High Dynamic Range Image Construction and Noise Reduction Using Differently Exposed Images

Anton Öhrn

Master's thesis
2014:E50

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

High Dynamic Range Image Construction and Noise Reduction Using Differently Exposed Images

Anton Öhrn
Anton.ohrn@gmail.com

September 19, 2014

Master's thesis work carried out at Microsoft.

Microsoft supervisor: Anders Stålring, Anders.stalring@microsoft.com
LTH supervisor: Magnus Oskarsson, magnuso@maths.lth.se
Examiner: Kalle Åström, kalle@maths.lth.se

Abstract

This thesis discuss how to use mutual information in differently exposed images to improve noise reduction. It also investigates how one can create an high dynamic range (HDR) image from multiple differently exposed images taken with a handheld camera while simultaneously cope with the problems this approach introduces.

The proposed method and workflow in this thesis is based on the non-local means algorithm that uses both image registration and a intensity transformation to takes advantage of the mutual information in the different images. It uses alpha expansion to select contiguous areas for the HDR construction and image blending to create seamless transitions between the images.

The noise reduction algorithm shows better results with an intensity based noise level compared to a constant one. The methods used to take advantage of the mutual information are proven to be inadequate as the result for using noise reduction in a single image is shown to be just as good. The HDR image construction using both alpha expansion and image blending works well. For smaller movements the approach shows good result and even works as an anti-ghosting algorithm but for larger movements ghosting artifacts are introduced.

Keywords: HDR, HDR construction, Noise reduction, portable device imaging, image analysis.

Acknowledgements

A big thank goes to my supervisors both at Microsoft, Anders Stålring, and at LTH, Magnus Oskarsson who both were supportive and helpful throughout the creation of this master thesis and gave constructive feedback on my ideas. Thanks to Johan Windmark at Microsoft in Lund who helped me when my supervisors were on vacation and also proof read this thesis. Thanks to the core team at the Microsoft office in Lund who made me feel very welcome during my 6 month project. I would also like to thank Christian Makela at Microsoft in Tampere, Finland, who helped me clear out some questions regarding noise reduction in general and how to handle the Bayer data. A final thanks goes to my family and friends for their support.

Contents

Chapter 1
Introduction

1.1 Background

All images taken with a digital camera today contain some kind of noise. Depending on the hardware used and the light condition in the scenery this noise will be more or less present. The dominant part of the noise in an image comes from hardware errors which can be reduced by improving the hardware itself. However, the cost of doing so as well as the size restriction for smaller devices, such as a smart phone, will inevitably put restrictions upon the improvements. These aside it is still impossible to remove all noise completely on the hardware level. This is due to the nature of light itself which is not that easily measured. The cause for this is that light can be considered as "packages" of photons that does not follow a steady stream but are in fact random in their behaviour. This effect is not noticeable for any normal light source as these sends out billions of photons per second. If one would instead study a source which sends out a very small amount of photons these fluctuations would be noticed. One must instead rely on different algorithms in order to reduce the noise and produce an image of good quality.

When it comes to noise reduction in images the state of the art algorithm has been the same algorithm for a couple of years, namely the Block-Matching and 3D filtering algorithm (BM3D) by K. Dabov et al. [11]. This algorithm will be explained in detail in the theory chapter 2.

Noise reduction for multiple images is something that is also of interest. One such case is that of a volume containing multiple images, such as a CT-scan. In this case an algorithm based upon BM3D is commonly used namely BM4D, [14]. Another multiple images case is that of a video sequence which has been done in a version of BM4D namely V-BM4D, [15].

A special case for multiple images is that of a sequence of images with different exposure. When it comes to same exposure images the mutual information can be used without a problem, however for high dynamic range (HDR) sequences of multiple images the exposure difference can render a normal approach useless. The image does however contain information that can be used for noise reduction. To take advantage of this information will require some effort and more clever methods. This has been done before by C. Aguerrebere et al. [9] which was an insperation for this thesis.

HDR images are images with a higher resolution in intensity space than a standard digital image. This means that HDR images can more accurately reproduce the scene of which the photograph was taken. A consequence of this is that one gets more details in both dark as well as bright areas. These areas would otherwise be completely black (under exposed) or completely white (over exposed). One way to construct an HDR image is to take multiple images with different exposure time and then merge them into a single image. If the images are taken with a non-fixed camera such as a handheld device there will be movements across the HDR sequence which needs to be accounted for. If there is also a time difference between the images a possibility of ghosting artifacts are introduced.

HDR often gets confused with tone mapping. This comes from the fact that HDR images are often tone mapped to show the content. Tone mapping is a set of image processing algorithms that enhances the image in different ways. They alters the content of the image by for example increasing the intensities of the colors. HDR on the other hand increases the information in the image by extending the intensities possible. This makes HDR images impossible to view on a normal monitor as it can not display the intensities that they contain. In this thesis all intensities witch are outside the boundaries are mapped into the viewable range to preserve these structures.

A problem when it comes to both noise reduction and HDR image construction is how to use the different images in a good way. This problem gets even harder when one considers the case of handheld devices such as a smart-phones or digital cameras. For such devices the images are not stationary but will have differences between them. One such difference comes from the movements the photographer might exert the camera to. These movements must be taken into consideration as the HDR is constructed. One approach might be to use an image registration algorithm. This will result in potentially smaller search areas for the noise reduction algorithm as well as making the HDR construction easier.

The construction of HDR images is another problem which has been looked upon multiple times. When it comes to a static scene it is something that can be done and the result is for the most cases good. However when the pictures are taken with a handheld device they are bound to have, as mentioned before, differences between the images from rotations and movements. Another source of concern is that of moving objects inside the image sequence such as humans, cars, branches, leaves, etc. these movements can, if not handled correctly, result in so called ghosting effects. Ghosting effects are shown in the final image as areas that becomes blurry.

1.2 Previous Work on the Subject

One of the major inspirations for this thesis project is the paper "Simultaneous HDR image construction and denoising for dynamic scenes", by Cecilia Aguerrebere, Julie Delon, Yann Gousseau and Pablo Muse. The data used, code as well as the paper can be accessed at the web page [4]. In the paper it is discussed how HDR construction can be made as well as denoising. The report simultaneously cops with three problems: irradiance estimation noise, camera motion (hand-held camera) and multiple objects motion (dynamic scenes).

As for HDR construction there are a lot of papers dealing with it and how to construct it. For a completely static scene, the camera and the objects in the scene are fixed, there has been many solutions to this problem. In more recent approaches working with raw data as well as a known model for the camera, a statistical estimation problem for the construction based on the maximum likelihood estimation (MLE) is shown to be optimal [5, 13, 3].

1.3 Aim of the Thesis

The aim of this thesis is to investigate how to use the mutual information that multiple different exposure images contain to improve the noise reduction when producing an HDR image. The goal is to create an improved image compared to the single image case and to also compare it to the algorithms used today.

Another goal is to examine how one can create HDR images from images taken with a handheld device while simultaneously coping with any problems introduced.

1.4 Overview of the Thesis

The thesis start in chapter 2 by introducing the more specific theory needed to understand the algorithm and general theory related to the problem. In Chapter 3 the Methods and workflow of the algorithm is introduced and explained. Why a certain algorithm was chosen for a step instead of another is also motivated. In Chapter 4 the results and evaluation of the result is done. In Chapter 5 conclusions from the results are drawn and improvements are discussed. Lastly Chapter 6 contains future work to improve the algorithm further.

Chapter 2
Theory

This chapter contains the specific theory needed to understand some of the concepts of image analysis used in Chapter 3 and 4. The concept of noise, why it exists, were it comes from and how one can reduce it is stated as well as how to construct HDR images.

2.1 Image Data

Producing high quality pictures with a digital camera is nowadays something everyday users do not have to think about. However, the amount of effort going into the processing of an image from the sensor to the final image is much more than one might suspect. An example of an entire image processing pipeline can be seen in Figure 2.1. In this case shown as a Bayer pattern, the colorful matrix at the top left to the final example image at the bottom right. An image sensor's task is capturing light and converting it into electrical signal. As the image sensor is based upon reading the incoming light one is bound to have artifacts in the recorded data, and these artifacts are commonly known as noise. The noise in an unprocessed image comes from many different sources and have several different characteristics, these will be discussed in more detail in the following sections, [9].

2.2 Noise in Images

Noise in images taken with a digital camera can have several different origins and can look in many different ways. In this subsection different kind of noise will be described and in some sense their respective origin will be evaluated. In many cases a common noise model is a Gaussian distributed noise. This will be a good first approximation but in order to improve the result one might need to take noise with other distributions into account. A noisy image as it is from the sensor can be seen in Figure 2.2 and it's noise reduced counterpart can be seen in Figure 2.3.

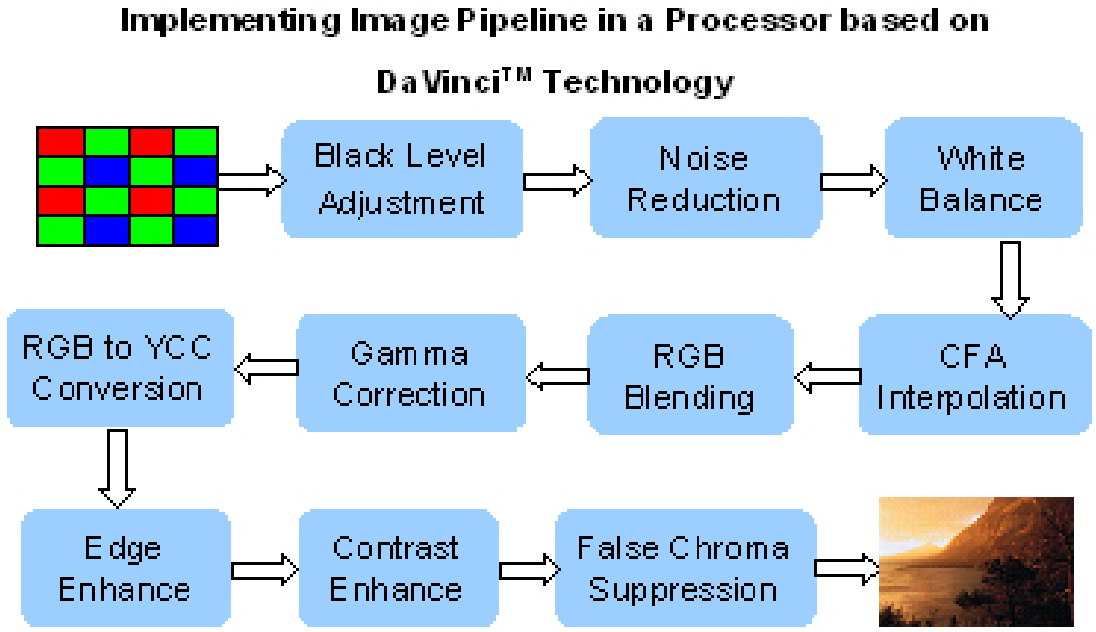

Figure 2.1: Example of a pipeline for a digital camera, from the sensors Bayer pattern to the final image displayed to the user. [24]

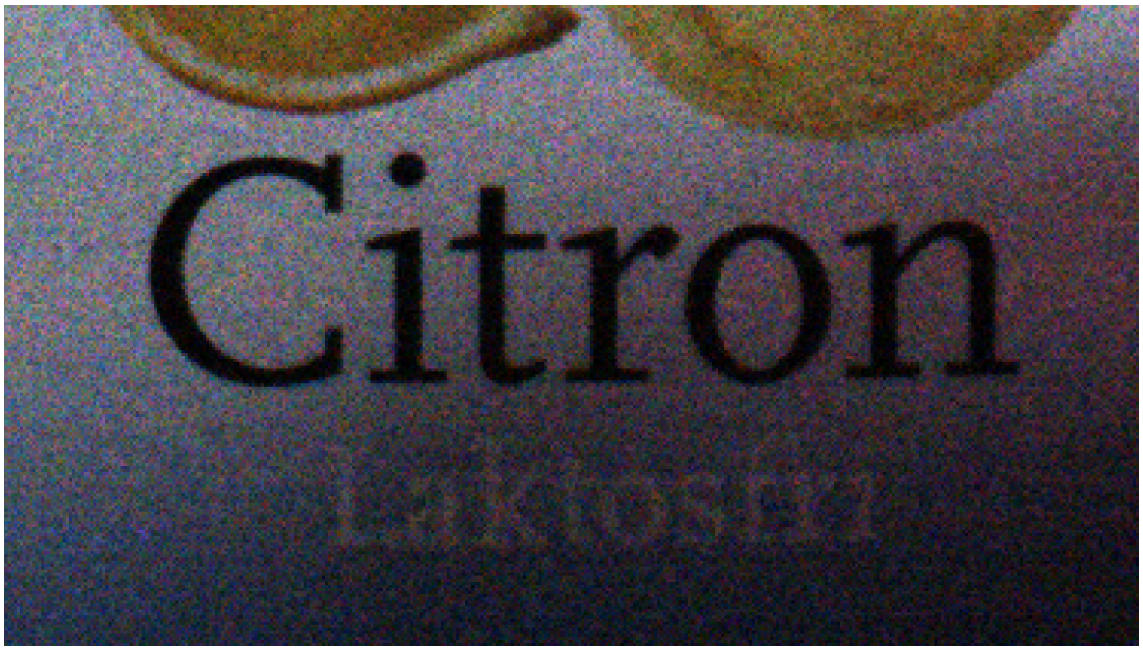

Figure 2.2: Figure showing a noisy image as it is from the CCD.

Figure 2.3: Figure showing a noise reduced image.

2.2.1 Gaussian Noise

Gaussian noise (or Normal distributed noise) is noise having a probability density function (PDF) equal to that of the Gaussian distribution. In other words the value that the noise can take follows a Gaussian distribution, which for a gray scale image takes the form of the following equation

$$p_G(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.1}$$

where x is the gray value of the pixel, μ is the expected value and σ is the standard deviation, therefore its variance is σ^2. Gaussian noise in the case of digital images can arise due to sensor noise caused by one or several of the following, poor illumination, high temperature and electronic circuit noise [6]. A example of Gaussian noise and the corresponding image histogram can be seen in Figure 2.5. An example of this image without any noise and its corresponding image histogram can also be seen in 2.4.

2.2.2 Impulse Noise

Impulse noise, also called Salt-and-Pepper noise, is noise that appear as sparsely located pixels that has an intensity value that deviates a lot from the surrounding. Often seen as a light pixels (salt) in darker areas, and dark pixels (pepper) in white areas. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission or other type of hardware related errors. If impulse noise is present in an image it is common that the values of the noise is large compare to the signal strength, [6]. An example of impulse noise and its corresponding image histogram can be seen in Figure 2.6, here a noise density d is given which is the approximate number of pixels effected by the noise.

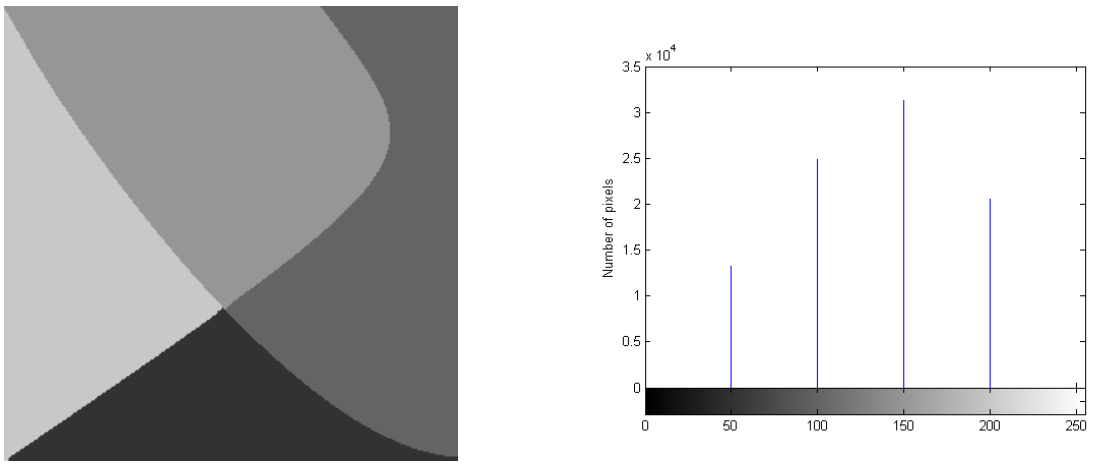

Figure 2.4: Noiseless image and its corresponding image histogram. Here y-axis contains the number of pixels and x-axis the value of the pixels (0-255).

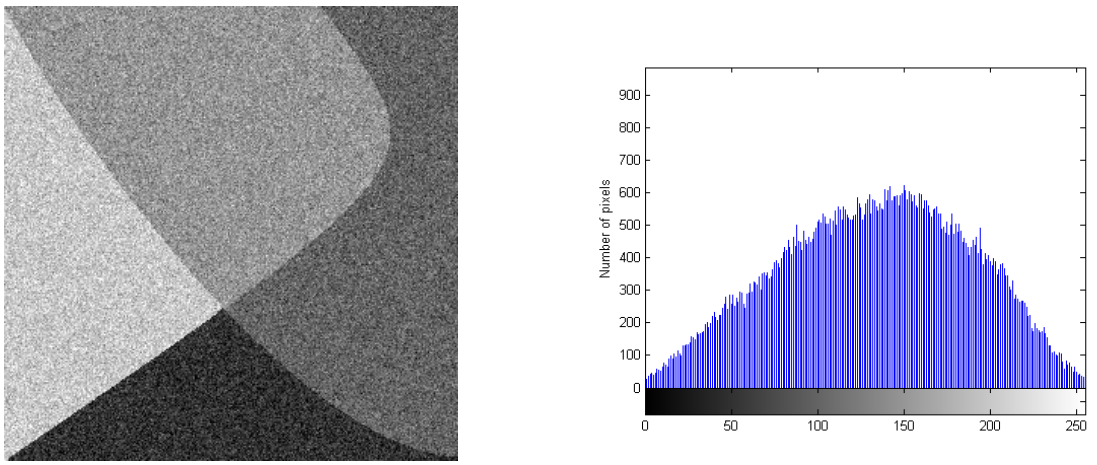

Figure 2.5: Figure containing an example of gaussian noise with expected value $\mu = 0$ and the variance $\sigma^2 = 3$ and its corresponding histogram , y-axis is counted number of pixels with a certain value that can be read on the x-axis (0-255).

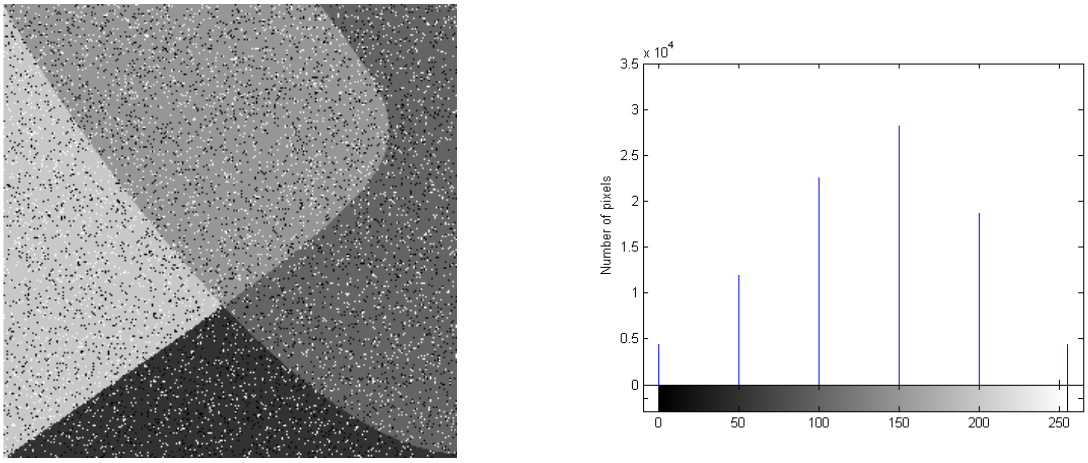

Figure 2.6: Figure containing an example of impulse noise with noise density $d = 0.1$, and its corresponding histogram. y-axis is counted number of pixels with a certain value that can be read on the x-axis (0-255). Here the axis x-axis has been widen in order to see the values at the endpoints.

2.2.3 Shot Noise

Shot noise is another type of noise that is of high importance when talking about HDR. Shot noise exists because of the nature of light itself is random. This randomness is not something that is easily detected by the naked eye as most light sources emits billions (if not more) photons each second. The randomness of shot noise makes it closest related to a Poisson distribution. The Poisson distribution for a discrete random variable X has the following formula

$$f(k;\lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{2.2}$$

where λ is the mean value, $\lambda > 0$, k are the number of occurrences, $k = 0, 1, 2....$

2.2.4 Usable Noise

Even though noise is an artifact that almost always is undesirable there are in fact cases where noise is used to enhance the image. This is of course not in the sense of signal to noise ratio (SNR) but in the way a human will interpret the image. Noise purposely added to an image to prevent discretization artifacts is called dither. Noise can also be added to an image in order to increase the acutance, i.e. the perceived sharpness of the image [20].

2.3 Model for noise

To create a model one can base it of the number of electrons, I, at a specific collection site.

$$I = T \int_{\lambda} \int_{y} \int_{x} B(x, y, \lambda) S_r(x, y) q(\lambda) dx dy d\lambda, \tag{2.3}$$

where T is integration time (in seconds), (x, y) are continuous coordinates on the sensor plane, $B(x, y, \lambda)$ is the incident spectral irradiance (Watts/unit area), and $q(\lambda)$ is defined as the ratio of electrons collected per incident light energy for the device as a function of wavelength λ. $S_r(x, y)$ is the spatial response of the collection size. Assuming that the noise is not wavelength dependent and considering all the noise types that can be present gives the final model

$$D = (KI + N_{DC} + N_S + N_R)A + N_Q. \tag{2.4}$$

The model consists of KI where K is a constant, N_{DC} is the number of electrons due to dark current which is a small electric current that flows through the device even when no photons are entering it. N_S is the zero mean Poisson shot noise with variance depending on the number of collected photons KI. N_R is static noise from amplifying units inside the camera, A is the combined gain of the amplifier and the camera circuitry. N_Q is a noise model from quantization process of going from voltage into a digital signal D, [9]. This model contains multiple sources which depends on each other which makes is unnecessarily hard to create a noise reduction algorithm upon. Instead, as introduced in the next section, a good model does not have to be as complex as shown above.

2.4 Poissonian-Gaussian Noise Model

As suggested by A. Foi et al. in [7], a simple but still fairly accurate model for the noise in a modern digital camera can be based on a Poissonian part, modeling the photon sensing, and a Gaussian part, for the remaining stationary disturbances. The reason one chooses to use both a Gaussian and a Poissonian component comes from the fact that the pixels in today's cameras are small and tightly packed to increase the pixel count. This results in a higher susceptibility to photon noise, i.e. measurement errors on the pixel level. This source of noise is the single most significant contribution to the overall noise, [7]. Photon noise varies with the intensity as well, which leaves an approximation out of the question as one can not make sure there is enough data for any given intensity. The following equations and assumption upon the data is done in [7], Let us consider the generic signal-dependent noise observation model of the form

$$z(x) = y(x) + \sigma(y(x))\xi(x), \tag{2.5}$$

where $x \in X$ is the pixel position in the domain X, $z : X \to \mathbb{R}$ is the observed (recorded) signal, $y : X \to \mathbb{R}$ is the original (unknown) signal, $\xi : X \to \mathbb{R}$ is zero-mean independent

random noise with the standard deviation equal to 1, and $\sigma : \mathbb{R} \to \mathbb{R}^+$ is a function of y that gives the standard deviation of the overall noise component. From $E\{\xi(x)\} = 0$ follows that $E\{z(x)\} = y(x)$, i.e. the original signal can be defined as the expected value of the noisy observations. Consequently, one has that $std\{z(x)\} = \sigma(E\{z(x)\})$, i.e. the standard deviation of the noise is a function, namely σ, of the expectation of the noisy signal.
In the modeling, one assumes that the noise term is composed of two mutually independent parts, a Poissonian signal- dependent component η_p and a Gaussian signal-independent component η_g:

$$\sigma(y(x))\xi(x) = \eta_p(y(x)) + \eta_g(x). \tag{2.6}$$

In terms of distributions, these two components are characterized as follows,

$$\chi(y(x) + \eta_p(y(x))) \sim \mathcal{P}(\chi y(x)), \quad \eta_g(x) \sim \mathcal{N}(0, b),$$

where $\chi > 0$ and $b \geq 0$ are real scalars and $\mathcal{P}$ and $\mathcal{N}$ denote the Poisson and Gaussian distributions. From the elementary properties of the Poisson distribution, one obtains the following equation for the mean and variance

$$E\{\chi(y(x) + \eta_p(y(x)))\}) = var\{\chi(y(x) + \eta_p(y(x)))\} = \chi y(x).$$

Since $E\{\chi(y(x) + \eta_p(y(x)))\}) = \chi y(x) + \chi E\{\eta_p(y(x))\}$ and $\chi^2 var\{\eta_p(y(x))\} = \chi y(x)$, it follows that $E\{\eta_p(y(x)))\}) = 0$ and $var\{\eta_p(y(x)))\} = y(x)/\chi$. Thus, the Poissoninan η_p has variance that depends on the value of $y(x)$, $var\{\eta_p(y(x))\} = ay(x)$, where $a = \chi^{-1}$. The Gaussian component η_g has instead constant variance equal to b.
Consequently, the overall variance of z in 2.5 has the affine form

$$\sigma^2(y(x)) = ay(x) + b, \tag{2.7}$$

which gives the standard deviation σ as the square root

$$\sigma(y(x)) = \sqrt{ay(x) + b}, \tag{2.8}$$

and, in particular, $\sigma(0) = \sqrt{b}$ and $\sigma(1) = \sqrt{a + b}$.

The Poissonian-Gaussian model (2.5 - 2.6) is naturally suited for the raw-data of digital imaging sensors. The Poissonian component η_p models the signal-dependent part of the errors, which is essentially due to the photon noise, while the Gaussian η_g accounts for the signal-independent errors such as electric and thermal noise. For more information on the noise model used and the algorithm used to estimate the noise in the images the reader can look up [7] in the bibliography.

2.5 Exposure

Exposure is the amount of light that hits the sensor, i.e the amount of photons that hits the sensors per unit area. The exposure is determined by shutter speed (the time in which one lets the light hits the sensor) and lens aperture (the effect of the lens through which the light travels before it hit the sensor). The exposure is however mostly effected by the luminance of the scene which one tries to capture. This means that if the scene is dark the exposure will be low and one needs to capture light for a longer time period in order to achieve the same detail visibility as a well lit scenery.

In the general case one uses the notion of "Exposure Value" or EV to express the difference in exposure. The general equation for this is the following

$$EV = \log_2 \frac{N^2}{t},$$

where N is the relative aperture (also called the f-number) and t is the exposure time or shutter speed in seconds. To change the EV of the images in HDR one changes the exposure time. This is often done in the following way. One takes an ordinary image with standard settings corresponding to the luminance of the scene. This is referred to as the $0EV$ image. Then a set number of images are taken on both sides of the $0EV$ image, the most common case are 1 image on each side. This is not a must, one can take multiple images on either side of the reference one. For the most part the images are taken with the same number of images on each side of the $0EV$ image but this can change depending on the light conditions of the scene. With $\pm 1.0EV$ is about a factor 2 difference of the exposure time, $\pm 2.0EV$ is about factor 4 difference.

$\pm 0.5EV$	$\frac{1}{880}$s	$\frac{1}{1284}$s	$\frac{1}{1764}$s
$\pm 1.0EV$	$\frac{1}{618}$s	$\frac{1}{1250}$s	$\frac{1}{2439}$s
$\pm 2.0EV$	$\frac{1}{356}$s	$\frac{1}{1248}$s	$\frac{1}{4902}$s

In the table above one can see that the different exposure times correspond to factor mentioned above. As for the $\pm 0.5EV$ case the factor is around 1/3, i.e. 2/3 times the $0EV$ case for the $-0.5EV$ and 4/3 times $0EV$ for the $+0.5EV$ case.

As a part of the algorithm used in this thesis (see Chapter 3 for details on the algorithm), one wants to map the different intensities to the same intensity in order to use them for denoising. With too high exposure difference this is more or less impossible as too much of the information is lost. One solution might be to use very low difference in exposure say $\pm 0.5EV$. The histograms will look much more similar but this will come at the cost of the dynamic range, i.e. one will not get as great of an improvement in darker and brighter areas as one wants. In this report a compensation is made by taking all images with $\pm 1.0EV$. Another assumption used is that the number of images are set to 3. For further HDR effect one might consider to change this number to 5 with $0EV$,$\pm 1.0EV$,$\pm 2.0EV$, this will work with the same principle, however the computational effort will be increased. As of now the number of images have been fixed to 3.

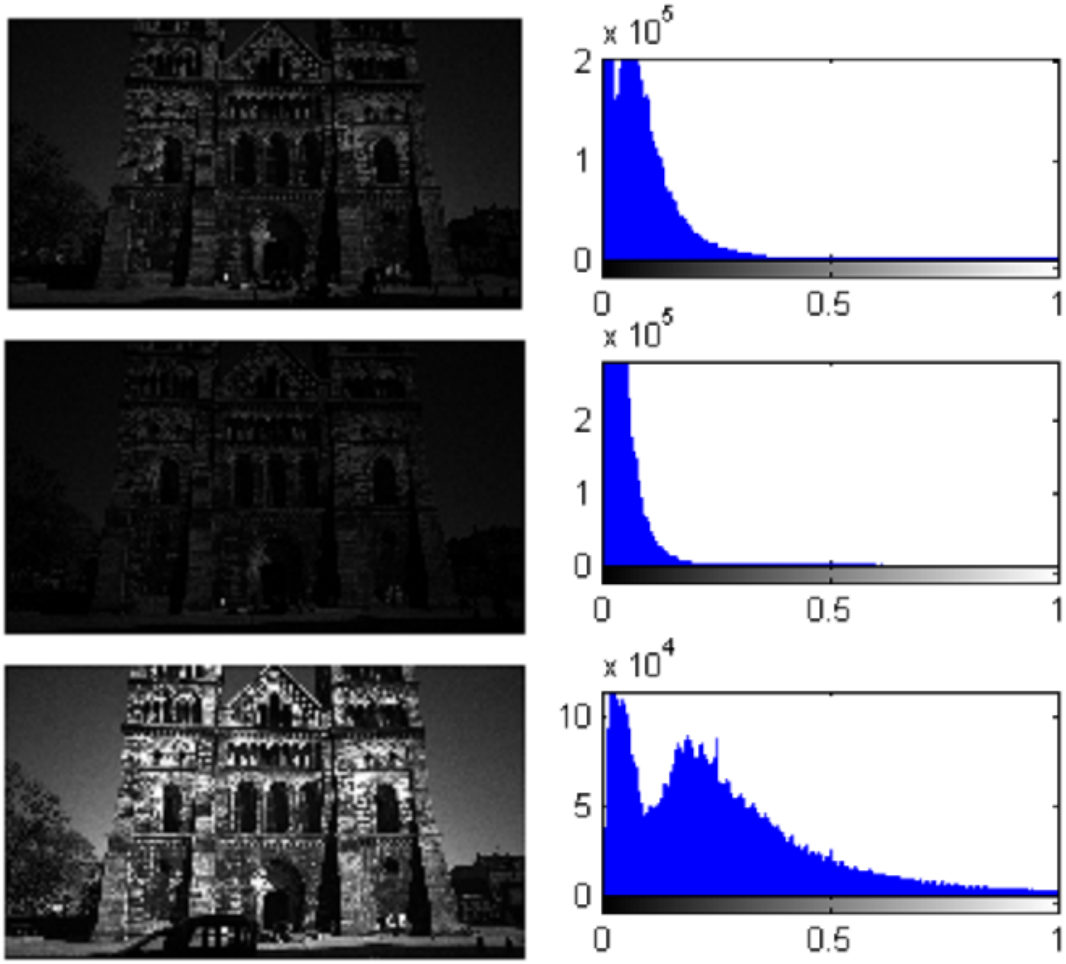

Figure 2.7: Figure containing the gray version of the $0EV$ (top), $-1EV$ (middle) and $+1EV$ (bottom) as well as the corresponding histogram.

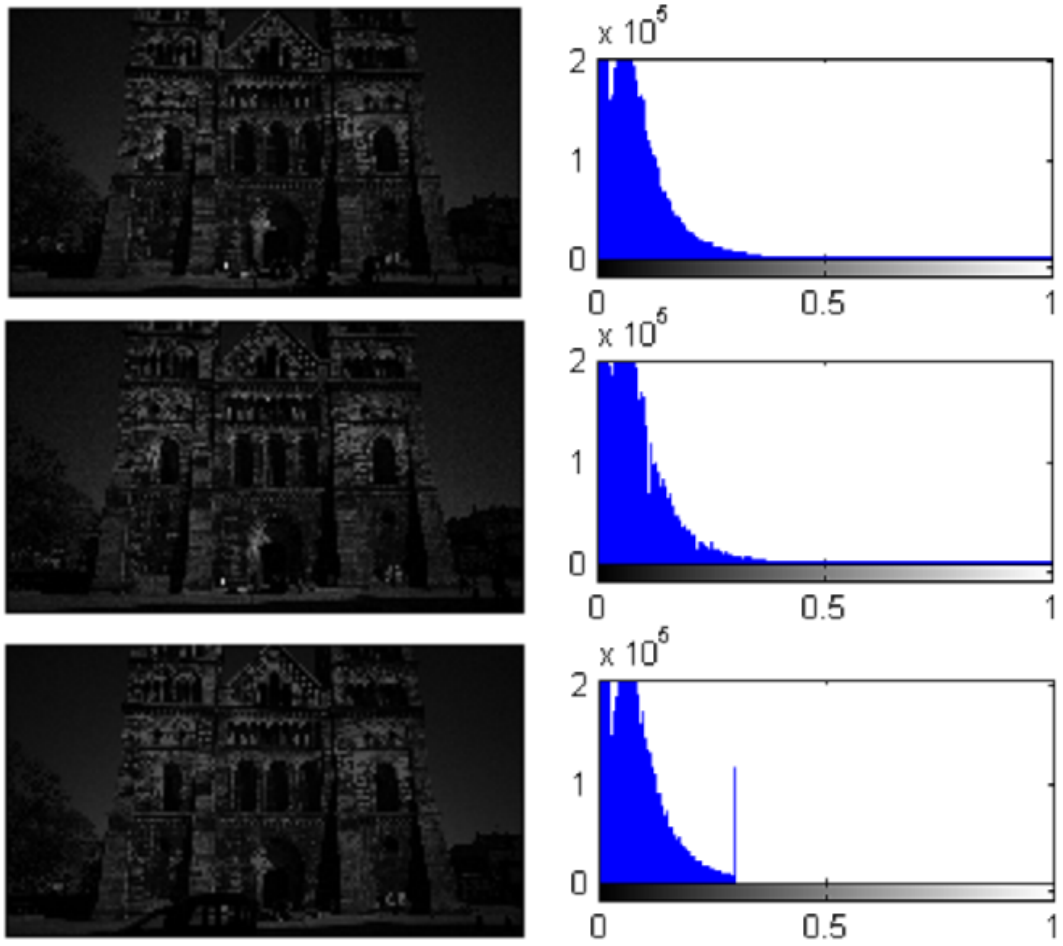

Figure 2.8: Figure containing the gray version of the $0EV$ (top), $-1EV$ (middle) and $+1EV$ as well as the corresponding histogram after the gray level transformation has been done.

2.6 Image Registration

Image registration is the process of finding a transformation that maps different images into the same coordinate system. The coordinate system is often set to be one of the images that needs registration. One of the ways you can do this, also the one selected during the algorithm, is to use image features. The easiest way to illustrate an image registration using features is by showcasing the work-flow of the algorithm. In Figure 2.10 one can see the difference between two images that one wants to do noise reduction on. But in order to do so one first need to register, the method of image registration can be seen in 2.10.

2.6.1 The Parallax Error

When using multiple images taken from the same scene the so called *parallax error* can arise. The parallax error is the visibility difference in a scenery that comes from watching it from two different viewpoints. An illustration of this phenomenon can be viewed in Figure 2.9. If two different images taken of the same scene but the viewpoint is too different or the object is very close an image registration might be impossible to perform. In most cases for this thesis this will not be a problem as both the viewpoint and the "distant background" usually is stationary. Objects of smaller size will not cause a big problem as one will see. Larger objects that cover more or less the entire view field that has a larger movement will make any registration impossible. If the camera center is stationary or if the scene is planar no parallax error can occur.

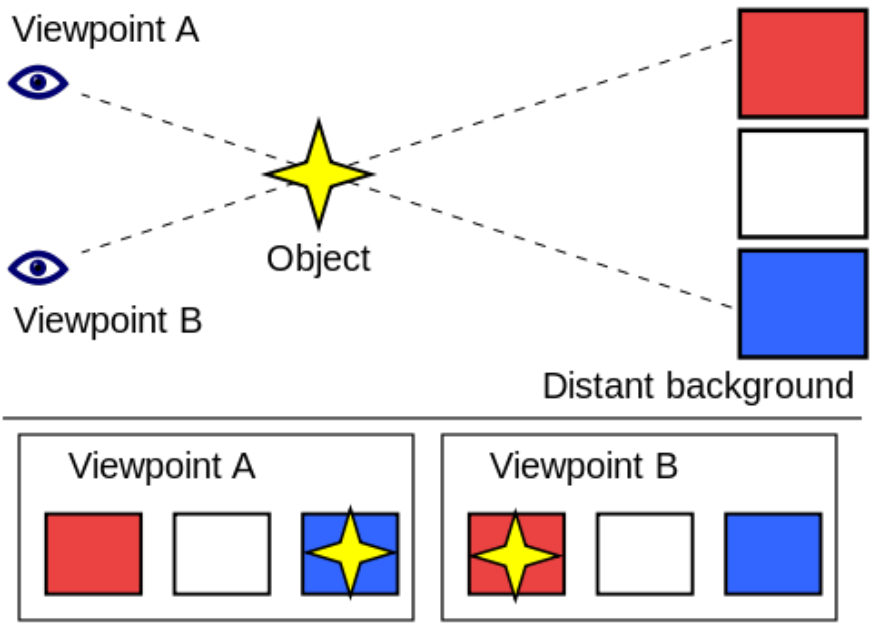

Figure 2.9: A simplified illustration of the parallax error. When viewed from "Viewpoint A", the object appears to be in front of the blue square. When the viewpoint is changed to "Viewpoint B", the object appears to have moved in front of the red square, [22].

Figure 2.10: Image showcasing two images, top left and top right, that through an affine transformation using the featured showed in the bottom left sub image are registered, bottom right.

2.6.2 Image Transformation

An image transformation is a function that takes the coordinates of one image, the target, and maps it to another image, the source, reference frame (coordinate system). This results in that objects in the both images overlap, the result of such a transformation can be seen in Figure 2.10. However, the images might have areas that do not overlap even with an optimal registration. This since part of the image might not even be in the other image due to moving obejects etc.

The kind of transformation needed to perform to map the target image to the source image depends entirely upon the difference between the images. The following are feasible examples of geometric transformations.

The most simple of transformation is that of a movement in either x, y or a combination of both. This is called an **translation** and has the following matrix form,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

where x and y are the old coordinates of the target system, t_x and t_y are the translations in x and y direction respectively denoted by the subscript. The coordinates are in their homogeneous form, hence the 1 at the bottom row. This means that the both coordinates exist in the same plane. A **translation** preserves the orientation of the image, the image will in other words not rotate or be scaled in any way.

The second transformation is a so called **Euclidean** or **Rigid** transformation. It is the same as the **Translation** transformation but with an added rotation resulting in the following matrix form,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

where θ is the angle needed in order to align the images. No lengths are altered by this transformation.

The third transformation is a so called **Similarity** transformation, it is the same as the **Euclidean** transformation but with an added scaling. This results in that lengths no longer are preserved but changed according to the scale r, angles are however preserved.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} r\cos\theta & -r\sin\theta & t_x \\ r\sin\theta & r\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Affine transformation preserves parallelism and has the following form

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & t_x \\ a_{10} & a_{11} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

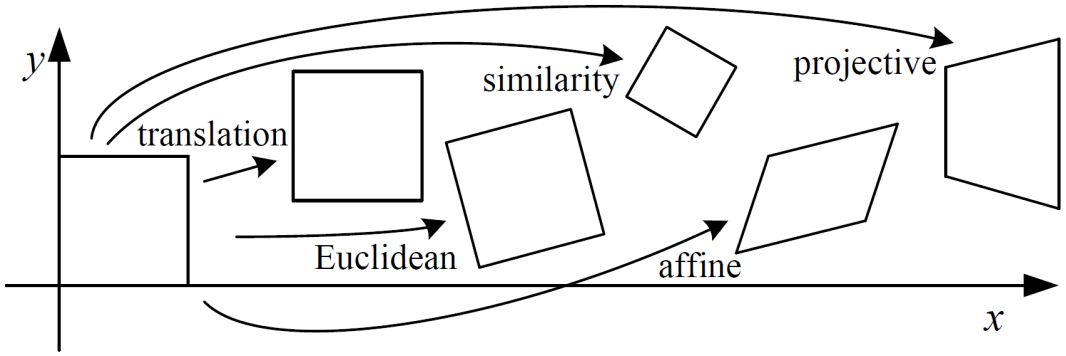

Figure 2.11: Basic set of 2D planar transformations, [19].

The last transformation introduced is a so called **Projective** transformation and it preserves straight lines,

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

2.6.3 Feature-based image registration

The first step of the registration mentioned previously is to find features in both the images. An image feature can be found in multiple ways, a common way to do this is to search for edges or corners. This features are then mapped to the most likely corresponding feature in the other image.

The approach used in this thesis can be seen in Figure 2.10. The features were found using the *MATLAB* function *detectSURFFeatures* and *extractFeatures*, these are features of the type SURF(Speeded Up Robust Features), [10]. The process then proceed to map all the different features, used in this thesis was the *MATLAB* function *matchFeatures*. These features are matched to each other in a most likely correspondence sense using nearest neighbour ratio, [12]. One problem that can arise in this step is that the mapping also find features that are very far apart if the features are well matched, such a match can be seen in Figure 2.10 in the middle right sub image. It is therefore important to set a maximum distance between the features, the features that exceed this distance can not be used as they do not correspond to the rest of the features. The features mapped with a larger distance then the maximum allowed are called outliers, the ones mapped within this distance are called inliers. The selected inliers can be seen in Figure 2.10 in the bottom left corner.

After the inliers are found the algorithm then finds a transformation matrix that can transfer the images to the same space. This was done using the *MATLAB* geometric estimation function *estimateGeometricTransform*, keeping one of the images stationary and applying the estimated transform to the other image. This transform is extracted from the features by estimating values in the image transformation matrix that makes the equation

add up. The transformation between the images can have different forms but a rotation and a translation is often the best correction to make, this is as previously stated the affine transformation. If the view-angle is changed between the time the pictures were taken a projective transformation will add a non-uniform scaling to the transformation matrix and take care of that distortion as well.

A problem with the feature mapping approach is the importance of well spread features across the image. If the feature detector fails to find features that cover more or less the entire image there is a risk that the transformation estimation will be faulty for areas with unmapped features. In the showcase, Figure 2.10, the features are well spread thus yielding a good transformation which can be seen in the bottom right sub image. The green and purple parts are parts that differ in the images, green is in one but not the other and vice verse for purple. The big colored areas on the edges are areas that is part of the source image but not in the reference image.

2.7 Bayer Filter

A Bayer filter is a mosaic of color filters placed over the image sensor of a digital camera. By doing so only the intensity of a specific wavelength is measured at each point of the sensor, see Figure 2.13. This results in gray-color images in the same image which which can be merged into a single color image. The reason this is beneficial is because one only needs one sensor for all colors were as one would need multiple sensors if the colors were to be measured separately. This is the standard way of recording digital color images.

A Bayer pattern is the way the mosaic pattern of the Bayer filter is designed. The most common Bayer pattern is composed of 3 different colors arranged in a matrix formation. This can be seen in Figure 2.12. The pattern consists of the upper left 2 by 2 matrix constellation repeated over the entire image sensor. In the general case one has 50% green, 25% blue and 25% red. One reason for using two green channels is that the green color tends to contain a larger amount of small structures such as leafs or grass, compared to blue areas that are often larger like for example the sky, and red most likely man made structures, houses, cars etc. It is therefore helpful to have double the amount of data for areas more prone to noise. Another reason is that green is in the middle of the visible light range making it easier to see noise in green areas [17]. This also result in that the human eye is more sensitive to variation in shades of green as it is more closely correlated with the perception of light intensity of a scene [18]. The colors in Figure 2.12 are arranged in a 'BGGR' pattern. There are however many different kinds of patterns. There are also different constellations of arrangements as the sensor might not be formed as a square at all.

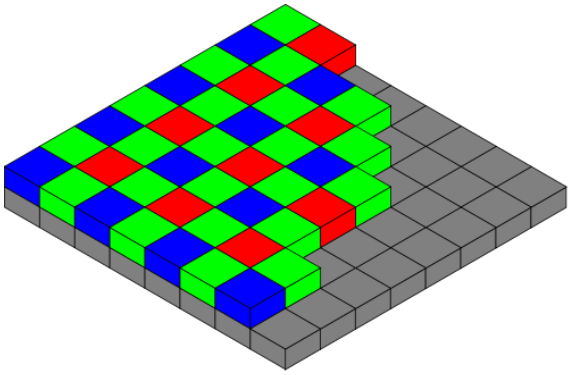

Figure 2.12: Bayer pattern arranged on top of a sensor, [18].

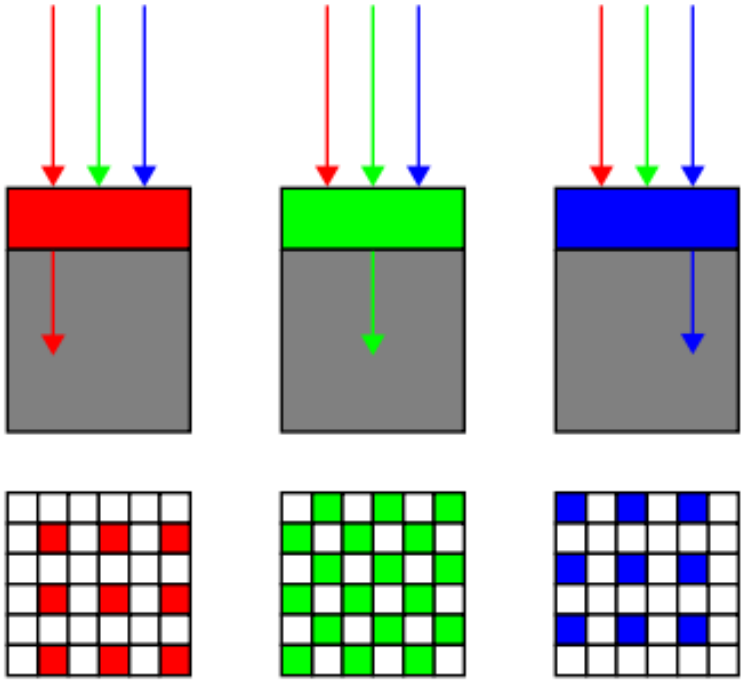

Figure 2.13: Bayer pattern profile showing how the different color channels are produced [21]

2.8 Noise Reduction Algorithms

The following sections contain theory of both the non-local means (NLmeans) and the Block-Matching and 3D Filtering (BM3D) algorithms. These algorithms work upon the principle of finding similarities in the images using so called block matching which will also be discussed.

2.8.1 Block Matching

The basis of both BM3D and the non-local means algorithm is to find similarities in the image. This is done by searching the areas surrounding the current pixel one is looking at. This method is usually referred to as block matching or patch matching. One would like to find a 2D neighborhood of a given size that matches the 2D area surrounding the pixel currently being noise reduced.

To be able to evaluate the similarity between the signal fragments the inverse of a distance measurement is often used. This results in a high similarity if the distance (the difference) between the signals is small. Various distance measurements can be used, such as the weighted Euclidean distance used in [1]. Grouping using patch matching is based upon the assumption that one has a lot of self-similarity in the viewed signal. For the image case this is often true when viewing anything man-made. This comes from the fact that humans tends to build straight buildings, roads, furniture, etc. If one has an object, it is likely that this object and the edges of said object is going to be extended over an area, thus making many matches possible. Patch matching can be used over the entire image but the computational cost will increase as one increases the search area.

2.8.2 Non-Local Means

One basic model for image denoising can look as the following one,

$$v(x) = u(x) + n(x), \tag{2.9}$$

where $v(x)$ is the observed noisy image, $u(x)$ is the true valued image and $n(x)$ is the noise. The noise will in most cases consist of several different distributions based on the pixel value x. It is often single or few pixels that alternate from the overall surrounding that one wants to get rid of but still keep structures that varies on a pixel to pixel basis. The denoised value at x is a mean of the values of all pixels whose Gaussian neighborhood looks like the neighborhood of x. The non-local means algorithm is based upon the previously mentioned block matching, in the way that one searches for similar blocks in a surrounding. The similar blocks are then used to reduce noise. This idea is more or less based on the principle of taking multiple measurements. In this case however, one use multiple similarities in the same measurement to reduce the noise, [1].

Given a discrete noisy image $v = \{v(x)|x \in X\}$, the estimated value $NL[v](x)$, for a pixel x, is computed as a weighted average of all the pixels in the image,

$$NL[v](x) = \sum_{j \in X} w(x, j)v(j), \tag{2.10}$$

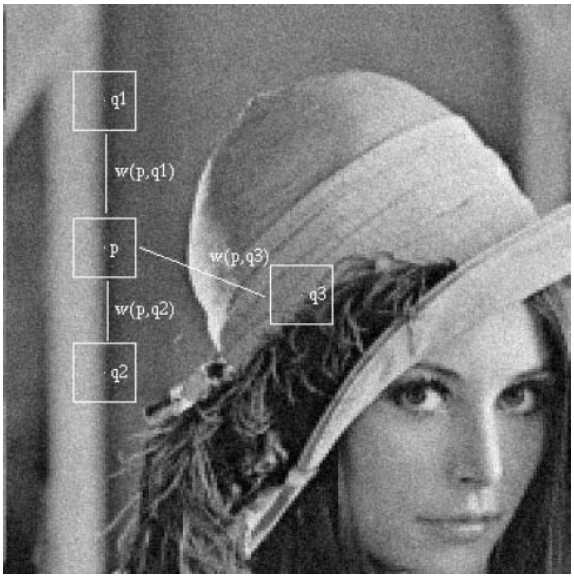

Figure 2.14: Scheme of NL-means strategy. Similar pixels neighborhoods give a large weight, $w(p, q_1)$ and $w(p, q_2)$, while much different neighborhoods give a small weight, such as $w(p, q_3)$, [1].

where the family of weights $\{w(x, j)\}_j$ depends on the similarity between pixel x and j, and satisfies the conditions $0 \leq w(x, j) \leq 1$ and $\sum_j w(x, j) = 1$. The similarity between two pixels x and j depends on the similarity of the intensity gray level vectors $v(\mathcal{N}_x)$ and $v(\mathcal{N}_j)$, where $\mathcal{N}_k$ denotes a square neighborhood of fixed size and centered at a pixel k. This similarity is measured as a decreasing function of the weighted Euclidean distance, $\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|^2_{2,a}$, where $a > 0$ is the standard deviation of the Gaussian kernel. The application of the Euclidean distance to the noisy neighborhoods raises the following equality

$$E\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|^2_{2,a} = \|u(\mathcal{N}_x) - u(\mathcal{N}_j)\|^2_{2,a} + 2\sigma^2.$$

This equality shows the robustness of the algorithm since in expectation the Euclidean distance conserves the order of similarity between pixels.

The pixels with a similar gray level neighborhood to $v(\mathcal{N}_x)$ have larger weights on the average. An example can be seen in 2.14. These weights are defined as,

$$w(x, j) = \frac{1}{Z(x)} e^{-\frac{\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|^2_{2,a}}{h^2}},$$

where $Z(x)$ is the normalizing constant defined as

$$Z(x) = \sum_j e^{-\frac{\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|^2_{2,a}}{h^2}},$$

and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

2.8.3 Block-Matching and 3D Filtering algorithm

The Block-matching and 3D filtering algorithm as presented by K. Dovan et al. in [11], is one of the most widely used denoising algorithms today. It is based on block-matching. The similarity between signal measurements is computed as the inverse of some distance measure. Hence, a smaller distance implies higher similarity. In the case of images these signal fragments are typically 2D neighborhoods. The blocks are then stacked into a 3D structure which can be processed. This concepts is denominated as grouping. The main principal of grouping is to enable the use of a higher-dimensional filter of each group, which takes advantage of any potential similarity between the blocks in the group to estimate the true signal in each of them.

The workflow of the algorithm, which can be seen in Figure 2.15, consists of 2 steps.

step 1: Basic estimate

a) Block-wise estimates: For each block in the noisy image two major operation are performed

i) Grouping: Search the image for matching blocks with block matching, then group all the matched blocks into a 3D array, a so called group.

ii) Collaborative hard-thresholding: Apply a 3D transform to the formed group, attenuate the noise by hard-thresholding of the transform coefficients, invert the 3D transform to produce estimates of all grouped blocks, and return the estimates of the blocks to their original position.

b) Aggregation: Compute the basic estimate of the true-image by weighted averaging all of the obtained block-wise estimates that are overlapping.

step 2: final estimate: Using the basic estimate a second run of the algorithm is performed. with an added collaborative Wiener filtering.

a) Block-wise estimates: similarly to step 1 the blocks are processed, this time however using a collaborative Wiener filtering instead of the collaborative hard-thresholding.

i) Grouping: Again using block-matching this time within the basic estimate to find the locations of the blocks similar to the currently processed one. Using these locations, form two groups (the stacked 2D blocks), one from the noisy image and one from the basic estimate.

ii) Collaborative Wiener filtering: Apply a 3D transform on both groups. Perform Wiener filtering on the noisy one using the energy spectrum of the basic estimate as the true (pilot) energy spectrum. Produce estimates of all grouped blocks by applying the inverse 3D transform on the filtered coefficients and return the estimates of the blocks to their original positions.

b) Aggregation: Compute a final estimate of the true image by aggregating all of the obtained local estimates using a weighted average.

The two main reasons for using the second step is firstly that the basic estimates improve the grouping by block-matching. Secondly, using the basic estimate as a pilot signal for the empirical wiener filter is much more effective and accurate than the simple hard-thresholding of the 3D spectrum of the noisy data. The mathematics behind the BM3D algorithm is not

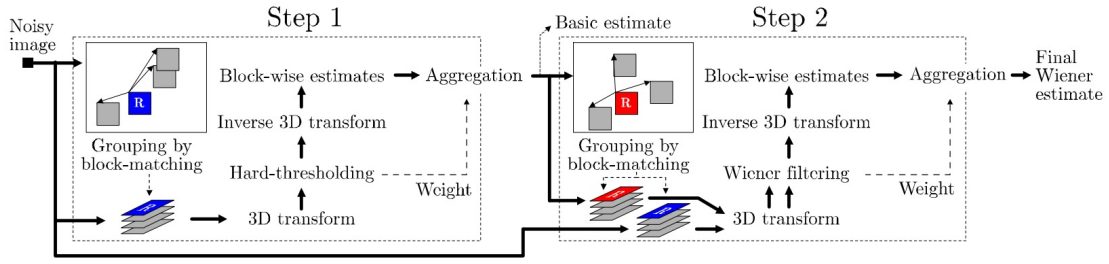

Figure 2.15: Flowchart of the BM3D denoising algorithm. The operations surrounded by dashed lines are repeated for each processed block (marked with 'R'). [11].

part of this thesis as the whole article would be needed to be included. To read the BM3D article as a whole see [11].

2.9 High Dynamic Range Construction

The following two sections are dedicated to the HDR construction portion of this thesis. The use of these algorithms will be clarified and explained in more detail in chapter 3.

2.9.1 Alpha Expansion

α – expansion is an algorithm used to categorize or assign a pixel $p \in P$ to a certain label f in a finite set $\mathcal{L}$ of labels. The way this is done is to create an energy function and minimizing it by re-assign the labels to the pixels.

$$E(f) = E_{smooth}(f) + E_{data}(f). \tag{2.11}$$

This will create coherent areas of same labeled pixels but keep pixels that deviate too much from the surrounding as the energy it takes to convert these would be too large. The used energy expression usually consist of a E_{data} part that has the following form

$$E_{data}(f) = \sum_{p \in P} D_p(f_p),$$

where D_p measures how well label f_p fits pixel p given the observed data. An example of $D_p(f_p)$ is $(f_p - I_p)^2$, where I_p is the observed intensity of pixel p, and f_p is the label intensity. Normally one works with quadratic smoothing terms

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q),$$

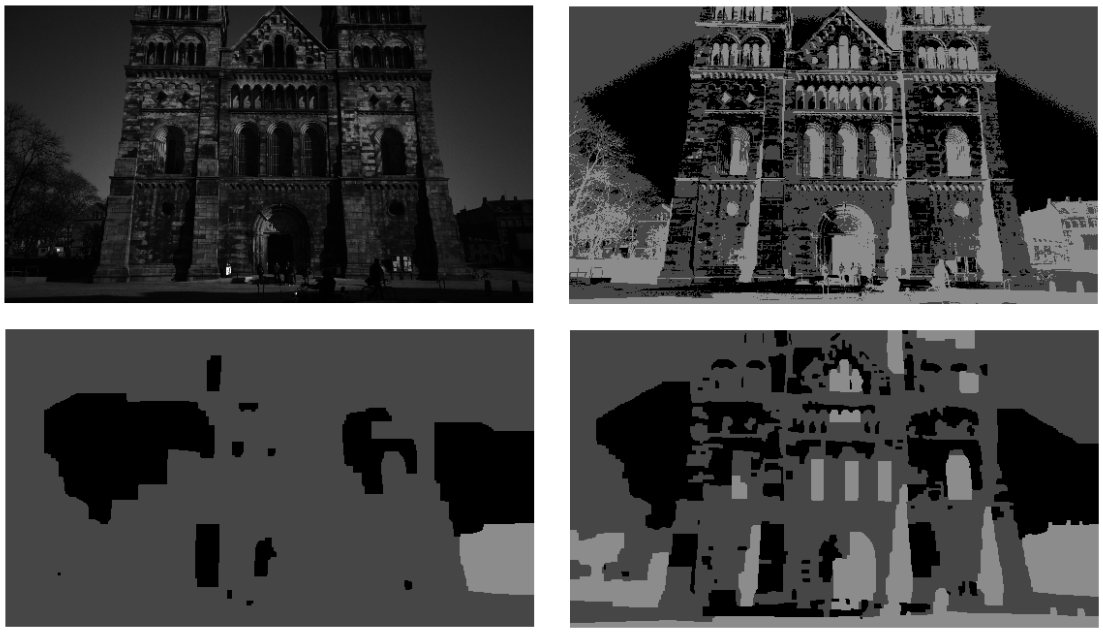

Figure 2.16: Figure comparing how the resulting image change depending on the selection of weight between E_{data} and E_{smooth}. Top right is the input image which, in this case, consists of a mean RGB of the $0EV$ picture. Top right is the result if two much emphasis is put on the E_{data} term, the images is put into the labels but do not change. Bottom left has the opposite problem, to much weight emphasis on the E_{smooth} term. The bottom right result is what can be considered as more balanced.

where $\mathcal{N}$ is the set of interacting pairs of pixels. In order for the energy minimization to converge constraints need to be put on how one picks the smoothing term. In this thesis the Potts interaction penalty $V_{p,q}(f_p, f_q) = \delta(f_p \neq f_q) \cdot \lambda$ is used. This interaction will introduce a penalty for neighboring pixels having different labels.

The two expressions, $E_{data}(f)$ and $E_{smooth}(f)$ are balanced in order to get the desired result. If all weight is put towards $E_{data}(f)$ the image will keep the structure and only label the pixels to the appropriate label depending on D_p. On the other hand if all emphasis is put on the $E_{smooth}(f)$ part one will create an image of only the dominant label in accordance to $V_{p,q}(f_p, f_q)$ over the image. An example of how the output changes depending on the term favored in the weighting can be seen in 2.16. In the article, [23], a detailed description of how this type of algorithm works and what kind of result it produces.

2.9.2 Image Blending

The concept of image blending is to seamlessly create transitions between images or areas which have been inpainted into an image. The difficult task when it comes to blending is to keep a good balance between smoothing out low-frequency variations and retaining sharp enough transitions to prevent blurring. There are several different kinds of methods

used but recently studies have focused on using the image gradient.

In order to be able to use the gradient for multiple images with different intensity the previously mentioned intensity transformation is performed before the image blending. This will make sure that the gradients are of the same magnitude and thus preventing lowering the contrast in areas picked from the low exposure image.

The Image blending algorithm used in this thesis have three main steps. The first step is to calculate the image gradient for all the images. One then select the gradients according to the map selected by the $\alpha-$expansion and integrate the image again. This integration can however not be done unless the function is a conservative field. In other words

$$\frac{\partial^2 I}{\partial x \partial y} = \frac{\partial^2 I}{\partial y \partial x},$$

this is however most likely not the case since the gradients are from multiple images. To be able to solve this one instead search for a function I whose gradient minimizes the integral

$$\iint F(\nabla I, G) dx dy, \tag{2.12}$$

where G is the gradient map from the α − expansion and $F(\nabla I, G) = \|\nabla I - G\|^2 = \left(\frac{\partial I}{\partial x} - G_x\right)^2 + \left(\frac{\partial I}{\partial y} - G_y\right)^2$. According to the variation principle, a function I that minimizes the integral must satisfy the Euler-Lagrange equation

$$\frac{\partial F}{\partial I} - \frac{d}{dx}\frac{\partial F}{\partial I_x} - \frac{d}{dy}\frac{\partial F}{\partial I_y} = 0,$$

which is a partial differential equation in I. Substituting F one obtains the equation

$$2\left(\frac{\partial^2 I}{\partial x^2} - \frac{\partial G_x}{\partial x}\right) + 2\left(\frac{\partial^2 I}{\partial y^2} - \frac{\partial G_y}{\partial y}\right) = 0.$$

Dividing by 2 and rearranging the terms gives the Poisson equation

$$\nabla^2 I = \text{div } G,$$

where ∇^2 is the Laplacian operator $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ and div G is the divergence of the vector field G, defined as div $G = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}$, [16].

To be able to integrate one also needs boundary conditions. The boundary condition chosen are of Neumann type and are put at the end of the image in the gradient direction. The boundaries are set so that the image gradient is zero at these edges.

Chapter 3
Methods and Approach

3.1 Overview

In this chapter a step-by-step description of the algorithm will be given according to the workflow which can be seen in figure 3.1. The chosen algorithm to deal with each of these tasks is discussed as well as some alternative algorithms.

Figure 3.1: The steps that the algorithm takes to achieve the HDR noise reduction.

The first step is the noise reduction. This step is performed early in the algorithm as one wants to work with as unaltered data as possible. In this thesis one works with multiple images which in the raw format share some common characteristics that will get modified if any processing were to be done before the noise reduction step. Another motivation to do noise reduction early in the algorithm is that if one alters the information in the image before the noise reduction one will also alter the noise. This would create even more dependencies in the noise, which will in return effect the noise reduction.

The second step is to process the images into the final viewable images. This step is done before the HDR construction as it was found to produce better results for the total workflow of this thesis.

The third step consists of merging the images into an HDR image by using the $\alpha-$expansion and image blending algorithms.

To simplify the work the following assumptions were made. The algorithm was fixed to take 3 input images taken with different exposure time. To be able to both incorporate the dynamic range but also be able to use the mutual information the input images were taken with taken with $-1EV$,$0EV$ and $+1EV$.

3.2 Noise Reduction

The first step in the workflow of the proposed algorithm is that of noise reduction. This step includes several sub-steps, which can be seen in 3.2.

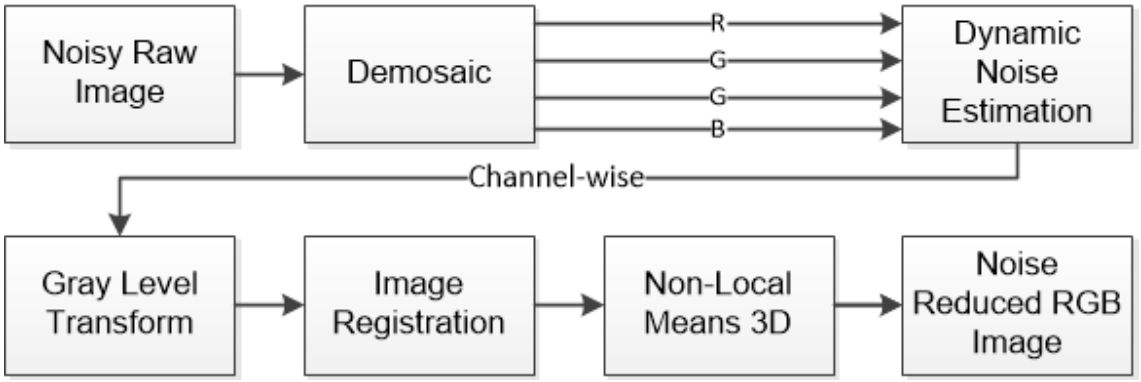

Figure 3.2: The noise reduction step broken down into a step-by-step overview

To take advantage of the mutual information in the different images it was concluded that searching through the vicinity of the pixel in each of the differently exposed images should be beneficial. The big concern is that when the intensity transformation is done not only the wanted information will be mapped but also the noise. This means that the darkest image, the image taken with the $-1EV$ setting, will have its noise increased by the corresponding factor k that mapped it to the $0EV$ image. When searching through the other images for corresponding patches and more importantly when information is used from these image this needs to be compensated for. Therefore an alternation to the weight function between patches were made which will be described in the section 3.2.5.

3.2.1 Demosaic

As mentioned in the theory section the sensor is arranged in a Bayer pattern. If one wants to access the four different color channels that lies within the Bayer pattern one can extract every other element according to the pattern constellation. This can, in the case of an 'grbg' pattern, be done with the following *MATLAB* code:

```
% Bayer pattern extraction from the pattern type 'grbg'.
Green1 = raw(1:2:end,1:2:end);
Red = raw(2:2:end,1:2:end);
Blue = raw(1:2:end,2:2:end);
Green2 = raw(2:2:end,2:2:end);
```

This approach will reduce the image resolution as the channels are treated separately and then merged. It can also suffer from artifacts along edges of objects in the image as the Bayer pattern does not have overlapping channels. The channels are instead slightly shifted which can result in small color errors were any edges are in the image.

These errors can be prevented by using a more advanced demosaic algorithm such as Hamilton Adams, [2]. The Hamilton Adams algorithm is based on an an interpolation method in order to keep the resolution of the raw image. This will however slow down the computations as more data needs to be handled. This can be prevented by down scaling the interpolated result from the Hamilton-Adams when doing computations.

In order to leave the noisy data as unaltered as possible one wants to work with the four different channels from the Bayer pattern individually. The reason four different channels are used and not three is because the two different green channels in the mosaic pattern shows a larger difference than expected. It was therefore chosen to keep the data unaltered during this step and only after the noise reduction take a pixel-wise mean to create a single green channel.

3.2.2 Dynamic Noise Estimation

As intensity varies over the image so does the noise. It would therefore be wrong to assume a constant noise level over the entire image. As described A. Foi et al. in [7] a noise model can be based on two different key components. Firstly a Poissonian part that models the photon sensing, and secondly a Gaussian part which models the remaining constant disturbances. The algorithm result in two parameters which corresponds to a linear estimation of the noise:

$$\sigma(x) = c \cdot x + m,$$

where σ is the standard deviation for the pixel, c is a positive scaling factor, x is the pixel value and m is a constant. If applied to all possible pixel values in the image one gets the corresponding noise variance curve, an example of this can be seen in figure 3.3. The actual noise level may very well fluctuate around this noise level curve leaving the some pixels either too much noise reduced or too little noise reduced. The code used is available

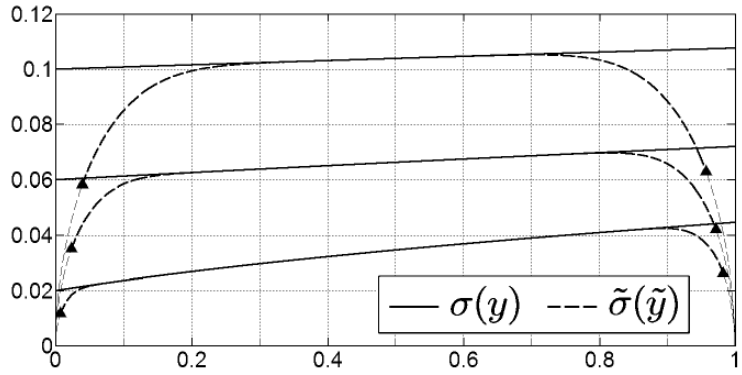

Figure 3.3: example of three different noise estimation levels. $\sigma(y)$ is the linear estimation and $\tilde{\sigma}(\tilde{y})$ are the clipped values, [7].

at the web page [8] and is based on the article [7]. To implement this kind of method from scratch was to time demanding and thus the code was used as a step in the algorithm instead. This approach was chosen as it gave considerable better result then the evaluated alternatives as one can see in chapter 4.

3.2.3 Image Registration

Image registration, as introduced in the theory section, is the method of aligning images. This is mainly done in the algorithm to reduce the needed search area for the noise reduction algorithm, as well as making the HDR construction easier. The reason the images are not aligned in the first place comes from the fact that they were taken with a handheld device. This can result in movement in both x and y direction, rotations both clockwise and counter-clockwise (in case of 3 or more images) and a scaling compared to the chosen reference image. The scaling is in most cases relatively small compared to the rotation and global movement so a similarity transformation might be enough. It is however easier to go with a higher order homography then needed to assure a better registration and thus estimate a projective transformation instead.

This is done using the *MATLAB* function $estimateGeometricTransform$ which takes as input the matched features for both images as well as an string specifying which transformation should be used. The output of the algorithm are the matched inlier features and a transformation matrix T according to the input. The transformation matrix T is used in two different ways in this thesis. The first case is that of noise reduction. For this case one does not apply the transformation to align the images as that would alter the image information. Instead one changes the search area according to the transformation for the corresponding image. The second case is the HDR construction which needs the images to be registered in order to choose the correct image at the correct place.

One major flaw with the feature estimation of the transformation matrix is that enough features might not be found and therefore result in a faulty transformation. This will effect the result in the noise reduction part of the algorithm as it will not find the corresponding

areas to take advantage of the mutual information. More importantly is the HDR construction part. If the images are not registered the seamless transition between the images will be lost as a faulty part of the other images will be in-blended. These parts might not even be producing an HDR image at all as the selection to create it are based on the reference image. This effect will be more clearly seen in the result chapter.

3.2.4 Intensity Transformation

To be able to compare the different images, not only a registration is needed to have overlapping images, but one would also like to have the same values in corresponding pixels across the images. As mentioned in the theory section an intensity transformation can be done. To do this one could normalize the images with the exposure time as suggested and also done in [5]. In this thesis another approach was used instead, namely that of mapping the images channel-wise to the same gray scale by finding a linear transformation of the form

$$I_{trans} = k \cdot I_{normal} + m, \tag{3.1}$$

where I_{normal} is the image one wants to map to, I_{trans} is the mapped image, k is a global scaling factor and m is a constant. This transform is found by searching for correspondences between the two images. To find the best match an area is selected in the target image, another smaller block is then selected from the source image within the area selected from the target and a convolution is performed in order to find the best match. This procedure is performed over the entire image resulting in a number of corresponding blocks from which one can then estimate a transform of the above type.

3.2.5 Non-Local Means 3D

To do noise reduction using multiple differently exposed images an algorithm based on the non-local means is suggested, the non-local means 3D.

Step 1: Patches for each pixel: For each pixel in the image an area corresponding to the $2 \cdot z + 1$ neighborhood is selected.

Step 2: Search for corresponding patches: Corresponding patches for the pixel are searched for within a maximum distance d between the pixels. This is done by transforming the pixel and the search area according to the transformation matrix found by the image registration.

Step 3: Calculating weights: The weight of the pixel to noise reduce the pixel is created similarly to the normal non-local means but modified to handle multiple images.

$$w(x, j, n, m) = \frac{1}{Z(x)} \cdot k(n, m) \cdot e^{-\frac{\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|_{2,a}^2}{h(x)^2}}, \tag{3.2}$$

x is the pixel one is currently noise reducing by comparing the gray level vector $v(\mathcal{N}_x)$ for the patch surrounding it with the gray level vector $v(\mathcal{N}_j)$ which surrounds the pixel j.

$k(n, m)$ is the intensity transformation factor between the image n which is the image one is currently noise reducing and m which is the image one is searching for similarities in. In the case were $n = m \Rightarrow k(n, m) = 1$.

$Z(x)$ is the normalizing constant defined as $\sum_j k_n e^{-\frac{\|v(\mathcal{N}_x) - v(\mathcal{N}_j)\|_{2,a}^2}{h(x)^2}}$

$h(x)$ is the intensity depend noise standard deviation and a is the standard deviation of the Gaussian kernel.

The weights used consists of a $k(n, m)$ part mapping the image to the same intensity by keeping one image stationary and scaling the other with a linear factor, a decaying exponential part which makes the weight small when the difference is large between the patches, a Gaussian kernel to normalize and an intensity based noise level $h(x)$ to make appropriate noise reduction based on the intensity. The noise reduced pixel is then calculated as the weighted average over all pixels and their surrounding patches compared with.

3.3 Image Processing

In this section most of the methods used have a main purpose to step-by-step take the image from the noise reduced raw type format that the sensor read and make it viewable in a way that the user expects it to be. These methods are not within the frame of this thesis but were incorporated to make the result more easily understood and viewable. The main workflow can be seen in 3.4. All of these steps were made by the help of a guide by Rob Sumner at the Department of Electrical Engineering, UC Santa Cruz, [18].

3.3.1 Linearizing

To ensure that the full dynamic range of the image is used a linearization can be done. This is also done to remove any non-linear transformation done by the camera for storage purposes. To linearize an image is to ensure that the maximum and minimum correspond to the limits allowed in the current format. For example, if the image is in the double range one will have a black level of 0 and a white level of 1. This is simple to do with only the values in the matrix that makes up the image. However a more accurate way of doing it is to take advantage of the metadata stored in the DNG file. A DNG file is information about how the image was taken and also contains information helpful in order to process

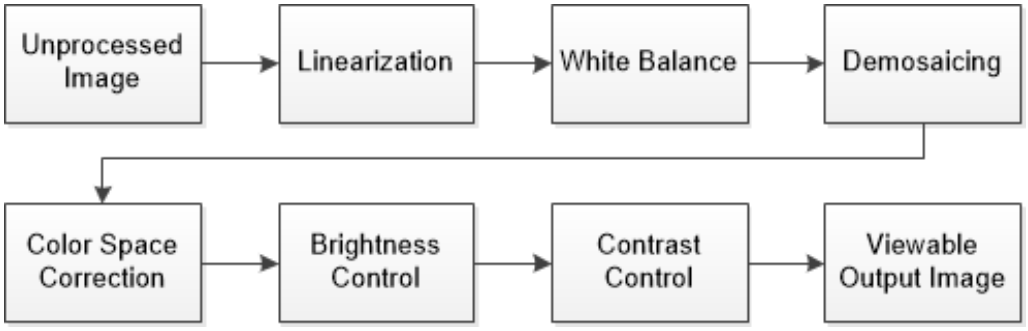

Figure 3.4: Step-by-step overview for the processing of an raw image.

the image. In some cases one can take advantage of a linearization table stored in the DNG file, if no such table is present one can instead base it as mentioned above on the stored white level and black level. In *MATLAB* this can look as follows:

```
if isfield(meta_info.SubIFDs{1},'LinearizationTable')
    ltab = meta_info.SubIFDs{1}.LinearizationTable;
    im = ltab(im+1);
end
black = meta_info.SubIFDs{1}.BlackLevel(1);
saturation = meta_info.SubIFDs{1}.WhiteLevel;
lin_im = (im-black)/(saturation-black);
lin_im = max(0,min(lin_im,1));
```

Here *im* is the read image file, meta_info is the metadata file read from the DNG data and *lin_im* is the linearized image data.

3.3.2 White Balance

To be able to white balance the image one might again take advantage of the metadata. Since only the ratio of the three colors matters in the final image, one can set one channel to have the multiplier 1. The *MATLAB* to perform white balance on the image can look as follows:

```
wb_multipliers = (meta_info.AsShotNeutral).^-1;
wb_multipliers = wb_multipliers/wb_multipliers(2);
wblin_im(:,:,1) = wb_multipliers(1).*lin_im(:,:,1);
wblin_im(:,:,2) = wb_multipliers(2).*lin_im(:,:,2);
wblin_im(:,:,3) = wb_multipliers(3).*lin_im(:,:,3);
wblin_im = wblin_im/max(wblin_im(:));
```

Here *AsShotNeutral* contains the inverse ratio of the three different color channels when the image was taken, and *lin_im* is the previously linearized image. The white balance multipliers are channel-wise applied to the image.

3.3.3 Color Space Conversion

After the current processing the image is in some sense already viewable and in RGB colors but not in a general RGB space. The colors are in the camera color space and

therefore needs to be corrected to a common RGB space to be viewed on a monitor. To get a more accurate and general color content in the image a color space conversion can be done. This can be done with two separate matrices, one that maps the image from current camera color basis to XYZ coordinates and another that maps from XYZ to the desired sRGB space. The multiplication of the matrices can of course be done first saving computation and resulting in an matrix with the following map.

$$A_{sRGB \leftarrow Cam} = (A_{Cam \leftarrow XYZ} A_{XYZ \leftarrow sRGB})^{-1}.$$

This can however result in that the colors are non-uniformly multiplied which can lead to a ruined white balance. To solve this issue one can make sure that matrix multiplication holds true

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}_{Cam} = \begin{bmatrix} A_{Cam \leftarrow sRGB} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}_{sRGB},$$

this can be done by normalizing the rows of $A_{Cam \leftarrow sRGB}$.

The matrix $A_{Cam \leftarrow XYZ}$ can be found from the metadata at meta_ info.ColorMatrix2. For the $A_{XYZ \leftarrow sRGB}$ there are many different options that might work depending on how the picture is going to be viewed, but the most common is

$$A_{XYZ \leftarrow sRGB} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix}.$$

This one is used in particular because it is adapted to a common viewing environment such as an computer monitor.

3.3.4 Brightness and Gamma Correction

The image is still a linear image with values relating to what the sensor read. This may not be in a range appropriate for being displayed, often it is to dark to be able to see anything. To make the image look better a non-linear transformation may be performed. This can be done by using a power function on the pixel values, a so called gamma correction. A common gamma correction is $p_{non-linear} = p_{linear}^{\gamma}$ where $\gamma = \frac{1}{2.2}$.

The dark areas in the image might still be too dark. This can be dealt with by using a brightening measure. As a brightening measure one can for example, increase the mean luminance of the image. A fairly arbitrary scale is to scale it so that the mean luminance is 1/4 the maximum. This will ensure that darker areas are more viewable but it will also effect the HDR construction.

3.4 High Dynamic Range Construction

Figure 3.5: HDR construction step-by-step overview

The last step the proposed algorithm is that of HDR image construction. In this case the steps can be seen in Figure 3.5. Working with the assumption of three image that are all noise reduced and has been processed according to the previous section the HDR steps can be performed. The reason that the images are processed before constructing the HDR image is that one has different white balance for the differently exposed images. That is, one can not apply one white balance but would have to apply it to the areas separately according to map produced by the $\alpha-$ expansion algorithm. There were also problems with the final image becoming gray and not at all as colorful as one would want. This mainly from the previously mentioned $\alpha-$ expansion but probably also from the image blending algorithm. It was thus deemed easier to just process the images then and then do the HDR construction after that.

3.4.1 Alpha Expansion

When it comes to choosing areas suitable to create an HDR image with there are a bundle of approaches that might be considered. The first one that comes to mind might be to simply set three labels in different gray colors and then assign each pixels to the closest labels. This approach suffer from one major drawback. At each place were one has a transition between the labels there will be pixels fluctuating between two labels. This comes from the fact that the boundaries are not exact and if the values are on the edge the labels will change on a pixel-to-pixel basis. These fluctuations can reduce the result for the image blending as the gradients will shift more frequently. This can be worked around by smoothing out these areas, on way to do so is to use the concepts of erosion and dilation. This will take care of fluctuation of the labels and create a more solid edge between the areas. One drawback with such a solution is that all areas are of equal importance and it will thus not take into consideration how big the difference in intensity is of neighboring areas. The chosen approach in this thesis is to pose the problem as a multi-label optimization problem and minimize it using the $\alpha-$expansion method. This approach copsed with the mentioned problem. The higher the intensity difference is between two areas the more energy it takes to convert it into a single one. To be able to perform the $\alpha-$expansion one needs to have a gray color map which indicates the starting weights for each pixel. To do this the mean of the RGB channels of the 0EV image was used.

The result of the $\alpha-$ expansion can be seen in Figure 3.6. Here the gray levels in the produced image (the right figure) corresponds to the image that should be selected when

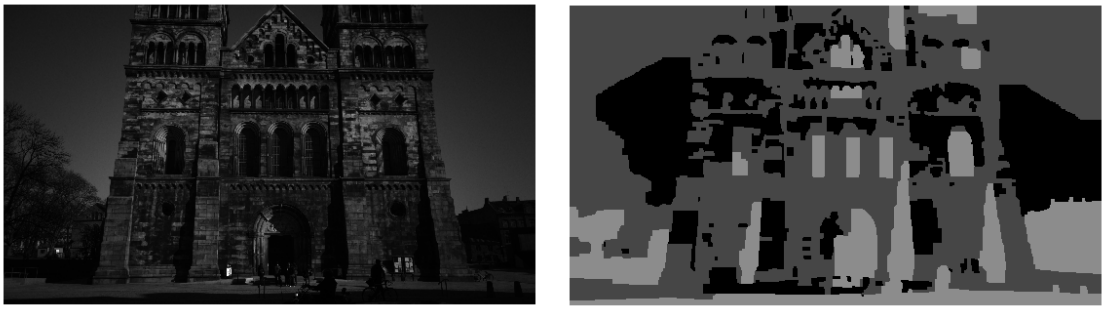

Figure 3.6: The input produced by taking the mean(R,G,B) of the image to the left, and the corresponding produced image by the α − expansion to the right.

constructing the HDR. The black color is the darkest image, i.e. the $-1EV$ image, the dark gray color corresponds to the $0EV$ image and the bright gray corresponds to the $+1EV$ image. The result is on the smooth side which will ensure a small amount of shifts for the gradients. Another major improvement with a smooth map is that some ghosting effects will be dealt with. This in the sense that areas that have many small structure differences will be selected from the same image. An example of this is grass or leaves which can have a brighter background that in some areas shines through. A large portion of these areas will be selected from the same area given a sufficiently high smoothness term.

Larger movements in the image can not be handled with only an α − expansion. This as the α − expansion will produce an image that has to be sufficiently smooth to handle any possible ghosting. To ensure no ghosting with large move across the sequence one would have to increase the smoothness term even more. This would lead to areas covering almost the entire image and would result in reduced dynamic range in the image. To solve this another approach, though not implemented, is to detect movement of larger type between the image and then select the $0EV$ image in these areas. Even though the produced image will suffer from lower dynamic range, having severe ghosting artifacts would be worse.

3.4.2 Image Blending

The created smooth gray color map with anti-ghosting as described in the previous section is now going to be used in order to create the final HDR image. The map indicates where each noise reduced image should fit in. To simply check the images pixel-wise choosing the corresponding image to the map will result in big edges and an image that is hard to correct. Instead the same image histogram transformation and image registration as before can be executed thus having the image in same color space and reference system. Now a pixel-wise selection based on the α − expansion might work but in order to get a seamless transition between the images the mentioned image gradient approach is used instead.

The gradient solution is done channel-wise on the three RGB channels for simplicity. A solution that is depending on the gradients in all the images keeping the ratio can also be implemented. A problem when it comes to the gradient solution is that integrated image will not be inside the range of the inputs image but can be well outside it. A simple linearization can however be done to solve this.

Chapter 4
Result and Evaluation

In this chapter the result of the algorithm will be evaluated. The chapter is divided into three steps, noise reduction, image processing and HDR construction which will be evaluated separately.

4.1 Noise Reduction

In this section the images were processed according to the guidelines in image processing section, section 3.3. This was done to better represent how the final result would look like.

A common problem when it comes to noise reduction is that fine structures are smoothed. This is especially true for the non-local means algorithm as it is based on working with weighted means. In Figure 4.1 one can see the comparison between four different images. The top image is the noisy input image, the top-middle image is the non-local means 3D algorithm (NLM3D) which uses three differently exposed images for denoising, the bottom-middle one is the non-local means 1D (NLM1D) which uses only the image itself for denoising and the bottom image is the BM3D algorithm which was extended so that it searches over all three images to find matches. Worth noting is that BM3D algorithm took the mean of the noise as input but the other two algorithms work with intensity depending noise reduction. One can see that the structure are kept in all three images, the BM3D algorithm does however create some artifacts which makes straight lines appear in the mortar and the bricks in the wall, these lines make the image look unnatural but can be caused by an over smoothing in this area as it might have a lower noise level compared to the mean of the image.

An easier noise reduction case is that of larger areas with the same color. In Figure 4.2 one can see a part of the sky as well as the wall of the cathedral in Lund. Again the two non-local means algorithm performs well and not much noise can be noticed. In this case

the BM3D leaves noise after the process which indicates that this part contains more noise than the mean of the image does. This is as expected from the dynamic noise estimation which for higher intensities expects more noise than for lower.

The overall result is the same for the other cases of noise reduction. The most important part when doing noise reduction is to use a intensity based noise model to prevent over-smoothing in darker areas and a lack of noise reduction in bright ones.

4.1.1 High noise level reduction

If one instead looks upon the case were an unusual amount of noise is present one can see that the NLM3D algorithm will keep noise as it is interpret as pixel structures. Where as the NLM1D will smooth these out, an example of this can be seen in Figure 4.3. In this case the BM3D leaves artifacts in the image and some noise from the presumed faulty noise level. There is however a problem with using dynamic noise level in the NLM1D case. This can be seen in Figure 4.4 as a faulty color in the result image. This is not only true for this part of the image but the color tone of the image changes overall. The BM3D preforms well with keeping the text structure despite the different noise level.

4.2 Image Processing

In this section a step-by-step result will be shown for the process that the image goes through after the noise reduction step. Starting by showing an unprocessed image from the sensor top left corner of the Figure 4.5. In the top right corner the produced image after and the image has been linearized. The small difference between the images indicates that the image was already more or less linear. The second step is white balance, which can be seen to the middle left in the figure. Here the green color artifact has been taken away and the image as a whole look more natural. the image is now correct in the camera space colors but to be able to view it with an external monitor without loss of colors another step is needed namely the color conversion. This step can be seen in the middle right of the figure. Color Conversion takes the image from camera space to the "sRGB" space which is a common color space for monitors. The image is now processed and can be viewed with correct white balance and in the right color space. However one can do two additional process to enhance the image and brighten it, brightness correction to the bottom left and gamma correction to the bottom right. These two steps can be left out and a tonemapping can be performed instead to brighten the image. It was however beneficial for the HDR construction to use gamma corrected images. The brightness correction did not effect the result as a intensity transformation is already performed in the HDR construction steps.

Figure 4.1: Noise reduction comparison for a high noise image. Top is the noisy image, upper middle is the NLM3D algorithm, Bottom middle is the NLM1D algorithm and at the bottom is the BM3D with a constant noise level.

Figure 4.2: Noise reduction comparison for a high noise image. Top is the noisy image, upper middle is the NLM3D algorithm, Bottom middle is the NLM1D algorithm and at the bottom is the BM3D with a constant noise level.

Figure 4.3: Noise reduction comparison for a high noise image. Top is the noisy image, upper middle is the NLM3D algorithm, Bottom middle is the NLM1D algorithm and at the bottom is the BM3D with a constant noise level.

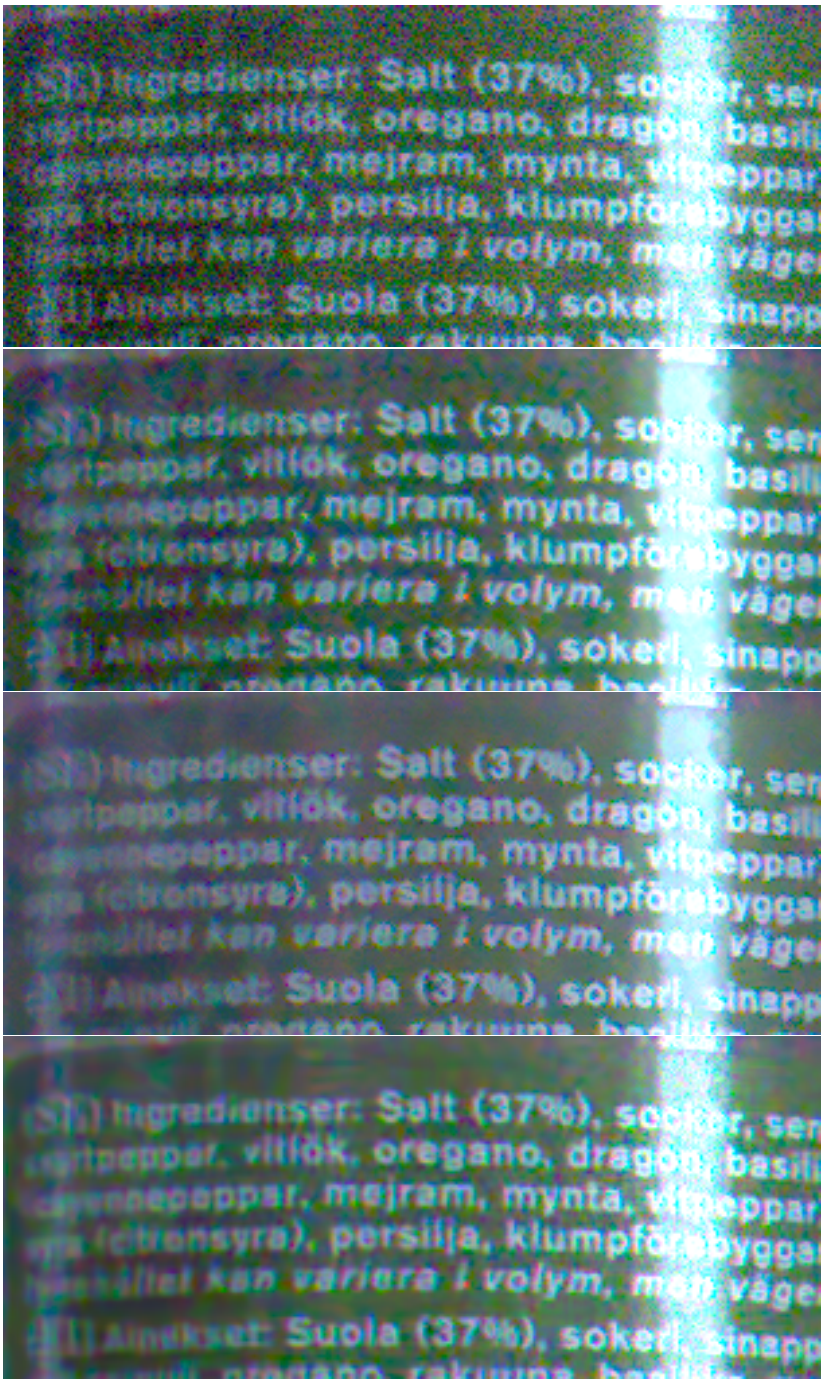

Figure 4.4: Noise reduction comparison for a high noise image. Top is the noisy image, upper middle is the NLM3D algorithm, Bottom middle is the NLM1D algorithm and at the bottom is the BM3D with a constant noise level

Figure 4.5: Images showing the workflow according to the image processing seen in 3.4, Top right shows the raw image from the sensor, the only processing done is that of noise reduction. Top right shows the Linearized data, i.e the darkest value is 0 and the brightest one is 1. Middle left image shows the image after white balance on each respective channel has been performed. Middle right image shows the image after the color has been corrected to the "right" space, i.e more suitable to be viewed from a computer screen. Bottom left image shows the image after brightness correction has been made. Bottom right shows the final image after another correction has been made in form of gamma correction.

4.3 High Dynamic Range Construction

The result from the HDR image construction algorithm can be seen in the Figures 4.6 - 4.10. Here the 3 differently exposed images, the $\alpha - expansion$ map and the resulting HDR image can be seen.

In Figure 4.8 one can see the larger ghosting effects from the time difference between the images. This can be dealt with but as it is outside the scope of this thesis these areas were edited manually in the $\alpha - expansion$ map. The result can be seen in Figure 4.11.

In Figure 4.11, two different kind of ghosting artifacts are shown. The image at the top left showcase a person whom the $\alpha -$ expansion map did not capture entirely. This results in areas which are cut off and only half the person is shown. This can be avoided by either selecting the entire person to be in the final image or select so that he is not in the image at all. The former was implemented by editing the $\alpha -$ expansion map manually, the result can be seen in the top right image.

The second type of ghosting artifact can be seen in the middle left image. Here the person has moved not only between the images but also between the mapped areas in the images resulting in a cloning artifact. To avoid this one can change the map for one of the areas and thus removing the clone. Once again the $\alpha -$ expansion map was edited manually before the HDR construction.

Smaller artifacts are partly taken care of by the $\alpha -$ expansion, but for smaller movements which cover larger portions of the image one need to select the area from the same image to avoid ghosting. An example of this are the branches in the top right of the bottom image. This artifact can be harder to deal with as one has a dark object which one would like to choose from the +1EV image and a bright object in the background which one would like to choose from the -1EV image. In top image of Figure 4.12 one can see that the branch is smoothed out by the $\alpha -$ expansion that instead selects the +1EV for the building and 0EV for the sky in the background. This does however not help as there are two different images selected in the area which the branch extends. To avoid this ghosting altogether this entire area is selected from the 0EV image. The resulting image can be seen as the bottom image in Figure 4.11. The editing of the $\alpha -$ expansion map can be seen in 4.12

Figure 4.6: The image showcases the selection of the three noise reduced and processed images to make the HDR image. The Top left shows the processed $0EV$ image. Top right shows the processed $-1EV$ image. Middle left shows the processed $+1EV$ image. Middle right shows the selection corresponding to the $\alpha - expansion$ map, see 3.4.1. The bottom image shows the final HDR image result.

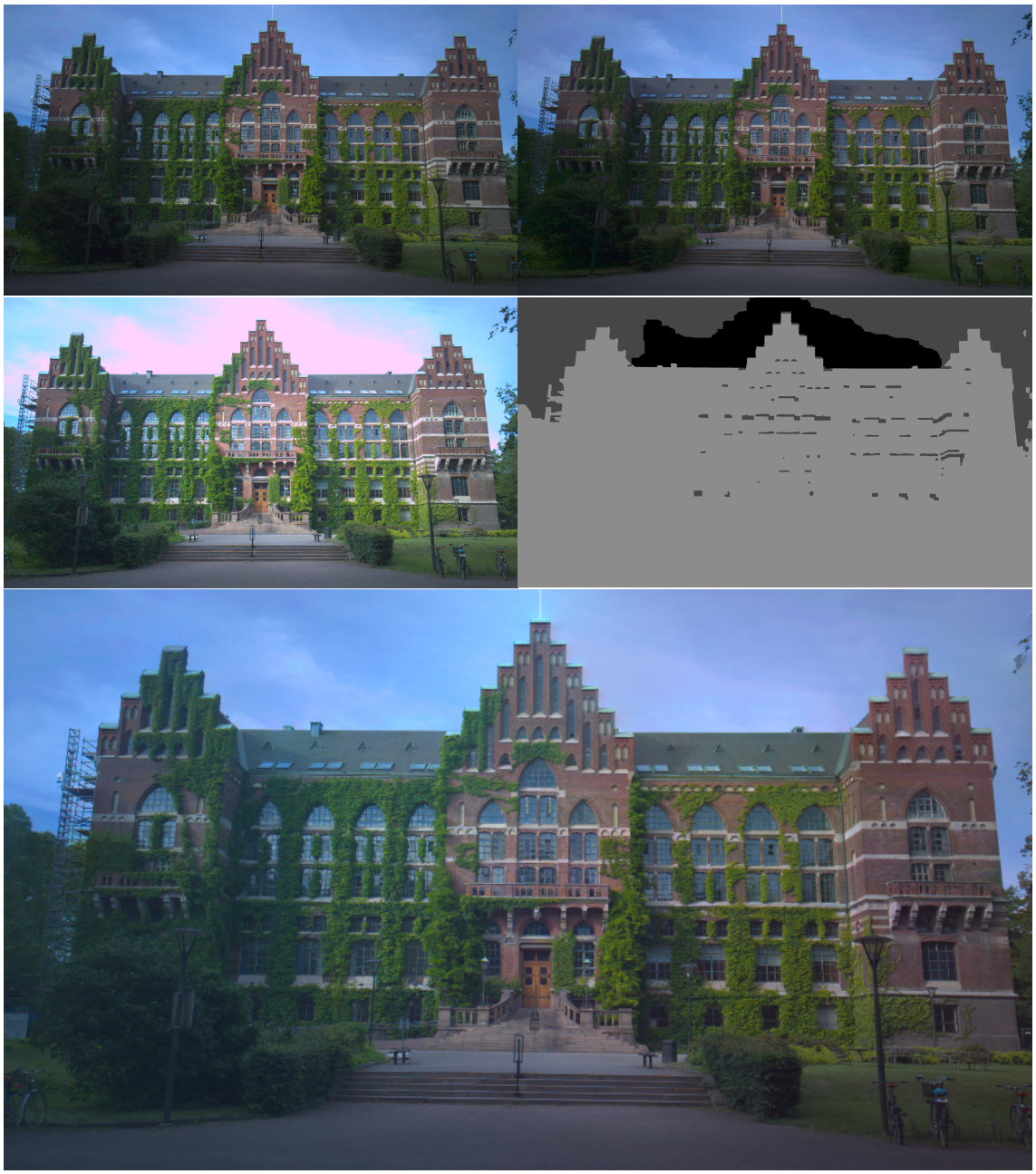

Figure 4.7: The image showcases the selection of the three noise reduced and processed images to make the HDR image. The Top left shows the processed $0EV$ image. Top right shows the processed $-1EV$ image. Middle left shows the processed $+1EV$ image. Middle right shows the selection corresponding to the $\alpha-expansion$ map. The bottom image shows the final HDR image result.

Figure 4.8: The image showcases the selection of the three noise reduced and processed images to make the HDR image. The Top left shows the processed $0EV$ image. Top right shows the processed $-1EV$ image. Middle left shows the processed $+1EV$ image. Middle right shows the selection corresponding to the $\alpha-expansion$ map. The bottom image shows the final HDR image result.

Figure 4.9: The image showcases the selection of the three noise reduced and processed images to make the HDR image. The Top left shows the processed $0EV$ image. Top right shows the processed $-1EV$ image. Middle left shows the processed $+1EV$ image. Middle right shows the selection corresponding to the $\alpha-expansion$ map. The bottom image shows the final HDR image result.

Figure 4.10: The image showcases the selection of the three noise reduced and processed images to make the HDR image. The Top left shows the processed $0EV$ image. Top right shows the processed $-1EV$ image. Middle left shows the processed $+1EV$ image. Middle right shows the selection corresponding to the $\alpha-expansion$ map. The bottom image shows the final HDR image result.

Figure 4.11: The images showcase some larger ghosting effects and how they look after they have been manually corrected. The top left and top right images shows how a person has been severely cut from the image as the α − expansion cut into this area. In the pair of images in the middle a "cloning" type of artifact can be seen and the result of removing it. The bottom picture shows the result after the anti-ghosting has been done.

Figure 4.12: Image showing a map which introduces ghosting (top) and an image showing the manually edited map to prevent these artifacts (red circles) in bottom picture.

Chapter 5
Conclusions

When it comes to noise reduction the use of mutual information for differently exposed images is hard. Not only are the areas for which the mutual information can be used limited but it also requires a good intensity transform to be able to benefit from it. The intensity map used in this thesis is a linear one thus errors can be introduced as it might not be a linear intensity difference between the images, an example of this is areas which are over- or under-exposed. Another problem with applying a transformation is that it is impossible to map a single intensity into multiple ones. This from the fact that one does not know what pixel should get what value. This results in a faulty map with a risk that the map effects the noise. An approach which reduces this risk is to transform the images to a common intensity level by using information in the DNG file such as the exposure time. This combined with a dynamic noise estimation on the mapped data would most likely result in a more consistent noise reduction.

A sufficient approach when it comes to the case of HDR image construction is to noise reduce the part of each image that is going to be used according to the $\alpha - expansion$ map. This both speeds up the algorithm by using a faster noise reduction method but also only reduce the noise in used areas. As the least noise is found in the +1EV image so to use that to a larger extent would probably be beneficial. The most important part when it comes to noise reduction however is to have an intensity based noise model that can adept and not over-smooth or leave noise unprocessed.

When it comes to HDR construction the method performs well. It is beneficial as it reduces smaller ghosting artifacts by using the $\alpha - expansion$. For larger movements across the image sequence it performs poorly but in order to handle these types of movements more advanced methods are needed. One drawback is that the method is dependent on a good registration or errors will be introduced where the images change. The algorithm can be made to work completely automatically which makes it beneficial for any implementation purposes.

To use HDR construction when it comes to dynamic scenes or with a handheld device which moves between the images should be avoided if possible as the result will suffer. To instead have a static scene and stationary camera will create a better result.

Chapter 6
Future Work

As this thesis covers multiple areas there are many things that can be improved and developed, both when it comes to the mutual information in the HDR image sequence and also for the image gradient solution for the HDR construction. As mentioned throughout the report each step selected is done in none optimized code. The steps were often chosen to get the work done prior to selecting an optimal solution. Hence most of the implementations are based around *MATLAB* implementations. This provides a good base with general methods that work well, but for the purposes used in this thesis other more problem specific methods should be investigated.

By starting instead with the $\alpha - expansion$ calculation one could reduce the computation time of the noise reduction by only reducing the areas which would be used in the final HDR construction. To take care of ghosting artifacts multiple approaches would be interesting to investigate. One discussed during the work of the thesis was to make modification to the $\alpha - expansion$ algorithm so that the weight take care of some ghosting. Another approach would be to use a movement detection between the images an select areas were movement is detected from the 0EV image or even select the image which alters the $\alpha - expansion$ the least to preserve as much dynamic range as possible.

One ideas which was not implemented due to time duration of the thesis was that of a BM3D algorithm with a similar workflow used in this report. Implementing a BM3D algorithm from scratch and alter it to fit the needs in this report was simply not feasible.

Another future work that would be of interest is to noise reduce with more images and compare the result. How does it overall compare to multiple images taken of the same scene and same exposure. Perhaps is another noise reduction algorithm more appropriate when working with differently exposed images compared to the single image case.

Doing a tonemapping in the gradient domain on the merged gradient would be benefi-

cial as one might still do this on the final image. This would save computational time as a derivation and integration is removed. It would also be interesting to see how a tonemapping algorithm can take advantage of the $\alpha - expansion$ map to perform an even better tonemapping on the image.

Bibliography

[1] B. Coll A. Buades and J. Morel. “a non-local algorithm for image denoising”. *Proceedings of 2005 IEEE Computer Society Conferance on Computer Vision and Pattern Recognition*, vol. 2:pp. 60–65, 2005.

[2] J.E. Adams and J.F Hamilton. “ design of practical color filter array interpolation algorithms for digital cameras ”. *Proceedings of SPIE 3028*, pages 117—125, 1997.

[3] Y. Gousseau C. Aguerrebere, J. Delon and P. Mus. “best algorithms for hdr image generation. a study of performance bounds”. *SIAM Journal on Imaging Sciences.*, 7:1–34, 2010.

[4] Y. Gousseau P. Muse C. Aguerrebere, J. Delon. “simultaneous hdr image reconstruction and denoising for dynamic scenes”. `http://perso.telecom-paristech.fr/~gousseau/hdr_denoising`.

[5] Y. Gousseau P. Muse C. Aguerrebere, J. Delon. “simultaneous hdr image reconstruction and denoising for dynamic scenes”. *IEEE International Conference on Computational Photography (ICCP)*.

[6] P. Cattin. Image restoration: Introduction to signal and image processing. `http://miac.unibas.ch/SIP/06-Restoration.html`.

[7] M. Trimeche V. Katkovnik Foi, A. and K. Egiazarian. “practical poissonian-gaussian noise modeling and fitting for single-image raw-data”. *IEEE Trans. Image Process*, vol. 17.

[8] M. Trimeche V. Katkovnik Foi, A. and K. Egiazarian. “signal-dependent noise modeling, estimation, and removal for digital imaging sensors”. `http://www.cs.tut.fi/~foi/sensornoise.html`.

[9] Glenn E. Healey and Raghava Kondepudy. “radiometric ccd camera calibration and noise estimation”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

[10] Tinne Tuytelaars Luc Van Gool Herbert Bay, Andreas Ess. "surf: Speeded up robust features". *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[11] V. Katkovnik K. Dabov, A. Foi and K. Egiazarian. "image denoising by sparse 3d transform-domain collaborative filtering". *IEEE Trans. Image Process*, vol. 16:pp. 2080–2095, August 2007.

[12] David Lowe. "distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] M. Wand C. Theobalt M. Granados, B. Ajdin and H. Seidel. Optimal hdr reconstruction with linear digital cameras. In *CVPR*, pages 215–222, 2010.

[14] K. Egiazarian M. Maggioni, V. Katkovnik and A. Foi. "a nonlocal transform-domain filter for volumetric data denoising and reconstruction". *IEEE Trans. Image Process*, vol. 22.

[15] K. Egiazarian M. Maggioni, V. Katkovnik and A. Foi. "video denoising, deblocking and enhancement through separable 4-d nonlocal spatiotemporal transforms". *IEEE Trans. Image Process*, vol. 21(no. 9).

[16] D. Lischinski R. Fattal and M. Werman. "gradient domain high dynamic range compression". *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 21:249–256, 2002.

[17] Steven K. Shevell. *"The Science of Color"*. Elsevier, 2003.

[18] Rob Sumner. Processing raw images in matlab. `http://users.soe.ucsc.edu/~rcsumner/rawguide/RAWguide.pdf`.

[19] R. Szeliski. *"Computer Vision: Algorithms and Applications"*. Springer, 2010.

[20] Robert A Ulichney. "halftone characterization in the frequency domain". *IS&T's 47th Annual Conference/ICPS*, 1994. `http://home.comcast.net/~ulichney/CV/papers/1994-freq-characterization.pdf`.

[21] wikipedia.org. Bayer pattern. `http://en.wikipedia.org/wiki/Bayer_pattern`.

[22] wikipedia.org. Parallax. `http://en.wikipedia.org/wiki/Parallax`.

[23] O.Veksler Y. Boykov and R.Zabih. "fast approximate energy minimization via graph cuts". *IEEE Trans. on pattern analysis and machine intelligence*, vol. 23(no. 11).

[24] Jianping Zhou. Image pipeline: Fine-tuning digital camera processing blocks. `http://www.eetimes.com/document.asp?doc_id=1273744`.

Master's Theses in Mathematical Sciences 2014:E50
ISSN 1404-6342

LUTFMA-3266-2014

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/