Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Studies on Mechatronics

Camera Auto Exposure Control for VSLAM Applications

Autumn Term 2010

Supervised by:
Christoph Huerzeler
Marcel Tresanchez
Janosch Nikolic

Author:
Michael Muehlebach

Acknowledgement

I would like to thank Christoph Huerzeler, Marcel Tresanchez and Janosch Nikolic for the interesting discussions we had and for their help and useful inputs throughout this study.

Contents

Abstract

This report gives an introduction and overview of the state-of-art of auto exposure algorithms and image sensors, as they are closely related to auto exposure. Furthermore it points out the advantages and disadvantages of the presented algorithms and shows their application. And last but not least the algorithms are evaluated in connection with an industrial VSLAM application and a method to incorporate active illumination is proposed.
Keywords: auto exposure, active illumination, visual odometry
...

Symbols

Acronyms and Abbreviations

AE	auto exposure
CDS	correlated double sampling
CCD	charge coupled device
CMOS	complementary metal oxide semiconductor
EV	exposure value
$\mathcal{F}$, FT	Fourier transform
FPN	fixed pattern noise
HDR	high dynamic range
UAV	unmanned aerial vehicle
SNR	signal to noise ratio
SVD	singular value decomposition
VSLAM	visual simultaneous localization and mapping

Chapter 1

Introduction

1.1 Motivation

In recent years the whole world has been digitalized. Digitial cameras evolved and revolutionized the photo-industry with a lot of new possibilities. As a consequence a need for auto exposure algorithms emerged.
Although nowadays, almost all cameras have an auto exposure control integrated, for demanding applications, such as visual odometry, the provided algorithm is not sufficient and an external auto exposure control has to be implemented.
With the project AIRobots, an aerial service robot for industrial applications is designed. For localization of the UAV, two cameras are used. Due to difficult light conditions and to ensure that both cameras have the same exposure setting, an external auto exposure control has to be designed.

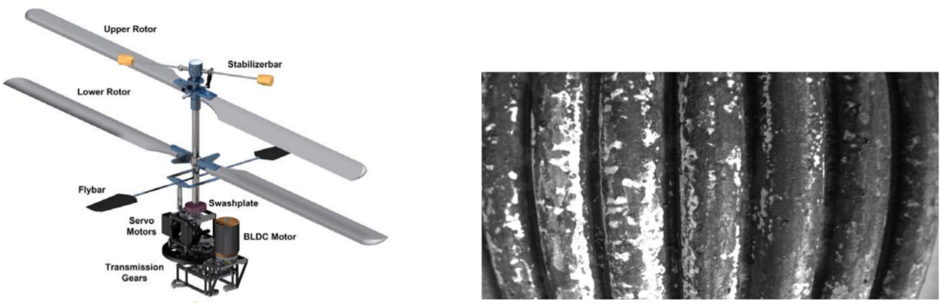

Figure 1.1: AIRobots project: Left a prototype of the UAV, right a typical environment.

1.2 Auto exposure

The brightness intensity of a scene may vary between .001 $\frac{cd}{m^2}$ and 100'000 $\frac{cd}{m^2}$ and results in a dynamic range of 160 dB. However, the dynamic range (for a definition see Appendix A) of a typical digital camera is approximately 55 dB. This makes auto exposure a quite demanding task. On one side the image signal might sink below the noise level, resulting in underexposure and on the other side very bright parts of the scene may be clipped. Therefore auto exposure can be seen as shifting the dynamic range of the camera over the one of the scene in order capture as much useful information as possible. What useful information is, may vary a lot between applications and as a result there exists no overall optimum auto exposure algorithm.

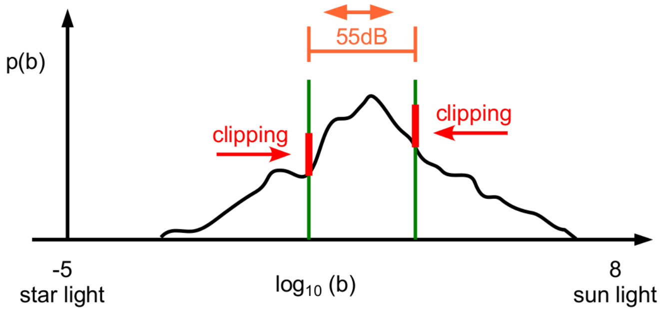

Figure 1.2: Range histogram p(b) of the scene, [1].

Auto exposure is in most cases also closely related to the type of image sensor. For example with a simple mean-value based algorithm, the whole functionality of a sophisticated multi slope CMOS sensor cannot be exploited. As a consequence, depending on the application, either the sensor or the algorithm has to be adapted. Furthermore, there are algorithms, which work only with a certain type of sensor. In Appendix A, one can find more information about image sensors.

To perform auto exposure, there are usually three parameters, which can be modified: AGC, electrical shutter and the f-number, i.e. the iris. However by increasing the magnitude of the signal, it' s noise is increased as well. As a consequence AGC is rarely used for the purpose of auto exposure.
The exposure value EV is defined by equation 1.1, according to [2].

$$EV = \log_2(\frac{F^2}{T}) \tag{1.1}$$

Where F is the f-number and T the exposure time. Since the exposure value is much more sensitive to a change of the f-number, most algorithms use the exposure time T to control auto exposure, [3]. Furthermore, not all cameras provide the possibility to change the f-number.

Chapter 2

Basic auto exposure algorithms

In this chapter the simplest approaches for auto exposure are explained and discussed. This simplicity is not necessarily a disadvantage. They can easily be combined with other, more sophisticated ideas, which leads to robust and fast auto exposure algorithms.

2.1 Mean value algorithms

These algorithms try to maintain the mean pixel-brightness of an image to a certain brightness value. The target mean value is often mid tone, (i.e. 128 for 8 bits). According to [17], [18] the brightness level of an image can be expressed by:

$$Bl = kLGTF^{-2} \tag{2.1}$$

where:

Bl	Mean brightness level of image
k	Constant
L	Mean luminance of the scene
G	Gain of the AGC-circuit
T	Exposure time
F	F-number

However, equation 2.1 assumes a linear dependency between the mean value brightness of the image and the one of the scene. For a non linear image sensor, this is certainly not true. But even for a linear image sensor, this is not the case for most scenes: Since the brightness range of the camera is usually smaller than the one of the scene, the image sensor saturates and the low and high brightness levels are clipped away.

2.1.1 Non iterative algorithm for linear image sensor

If we assume, that k, G and F where known, one could solve equation 2.1 once for L and once for T. As a result, one could first capture an image with a known exposure Time T, then calculate the luminance of the scene L and finally calculate the exposure Time T_{n+1} for the next frame in order to get a picture with the desired mean value brightness (it is assumed, that the luminance of the scene does not change between two frames). This is exactly the approach proposed in [17].
To speed up the algorithm, look up tables are prepared in advance. From the values of Bl, G, T and F the appropriate value for T_{n+1} can immediately be looked up.

Unfortunately the relation of 2.1 is just an approximation, as for example k varies with the brightness value. Furthermore the image sensor cannot capture the whole brightness range of the scene, which leads to clipping and therefore L cannot be determined exactly.

The strengths and weaknesses of [17] are summarized next:
Advantages:

- fast
- simple
- can react fast to sudden change of lighting conditions

Disadvantages:

- assumes a linear response curve of the image sensor, which is even for a linear image sensor only approximately the case
- look up table needs to be adjusted for every image sensor (since every image sensor has for example a slightly different k)
- neglects clipping distortions and cannot handle scenes with a high brightness range
- only adjusts mean value brightness of the image, for special lighting conditions (for example excessive front lighting, back lighting) this results in under- or overexposed image regions.

2.1.2 Iterative algorithm for linear image sensor

In order to improve the previous approach [18] finds an iterative process:

Taking the logarithm of equation 2.1 leads to

$$\log_2(Bl) = \log_2(k) + \log_2(L) + \log_2(G) - \log_2(\frac{F^2}{T}) \tag{2.2}$$

$$\log_2(Bl) = \log_2(k) + \log_2(L) + \log_2(G) - EV \tag{2.3}$$

with EV according to equation 1.1 (Introduction).

Equation 2.3 can be rewritten as an iterative process with:

$$\log_2(Bl_n) = \log_2(k) + \log_2(L) + \log_2(G) - EV_n \tag{2.4}$$

and

$$\log_2(Bl_{n+1}) = \log_2(k) + \log_2(L) + \log_2(G) - EV_{n+1} \tag{2.5}$$

where it is assumed that k, L and G do not change between to frames. which leads finally to:

$$EV_{n+1} = EV_n + \log_2(Bl_n) - log_2(Bl_{desired}) \tag{2.6}$$

In most cases $Bl_{desired}$ is taken as mid tone, i.e. 128 for 8 bit.
Again, in real implementation the logarithmic term of the mean brightness level is stored in a look up table to speed the algorithm up.

This leads to the following evaluation:

Advantages:

- fast
- simple
- can react fast to sudden change of lighting conditions
- although best results will be achieved with a linear image sensor, this algorithm works with every type of image sensor

Disadvantages:

- assumes a linear response curve of the image sensor, which is even for a linear image sensor only approximately the case
- neglects clipping distortions and cannot handle scenes with a high brightness range
- only adjusts mean value brightness of the image, for special lighting conditions (for example excessive front lighting, back lighting) this results in under- or overexposed image regions.

2.1.3 Algorithm for non linear image sensor

[3] proposes an mean-value based algorithm for non linear image sensors. It is based on the bisection method (numerical analysis).

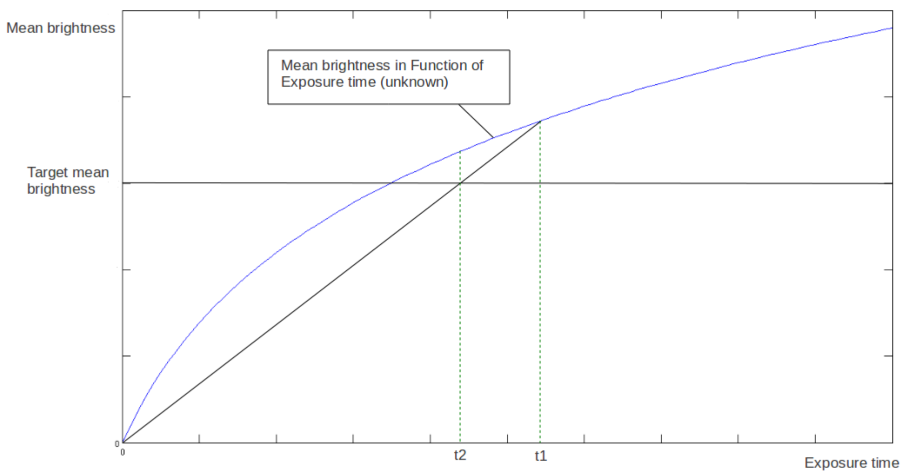

Figure 2.1: Illustration of the bisection method. Note that the function of exposure time to mean value image brightness (blue curve) is unknown.

Figure 2.1 illustrates the idea: First an image with an arbitrary exposure time t_1 is taken and its mean brightness b_1 is calculated. Secondly it is clear that zero exposure time leads to a completely dark image. This information is used to calculate the next exposure time, by the simple relation:

$$t_2 = \frac{b_{target}}{b_1} \tag{2.7}$$

where b_{target} is the target mean brightness value of the image. These two steps are continuously repeated.
This method has the following advantages and disadvantages:

Advantages:

- fast
- simple
- works with any image sensor

Disadvantages:

- only adjusts mean value brightness of the image, for special lighting conditions (for example excessive front lighting, back lighting) this results in under- or overexposed image regions.

2.2 Weighted mean value algorithms

2.2.1 Fixed weighting areas

The simplest extension is found in [19]. The idea is to divide the image into several different areas (see Figure 2.2). It is assumed that the main object lies in the regions 1 and 4. If the main object back lighted, a normal mean value based algorithm cannot perform a satisfactory exposure. In that case the algorithm weights the brightness values of the pixels which lie in region 1 and 4 more than the others. This results in a better exposure of regions 1 and 4.

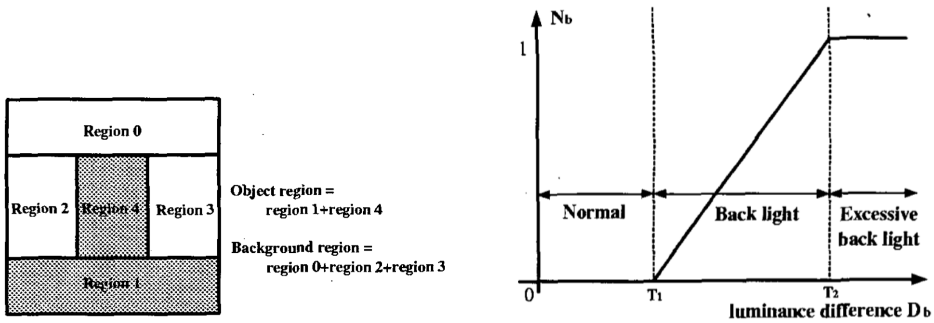

Figure 2.2: Different image regions ($R_0, ..., R_4$), and function of D_b. The thresholds T_1 and T_2 are set based on empirical observations, [19].

Back lighting is detected by the coefficient D_b:

$$D_b = (R_0 + max(R_2, R_3)) - (R_1 + R_4) \tag{2.8}$$

where R_i is the mean brightness in region i, see figure 2.2.
The amount of back lighting is given by N_b. Therefore region 1 and 4 are weighted more, according to N_b than the other regions.
However, it is important to notice, that this approach works only if the main object is in region 1 and 4. Obviously, this is in general not the case.
In [20] a similar algorithm is proposed. The image is divided into slightly different sections. In contrast to [19], logical laws are formulated between the different areas in order to detect extreme light conditions and implemented as fuzzy logic. This

might speed up the algorithm a bit, but robustness is lost. With fuzzy logic the algorithm can only handle a finite number of different situations, recognized by the developer of the auto exposure system.

Finally both methods are evaluated:

Advantages:

- fast
- simple
- works with any image sensor

Disadvantages:

- non flexible, main object is assumed to lie in the center!

2.2.2 Flexible weighting areas

[18] proposes an algorithm with flexible weighting areas. For normal lighting conditions it performs an iterative mean value based exposure adjusting (see above). Under front lighting or back lighting there is usually a high contrast between the main object (if there is one) and the background. As a result, if the algorithm detects high contrast differences within the image, the first derivative of the brightness of each pixel is analysed in order to detect the main object. Once the main object is determined, the algorithm proceeds in the same way as [19], i.e. weighting the main object region more than the background, in case of front/back lighting conditions.
This advanced algorithm leads to the following evaluation:

Advantages:

- can handle difficult light conditions
- works with any image sensor
- flexible, can identify a main object

Disadvantages:

- more complex, main object identification leads to a slower algorithm
- still not implemented in real time
- there has to be a main object. However difficult light conditions can also arise without main object, just by a scene with a high brightness range.

2.3 Hist - Parameter

[33] introduces a new parameter (HIST) to determine the amount of compensation for excessive front or back light scenes. The HIST parameter, as a function of the brightness b is defined as:

$$HIST(b) = \frac{number\ of\ pixels\ that\ have\ a\ brightness\ higher\ than\ b}{total\ number\ of\ pixels} \tag{2.9}$$

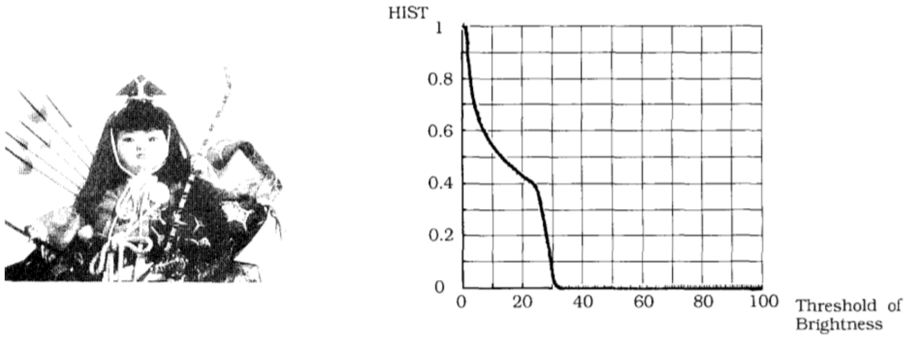

Figure 2.3: A front lighted image with the correspondent HIST - curve, [33]

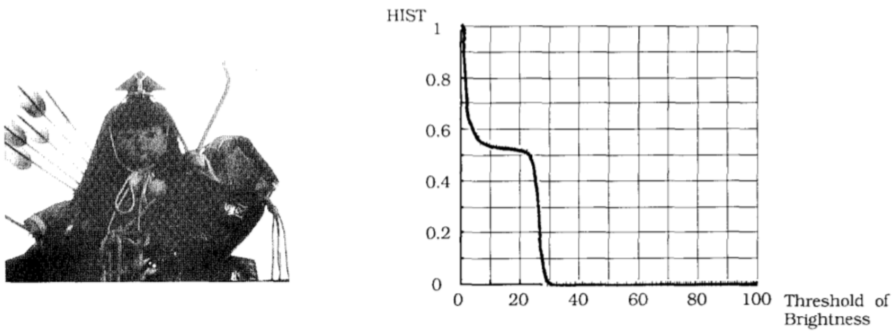

Figure 2.4: A back lighted image with the correspondent HIST - curve, [33]

By analysing the HIST-parameter of different images (see figure 2.3 and 2.4), one can find out, that the height of the flat part from the x - axis represents the ratio of dark to bright areas, where as the slope of the flat part represents the contrast between the bright and dark region (the higher, the less contrast).

In contrast, a well exposed image is shown in figure 2.5. To perform auto exposure, the following parameters are introduced:

H_{mean}	Hist value of the mean image brightness
H_{half}	Hist value of half the mean image brightness
H_{twice}	Hist value of twice the mean image brightness
H_{diff}	$:= \min(H_{twice} - H_{mean}, H_{mean} - H_{half})$

Note that H_{mean} represents the ration of dark area to bright area and H_{diff} represents the contrast between the object and the background.
The authors collected a lot of data from various shooting conditions and found a correlation between shooting conditions and H_{mean} and H_{diff}. Therefore a set of fuzzy logic rules where derived in order to create a fast and robust algorithm.
Although there exists more sophisticated ideas, the HIST-Parameter is definitively convincing.

This leads to the following advantages and disadvantages:

Advantages:

- simple, fast, flexible
- can handle difficult light conditions
- works with any image sensor

Disadvantages:

- focuses on whole image, no weighting of regions with higher interest possible

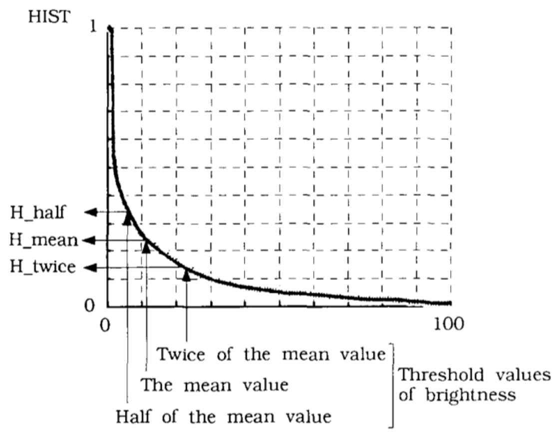

Figure 2.5: A well exposed image with the correspondent HIST - plot, [33].

Chapter 3

Algorithms using luminance histograms

3.1 Luminance and brightness histograms

Luminance and brightness histograms are synonyms. To be very strict they should be called luminosity histograms, because luminance is defined as a photometric measure of the luminous intensity per unit area of light travelling in a given direction and has the SI unit of candela per square meter ($\frac{cd}{m^2}$) [22]. In contrast, luminosity means perceived brightness, [23] and is calculated by:
$Rec.601 : L = 0.299R + 0.587G + 0.114B$
Furthermore there exist also histograms for each color channel. By adding the histograms for each color channel, one will get the RGB - histogram. On the other side, color pixels in luminance histograms are first converted to luminosity and then added to a histogram. For the purpose of auto exposure luminance histograms are used, since every pixel is considered only once.

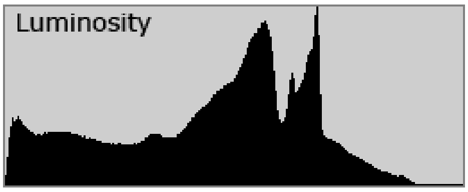

Figure 3.1: Picture with corresponding histogram, [23].

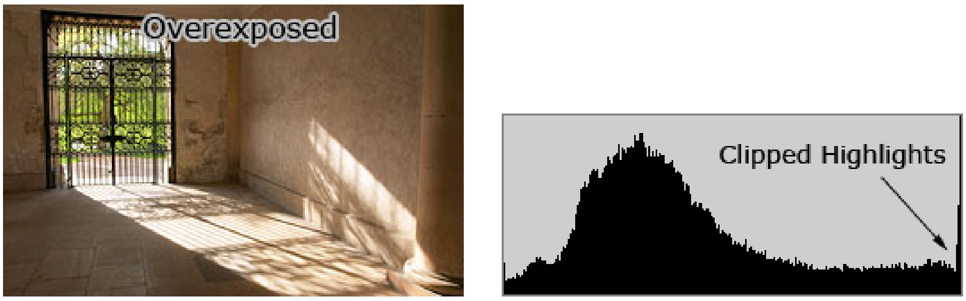

Figure 3.2: Overexposed picture with corresponding histogram, [23].

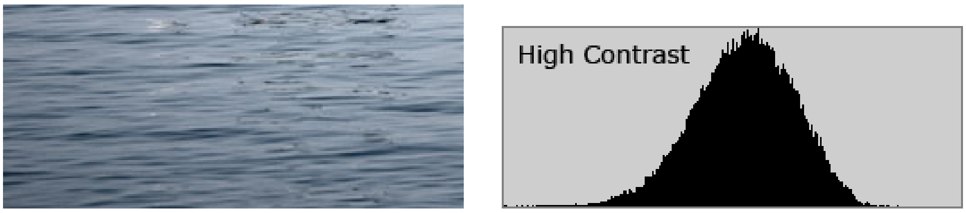

Figure 3.3: High contrast picture with corresponding histogram, [23].

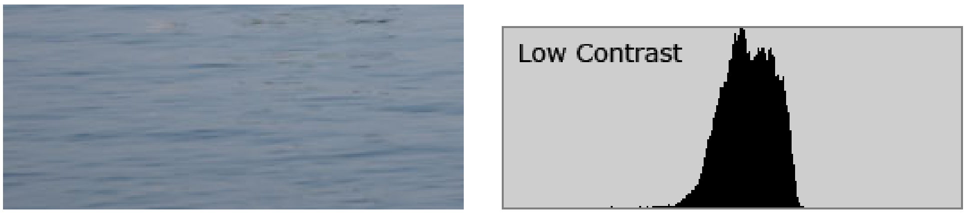

Figure 3.4: Low contrast picture with corresponding histogram, [23].

Figure 3.1 shows a scene with the corresponding luminance histogram. Although the scene has a high brightness range, no clipping occurred and the picture is well exposed.
Figure 3.2 shows a overexposed image. The corresponding luminance histogram shows that clipping occurred, at the high and at the lower end. The bar representing the maximal captured brightness is higher than the one representing the minimal captured brightness and therefore the picture is overexposed. Luminance histograms providing only one tiny, but high peak belong to images with low contrast and low information, where broad peaks belong to images with high contrast and high information. Figure 3.3 and figure 3.4 are illustrating this phenomena.
These figures should also illustrate the fact, that luminance histograms are a very powerful mean to determine, how well an image is exposed.

3.2 Peak analysis method

[24] presents a method, which tries to divide an image into regions of interest and regions of non interest. It is claimed that peak regions in a luminance histogram usually contain less gray-levels and the image entropy (see Appendix B) in those regions is small. They correspond to image regions with the same luminance value, have low complexity, contain less lines and edges and have usually low frequency components. They are defined as the region of no interest. In contrast, broad, flat histogram regions may contain more gray-levels, have more complexity and a high image entropy. They are defined as the region of interest.

Figure 3.5: Underexposed image, its corresponding histogram is found in figure 3.6, [24].

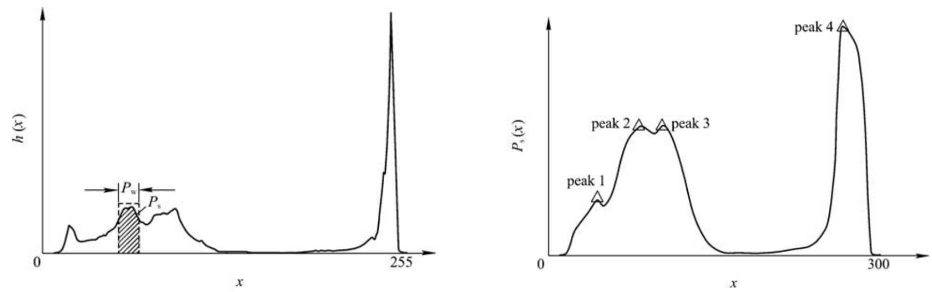

Figure 3.6: Finding peak regions in histogram. Left: convolving process, Right: convolved results, [24].

In order to find the width of peak regions, the luminance histogram is convolved by a unit sequence with width P_w, a predefined value. A good choice for P_w is crucial and it turns out that for 8bit resolution, $P_w = 16$ is optimal. Actually P_s, the width of a peak is defined as the height of the local maxima of the convolved luminance histogram divided by the total number of pixels. To prevent luminance flicker, the first two largest peak widths are considered and denoted as P_{bk}, the peak in the

darker region and P_{br}, the peak in the brighter region.
Furthermore the mean brightness of the scene L is calculated by (compare to 2.1 on page 3):

$$L = \frac{B_l F^2}{kGT} \tag{3.1}$$

, where k is set in order to normalize L, i.e. $L \in [0, 1]$.
In a second step the peak regions P_{br} and P_{bk} are weighted separately, by:

$$W = a_i + b_i(P_s - c_i)^2 \tag{3.2}$$

where a_i, b_i and c_i denote constant parameters and P_s is the size of a peak region, that means either P_{br} or P_{bk}. Figure 3.7 shows different plots of W.

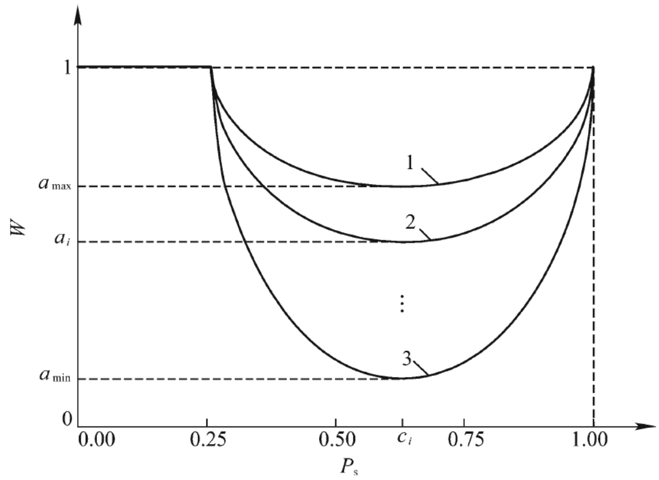

Figure 3.7: Set of weighting curves, 1:if $L = 0$ for P_{br} or if $L = 1$ for P_{bk}, 2: if $L = 0.1$ for P_{br} or if $L = .9$ for P_{bk}, 3: if $L = 1$ for P_{br} or if $L = 0$ for P_{bk}, [24].

In a third step the final mean weighted luminance value M_W is calculated by:

$$M_W = \frac{P_{acc} - \sum_{i=1}^{2} P_{acc,i}(1 - W_i)}{N(1 - \sum_{i=1}^{2} P_{si}(1 - W_i)} \tag{3.3}$$

where N is the total number of pixel, P_{si} is the size of a peak region and (for 8bit resolution)

$$P_{acc} = \sum_{j=0}^{255} jh(j) \tag{3.4}$$

$$P_{acc,i} = \sum_{j=x_i}^{x_i+P_w-1} jh(j) \quad (i = 1, 2) \tag{3.5}$$

x_i denotes the starting position of the peak region, i.e. the left end point of the corresponding interval in the luminance histogram.
To find the next exposure time, this algorithm continues like a normal weighted mean value algorithm, i.e. M_W adjusted to mid tone (approximately 128 for 8bit).

Discussion

As mentioned before, the key idea is to emphasize the weight of the flat luminance histogram regions (low peaks), by reducing the weight on the high peak regions. Therefore the weighting parameter W is high ($W = 1$) for low peaks, has then a minimum, depending on the predefined parameter $c_i \in [0.5, 1]$ and finally reaches one again, because a peak value close to one means that almost all pixels are in the peak region (a totally under- or overexposed image). The height of the minimum is given by a_i. A mean luminance L close to one means difficult lighting conditions (a really bright image) and therefore the regions of interests, i.e. the flat regions have to be emphasized a lot, which results in a small a_i for the weighting curve of P_{br} and a high a_i for P_{bk}. If the mean luminance L is close to zero, again a difficult lighting condition has to be handled and therefore a high a_i for P_{br} and a small a_i for P_{bk} are taken.
If one takes a closer look at equation 3.3, it can be noticed that if $W_i = 1 \;\; for \; i = 1, 2$, M_W is the normal mean brightness. In contrary if $W_i = 0 \;\; for \; i = 1, 2$, M_W is the mean brightness value of the image, when the peak regions are excluded. Therefore the parameter W is used to control the influence of the peak regions on the mean weighted brightness value M_W.
For an efficient implementation, instead of the weighting curves a set of fuzzy logic rules can be applied. This optimizes the algorithm in terms of speed and still preserves the basic idea.

The presented algorithm has the following advantages and disadvantages:
Advantages:

- very robust, can deal with difficult light conditions
- flexible, there must be no main object, respectively the main object does not have to lie in a particular image region.
- works with any image sensor
- low computational effort needed (if using fuzzy logic)
- similar to human vision

Disadvantages:

- clipping is neglected
- (no feedback from feature tracking algorithm possible)

3.3 “Optimum” auto exposure method

This method is presented by [25].
For a (piecewise) linear image sensor the exposure time can be seen as the slope of the response curve, i.e. the accumulated charge (which is proportional to the brightness of the image pixel) in function of the photo current (which is proportional to the incident brightness). Figure 3.8 shows the response curve of an linear image sensor.

The captured brightness range of the scene is therefore $\in [b_{min}, b_{max}]$. The key idea behind this method is very simple: first a brightness histogram from the whole scene is created, by multiple images and then the slope α, i.e. the exposure time is

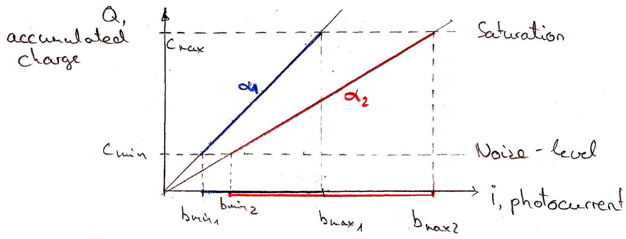

Figure 3.8: Linear response curve. Note that the steeper slope α_1 corresponds to a longer exposure time, where as α_2 corresponds to a shorter exposure time.

taken in order to maximize the number of well exposed pixels. That means, that most brightness levels of the scene are located between $[b_{min}, b_{max}]$.
To create the high dynamic range brightness histogram of the scene, multiple pictures with different exposure times are taken. However in [25], the response of the camera is assumed to be linear, which is not perfectly true. It is stated that if the images are exposed such that their brightness ranges touch each other, approximately three images are needed to capture all brightness levels of an entire scene.
In a second step, to calculate the exposure time, which maximizes the amount of well exposed pixels, the integral

$$\max_{\alpha} \int_{\frac{c_{min}}{\alpha}}^{\frac{c_{max}}{\alpha}} p(b)db \tag{3.6}$$

is maximized, by choosing the optimal value for α.
$p(b)$ denotes the number of pixels of the scene at the brightness level b, c_{min} the noise level and c_{max} the saturation level of the image sensor.
However, finding this maximum is not trivial, requires a numerically robust algorithm and takes some computation time.

Furthermore, this idea can be extended to multiple slope, sometimes also called "clipped" image sensors (see Appendix A for more information about image sensors).
Figure 3.9 shows the response curve of an image sensor with two different slopes.

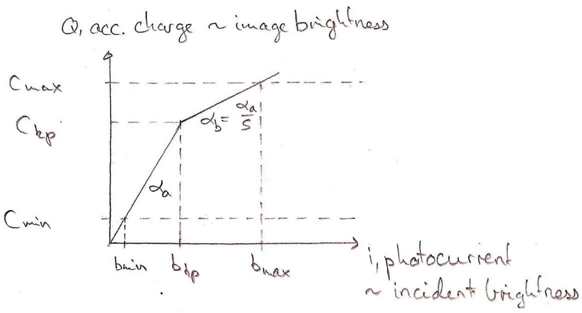

Figure 3.9: Multiple slope response curve.

The point where the slope changes is called kneepoint and is characterized by the parameters $s \in [1, \infty]$ and $\beta \in [0, 1]$. The slope of the second segment is given by $\alpha_b = \frac{\alpha_a}{s}$ and the height of the kneepoint by $c_{kp} = \beta c_{max}$. Therefore:

$$b_{min} = \frac{c_{min}}{\alpha_a} \tag{3.7}$$

$$b_{kp} = \frac{\beta c_{max}}{\alpha_a} \tag{3.8}$$

$$b_{max} = \frac{s c_{max} + (1-s)\beta c_{max}}{\alpha_a} \tag{3.9}$$

Furthermore the quantization error can be considered (for n bit output):

$$\Delta b_a = \frac{c_{max}}{\alpha_a 2^n} \tag{3.10}$$

for the first segment and

$$\Delta b_b = \frac{s c_{max}}{\alpha_a 2^n} \tag{3.11}$$

for the second segment. As a consequence, the quantization error raises with the ratio s of the slopes.
Finally equation 3.6 can be rewritten as:

$$\max_{\alpha_a, s, \beta} \int_{b_{min}}^{b_{kp}} p(b)db + s^{-m} \int_{b_{kp}}^{b_{max}} p(b)db \tag{3.12}$$

where equations 3.7, 3.8, 3.9 are used and the term s^{-m} punishes quantization errors with the constant $m \in [0, 1]$.

Maximizing equation 3.12 is even more complex. But on the other side, this method exposes the maximum amount of pixels well. Information losses, which occur when pixels are clipped are minimized.
However, it is not always the case, that the whole image is of interest. Maybe just a good exposure of the main object or a certain region is of interest. In that case, clipped brightness levels in the background would even enhance contrast and therefore this method would be insufficient.
Furthermore, almost all information contained in the frames made in order to capture the brightness range of the scene (first step) is discarded. This information is just used to calculate the “optimal” exposure time. However condensing this information in a high dynamic range image is computationally less intense as maximizing equation 3.12, but would contain theoretically all the information of the scene[1].
Summarizing, the following advantages and disadvantages can be outlined:

Advantages:

- very robust, can deal with difficult light conditions
- works also with multiple slope sensors
- information loss regarding the whole image is minimized
- quantization errors can be considered

[1] On the other side other problems arise with high dynamic range images, see chapter 4

Disadvantages:

- high computational effort needed
- a brightness histogram of the scene has first to be measured, this may cost up to three or four frames
- overall exposure, algorithm does not weight certain image regions more

Chapter 4

Multiple picture algorithms

In this chapter methods, using multiple images of the same scene with different exposure times are presented. First the so called fusion methods, which are not using the exact image sensor response function and mostly use only two frames are explained. In the second part, this idea is generalized to high dynamic range images.

4.1 Image fusion

[31], [32] propose both to use image fusion, to achieve a better auto exposure control. In [32] first an normal mean based exposure algorithm is performed. Then two successive video frames with half exposure and double exposure time are fused according to:

$$Y_F(x) = \begin{cases} M_F + \frac{Y_H(x) - M_H}{S_H} & if(Y_H(x) > M_H)and(Y_D(x) > M_D) \\ Y_D S_D & if(Y_H(x) < M_H)and(Y_D(x) < M_D) \\ \frac{Y_H(x) + Y_D(x)}{2} & else \end{cases} \tag{4.1}$$

$$M_F := \frac{M_D + M_H}{2} \tag{4.2}$$

$$S_D := \frac{M_F}{M_D} \tag{4.3}$$

$$S_H := \frac{N - M_F}{N - M_H} \tag{4.4}$$

with:

M_D	median brightness of the image with half the exposure time
M_H	median brightness of the image with double the exposure time
M_F	median brightness of the fused image, is set as the mean of the two medians of the two frames with different exposure time
N	number of brightness levels, i.e. 256, for 8bit
$Y_D(x)$	x'th pixel brightness value of the image with half exposure time
$Y_H(x)$	x'th pixel brightness value of the image with double exposure time
$Y_F(x)$	x'th pixel brightness value of the fused image

Therefore the pixels of the two successive images are fused linearly. However, the response function even for a linear image sensors is not perfectly linear and as a result, this technique works only well for mid tones.
With the right image sensor fusion can be implemented really efficiently. The so called dual sampling approach, see [12], Appendix A, adds an additional capacitor bank to a CMOS sensor in order to maintain a high frame rate.
Finally the following advantages and disadvantages can be outlined:

Advantages:

- can capture a higher brightness range of the scene, less clipping
- can be implemented very efficiently

Disadvantages:

- works only with linear image sensors
- works only well in the region, where the sensor response curve is linear, i. e. mid tones
- cannot totally eliminate clipped regions
- adjusts the mean brightness of the whole image, if just certain areas are of interest, the contrast is even reduced due of fusion!

Embedded fusion algorithm

In [32] the fusion process is embedded into a more complex algorithm. Fusion is only used for difficult light conditions, which are detected by comparing the mean and the median. This makes sense, because the median is much less sensitive two distortions than the mean.
This algorithm can therefore quickly estimate an appropriate exposure and then, if difficult light conditions occur improve the exposure settings to achieve a robust auto exposure. A drawback is that only the whole image is well exposed. If one is interested only in certain image regions (for example in feature tracking), fusion can even reduce the contrast in those regions.

4.1.1 Fusion maximizing information entropy

In [41] multiple images are fused together. The aim is to maximize the information content, thus to maximize information entropy, see Appendix B. This is done by dividing the images into subsections of 50x50 pixels (the subsection size was determined empirically) and calculating the information entropy of each block. In a second step the final image is assembled only by the blocks with the maximal information entropy. To prevent sudden change of brightnesses a smoothing kernel is applied and the blocks are merged together. This approach is very interesting, since only image parts are fused together, which still maintains a high contrast.
Therefore, the following advantages and disadvantages can be brought together.

Advantages:

- can capture a high brightness range of the scene
- eliminates clipping
- maintains a high contrast, due to partial fusion

- can be combined with an auto exposure algorithm based on information entropy (see 5.2), which would save computation time

Disadvantages:

- works only with a linear image sensor
- higher computation effort needed compared to basic fusion algorithm
- needs different exposures of the same scene

4.2 High dynamic range images

A high dynamic range image is made by condensing the information of multiple pictures with different, fixed exposure times into one picture, the radiance map. Figure 4.1 shows an example of a high dynamic range image.

Figure 4.1: Example of a HDR-image, [26].

The reason for discussing high dynamic range images is, that by using this technique, no more auto exposure has to be done. Moreover, a radiance map contains theoretically all the information of the scene.
Radiance maps cannot be displayed, since there exist no display devices, which can handle such high brightness ranges. In order to display such a HDR - image, methods known as tone mapping are performed. But because they are not related to auto exposure, they are not further discussed. This chapter presents only roughly the main idea behind this method. More informations, extensions and variations are found in the bibliography: [27] gives a good overview and [28], [29] show practical applications of this method.

4.3 Algorithm for creating radiance maps

The whole algorithm can be divided into two steps. First the actual response of the image sensor has to be found. This step is only performed once. Then using the results from the first step multiple images of the same scene are condensed into the radiance map.

4.3.1 Image sensor response

It is assumed that the following relation holds:

$$Z_{ij} = f(I_i \Delta t_j) \tag{4.5}$$

where Z_{ij} is the i'th pixel value of the j'th image, I_i is the brightness of the scene, and Δt_j is the exposure time of the j'th image. The aim is to find f and I_i, where Z_{ij} and Δt_j are known.
By inverting 4.5 and taking the logarithm, one gets:

$$\ln(g(Z_{ij})) = \ln(f^{-1}(Z_{ij})) = \ln(I_i) + \ln(\Delta t_j) \tag{4.6}$$

By substituting $\log(f^{-1})$ by g (see equation 4.6) the problem can be rewritten as summation over all pictures and pixels:

$$\mathcal{O} = \sum_{i=1}^{P}\sum_{j=1}^{N}[w(Z_{ij})(g(Z_{ij}) - \ln(I_i) - \ln(\Delta t_j))]^2 + \lambda \sum_{z=Z_{min}+1}^{z=Z_{max}-1}(w(z)g''(z))^2 \tag{4.7}$$

Where N is the number of pixels, P the number of pictures, Z_{min}, Z_{max} the minium, resp. maximum pixel image brightness and λ a constant. The first term ensures that the solution satisfies the set of equations arising from equation 4.6 in a least square sense. The second term ensures that g is smooth. For g'' the discrete approximation $g(z+1)-2g(z)+g(z-1)$ is used. The scalar λ weights the smoothness term and has to be chosen appropriately for the amount of noise expected in the Z_{ij} measurements, [30]. $w(z)$ is a weighting function in order to weight the mid tone brightness levels more (because the mid tones are more important and reliable). Furthermore an additional constraint (a scale factor) has to be added. In [27] $g(Z_{mid}) = 0$ is proposed.
The solution to 4.7, i.e. the response curve g can be found by singular value decomposition (SVD).
Although this step requires some computation, it has only to be done once per image sensor and is therefore no big disadvantage of this method.

4.3.2 Algorithm for high dynamic range radiance map

Once the response curve g has been found, recovering the radiance map is straight forward. From equation 4.5, the radiance can be calculated as:

$$\ln(I_i) = g(Z_{ij}) - \ln(\Delta t_j) \tag{4.8}$$

To avoid information losses, all the pixel from all the captured image of the same scene are considered, where the mid - tones are weighted more:

$$\ln(I_i) = \frac{\sum_{j=1}^{N} w(Z_{ij})[g(Z_{ij}) - \ln(\Delta t_j)]}{\sum_{j=1}^{N} w(Z_{ij})} \tag{4.9}$$

The following advantages and disadvantages of this method can be summarized:
Advantages:

- very robust, can deal all light conditions
- works with any image sensor
- almost no information loss (still quantization losses)

Disadvantages:

- a lot of memory is needed to store a radiance map. Solutions to this problem are found in [27].
- multiple frames needed of the same scene
- cannot display radiance maps directly (needs tone mapping, which requires some additional computation effort)

Chapter 5

Advanced algorithms

5.1 Content based auto exposure

[34] presents an algorithm adapted to feature tracking applications. The presented system is applied to automobiles, where the goal is to detect lane markings to warn the driver. The key idea is to use the multiple slope feature of ("clipped") CMOS image sensors, see Appendix A for more details.
As a first approximation, the exposure time $\tilde{t}_{int}$ is found by a simple mean value based algorithm. Then the lane markings are detected and tracked. With this information the mean brightness value $\overline{c_l}$ of the lane markings is calculated:

$$\overline{c_l} = \frac{\sum c_k}{A_l} \tag{5.1}$$

Where A_l is the area of the lane markings. The mean brightness value of the surrounding road, $\overline{c_g}$ is calculated in the same way. In order to maximize the contrast between the road and the lane markings for the next frame, the mean value $\overline{c_l}$ is set to the maximum brightness level c_{max}, i.e. 256 for 8bit. Therefore:

$$c_{max} = \frac{T_{int}}{\tilde{t}_{int}} \overline{c_l} \tag{5.2}$$

or rewritten for the next exposure time T_{int} of the next frame:

$$T_{int} = \frac{c_{max}}{\overline{c_l}} \tilde{t}_{int} \tag{5.3}$$

The brightness level of the lane markings $\tilde{c_l}$ in the actual frame lie between:

$$\tilde{c_1} < \tilde{c_l} < \tilde{c_2} \tag{5.4}$$

In the next frame they will lie between:

$$c_1 < c_l < c_2 \tag{5.5}$$

with $c_1 = \frac{T_{int}}{\tilde{t}_{int}} \tilde{c_1}$ and $c_2 = \frac{T_{int}}{\tilde{t}_{int}} \tilde{c_2}$.
However, for a linear image sensor this would lead to an overexposure of certain areas of the lane markings. To prevent this, the multiple slope feature, shown in figure 5.1 is used.
The first reset time T_{slp2} and the first reset voltage V_2 are set in order to fit the brightness of the lane markings in the region of the second slope. Thus:

$$T_{slp2} = T_{int} - \frac{c_2 - c_1}{\tilde{c_2} - \tilde{c_1}} \tilde{t}_{int} \tag{5.6}$$

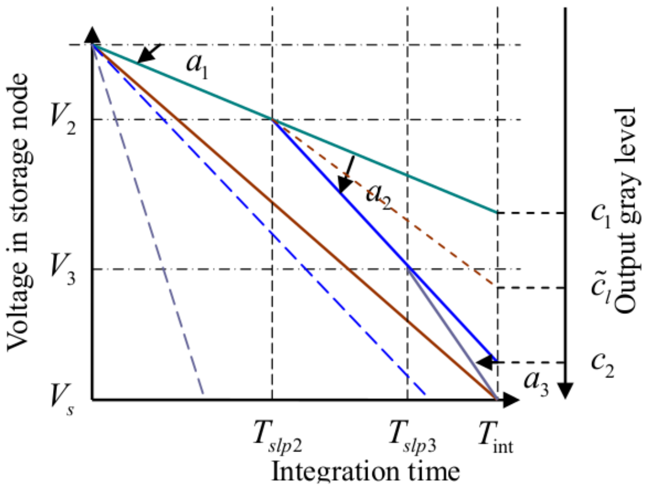

Figure 5.1: Multiple slope feature of a “clipped” CMOS image sensor, [34]

$$V_2 = (1 - \frac{c_1 T_{slp2}}{c_{max} T_{int}}) V_1 \tag{5.7}$$

where V_1 is the reset voltage of the image sensor.
For the third part, the angle a_3 is maximized. This gives us T_{slp3}, V_3 is calculated according to:

$$V_3 = \frac{T_{slp3} - T_{slp2}}{T_{int} - T_{slp2}} V_2 + \frac{T_{slp3} - T_{slp2}}{T_{int} - T_{slp2}} \frac{c_{max} - c_2}{c_{max}} V_1 \tag{5.8}$$

For the next step $\tilde{t}_{int}$ is replaced by T_{int} and the new set of parameters is calculated in the same way.
This procedure should lead in the ideal case to the response curve shown in figure 5.2. Note that all brightness ranges (the road, the lane markings and the sky) are well exposed. Clipping could only occur in really bright regions, for which we are not interested.

However, this algorithm does not consider quantization errors. Because the slope of the response curve decreases with higher brightness levels of the scene, the lane marking region and above all, the sky region suffers more from quantization errors. This is somehow troublesome, because the main areas of interest are the lane markings and not the road region, which has the best resolution.
Summarizing this algorithm, the following advantages and disadvantages can be brought together:

Advantages:

- fast (can be implemented in real time), simple
- can handle extreme light conditions
- although the algorithm focuses on enhancing the contrast between lane markings and the road, the exposure of the whole image is still not neglected, this adds robustness

Disadvantages:

- works only with multiple slope image sensors
- a good feature tracking of the lane markings is crucial, otherwise this algorithm will not work
- quantization errors are not considered

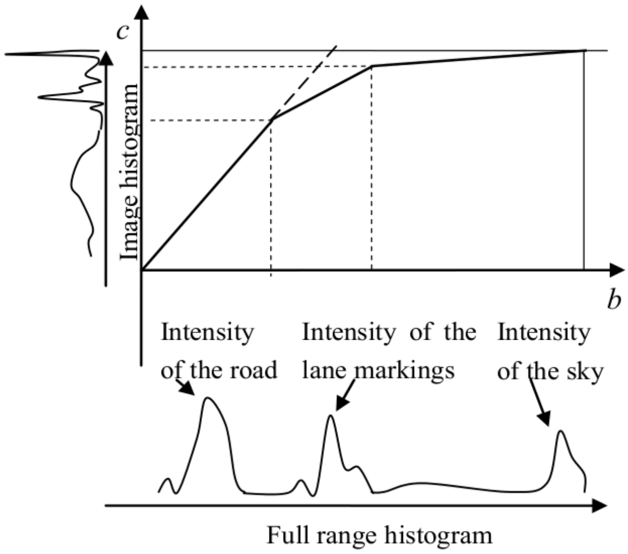

Figure 5.2: Multislope response curve in the ideal case, [34].

5.2 Auto exposure algorithm based on information entropy

A short introduction into the topic of information entropy is found in Appendix B. [35] proposes a very simple algorithm, which uses the key idea of image entropy. Information entropy can be used to measure the information content of an image. The fraction of amount of pixels at a certain brightness level over the total number of pixels is regarded as probability, i.e. the probability of a pixel having a certain brightness level. If every brightness level has the same probability, that means that the ratio of pixels at that brightness level i to the total number of pixels is the same for every brightness level, the information entropy is maximized. When considering the brightness of an image as a signal, a maximum information entropy means, that from the knowledge of the past values, no prediction for future values can be made. Thus, information entropy of an image is defined as:

$$H(X) = -\sum_{i=0}^{N} P(X_i)\log(PX_i) \tag{5.9}$$

with

$$P(X_i) = \frac{number\ of\ pixels\ at\ brightness\ i}{total\ number\ of\ pixels} \tag{5.10}$$

N denotes the total number of different brightness levels, i.e. 256 for a 8bit resolution.

Since every pixel is considered as discrete information source, maximizing information entropy would lead to the maximal information content of an image and therefore to the best exposure. As a result, for every frame the information entropy $H(X)$ is calculated and the next exposure time is chosen in order to maximize information entropy.

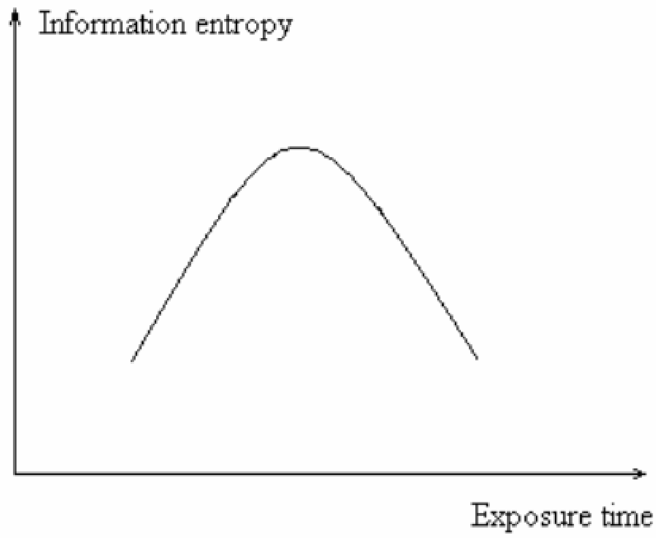

Figure 5.3: Relation between information entropy and exposure time, [35].

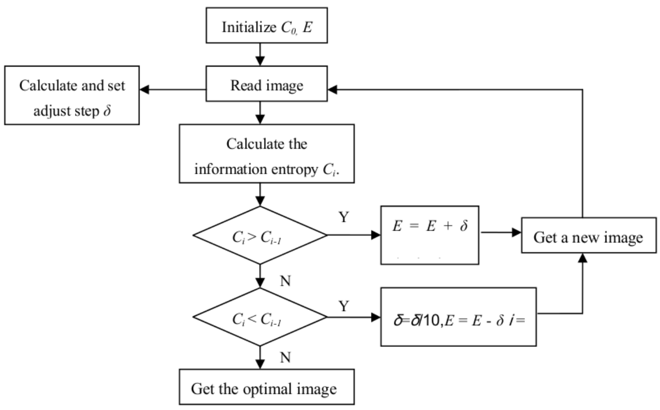

Figure 5.4: Mountain Climbing Servo Algorithm, [35].

[35] assumes that the relationship between information entropy and exposure time can be approximated by a parabola, with a single maximum, see figure 5.3. As a consequence [35] proposes the so called mountain climbing servo (MCS) algorithm to find the maximum. This algorithm is shown in figure 5.4.
However, this algorithm may be improved, as it makes no use of the parabola-assumption or of the knowledge about the integrated image sensor (already the secant or bisection methods would lead to a better performance).
Finally following advantages and disadvantages are summarized:

Advantages:

- simple, theoretically maximal information
- works with any image sensor

Disadvantages:

- image sensor non-idealities, such as clipping is not considered
- overall exposure, does not weight certain image regions more
- extreme light conditions cannot be handled

Optimizing multiple parameters

[42] propose to use information entropy to find the appropriate gain and exposure time. The idea in [42] is similar to the procedure in [35]. However, to find a unique solution an additional constraint has to be added. [42] proposes to search the maximum only along a path (see figure 5.5). Furthermore, this simplifies the algorithm and reduces the computational effort. In [42] the path is a straight line, characterized by exposure time=gain.

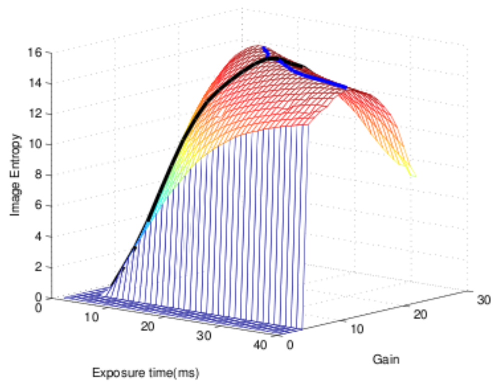

Figure 5.5: Searching the maximum information entropy along the dark path (exposure time=gain), [42].

On the other side a high gain increases noise and therefore the gain should be kept as small as possible. As a result gain should be punished more heavily, i.e. search for example along the path $t = \alpha G^2$, with t the exposure time, α a constant and G the gain.

Chapter 6

Comparison

6.1 Requirements

In this section the presented algorithms are evaluated in relation to a particular VSLAM application, namely to use an UAV for industrial inspection. The task is to provide well exposed images for the localization and stabilization process of the vehicle. To illuminate the environments, the robot is equipped with a controllable on-board light source. This allows active illumination of the surroundings to perform even a better image quality.
Therefore the auto exposure algorithm needs to full fill the following requirements:

- very low computational effort, auto exposure has to be performed in real time
- has to be able to handle difficult light conditions
- the whole image has to be exposed well, there is no main object
- has to react fast to high changes in light conditions, but must ignore small changes in order to facilitate the task of feature tracking
- has to control actively the illumination

6.2 Active Illumination

Unfortunately no literature about active illumination in connection with auto exposure has been found. Nevertheless, controlling the light source can improve the image quality and above all, the adjustment time of exposure.
The exposure time is limited only on high side: Too long exposure times lead to motion blur, due to the movement of the UAV. Because the depth of field (focus) is only dependent on the aperture, which is constant in this case, there is no limitation on the low side.
In contrast, there are always some shiny or radiant surfaces, which limits the maximum light intensity. Although in this particular case, the maximum light intensity may be very high, since there is only a small amount of radiant surfaces. However, this upper light intensity limit must be determined empirically.
Fortunately, no new control method has to be designed, since changing exposure time and changing light intensity have the same effect on the image. Changing the light intensity can be seen as rough adjustment (and must be handled with care) in contrast to changing exposure time, which represents the fine adjustment. A simple control algorithm would just watch the change in exposure time: if the correction of exposure time from one frame to the next is above a certain threshold, the light

source is adapted. This improves the adjustment time of image exposure.
A more advanced algorithm would adjust the light source more often. This could be realized for example by a PID (or even PI) controller. However, this controller must react very gently, i.e. have a high time constant (low gain (k_p) and high integrator time constant (T_i)) to control illumination carefully and slowly.
Additionally, if illumination is controlled actively, also the control algorithm of the exposure time must be slightly adapted and tuned, in order guarantee an efficient and synchronized exposure system.

6.3 Evaluation

6.3.1 Mean value algorithms

If a normal mean value based algorithm would be applied, it would certainly not full fill the task. However, mean value based algorithms are simple, need quite low computational effort and since there are only a few predictable parameters to tune, they may proceed in a iterative way. An iterative procedure is especially of use to facilitate feature tracking, because small light changes do not effect exposure much. Therefore, an iterative mean value based algorithm for auto exposure with the following modifications is adequate:

- Since the on board light source causes a light cone, certain part of the image may be totally dark. As a consequence the totally dark regions, outside of the light cone must not be considered. Therefore a weighting function must be established in order to exclude these regions.

- Secondly, feedback from the feature tracking process should be realized: Regions, where the distinctive features are expected are weighted more. Actually not only the feature itself but also a small surrounding must be weighted to guarantee a better feature detection.

But still, there are several drawbacks. For this application the Micron, MT9V022I77ATM CMOS image sensor is used, which provides the multiple slope feature shown in figure 6.1 (for more details, see Appendix A). A main point is, that by using a mean value based algorithm this additional multiple slope feature cannot be used. Using the multiple slope feature would lead to less clipping and more information of the scene could be captured.

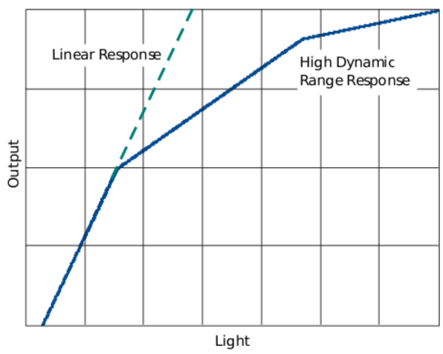

Figure 6.1: Multiple slope respones of the Micron MT9V022I77ATM image sensor with three piecewise linear, programmable segments, [43].

6.3.2 Peak analysis method

The key idea for this method was to emphasize the weight of the regions with high interest. These regions were found by the analysis of the image histogram. Since feature detection and tracking is performed in this application, the region of interest are already known and therefore it makes no sense to use the peak analysis method for this application.

6.3.3 "Optimum" auto exposure method

With this method a very robust auto exposure can be achieved. Furthermore the multiple slope feature of the image sensor could be used. But still, using this method would not be appropriate because of the following reasons:

- The computation time for this method is rather long and generating the brightness histogram of the scene may cost up to three frames. As real time auto exposure is crucial for this application, this method cannot be used.
- Additionally, multiple pictures of the same scene are needed in order to generate the brightness histogram of the scene. Since the robot is moving, this cannot be done.
- Furthermore, an additional weighting of the regions of high interest, which would improve feature detection, is not possible.

Therefore this method cannot be applied efficiently.

6.3.4 Image fusion

Since the UAV has to deal with difficult light conditions, this method has to be combined with an other auto exposure method. However, image fusion can be implemented really efficient with the appropriate image sensor. Nevertheless a high computational effort is needed, because at least twice as much information (from two frames) has to be analysed. Moreover, at least two frames of the same scene are needed and since the vehicle is moving fast, its image sensor has to provide a very high frame rate.
To maintain a high contrast for good feature tracking, the fusion algorithm maximizing information entropy, [41] would fit best. It should be combined with an auto exposure algorithm using information entropy and therefore the calculated information entropy could be used twice (once for fusion and once for additional auto exposure).
But still feedback from the feature tracking process in order to weight areas of interest more, is not possible. Furthermore image fusion method works only with linear image sensors and therefore the multiple slope feature of the applied image sensor cannot be used.

As a result this method could work fine for VSLAM tasks, provided fast computation algorithms, a high frame rate and the appropriate image sensor. However, in the presented application the latter requirement is not fulfilled and therefore this method cannot be applied efficiently.

6.3.5 High dynamic range images

For VSLAM applications these methods are not appropriate. Multiple frames of the same scene are needed, which is impossible to realize due to the motion of the robot. Additionally high dynamic range images are difficult to handle, because of their size. To perform fast image processing they would have to be compressed which would need additional computation time and loss of information.

6.3.6 Auto exposure based on information entropy

In many cases this algorithm represents a quite robust and simple choice. It can also be applied to any image sensor. But for this particular application, there exist a few draw backs:

- Algorithm requires high computational effort.
- Must be extended to use the multiple slope feature.
- Must be extended in order to enable feedback from feature tracking, i.e. weighting image areas differently.
- Must be extended to consider only the part within the light cone and neglect the outer, totally dark regions.

Therefore this algorithm does not suit the requirements and cannot be applied efficiently.

6.3.7 Content based auto exposure

For this particular application this method would suit best. It is the only approach found that can use the multi slope feature efficiently and in real time. Summarized, it has the following advantages:

- Algorithm needs low computational effort and works in real time.
- Feedback from the feature tracking is used to enhance contrast in the regions of interest. It can therefore focus on the well exposure of the regions of interest, without neglecting too much the other parts of the image.
- Can handle difficult light conditions, because of an efficient use of the multiple slope feature.

On the other side a robust feature tracking algorithm is a prerequisite, otherwise this method cannot focus on the regions of interest, which would result in non optimal exposure. This would again affect the feature tracking process and finally result in a vicious cycle.
But still, since a high brightness range of the scene is captured due to the multi slope feature, the above described phenomena should not occur.
Therefore this method is certainly adequate for this application. (It has also been developed and tested successfully for lane marking detection in auto mobiles, which has a lot of similarities to this application, [34].)

Chapter 7

Conclusion and Outlook

It is difficult to rate the presented algorithms in general, since all of them have different strengths and weaknesses. Choosing the appropriate method is therefore a difficult task and has always to be made depending on the image sensor, the actual application and its requirements.
Moreover, in a lot of cases, a combination of different approaches may deliver the best results. As a result it is very important to be aware of the different methods together with their advantages and disadvantages, since they represents the building blocks.

Concerning image sensors, CMOS technology is very promising. With CMOS a lot of additional features can be realized (for example multiple slopes). Recently Melexis developed image sensors, which are capable of performing local exposure: To every image region different exposure settings, i.e. integration times can be applied. For scenes with extreme light conditions or fast motions this can be very useful. For example if some parts of a dark scene are lighted, one could expose well total dark and bright parts in the light cone within the same image. In UAV vision this feature could be promising as well: While the horizon is staying still, regions on the side can move very fast. Therefore the image regions containing the horizon can have a longer exposure time to have more depth of focus, where regions containing fast moving objects must have a short exposure time to prevent motion blur.

But on the other side, there exists almost no auto exposure algorithm, that can handle these extra features. Therefore, still lot of improvements can be made by developing unique auto exposure algorithms that can use efficiently the full functionality of these innovative image sensors.

Appendix A

Image Sensors

As exposure is closely related to the image sensor of the camera, this section gives a rough overview about the different type of image sensors, their non idealities and finally the approaches that have been made to extend the dynamic range. It is not intended to explain the functionality in details.

A.1 CCD Image Sensor

CCD stands for charge-coupled devices. It was invented in 1969 at AT & T Bell Labs by Willard Boyle and George E. Smith and rewarded in 2009 with the Nobel Prize for Physics [4].
A CCD image sensor consists of an array of separate small photo sensors, the charge-coupled devices. Each of them has a potential well. When the photo sensor is exposed to incident light minority carriers are collected in the potential well. The amount of charge collected depends therefore on the exposure time. After exposure, the collected charges are shifted out from one potential well to the next. Finally the charge sequence is first converted into a voltage and then into a digital signal.

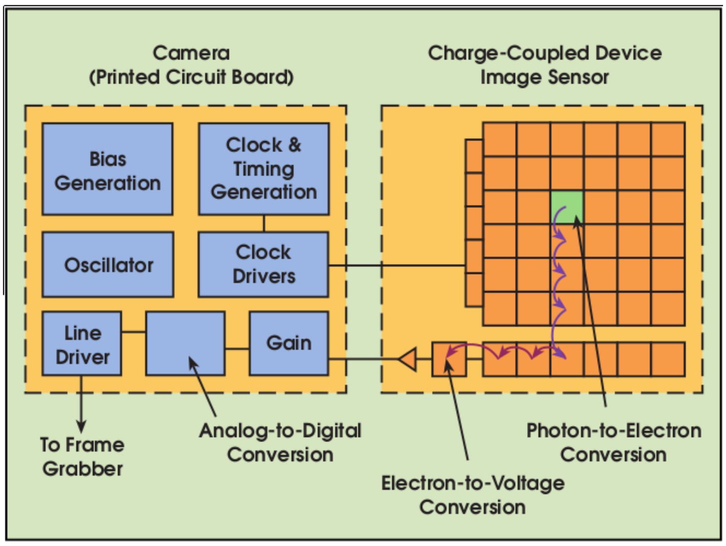

Figure A.1: Schematic of a camera with a CCD-image sensor. Note that most functions take place on the camera' s printed circuit board. [6]

The CCD-Technology is widely spread for high end applications, such as Medicine, Science and Astronomy. Properties of CCD-sensors are roughly summarized next, for more explanations consider [5], [6].

Advantages:

- high resolution (world record: 9216x9216 pixels)
- small pixel (world record: 2.4 μm)
- low noise

Disadvantages:

- multiple high supply voltages
- high power consumption
- blooming, smearing

A.2 CMOS Image Sensor

CMOS stands for complementary metal oxide semiconductor.
In recent years CMOS image sensors have been improved a lot. Nowadays they can easily keep up with CCD image sensors. A CMOS image sensor consists of an array of independent cells, called Active Pixel Sensor (APS). Two different types, APS with photodiode or photogate exists.

A.2.1 APS with photodiode

It is typically equipped with a photodiode and several transistors (See figure A.2). First a storage element, which can be implemented as a real capacitor or just as a parasitic capacitance of the photodiode and the transistors, is charged up to a known level U_{dd}, by TR. When incident light hits the photodiode D_{ph} a photocurrent I_{ph} is generated which discharges the storage element. After the exposure time, i.e. the integration time, the remaining charge is amplified with a MOSFET, T2. Finally with the row select T1 is activated, and the remaining amplified charge is read out with a voltage-follower [7], [10].

A.2.2 APS with photogate

Invented in 1993, the APS with photogate differs from the one with photodiode only in terms of the sensor-element, the read-out mechanism is the same. The photogate has a lot of similarities with the CCD image Sensor, [10].

A.2.3 CMOS - Summary

The CMOS - Technology is often used for industrial applications, which are space constrained, such as security cameras, bar code reader, consumer scanners, etc. Main advantages and disadvantages are presented next, interested readers are referred to [5], [6].

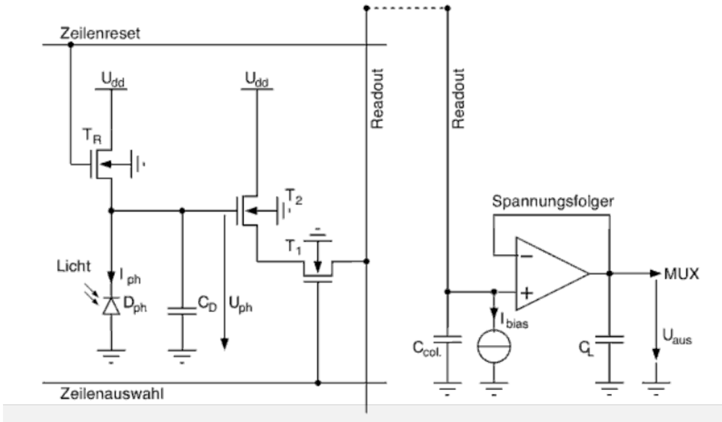

Figure A.2: Functionality of APS with photodiode, [10].

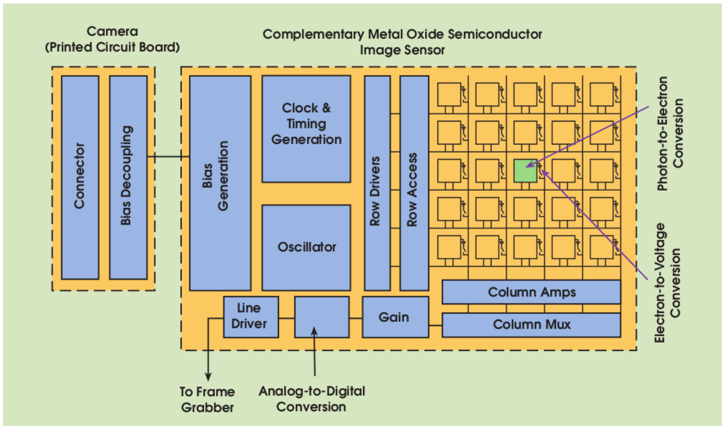

Figure A.3: Schematic of a camera with a CMOS sensor. Note that a CMOS-sensor converts charge to voltage at the pixel. [6]

Advantages:

- single voltage supply
- low power consumption
- no blooming, no smearing
- region of interest, random pixel access
- ability to integrate analog an digital processing on the same chip with the sensor

Disadvantages:

- dark current
- fixed pattern noise
- temporal noise

A.3 Sensor non-idealities

All image sensors suffers from different non idealities. According to [8] they can be divided in the following categories:

- dark current
- temporal noise
- offset and gain fixed pattern noise (FPN)
- spatial sampling and low pass filtering

Dark current

Dark current is the leakage current at the integration node, i.e. current not induced by photogeneration (incident light), but due to junction and transistor leakage, [8]. It varies widely across the image sensor array and can therefore not easily be removed.

Temporal noise

It consists mainly of three different noise sources, [8]: First, there is shot noise. Light can be seen as photons, which are emitted randomly from a light source. However, the mean value of emitted photons is proportional to the light intensity. Similarly, also current consists of randomly moving electrons. As a result, there is an uncertainty on the amount of particles hitting the image sensor and being converted into electrical charge in a given time period. Both processes can be modelled as Poisson distributions. Their standard deviation is often referred as shot noise. As a consequence, for n particles (charges or photons) the shot noise is $\sqrt{n}$, [7]. It is clear, that a longer exposure time decreases the amount of shot noise caused by photons.
Second, there are a lot non-idealities in the electronic circuit, such as Johnson white noise of the transistors or sample and hold (kT/C) noise.
Third, there is flicker noise (sometimes also called 1/f noise). It is related to the charge carrier traps in semiconductors, [7] and can be reduced by correlated double sampling, [5]. Temporal noise can also be lumped into three additive components: Integration noise, reset noise and readout noise. Reducing most temporal noise is only possible with larger pixel electronics.

Fixed pattern noise

Fixed pattern noise (FPN) is due to device and interconnect mismatches over the sensor. It can be divided into two components: offset and gain FPN, where Offset FPN can be reduced using correlated dual sampling (see below). FPN is much worse for CMOS image sensors, than for CCDs, due to multiple levels of amplification.

Image sensor model

As a summary of all non-idealities the following image sensor model is introduced, [8], [11]:
where:

$$Q(i) = \begin{cases} \frac{1}{q} i t_{int} & \text{for} \quad 0 < i < \frac{qQ_{sat}}{t_{int}} \\ Q_{sat} & \text{for} \quad i \geq \frac{qQ_{sat}}{t_{int}} \end{cases} \tag{A.1}$$

is the sensor transfer function, and

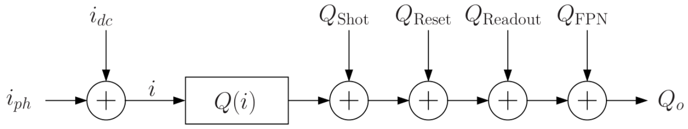

Figure A.4: Image sensor model, [8].

Q_{shot}	shot noise
Q_{reset}	reset noise
$Q_{readout}$	readout noise
Q_{FPN}	fixed pattern noise
q	elementary charge, $1.602 * 10^{-19}C$
Q_0	output charge
i_{dc}	dark current
i_{ph}	photo current

Correlated double sampling (CDS) is performed for noise reduction: The output is sampled twice, after reset and after the integration time. Both Samples are substracted from each other and therefore the offset FPN is eliminated, but readout noise is doubled and reset noise is either eliminated (CCD, photogate APS) or doubled (photodiode APS). Note: correlated double sampling is only applicable for linear sensors transfer function.

Signal to noise ration (SNR)

The signal to noise ratio SNR is defined as the ratio of the input signal power to the average input referred noise power and typically measured in dB, [8], [11].

$$SNR(i_{ph}) = \frac{i_{ph}^2}{\frac{q^2}{t_{int}^2}\left(\frac{1}{q}(i_{ph}+i_{dc})t_{int}+\sigma_r^2\right)} \quad for \quad 0 < i < \frac{qQ_{sat}}{t_{int}} \tag{A.2}$$

where: $\sigma_r^2 = \sigma_{reset}^2 + \sigma_{readout}^2 + \sigma_{FPN}^2$ is the read noise power.

Dynamic range

Dynamic range is a relative measure between the saturation level and the noise floor. It is often used to quantify the performance of an image sensor. However, two image sensors can have the same dynamic range, but perform differently in different brightness areas.

Thus:

$$DR = 20\log_{10}\left(\frac{i_{max}}{i_{min}}\right) = 20\log_{10}\left(\frac{\frac{qQ_{sat}}{t_{int}} - i_{dc}}{\frac{q}{t_{int}}\sqrt{\frac{1}{q}i_{dc}t_{int}+\sigma_r^2}}\right) \quad dB, [8] \tag{A.3}$$

where i_{min} is the minimum photo current, above the noise floor and i_{max} is the maximal photo current before saturation.

A.4 High dynamic range sensors

Several approaches have been made to extend the dynamic range of image sensors. One can extend the dynamic range either towards the darker or the brighter level.

Expanding towards the darker region, means dealing with low photocurrents and leads to a low SNR. Therefore, almost all the approaches presented extend the brightness range towards brighter levels. They can be divided into, [12]:

- companding sensors
- "clipped" sensors
- multimode sensors
- frequency-based sensors
- other methods

Companding sensors

They mainly make use of the subthreshold operation of MOS transistors in order to get a logarithmic sensor response and can be implemented in both, CCD and CMOS image sensors. The photocurrent I_p is given by:

$$I_p = I_{D0} \exp \frac{q}{nkT}(V_G - V_{SS} - V_T) \quad , [13] \tag{A.4}$$

where:

V_G gate voltage
V_S source voltage
V_T threshold voltage
q elementary charge, $1.602 * 10^{-19}C$
k Boltzmann' s constant
C_o capacitance of gate oxide layer
C_D capacitance of depletion layer
I_{D0} constant, see [13] for more details
$n = \frac{C_o+C_D}{C_o}$

And therefore:

$$V_G = V_{SS} + V_T + \frac{nkt}{q} \ln(I_p/I_{D0}) \tag{A.5}$$

is logarithmically dependent on the photocurrent, [13].

This configuration allows to read out the voltage V_G without integration. As a consequence, a very high frame rate can be achieved and no exposure - control has to be integrated.
A disadvantage of these pixels is, that logarithmic compression leads to low contrast. Several methods exists, [15], where linearity around the operation point is exploited. The other problem is, that the mismatch between the transistors will cause a non-linear FPN, which cannot easily be removed. Since the sensor works in continuous time, there is additional noise. However, by including a small capacitance, I_p is integrated and averaged, in order to minimize the additional noise and improve the SNR, [13].

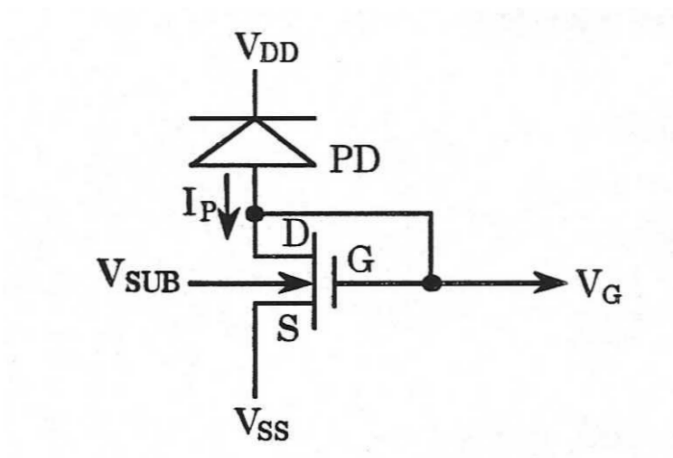

Figure A.5: Logarithmic converter, [13].

"Clipped" Sensors

This method is also called "Well Capacity Adjusting" or multiple slope method. First the application for CCDs is described:
If in a CCD a potential well is overfilled, minority carriers can diffuse to the neighboring potential wells. This phenomena is called blooming and is prevented by an overflow gate. In order to produce an logarithmic response, the voltage applied to the overflow gate changes over the integration period, thus changing the height of the potential barrier. Via this control of the potential barrier an arbitrary compression characteristic can be achieved, i.e. also a logarithmic one, [15].

In CMOS technology, the reset signal b(t) is adjusted multiple times during integration, see figure A.6.

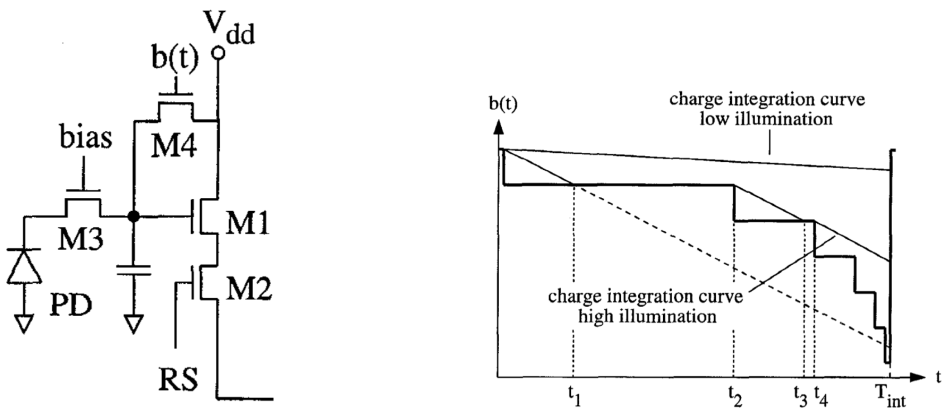

Figure A.6: Logarithmic CMOS-pixel and its charge integration curve, [16]

The gate voltage b(t) at M4 establishes a potential barrier to electron flow. If charge level of the node exceeds the barrier level, the excess charge flow to the drain. Therefore the charge integration is limited to the hight of b(t) which leads to a logarithmic sensor response, see figure A.7, [16].

For the CCD technique a drawback is, that although the principle is simple, ad-

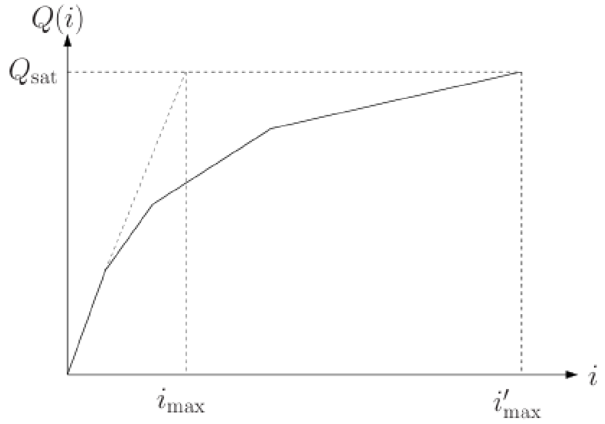

Figure A.7: Sensor response function, where i is the photocurrent (directly proportional to the incident brightness), and Q the accumulated charge (proportional to the image pixel brightness), [8].

ditional circuitry is required to make it work in the proper way, which increases complexity, [7]. Furthermore the overflow drain either requires additional space, or if build vertically, decreases the visible long-wavelength sensitivity of the photosensor, [12].
For the CMOS-pixel the main problem is mismatch between the transistors, resulting in a non uniform reset and leading to additional noise, [7]. But luckily CDS is still applicable, which can be used to reduce the amount of offset noise.

Multimode sensors

This sensor uses additional bipolar transistors to boost the photocurrent and has only been used for CMOS sensors. Proposed is a sensor with two additional bipolar transistors: For low light, both are activated, for midtones only one and for high intensities both are inactivated. However, these transistors cannot be activated/deactivated during integration time. As a result this approach is only applicable for either dark scenes or bright scenes and cannot be used to capture high dynamic range scenes. Furthermore the additional transistors results in a low fill factor and are causing additional noise, [12].

Frequency based sensors

This approach works only with CMOS sensors and uses an simple integrate and reset circuitry to directly convert optical energy into a pulse frequency. The frequency of the pulse is still proportional to the photocurrent, but has no saturation limit. As a result, a very high brightness range can be captured by the image sensor. Although the principle seems simple and effective, the sensor produces a lot of noise due to transistor mismatch. Additionally it needs a delay element in order to ensure a proper reset, which decreases the fill factor, [12].

Other methods

There exists a few other approaches for expanding the dynamic range of image sensors. One of the most recent for CMOS is, to control the exposure settings locally. That means that almost every pixel can have a different response curve. This feature is especially useful for object tracking. It allows to perform a good exposure for the object, which has to be located, without drawbacks for the remaining scene, [7].

In locally adaptive image sensors, the responsivity of a pixel is adapted as a function of the mean intensity detected in the neighbourhood of the pixel and the resulting image is with considerable effort computed externally, [7].
The Time-to-saturation approach measures the time at which each pixels reaches saturation, instead of an accumulated charge. The main disadvantage is a low fill factor and no CDS, [8], [7].

Multiple Capture: Dual sampling has been used for CCD and CMOS sensors. It basically takes two images, one with a long and one with a short exposure time. In a second step the two images are fused either linear or nonlinear. The main drawback is that two integration times are used. But on the other side, since the sensor is linear, CDS can be done, the good noise characteristic is preserved and the fill factor is high, [12].

Conclusions - Summary

Different methods have been shown to extend the dynamic range of image sensors. The table A.4 is roughly summarizing the presented methods. It shows different properties of logarithmic sensors, always compared to linear ones. Notice this table is just an estimation based on the functional principle and on the comparison of typical SNR-diagrams of each sensor approach.

Table A.1: Summary of high dynamic range sensors, Where Cm.S: companding sensor, Cp.S: "clipping Sensor", M.M.S: multi mode sensor, F.S: frequency based sensor, L.E: local exposure, M.C: multiple capture

	Cm.S	Cp.S	M.M.S	F.S	L.E	M.C
i_{min}	+	same	same	+	same	same
i_{max}	++	+	+	++	+	+
linearity	no	(CMOS:yes,CCD:no)	yes	no	yes	yes
SNR	−	-	-	−	same	same
fill factor (pixel)	same	-	-	−	-	same

All the mentioned techniques have their specific advantages and draw-backs and as a result it is impossible to determine the overall best image sensor. The choice of an image sensor has always to be related to the specific application. For example, an image sensor for a VSLAM-application is totally different to one for a consumer scanner. A good sensor for the first application would be a "clipping" CMOS (because of the high frame rate, high range and the high SNR) and for the latter a companding CCD (because of the high range, simplicity and the continuous behaviour).

Appendix B

Information entropy

B.1 Introduction and Definition

The information entropy was first introduced by C. E. Shannon in 1948, by publishing his paper "A mathematical theory of communication", [36], [37]. He wanted to study communication in the most abstract way. He thought that the information of a message can be thought as the choice of one particular message out of a set of messages. For a discrete signal the set of messages would be finite and for a continuous signal infinite. The advantage of this description is, that no assumption has been made about the type of message and its transmission (could be for example transmitted by a wave, through a noisy or noiseless channel, . . .).
For a discrete random variable $X : \Omega \rightarrow \Sigma = [0, 1, ..., N]$ the information entropy is defined as:

$$H(X) = \sum_{i=0}^{N} P(X = x_i) \log(\frac{1}{P(X = x_i)}) \tag{B.1}$$

Note, that also the brightness values of an image can be seen as outcomes of a random variable and similarly, the procedure of image acquisition can be seen as transmitting information from a scene to a chip.
Intuitively the definition of the information entropy makes a lot of sense (see also figure B.1):

- if $P(X = x_i) = \frac{1}{N} \ \ \forall i$ (each outcome is equally probable), then $H(X) = \log(N)$, which corresponds to the Hartley's measure of information.
- if $P(X = x_k) = 1$ and $\forall i \neq k, \ \ P(X = x_i) = 0$, then $H(X) = 0$, because it is a priori clear, which message is going to be transmitted and therefore, no information is exchanged.
- if $P(X = x_i) \neq 0, \ \ \forall\ i\ \in \tilde{\Sigma}$ and $P(X = x_i) = 0, \ \ \forall\ i\ \notin \tilde{\Sigma}$ where $\tilde{\Sigma}$ is a subset of $\Sigma = [0, 1, ..., N]$, then $H(X) = \sum_{i \in \tilde{\Sigma}} P(X = x_i) \log(\frac{1}{P(X=x_i)})$, because no information is transmitted from the information sources with zero probability.

Furthermore Shannon could proof the usefulness of his information entropy definition by the Source Coding Theorem. Simplified and summarized, it states that:

Source Coding Theorem: *Suppose a source of entropy H. Then a message of length N symbols can be encoded (compressed) to approximately H N bits and then decoded successfully with high probability. This cannot be done with fewer than H N bits. In order to decrease the probability of failure we need to increase N*, [37].

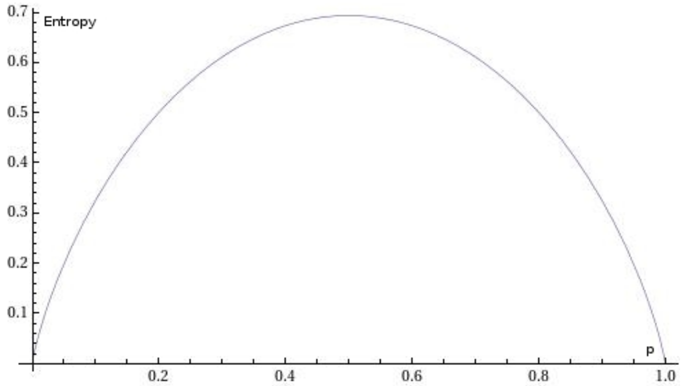

Figure B.1: Entropy (y-axis) of a random variable with two possible outcomes (probability p (x-axis) respectively 1-p). p=0.5 would be the entropy of fair coin tossing and because both outcomes are equally probable, entropy is maximized.

Further information on the proof of the Source Coding Theorem and on communication over noisy channels is found in [36] and summarized in [37].

B.2 Information entropy applied in image processing

This section presents an application of information entropy to show its usefulness. In image processing entropy can be used to quantify the amount of information of a picture and to determine the areas of high and low information, which is useful for feature tracking. This approach has been pursued in [38], [39].
The key idea beind this algorithm is to proceed a series of fine to coarse transformations ($f(x, y, t)$) on the image to observe the change in information entropy. Because these fine to coarse transformations mainly affect the regions with high information, observing the decrease in entropy leads to feature detection.
These fine to coarse transformations can be modelled as diffusion process. Then the entropy production rate, ($\frac{\partial H}{\partial t}$) and the density of entropy production (σ) can be determined for every transformation at every pixel (see figure B.2). Afterwards the entropy production rate is integrated (summed in the discrete case) over all transformations, which is defined as the activity of every pixel. Feature tracking is finally done, by classifying pixels by their activity, according to multiple thresholds (see figure B.3).

Fine to coarse transformations can be modelled by the diffusion equation:

$$\frac{\partial f(x, y, t)}{\partial t} = \Delta f(x, y, t) \tag{B.2}$$

where $f(x, y, t)$ represents the normalized brightness level of the pixel at position x, y after the t'th transformation. Equation B.2, as well as all following equations has to be thought discretized, with $t = [0, 1, 2, ..., N]$, $x = [0, 1, 2, ..., x_{max}]$,$y = [0, 1, 2, ..., y_{max}]$.
The solution to the heat equation may be found by Fourier-transformation, [40]:

$$f(x, y, t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x', y', t = 0)\frac{1}{4\pi t}\exp^{-\frac{(x-x')^2}{4t}}\exp^{-\frac{(y-y')^2}{4t}}\,dx'dy' \tag{B.3}$$

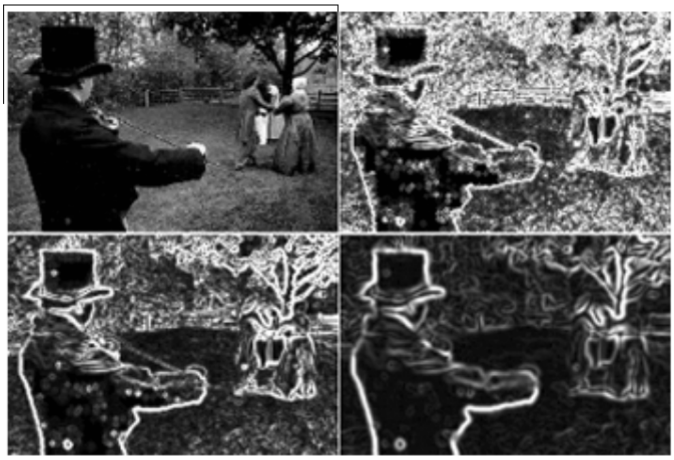

Figure B.2: Top left: original image, then the density of entropy production (σ) plotted after 1, 5, 10 iterations of the diffusion process. Bright points indicate high density of entropy production, [38]. To be exact, the absolute value $|\sigma|$ is plotted, since entropy decreases.

Figure B.3: Activity of the image in figure B.2, obtained after 20 iteration (anisotropic diffusion). Left: regions of high activity, right: regions of low activity, [38].

which is nothing but a convolution:

$$f(x, y, t) = f(x, y, t = 0) * K(x, y, t) \tag{B.4}$$

with

$$K(x, y, t) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}} \tag{B.5}$$

whose Fourier Transform is:

$$\mathcal{F}(K) = \frac{1}{2\pi} e^{-t(u^2+v^2)} \tag{B.6}$$

Considering the frequency domain the solution of the diffusion equation (B.2) can be seen as a low pass filtering of the initial condition (the original image) and therefore it is adequate to model fine to coarse transformations by the diffusion equation. Note that high frequencies, in context of image processing denote regions with high contrast and details, whereas low frequencies represent uniform brightness areas with low information content. Figure B.4 shows different plots of the Fourier Transform of $K(x, y, t)$ at different times (transformation steps).

If we now use the definition of Shannon's entropy:

$$H(f) = -\int_0^{y_{max}} \int_0^{x_{max}} f(x, y, t) \ln(f(x, y, t)) dx dy \tag{B.7}$$

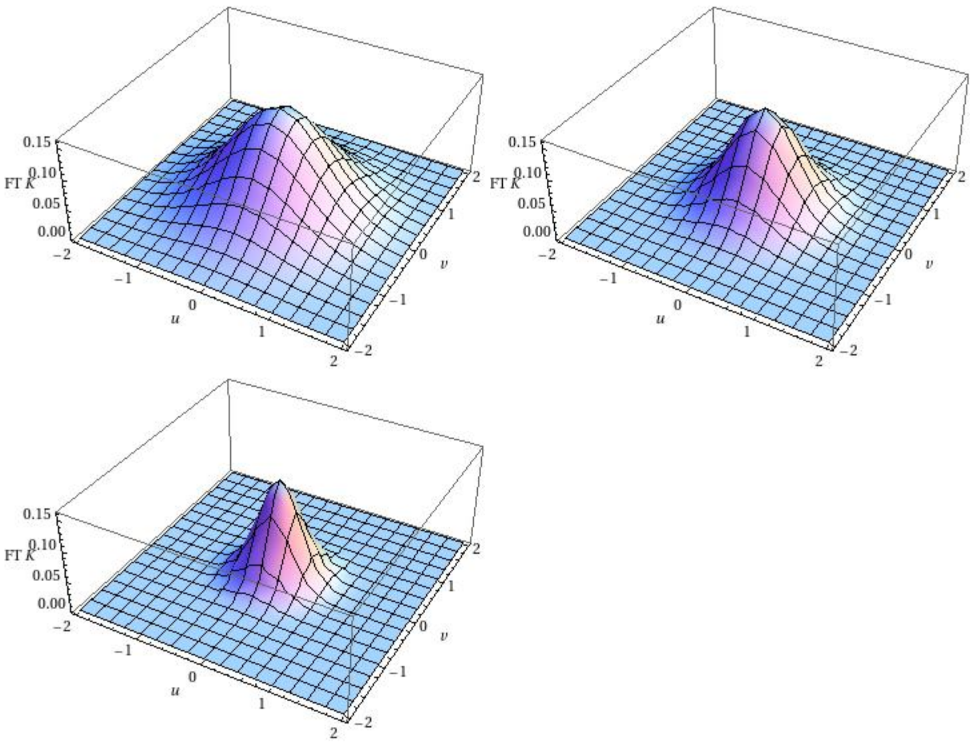

Figure B.4: Fourier Transform of $K(x, y, t)$, plotted from left to right at t=1, t=2 and t=5. Note that negative frequencies do not have to be considered in this content ($K(x, y, t)$ is real valued).

it can be shown, [39], that

$$\frac{\partial H}{\partial t} = \int_0^{y_{max}} \int_0^{x_{max}} f(x, y, t)\sigma(x, y, t)dxdy \tag{B.8}$$

with the density of entropy production

$$\sigma(x, y, t) = \frac{\nabla f(x, y, t) * \nabla f(x, y, t)}{f(x, y, t)^2} \tag{B.9}$$

and also that $\frac{\partial H}{\partial t}$ is really the change of relevant information contained in an image. The term density of entropy production comes from a comparison of equation B.8 to the entropy change in a thermodynamical sense.
Finally activity is defined as

$$a(x, y) = \int_0^N f(x, y, t)\sigma(x, y, t)dt = \int_0^N \frac{\nabla f(x, y, t) * \nabla f(x, y, t)}{f(x, y, t)}dt \tag{B.10}$$

Using activity, image regions can be distinguished and divided into regions with low, medium and high activity and therefore the information content of a certain region can be quantified. Furthermore this algorithm can easily be extended to color images, [38] and can even consider the interactions between the different color channels.
Luckily, the function $K(x, y, t)$ is rapidly decaying, but still, [38] states, that at least $N = 10$ to $N = 20$ iterations of the diffusion equation must be calculated. For calculating $f(x, y, t)$ one can apply two methods: Either equation B.2 is discretized

and solved iteratively or solved in the frequency domain (B.4). The latter approach is much preferable, because in that case the integrand of the activity could be directly calculated in the frequency domain:

$$\frac{\nabla f(x,y,t) * \nabla f(x,y,t)}{f(x,y,t)} \xrightarrow{\text{FT}} -\mathcal{F}(f(x,y,t))(u^2+v^2) \tag{B.11}$$

By considering equation B.4 and B.6 the integrand of the activity can be calculated in the frequency domain by:

$$\frac{\nabla f(x,y,t) * \nabla f(x,y,t)}{f(x,y,t)} \xrightarrow{\text{FT}} -\frac{\mathcal{F}(f(x,y,t=0))}{2\pi} e^{-t(u^2+v^2)}(u^2+v^2) \tag{B.12}$$

Since integration is a linear operation, the inverse Fourier transform and the integration over dt can be interchanged. Therefore, the activity can be rewritten in a compact form as:

$$a(x,y) = \mathcal{F}^{-1}(-\int_0^N \frac{\mathcal{F}(f(x,y,t=0))}{2\pi} e^{-t(u^2+v^2)}(u^2+v^2)dt) \tag{B.13}$$

Furthermore the integration over dt can be carried out:

$$a(x,y) = \mathcal{F}^{-1}(-\frac{\mathcal{F}(f(x,y,t=0))}{2\pi}(1-e^{-N(u^2+v^2)})) \tag{B.14}$$

Equation B.14 is nothing but a high pass filtering of the initial image, which is somehow a very intuitive result. Considering the limits, $N \to 0,\ a(x,y) = 0,\ \forall x, y$ and for $N \to \infty$ only the DC component is filtered out, which is again intuitively clear. Figure B.5 shows the high pass filtering, by function $1 - e^{-N(u^2+v^2)}$ for different N.

B.2.1 Implementation and Results

By means of the fast Fourier transform, equation B.14 can be evaluated quite fast. But still, the 2D fast Fourier transform algorithm for a $N \times N$ matrix has complexity $\mathcal{O}(N^2 log(N))$. Since the computation of the activity requires a 2D Fourier transform, a element wise $N \times N$ multiplication in the frequency domain and an inverse 2D Fourier transform, the overall complexity is still $\mathcal{O}(N^2 log(N))$.

As a first verification the Algorithm was implemented in Matlab (See C). The code was tested on the images, shown in figures B.6, B.7, B.8. To reduce noise artefacts, the contour plot of the activity is displayed on a logarithmic scale. The parameter N was chosen arbitrarily $N = 20$ and it may be possible to achieve better results with a different N. Generally speaking, a smaller N leads to a more “agressive” filtering (more iterations of the diffusion process), while a higher N filters more “gently”, (less iterations of the diffusion process).
These figures show that activity is actually a really powerful mean of feature detection and classification of image regions according to their information content. However, the main drawback is the complexity. For example in VSLAM applications, feature detection must be performed very fast and in this case the implementation of this algorithm must improved.

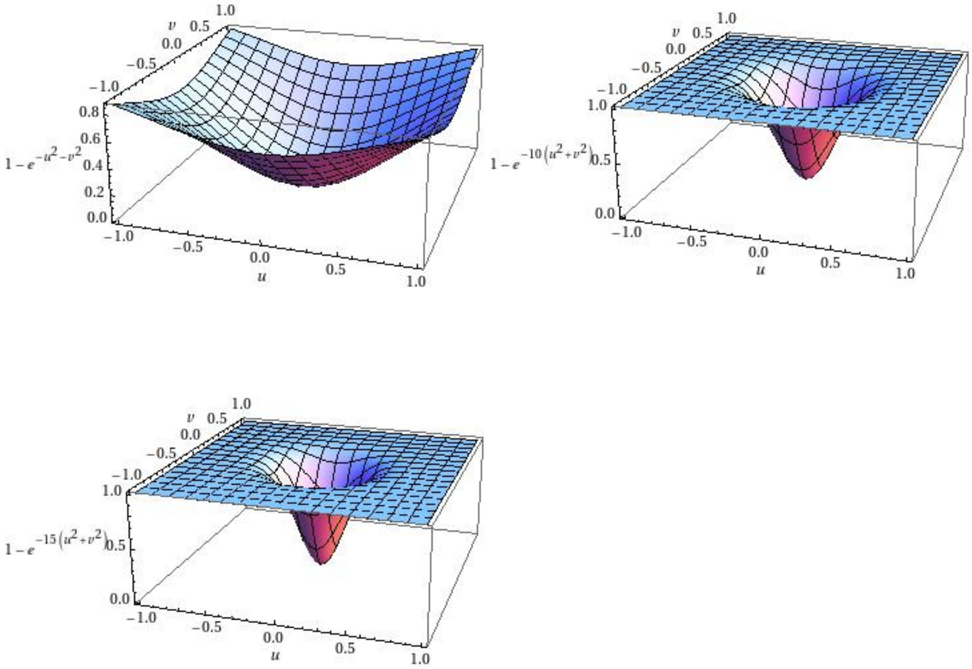

Figure B.5: Function $1 - e^{-N(u^2+v^2)}$ is plotted from left to right for N=1, N=10, N=15.

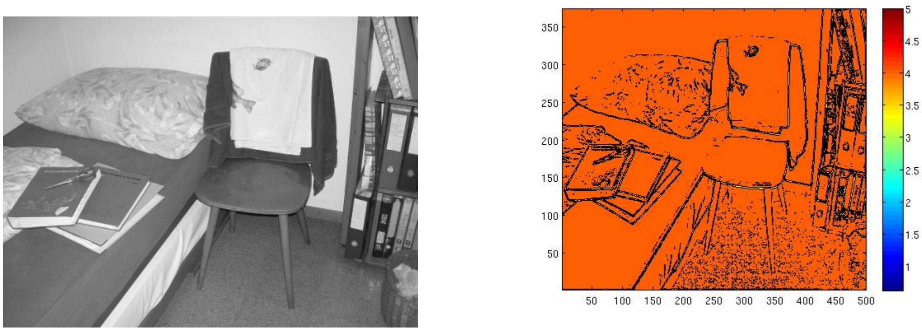

Figure B.6: Image with corresponding activity contour plot (displayed on a logarithmic scale).

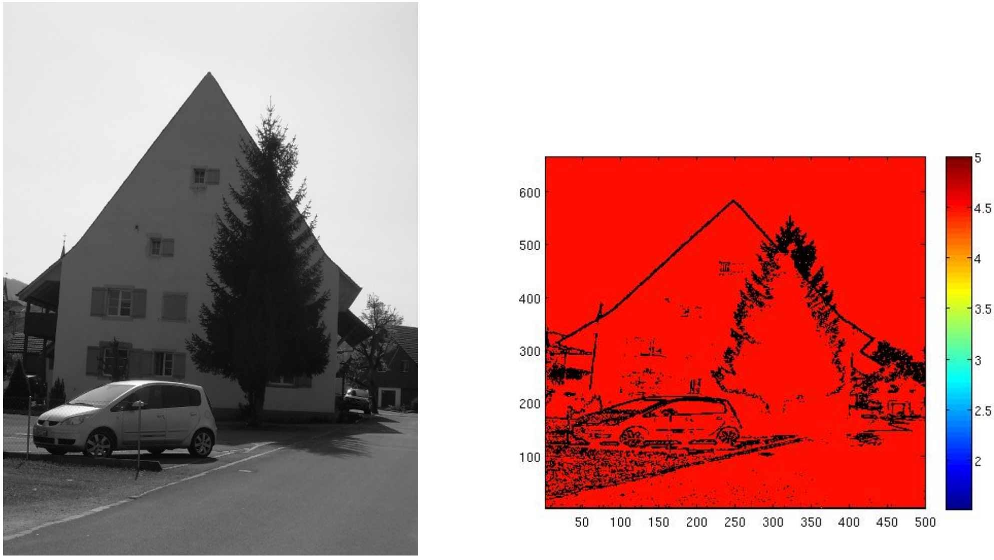

Figure B.7: Image with corresponding activity contour plot (displayed on a logarithmic scale).

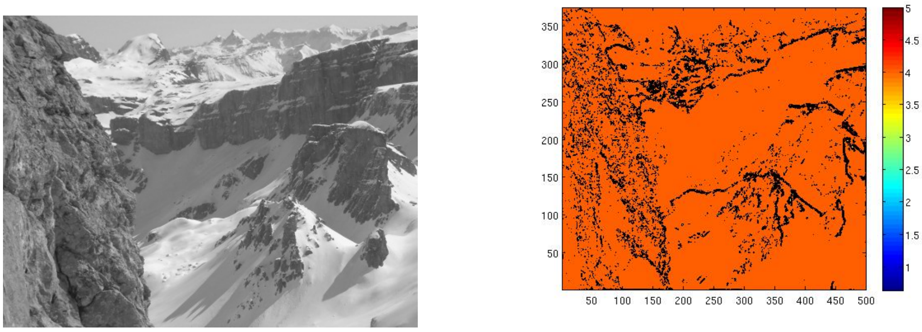

Figure B.8: Image with corresponding activity contour plot (displayed on a logarithmic scale).

Appendix C

Source Code of Activity.m

```
% This function calculates the activity for
% every pixel of the image 'image_name' and
% returns it stored in act.
% mmin, represents the minimum value of act
% mmax, represents the maximum value of act
% Furthermore, the time for computation of
% the activity is displayed.
%
% Written by Michael Muehlebach,
% December 2010

function [act,mmin,mmax]=Activity(image_name)

%load image
im=imread(image_name);

%maximum time (number of iterations)
T=20;

%convert image to grayscale
gs=rgb2gray(im);

%size of the image
N=length(gs(1,:));
M=length(gs(:,1));

N2l=ceil(N/2);
N2u=ceil((N+1)/2);
M2l=ceil(M/2);
M2u=ceil((M+1)/2);

%calculate filtering function, which
%is just dependent on M, N, T ...
%(for a real implementation this
%matrix could be stored in advance)

%calculation of u^2
u=[((0:1:(N-1))-N/2).^2];
up2=ones(M,1)*u;

%calculation of v^2
v=[((0:1:(M-1))-M/2).^2]';
vp2=v*ones(1,N);

%coefficient
M_r=(ones(M,N)-exp(-T*(up2/N^2+vp2/M^2)));
```

```
%begin time measurement...
tic

%DFT
ft=fft2(gs);

%rearrangement of fourier coeff:
ftr=zeros(M,N);
ftr(:,N2u:end)=ft(:,1:N2l);
ftr(:,1:N2u-1)=ft(:,N2l+1:end);

ft(M2u:end,:)=ftr(1:M2l,:);
ft(1:M2u-1,:)=ftr(M2l+1:end,:);

%result in the frequency domain
ftr=ft.*M_r;

%rearrangement of fourier coeff
ft(M2l+1:end,:)=ftr(1:M2u-1,:);
ft(1:M2l,:)=ftr(M2u:end,:);

ftr(:,N2l+1:end)=ft(:,1:N2u-1);
ftr(:,1:N2l)=ft(:,N2u:end);

%IDFT
act=real(ifft2(ftr));

%end time measurement...
toc

%calculating maximum, minimum activity
mmin=min(min(act));
mmax=max(max(act));
```

Bibliography

[1] S. Schulz, M. Grimm, R. Grigat: *Using Brightness Histogramm to perform Optimum Auto Exposure*. In WSEAS Transactions on System and Control, February 2007, Issue 2, Volume 2.

[2] JiaYi Liang, YaJie Qin, ZhiLiang Hong: *An Auto-exposure Algorithm for Detecting High Contrast Lighting Conditions*. In IEEE, 2007.

[3] Myung Hee Cho, Soek Gun Lee, Byung Deok Nam: *The fast Auto Exposure Algorithm based on the numerical analysis*. In SPIE Conference on Sensors, Cameras, and Applications for Digital Photography, 1999.

[4] `http://en.wikipedia.org/wiki/Charge-coupled_device`.

[5] Dr. N. Blanc: *CCD vs. CMOS image sensors*.

[6] Dave Litwiller: *CCD vs. CMOS: Facts and Fiction*. In PHOTONICS SPECTRA, January 2001.

[7] Arnaud Darmont: *Methods to extend the dynamic range of snapshot active pixel sensors*. In SPIE, 2008.

[8] Abbas El Gamal: *High Dynamic Range Image Sensors*. Stanford University. `isl.stanford.edu/~abbas/group/papers_and_pub/isscc02_tutorial.pdf`

[9] K. Takada, S. Miyatake: *Logarithmic converting CCD Line sensor and its noise characteristics*. In Proc. IEEE Workshop on CCD, 1999.

[10] Angelika Erhardt: *Einfuehrung in die Digitale Bildverarbeitung: Grundlagen, Systeme und Anwendungen*. Vieweg+Teubner. Seite 42.

[11] Sam Kavusi, Abbas El Gamal: *Quantitative Study of High Dynamic Range Image Sensor Architectures*. Stanford University, 2004.

[12] Orly Yadid-Pecht: *Wide-dynamic-range sensors*. In Society of Photo-Optical Instrumentation Engineers.

[13] K. Takada, S. Miyatake *Logarithmic-Converting CCD Line Sensor*. In IEEE Workshop on CCDs and Advanced Image Sensors, 1997.

[14] : *Adaptive photoreceptor with wide dynamic range*. In IEEE Intl. Symp. on Circuits and Systems, 1994.

[15] M. Sayag: *Non-linear photosite response in CCD imagers*. US Patent No. 5055667, 1984.

[16] S. Decker, R. McGrath, K. Brehmer, C. Sodini: *A 256x256 CMOS imaging arraywith wide dynamic range pixels*. In Proc ISSCC, 1998.

[17] Tetsuya Kuno, Hiroaki Sugiura: *A new automatic exposure system for digital still cameras.* In IEEE Trans. on Consumer Electronics, Vol. 44, No. 1, February 1998.

[18] Jia Yi Liang, YaJie Qin, ZhiLiang Hong: *An Auto-exposure Algorithm for Detecting High Contrast Lighting Conditions.* In IEEE, 2007.

[19] June-Sok Lee,You-Young Jung, Byung-Soo Kim, Sung-Jea Ko: *An advanced video camera system with robust AF, AE and AWB.* In IEEE Trans. on Consumer Electronics, Vol. 47, No. 3, August 2001.

[20] Toshinobu Haruki, Kenichi Kikuchi: *Video camera system using fuzzy logic.* In IEEE Trans. on Consumer Electronics, 1992.

[21] Quoc Kien Vuong, Se-Hwan Yun, Suki Kim: *A New Auto Exposure System to Detect High Dynamic Range Conditions Using CMOS Technology.* In IEEE, 2008.

[22] *http://en.wikipedia.org/wiki/Luminance*

[23] *http://www.cambridgeincolour.com/tutorials/histograms2.htm*

[24] Haitao Yang, Yilin Chang, Jing Eang, Junyan Huo: *A new automatic exposure algorithm for video cameras using luminance histogram.* In Front. Optoelectron. China 2008.

[25] Simon Schultz, Marcus Grimm, Rolf-Rainer Grigat: *Using Brightness Histogram to perform Optimum Auto Exposure.* In WSEAS Transactions on System and Control, February 2007.

[26] Guy J Brown: *High Dynamic Range Digital Photography.* In RPS Journal, 2006.

[27] S. Battiato, A. Castorina, M. Mancuso: *High dynamic range imaging for digital still camera: an overview.* In Journal of Electronic Imaging, July 2003.

[28] Mark A. Robertson, Sean Borman, Robert L. Stevenson: *Dynamic range improvement through multiple exposures.* In IEEE, 1999.

[29] Yongjie Piao, Wei Xu: *Method of Auto Multi-Exposure for High Dynamic Range Imaging.* In IEEE, 2010.

[30] Paul E Debevec, Jitendra Malik: *Recovering high dynamic range radiance maps from photographs.* In Proc. ACM SIGGRAPH, 1997.

[31] Wen-Chung Kao, Chien-Chih Hsu, Chih-Chung Kao, Shou-Hung Chen: *Adaptive Exposure Control and Real-Time Image Fusion for Surveillance Systems.* In IEEE, 2006.

[32] Quoc Kien Vuong, Se-Hwan Yun, Suki Kim: *A New Auto Exposure System to Detect High Dynamic Range Conditions Using CMOS Technology.* In IEEE, 2008.

[33] Suji Shimizu, Toshiharu Kondo, Takeshi Kohashi, Masaaki Tsuruta, Teruyoshi Komuro: *A new Algorithm for Exposure Control based on Fuzzy logic for Video Cameras.* In IEEE Transactions on Consumer Electronics, 1992.

[34] Shengdong Pan, Xiangjing An: *Content-based Auto Exposure Control for On-board CMOS Camera.* In IEEE, 2008.

[35] Zhang Chi, You Zheng, Yu Shijie: *An Automatic Exposure Algorithm Based on Information Entropy*. In Proc. of SPIE, 2006.

[36] C. E. Shannon: *A mathematical theory of communication*. In The Bell System Technical Journal, 1948.

[37] Parvez Ahammad, Konstantinos Daskalakis, Omid Etesami, Andrea Frome: *Claude Shannon and "A Mathematical Theory of Communication"*. In `www.essrl.wustl.edu/~jao/itrg/shannon.pdf`.

[38] Mario Ferraro, Giuseppe Boccignone, Terry Caelli: *Entropy-based representation of image information*. In Pattern Recognition Letters 23, 2003.

[39] Mario Ferraro, Giuseppe Boccignone, Terry Caelli: *On the representaion of image structures viea scale-space entropy conditions*. In IEEE Trans. on Pattern Analysis, 1998.

[40] G. Felder: *Partielle Differentialgleichungen fuer Ingenieurinnen und Ingenieure*. In `http://www.math.ethz.ch/u/felder/Teaching/PDG`.

[41] Q. R. Razlighi, N. Kehtarnavaz: *Correction of Over-Exposed Images Captured by Cell-Phone Cameras*. In IEEE, 2006.

[42] Huimin Lu, Hui Zhang, Shaowu Yang, Zhiqiang Zheng: *Camera Parameters Auto-Adjusting Technique for Robust Robot Vision*. In IEEE International Conference on Robotics and Automation, 2010.

[43] *Micron MT9V022I77ATM, 1/3 Inch, Wide-VGA CMOS Digital Image Sensor*. Micron Technology, 2006