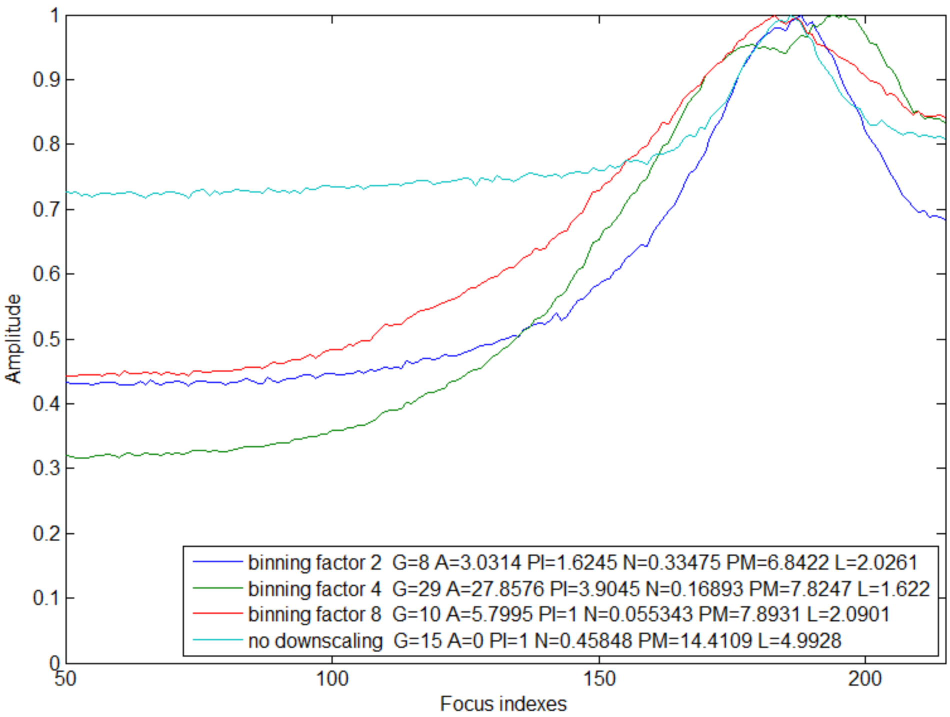

***Figure 46** Focus curves of AF box with different binning factors.*

It can be compared how subsampling, bicubic scaling and pixel binning affects to goodness values of AF box by looking table **Virhe. Viitteen lähdettä ei löytynyt.**. However it's important to keep in mind, when comparing those downscaling methods against each other, that subsampling and binning factors are fully corresponding, but when compared to bicubic scaling only factors 2 and scaling multiplier 0.5 are corresponding. From table 9 can be noticed that too heavy downscaling makes things worse compared to smaller downscaling. Subsampling and binning factors 8 are already too much for AF box. Actually in binning this isn't obvious from goodness values, because as can be seen in figure 46 there is something strange happening with binning factor 4. The focus spike has spread around the real focus and created to focus spikes next to it. This kind of unexpected behavior is even more common with pixel binning and it's dealt later on in this chapter.

***Table 9** Goodness values for AF box with different downscaling methods and without added noise.*

no downscaling	15	scaling multiplier 0.5	7
binning factor 2	8	scaling multiplier 0.75	9
binning factor 4	29	subsampling factor 2	10
binning factor 8	10	subsampling factor 4	10
scaling multiplier 0.3	8	subsampling factor 8	11

However generally factor 8 increases magnitude of out-of-focus area compared to factor 4. This means that too much information about edges is lost. If bicubic scaling would have been studied with small enough multipliers, the same effect should have been encountered. Nevertheless downscaling with factors 8 perform better in AF box than case without downscaling.

From table 10 can be noticed that in more natural scenes averaging in bicubic scaling and pixel binning makes those two better downscaling methods than subsampling, at least when original images are downscaled to half of the original size. In high frequency images subsampling may give better results. In noiseless case of AF box difference in goodness value isn't very big, but magnitude of out-of-focus areas differs more as can be seen in figure 46.

Table 10 *Goodness values of all scenes with comparable downscaling.*

AF box, binning factor 2	8	light studio, scaling multiplier 0.5	5
barcode, binning factor 2	2	Siemens star, scaling multiplier 0.5	1
light studio, binning factor 2	5	AF box, subsampling factor 2	10
Siemens star, binning factor 2	1	barcode, subsampling factor 2	1
AF box, scaling multiplier 0.5	7	light studio, subsampling factor 2	9
barcode, scaling multiplier 0.5	2	Siemens star, subsampling factor 2	1

From tables 11 and 12 can be noticed that if noise is added to images, downscaling with bigger scaling multiplier and binning factor reduces more noise and makes focus curves better, if accuracy stays suitable. From table 13 can be seen that also subsampling makes focus curves better, because skipping noisy samples reduce more noise compared to edge content. Of course too big amounts of noise ruin also benefits from scaling and naturally with too big scaling too much important information about image can be lost. Downscaling with bigger factors and multipliers has tendency to decrease the differences in height between the noisy and noiseless focus curves. Especially pixel binning has this feature. This just means that power of white noise becomes less meaningful if big enough blocks are averaged.

Table 11 *Goodness values of AF box with bicubic scaling and different amount of noise.*

scaling multiplier 0.75, Gaussian 0.0001	17	scaling multiplier 0.5, impulse 0.0001	7
scaling multiplier 0.75, Gaussian 0.0005	28	scaling multiplier 0.5, impulse 0.001	14
scaling multiplier 0.75, Gaussian 0.001	35	scaling multiplier 0.5, impulse 0.01	20
scaling multiplier 0.75, impulse 0.0001	11	scaling multiplier 0.3, Gaussian 0.0001	7
scaling multiplier 0.75, impulse 0.001	15	scaling multiplier 0.3, Gaussian 0.0005	12
scaling multiplier 0.75, impulse 0.01	24	scaling multiplier 0.3, Gaussian 0.001	16
scaling multiplier 0.5, Gaussian 0.0001	10	scaling multiplier 0.3, impulse 0.0001	15
scaling multiplier 0.5, Gaussian 0.0005	17	scaling multiplier 0.3, impulse 0.001	11
scaling multiplier 0.5, Gaussian 0.001	19	scaling multiplier 0.3, impulse 0.01	17

Pixel binning with higher binning factors has tendency to distract accuracy as can be seen from table 14 and figures 46, 47 and 48. Because of averaging certain amount of pixels without any weighting, it may cause unpredictable distortions to focus curve. For example for AF box binning factor 4 causes local minimum to focus point, where should be global maximum.

***Table 12** Goodness values of AF box with pixel binning and different amount of noise.*

binning factor 2, Gaussian 0.0001	13	binning factor 4, impulse 0.0001	29
binning factor 2, Gaussian 0.0005	20	binning factor 4, impulse 0.001	15
binning factor 2, Gaussian 0.001	23	binning factor 4, impulse 0.01	38
binning factor 2, impulse 0.0001	8	binning factor 8, Gaussian 0.0001	10
binning factor 2, impulse 0.001	13	binning factor 8, Gaussian 0.0005	12
binning factor 2, impulse 0.01	21	binning factor 8, Gaussian 0.001	13
binning factor 4, Gaussian 0.0001	24	binning factor 8, impulse 0.0001	11
binning factor 4, Gaussian 0.0005	26	binning factor 8, impulse 0.001	13
binning factor 4, Gaussian 0.001	27	binning factor 8, impulse 0.01	15

***Table 13** Goodness values of AF box with subsampling and different amount of noise.*

subsampling factor 2, Gaussian 0.0001	17	subsampling factor 4, impulse 0.0001	10
subsampling factor 2, Gaussian 0.0005	27	subsampling factor 4, impulse 0.001	12
subsampling factor 2, Gaussian 0.001	36	subsampling factor 4, impulse 0.01	23
subsampling factor 2, impulse 0.0001	12	subsampling factor 8, Gaussian 0.0001	20
subsampling factor 2, impulse 0.001	14	subsampling factor 8, Gaussian 0.0005	18
subsampling factor 2, impulse 0.01	23	subsampling factor 8, Gaussian 0.001	19
subsampling factor 4, Gaussian 0.0001	18	subsampling factor 8, impulse 0.0001	12
subsampling factor 4, Gaussian 0.0005	22	subsampling factor 8, impulse 0.001	62
subsampling factor 4, Gaussian 0.001	23	subsampling factor 8, impulse 0.01	33

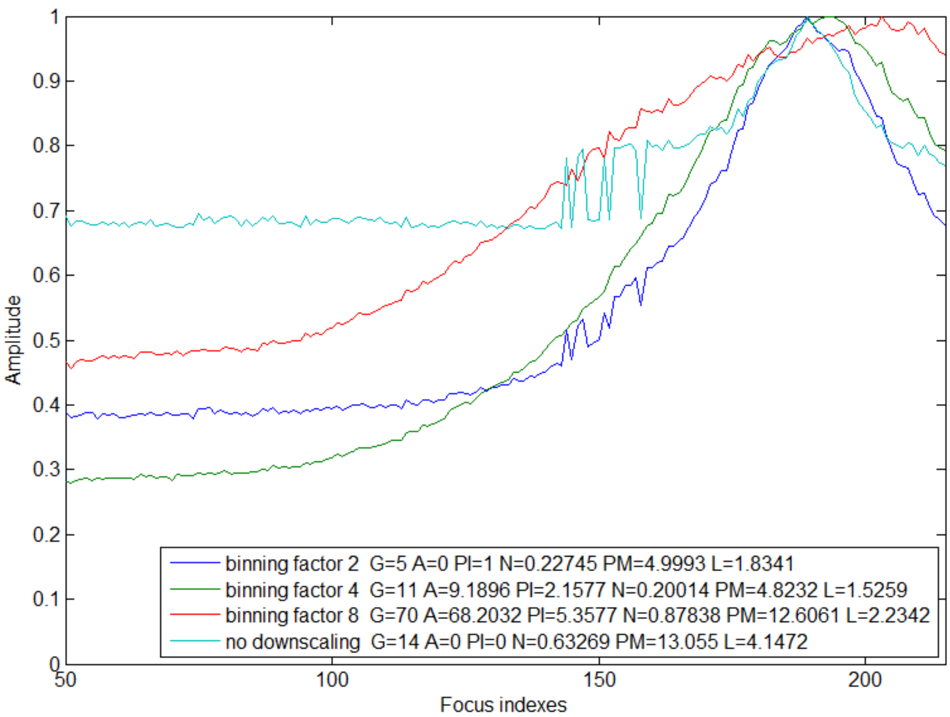

***Figure 47** Focus curves of light studio with different binning factors.*

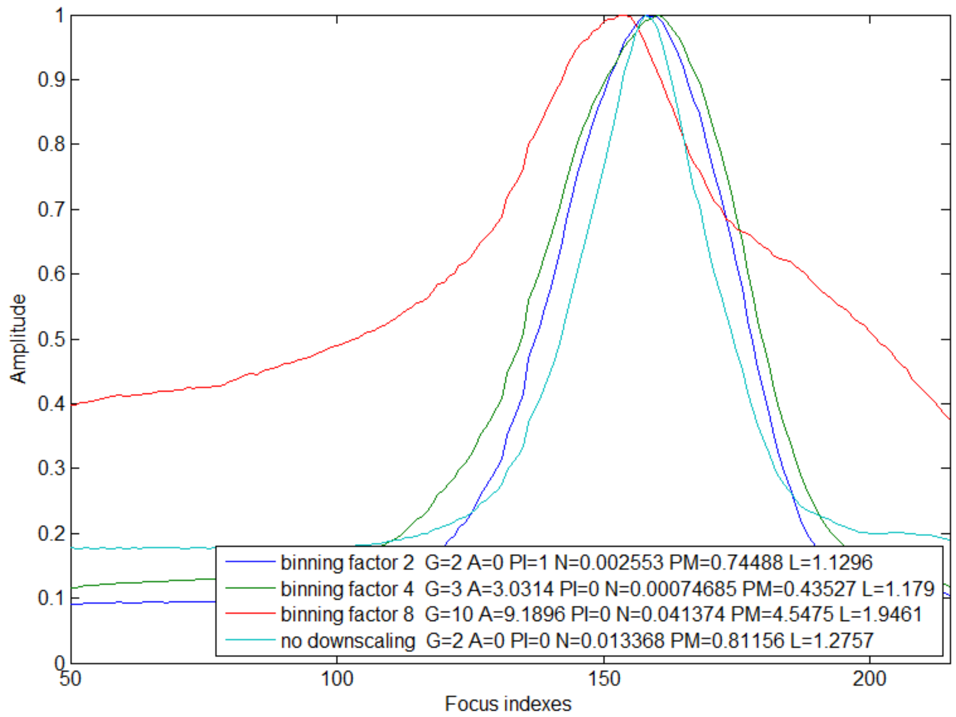

Figure 48 *Focus curves of barcode with different binning factors.*

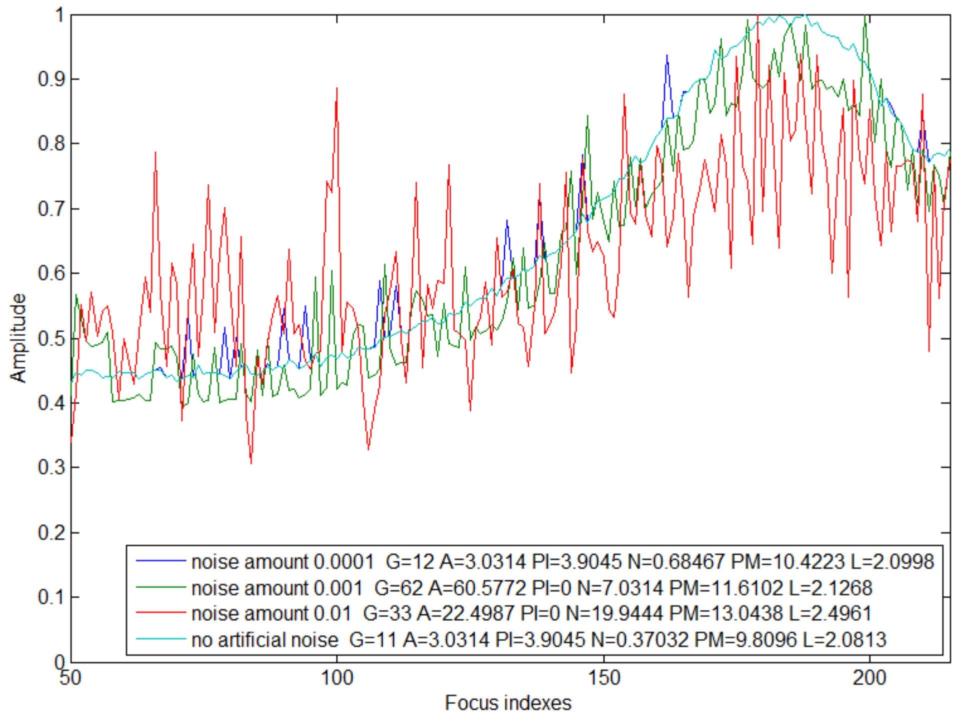

Figure 49 *Focus curves of AF box with subsampling factor 8 and different amounts of impulse noise.*

In figure 50 is demonstrated how averaging in pixel binning affects to in-focus image of Siemens star. These kinds of behaviors are much weaker in other downscaling methods. However fluctuation in subsampling may also cause inaccuracy and overall much worse focus curves, especially with higher amounts of noise. Impulse noise may cause very big fluctuations with subsampling as can be seen in figure 49. It can be seen generally, that accuracy is more likely to get worse with bigger downscaling and subsampling with impulse noise is emphasizing this effect. Other downscaling methods aren’t so sensitive to impulse noise, because of averaging.

Table 14 *Goodness values of barcode, light studio and Siemens star with pixel binning and without added noise.*

barcode, no downscaling	15	barcode, binning factor 4	3
light studio, no downscaling	2	light studio, binning factor 4	11
Siemens star, no downscaling	8	Siemens star, binning factor 4	1
barcode, binning factor 2	2	barcode, binning factor 8	10
light studio, binning factor 2	5	light studio, binning factor 8	70
Siemens star, binning factor 2	1	Siemens star, binning factor 8	3

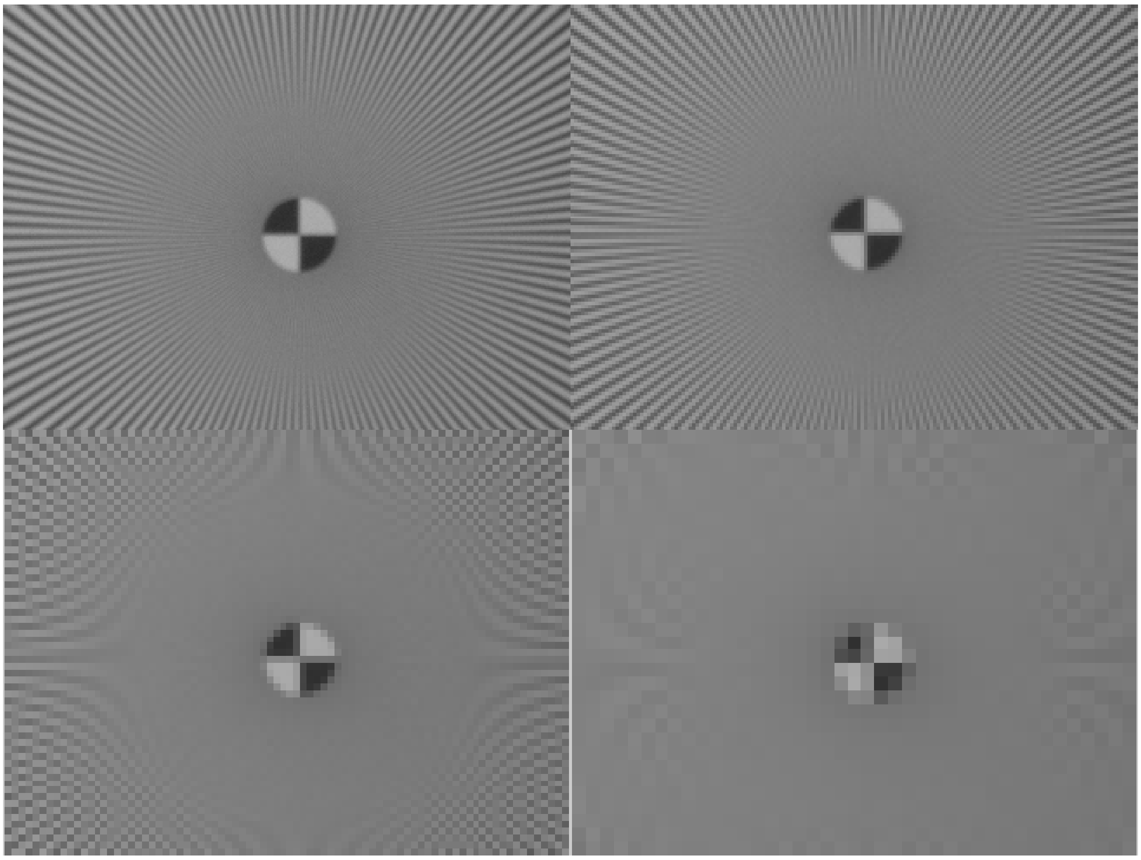

Figure 50 *Image of pixel binning effects for in-focus image of Siemens star. From left to right and top to bottom are presented images without pixel binning, with binning factor 2, with binning factor 4 and with binning factor8. Downscaled images are upscaled to original size with nearest neighbor interpolation.*

When bigger downscaling is used differences between downscaling methods become more unpredictable. There may be some variations in performance within different scenes between different downscaling methods, but usually bicubic scaling gives best results, pixel binning second best and subsampling worst results. Overall it can be said that small downscaling makes focus curves of noisy images better. For example AF box with Gaussian noise of variance 0.001 has goodness value 81. Subsampling that image with subsampling factor of 2 gives 36 as goodness value, which is already below the border value of 40. Still it has to be kept in mind that too big scaling can make things even worse. It's also worth mentioning that these methods create noise fluctuations without anti-aliasing. Fluctuations created by subsampling are even higher, because there isn't any averaging. However in downscaling studies almost all focus curves have acceptable goodness values. There are only few focus curves that have goodness values over 40 and they all suffer from poor accuracy values. It could be good idea to use just part of the sensors of camera when calculating AF statistics. Also some kind of averaging could be very useful. Anyway this is also a processing step that shouldn't take too much time. However some downscaling is always done, when image is shown on viewfinder, but in that point there is also lot of other processing done.

4.5 Color correction

From curves it can be noticed that white balancing improves focus curves with CCM multiplier 0 compared to case without any color correction. However difference in those curves is based on averaging of the two green color components. This averaging gets rid of some noise. If curve without color correction would be presented as average of two green channels, curve would be same as white balanced curve. It could be said that comparing those two is little unfair. Still it gives good idea how much even simple averaging of two green channels improves focus curve. In light studio and barcode those two curves wouldn't be exactly same, because green color component isn't there the weakest one. This means that for every differently focused image white balance gains are recalculated and green component is multiplied with value little higher value than 1 and that value is altered for every image. This alteration is caused by noise in image.

It can be also seen that using other than unitary color correction matrix makes focus curves worse, because it amplifies noise. Even Gaussian noise of variance 0.0001 with color correction multiplier 2 in AF box increases goodness value over the border value of 40 as can be seen from table 15. With bigger Gaussian noise variances CCM makes situation even worse as can be seen from figure 51. There CCM completely ruins the focus curve. However if AF algorithm is very smart, it might even find small focus spike from right focus point, but it is very small and biggest focus values are already achieved in much lover focus indexes.

__Table 15__ Goodness values of AF box with different color corrections and noises.

no color correction	15	CCM multiplier 2, Gaussian 0.001	191
CCM multiplier 0	14	no color correction, Gaussian 0.001	81
CCM multiplier 1	18	CCM multiplier 0, impulse 0.0001	15
CCM multiplier 2	22	CCM multiplier 1, impulse 0.0001	19
CCM multiplier 0, Gaussian 0.0001	21	CCM multiplier 2, impulse 0.0001	22
CCM multiplier 1, Gaussian 0.0001	34	no color correction, impulse 0.0001	16
CCM multiplier 2, Gaussian 0.0001	50	CCM multiplier 0, impulse 0.001	17
no color correction, Gaussian 0.0001	25	CCM multiplier 1, impulse 0.001	21
CCM multiplier 0, Gaussian 0.0005	39	CCM multiplier 2, impulse 0.001	23
CCM multiplier 1, Gaussian 0.0005	75	no color correction, impulse 0.001	18
CCM multiplier 2, Gaussian 0.0005	134	CCM multiplier 0, impulse 0.01	29
no color correction, Gaussian 0.0005	54	CCM multiplier 1, impulse 0.01	117
CCM multiplier 0, Gaussian 0.001	55	CCM multiplier 2, impulse 0.01	117
CCM multiplier 1, Gaussian 0.001	129	no color correction, impulse 0.01	26

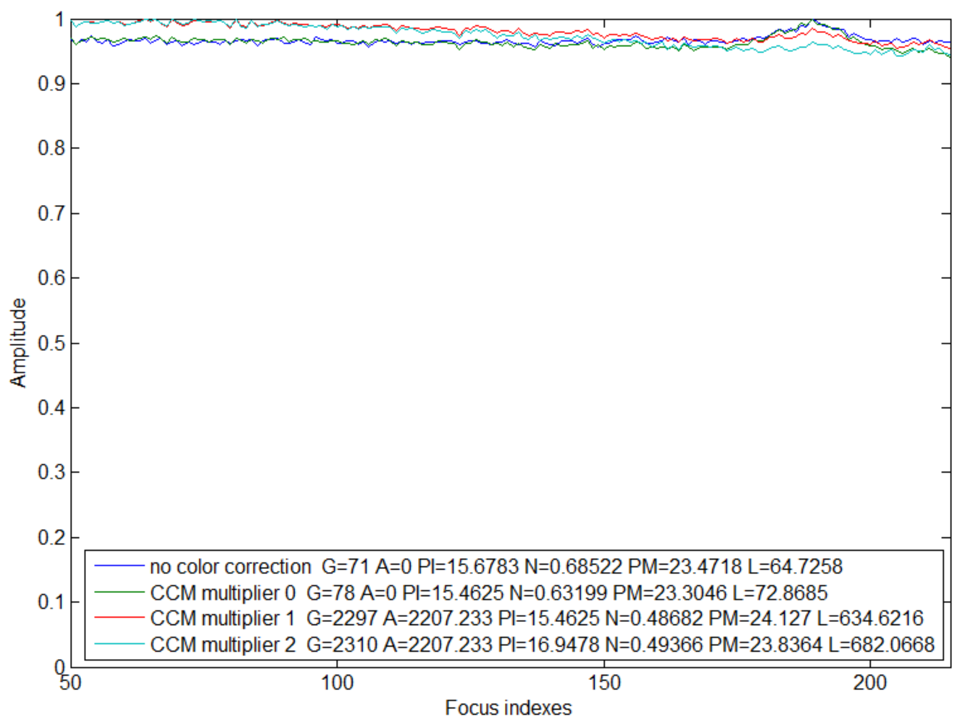

__Figure 51__ Focus curves of light studio with different CCM multiplier and Gaussian noise of variance 0.001.

In impulse noise studies effects aren't so big. It's mainly because added impulse noise is truncated after CCM back to original value. This means that only noise existing in images originally is amplified. It causes bigger differences to out-of-focus images and lowers the difference of in-focus and out-of-focus images in focus values. In impulse noise cases almost all goodness values stay under 40. Only poor accuracy in AF box with impulse noise of density 0.01 causes worse goodness values as can be seen from figure 52. There accuracy is 111.2 which mean inaccuracy of 19 lens positions. If AF algorithm can't filter that out it definitely means blur images.

Overall it's easy to say that color correction should be done after AF statistics calculations if possible. Especially if noise is very existent CCM may completely ruin the achieved focus and make images blur.

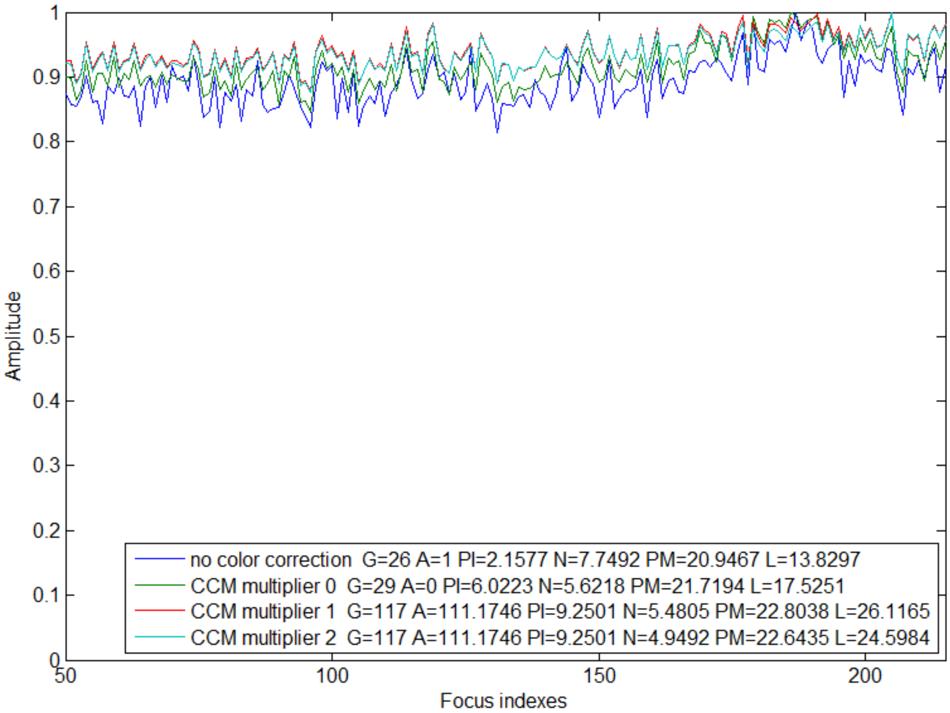

Figure 52 *Focus curves of AF box with different CCM multipliers and impulse noise of density 0.01.*

4.6 Filter size

From curves it can be found out that filter size doesn't play too big role in AF statistics calculations, still it has some effect on focus curves. Performance depends from the content of the scene. One-dimensional horizontal filters perform better compared to two-dimensional filters in barcode, whose only real content is vertical stripes. This can be seen in figure 53. 11x1 and 9x1 sized filters perform better than biggest two-dimensional 11x11 filter. Differences are still pretty small because all the filters perform well in that scene. In goodness value difference between best and worst filters is just 1. These kinds of scenes, which contain only vertical stripes, are pretty rare. Other extremity would be scene with horizontal stripes. In that case one-dimensional horizontal filter couldn't find right focus point at all.

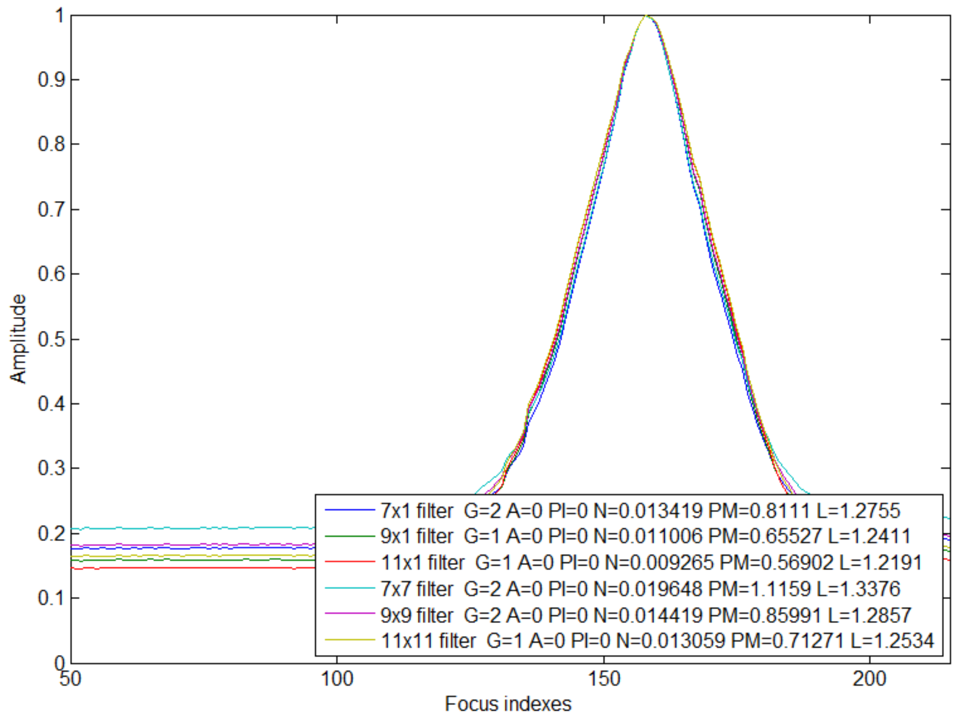

Figure 53 *Focus curves of barcode with different filter sizes.*

In all other scenes two-dimensional filters outperform their one-dimensional counterparts. In light studio the gap between one-dimensional and two-dimensional filters is biggest as can be seen in table 16. In AF box 7x7 performed as well as 11x1, while other two-dimensional filters were better. In goodness values differences are again really small. The best goodness value in AF box is 11, while the worst value is 15. In Siemens star there isn't big difference between curves. Still two-dimensional filters perform better than one-dimensional filters. However all studied filters had goodness value 1.

Table 16 *Goodness values of light studio with different filter sizes.*

7x1 filter	14
9x1 filter	13
11x1 filter	12
7x7 filter	10
9x9 filter	9
11x11 filter	8

As conclusion it could be stated that two-dimensional filters perform better in real scenes than their one-dimensional counterparts. Still one need to keep in mind that implementing two-dimensional AF filters in the camera module may be too expensive compared to benefits achieved. There is tendency that bigger filters perform better to some extent, but too big filters may make things worse by using too much information of peripheral samples.

5 CONCLUSIONS

In this thesis is studied how executing certain image processing steps affect to autofocus calculations. AF algorithm isn't fully implemented, but focus curves are analyzed from point of view of AF algorithm.

This object was achieved by capturing images of different scenes with the studied smartphone in controlled environment. That phone is still prototype and the software is slightly outdated at the time of running tests and writing this thesis, which may have caused some strange phenomena to focus curves. With the phone images of each scene were captured with every possible lens position. Then these images were processed in different ways to achieve focus curves for certain studies. Finally achieved focus curves and influence of processing steps were analyzed.

From analyzes it can be noticed that noise has huge effect on focus curves. Especially Gaussian noise causes worse focus curves. Impulse noise may cause so big fluctuations to focus curves that AF algorithm may find wrong focus point as the best one. When SNR becomes better also better focus curves are achieved. However focus curves were still pretty good with decent amounts of noise. It should be remembered that normally device is held in hand, when images are captured. This may cause bigger fluctuation and noise to focus values. When this fact is also taken to account some kind of noise reduction is recommended to be executed before focus value calculation. This also means that all processing that increase noise should be done after calculating focus values if possible. Color correction is one example of this kind of processing that should be postponed in the pipeline.

Scaling has also interesting impact to focus curves. Too big scaling may lose too much important information, while smaller scaling increase SNR and gives better focus curves. However it has been found that scaling may have very unpredicted effects. For this reason it's recommend to do some scaling before focus calculation only if effects of used scaling method are carefully studied.

It was noticed that it's hard to choose focus window size, which would produce good focus curves from every scene. If fixed focus window size is desired, smaller focus window would probably perform generally better. However, adaptive focus window would potentially be much more optimal solution. It could adapt the form and size of focused object. It can be also noticed that generally bigger filter size is better. Difference between two-dimensional and one-dimensional filters was surprisingly small. However if there is horizontal edge content within scene, horizontal filter can't detect these edges. For these reasons it's recommend to use two-dimensional filters if possible. Best size for the filter should be found out in comprehensive testing.

For further research presented studies could be further tested. In an image processing pipeline there also exist other processing blocks whose effects for focus curves could be interesting to study. Also fully implementing different AF algorithms and testing their performance could be interesting. It's also important to test whether the goodness criteria works with focus curves of differently calculated AF statistics.

REFERENCES

[1] Abdullah, S.J., Ratnam, M.M & Samad, Z. Focus Measure in a Liquid-filled Diaphragm (LFD) Lens Using Passive Auto-focus Method. Control Automation Robotics & Vision, 2010, Singapore, December 7 – 10, 2010. pp. 1086 – 1092.

[2] Groen, F., Young, I. & Ligthart, G. Comparison of Different Focus Functions for Use in Autofocus Algorithms. Cytometry Part A 6(1985)2, pp. 81 – 91.

[3] Liang, Q. & Qu, Y. Quantifying focus criterion function comparison for two-stage autofocusing system. Visual Information Processing XXI. Baltimore, Maryland, USA, May 1, 2012. Proc. of SPIE Vol. 8399 83990X. 18 p.

[4] MacEvoy, B. Color Vision [WWW]. [referred 15.12.2013]. Available: http://www.handprint.com/LS/CVS/color.html

[5] Gonzales, R.C. & Woods, R.E. Digital Image Processing. Second edition. Upper Saddle River, New Jersey 2002, Prentice-Hall, Inc. 793 p.

[6] Roder, H. Amplitude, Phase, and Frequency Modulation. IEEE Proceedings of the Institute of Radio Engineers 19(1931)12, pp. 2145 – 2176.

[7] Cheng, J. Digital Camera Ssystem Simulator And Applications. Dissertation. Stanford, Kalifornia, USA, June 2003. Stanford University. Electrical Engineering. 115 p.

[8] Long, B. Complete Digital Photography, 3rd ed., Hingham, Massachusetts, USA 2005. Cengage Learning. 528 p.

[9] Singh, D. Fundamentals Of Optics. New Delhi 2010, PHI Learning Private Limited. 431 p.

[10] Bigelow, R. Comprehensive Guide to Mastering Digital Photography. May 19, 2011. Ron Bigelow Photography. 222 p.

[11] Pennebaker, W.B. & Mitchell, J.L. JPEG Still Image Data Compression Standard, New York, USA 1993. Van Nostrand Reinhold. 638 p.

[12] McHugh, S., Cambridge in colour. Camera Exposure [WWW]. [referred 15.12.2013]. Available: http://www.cambridgeincolour.com/tutorials/camera-exposure.htm

[13] Iannello, G., Onofri, L., Punzo, G. & Soda, P. An Efficient Autofocus Algorithm for Indirect Immunofluorescence Applications. Computer-Based Medical Systems, 2011, Bristol, England, UK, June 27 – 30, 2011. pp. 1 – 6.

[14] ITU-R BT.500-13. Methodology for the subjective assessment of the quality of television pictures. Geneva 2012. International Telecommunication Union. 44 p.

[15] Eskicioglu, A.M. & Fisher, P.S. Image Quality Measures and Their Performance. Communications 43(1995)12, pp. 2959 – 2965.

[16] Stathaki, T. Image Fusion: Algorithms and Applications. London, UK 2011. Academic Press. 520 p.

[17] Dinesh, R. & Navya, V. New Approaches to Edge Preserved Image Denoising. 2012 International Conference on Recent Advances in Computing and Software Systems, Kalavakkam, Chennai, India, April 25 – 27, 2012. IEEE. pp. 141 – 147.

[18] Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. Image denoising with block-matching and 3D filtering. Electronic Imaging '06, no. 6064A-30, San Jose, California, USA, January 15, 2006. Proc. SPIE Vol. 6064. 12 p.

[19] Ray, S.F.. 3rd ed., Oxford, UK 2002, Focal Press. 656 p.

[20] More information about color shading. DxO. [WWW]. [referred 15.12.2013]. Available: http://www.dxo.com/intl/more-information-about-color-shading
[21] Keys, R. Cubic Convolution Interpolation for Digital Image Processing. IEEE Acoustics, Speech and Signal Processing 29(1981)6, pp. 1153 – 1160.
[22] Nikitenko, D., Wirth., M. & Trudel, K. White-balancing algorithms in colour photograph restoration. Systems, Man and Cybernetics, 2007. ISIC, Montreal, Quebec, Canada, October 7 – 10, 2007. IEEE. pp. 1037 – 1042.
[23] Bianco, S., Gasparini, F. & Schettini, R. Combining Strategies for White Balance. Image Analysis and Processing Workshops, Modena, Italy, September 10 – 13, 2007. IEEE. pp. 175 – 178.
[24] Hubel, P.M., Holm, J., Finlayson, G.D. & Drew, M.S. Matrix Calculations for Digital Photography. Fifth Color Imaging Conference: Color Science, Systems and Applications, Scottsdale, Arizona, USA, November 17 – 20, 1997. Society for Imaging Science and Technology. pp. 105 – 111.
[25] Lim, S.H. & Silverstein, A. Spatially Varying Color Correction Matrices for Reduced Noise. Twelfth Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications, Scottsdale, Arizona, USA, November 9, 2004. Society for Imaging Science and Technology. pp. 76 – 81.
[26] Vora, O. & Herley, C. Trade-offs Between Color Saturation and Noise Sensitivity in Image. Image Processing, 1998. Vol.1. Chicago, Illinois, USA, October 4 – 7, 1998. IEEE. pp. 196-200.
[27] McHugh, S. & Cambridge in colour. Understanding Gamma Correction [WWW]. [referred 15.12.2013]. Available: http://www.cambridgeincolour.com/tutorials/gamma-correction.htm
[28] [WWW]. [referred 15.12.2013]. Available: http://www.image-engineering.de/
[29] Douglas A. & Kerr, P.E. Color Temperature. Issue 4 November 8, 2005. 10 p.
[30] IEC 61966-2-1. Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB. Geneve, Switzerland 1999. 76 p.
[31] Parks, T.W. & McClellan, J.H. Chebyshev approximation for nonrecursive digital filters with linear phase. Circuit Theory 19(1972)2, pp. 189 – 194.