Master Thesis

Quality Monitoring of Machine Vision Systems and Defect Detection in Polymer Products

carried out for the purpose of obtaining the degree of

Diplom-Ingenieur

submitted at TU Wien,
Institute of Chemical Technologies and Analytics

under the supervision of
Ao.Univ.Prof. Mag. Dr. Johann Lohninger

by

Michael Stibi

Vienna, April 2019

Michael Stibi

I confirm, that going to press of this thesis needs the confirmation of the examination committee.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

Vienna, April 2019 Michael Stibi

Abstract

Modern fabrication facilities refine methods for automated and rapid production of goods. Due to soaring velocities, visual inspection grows to be impractical for human workers. Machine vision systems have grown to be powerful tools for automated quality control of products. However, current technologies are limited to monitoring simple objects and to the application of proven algorithms. This thesis focuses on the detection of defects in two different polymer objects with challenging surfaces. Circular, transparent lids and opaque, highly reflective des are the subjects of investigation. For both problems at hand, stable setups for measurement are presented. Additionally, innovative algorithms to find defects are introduced. The first is attracting considerable interest due to its rotation-invariant and statistical approach. The second stands out by reason of using a colour space transformation and surface modelling before the utilization of well-proven thresholding. Furthermore, research on intrinsic quality monitoring of machine vision systems provides knowledge for fast and simple diagnosis of possible malfunctions. A one-fits-it-all template and several proposed algorithms account for a potential remote detection of aforementioned malfunctions.

Kurzfassung

Moderne Fertigungsstätten sind darauf bedacht, Methoden für die schnelle und automatisierte Herstellung von Waren, weiterzuentwickeln. Aufgrund immenser Fertigungs- und Laufbandgeschwindigkeiten ist die visuelle Kontrolle durch den Menschen eine immer größer werdende Herausforderung. Maschinelles Sehen hingegen hat sich für die automatische Qualitätskontrolle von Produkten zu einem mächtigen Werkzeug entwickelt. Existierende Technologien beschränken sich jedoch derzeit auf die Überwachung einfacher Objekte und die Anwendung simpler Algorithmen. Diese Diplomarbeit befasst sich mit der Erkennung von Fehlern in zwei verschiedenen Polymer-Objekten mit anspruchsvoller Oberfläche. Zum einen sind runde und transparente Deckel, zum anderen opake und stark reflektierende Teile Gegenstand der Untersuchung. Für beide Probleme wird ein stabiles Messsystem präsentiert. Zusätzlich werden innovative Algorithmen zum Auffinden der vorhandenen Defekte vorgestellt. Der erste Algorithmus besticht durch seinen rotationsinvarianten und statistischen Ansatz, der zweite durch Verwendung einer Farbraumtransformation und Oberflächenmodellierung vor der Anwendung des bewährten Schwellenwertverfahrens. Überdies befasst sich die Studie mit der Qualitätssicherung von Systemen für maschinelles Sehen und bietet Vorschläge für die schnelle und simple Diagnose möglicher Fehlfunktionen. Diese können durch das Anwenden von Algorithmen auf Photographien der entwickelten Schablone dezentral festgestellt werden.

Acknowledgements

First and foremost, I want to thank Hans Lohninger for his guidance in creating this work. Hans introduced me to programming, regularly freed his time for answering my countless questions and has always valued me for who I am. Our long discussions shaped my thinking and let me grow both professionally and personally. Heartfelt thanks to Dieter and Benedikt who made me feel welcome from the very first day and frequently resolved any issues regarding my programming. Furthermore, I want to thank Christoph L., Christoph G. and Stefan for their help on the experimental setup.

Nobody has ever accomplished anything fully without the influence of others. This thesis and my studies at TU Wien are not an exception. Therefore, I want to express my deepest gratitude to:

Julia, Kerstin, Franziska, Tanja, Fabian, Peter, Martin, Patrick and Felix, my oldest friends. My youth and adolescence could not have been any better. They were there in the woods when I was eight and they will be there on the bench when I am eighty.

Mag. Gebert and Mag. Roch-Reisenauer who lastingly shaped my ethical and scientific point of view and somehow directed me towards this fascinating discipline.

All the members of Sportclub Dörfl for the truly intense, joyful and painful moments we shared. The hours on the playground always took my mind off things and strengthened me continually.

My study colleagues Anna, Sophia, Charlie, Joseph, Julian, Max and Tobias for the numerous hours spent studying, discussing and learning from and with each other.

Andreas, Eva, Stefan, Sophia, Michael, George, Isabella, David, Raphael, Clemens, Justin, Sebastian, Chris, Pablo, Elena, Marton and Sandra, friends of a very special and lovely kind.

All artists who created musical masterpieces that kept me going in difficult times and have lifted my spirits even higher when I was already blissful. I am grateful for all the honest concerts and ecstatic memories that I will keep in my mind forever.

All the people I met during a hazy night out. Their ludicrous insanity has always comforted me.

Anton, Mathäus and especially Sasha for reminding me what life should be like, namely a fun ride without limits. Joaquin who is one of the most generous and kindest persons I know. Nils for being intellectually exciting and our rock. Thomas for encouraging me to pursue big dreams. We are going to create greatness together.

The people of Swing City for igniting my passion for Lindy Hop. I will always carry Argentina's shining light of love, chaos and madness in me.

Susanne and her extraordinary soul that is so deeply attached to my heart and mind.

Daniel for the tremendous fun, never-ending discussions and the myriad of adventures during our travels. Konstantin for consistently brightening my mood, his everlasting affection and for always being there. They blessed me both with unique friendship.

Dominik and Hannah, who would do anything to protect me. Anna who lives life according to her own rules, inciting me to do the same. Florian and Tobias, for their shared love for movies and great concerts. Carmen for being the greatest aunt one could wish for.

My parents Ingrid and Hans who gave everything they had so I could be anything I have wanted to be.

For my parents

Table of Content

List of Figures

List of Tables

List of Abbreviations

α	Type I error
β	Type II error
AD	Analogue digital
B&W	Black and White
CCD	Charged coupled device
CMOS	Complementary Metal Oxide Semiconductor
DL	Defected lids
GV	Grey value
H_0	Null hypothesis
H_1	Alternative hypothesis
MV	Machine vision
NDL	Non-defected lids
RGB	Red green blue
HSI	Hue saturation intensity

1. Introduction

In the early 1770s the manufacturers all over Europe began transitioning their production processes towards both time and workload efficient systems using machines that exceed human rate of production. 250 years later engineers have improved their factories to an extent, where manpower is becoming less important, whereas smart control and surveillance of rapid production systems is crucial to a company's production rates and thus success. Producers of packaging materials are assembling at such a high velocity, that quality control of the produced goods is becoming less and less feasible using the human body as a decision maker. Not only is a person's ability to detect defects in packaging product limited due to the rapidness of common conveyor belts, but more importantly one's interpretation of a defect might differ from another one's point of view. Moreover, visually monitoring thousands of parts that pass by at high speed is incredibly challenging and exhausting. Thus, it requires a high level of concentration that might not be maintained by a human worker.

Since manual inspection is prone to mistakes due to a lack of human ability and concentration, product quality cannot be fully assured. Consequently, waste rates may vary, costing unnecessary time and resource expenses. Naturally, to avoid all these factors of inconsistent quality control, producers seek to find automated solutions. A technology that is using an image acquisition system to process and extract information, giving a decision upon well-defined criteria whether an object has to be sorted out or not, is called machine vision (MV) system.

In chapter 2, the fundamentals of MV systems and statistical tests are explained. Furthermore, theory concerning image acquisition and processing is elucidated. This first part of the thesis contains basic understanding of image sensor functionality, representation of colours in a digital environment and image procession.

There is little scientific research in the field of the quality monitoring of the MV system itself. Producing companies however heavily depend on a well-developed quality management system, thus theoretical and experimental research was done to provide solutions on how to monitor MV hardware on its performance. Chapter 3 presents several probable malfunctions of MV systems, as well as the solutions of how to easily and remotely identify their origins.

One main aim of this work was to find defects in two use cases using a self-built lab-scale machine vision system. Chapter 4.1 introduces a unique MV setup and algorithm, that allows the rotation-invariant analysis of circular, transparent polymer lids. The procedure focuses on the statistical analysis and comparison of image data. In chapter 4.2, a possible setup and algorithm for the detection of defects in opaque, reflective parts is presented. The algorithm revolves around the idea of taking advantage of distinct colour space properties.

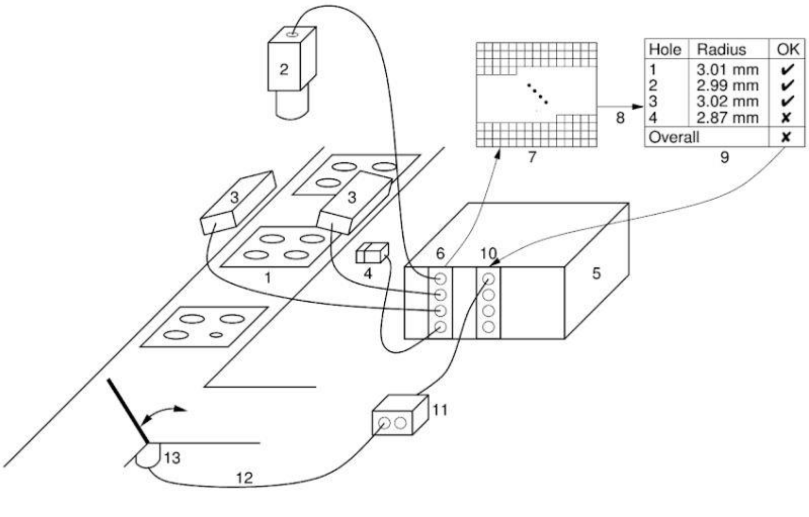
2. Theoretical Background

2.1. Machine Vision Systems

A process in which a machine, e.g. a computer, automatically processes an image and recognizes its content is called machine vision [1, p. 4]. Accordingly, a common MV system consists of the following parts, depicted in Figure 2.1:

- Camera
- Illumination
- Processing unit (computer)
- Sorting unit

The inspection process can be described as follows: parts of interest move along a transportation line or are being put in a certain position by a robot-arm. In the first case, a suitable sensor triggers the image acquisition, in the second case it's triggered when the robot-arm is in a well-defined position. When the image process is triggered, the illumination system sets off a flashlight and the camera opens its shutter, imaging the object of interest. The data of the electronic image is then sent to a central processing unit, where the photograph may be subject to some pre-processing steps, before its content is analysed using an algorithm. Afterwards, the obtained information is used to control a sorting unit, that eventually separates objects from the transportation line, that do not meet the criteria set by the operator.



Hole	Radius	OK
1	3.01 mm	✔
2	2.99 mm	✔
3	3.02 mm	✔
4	2.87 mm	✘
Overall		✘

Figure 2.1: Components of a typical machine vision system [2, p. 2]
1: object of interest; 2: camera; 3: illumination; 4: trigger sensor
5: processing unit; 13: sorting unit

Cameras operate by opening a shutter, exposing their sensors to light. The appearance of objects in the photograph and the image quality in general is defined by the camera's settings, its resolution and the lens' settings. The variation of the ISO value controls the sensitivity of the sensor, while the shutter time defines how much light falls onto the sensor. Object appearance is affected by the aperture, which influences the depth of field. For those, who have never explored the manifold world of photography, a wide variety of open books [3] and photography forums [4] are available to understand the most basic principles of photography and how the camera's settings affect the images taken.

2.2. Image Acquisition and Processing

2.2.1. Introduction

The key component of every machine vision system is the camera. A century ago, images were taken by exposing radiation sensitive materials like silver halide crystals to light. While this analogue technique still fascinates photographers around the globe, the consumer market relies on a faster and more stable method to obtain images.

The main part of a so called "digital" camera is a light-sensitive sensor, that we will call the camera sensor. Whereas analogue cameras collect the light they are exposed to using a chemical reaction, digital cameras convert the light that falls onto their sensor by exploiting the photoelectric effect and then saving the digitalised image on a digital memory. Digitization describes a process, where information, e.g. an analogue signal like the intensity of light, is converted into a computer-readable format [2].

2.2.2. Solid-State Sensors

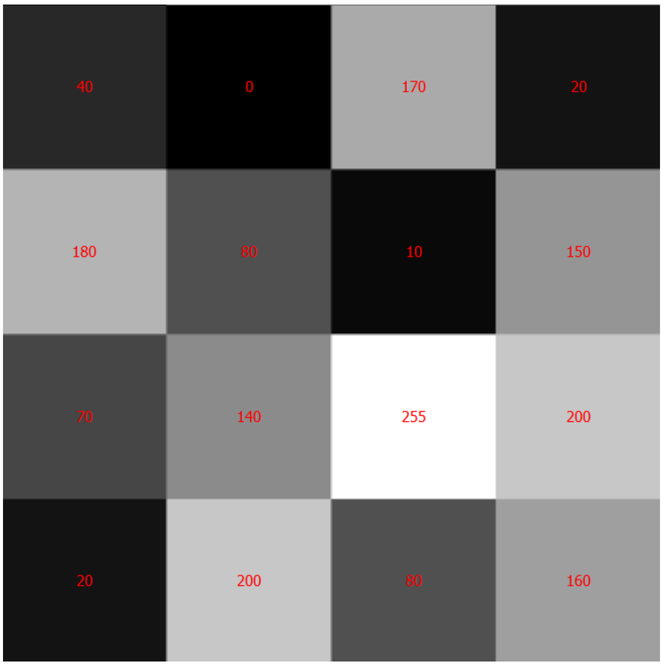
As previously mentioned, modern solid-state sensors take advantage of Einstein's Nobel prize winning discovery of the photoelectric effect. Those sensors are most commonly arranged in a matrix, where every light-sensitive element is called pixel. The intensity of light that falls onto a pixel is converted into a charge or a digital value that is then ready for processing. Image sensors are categorized according to their geometrical arrangement of their pixels, either in line or in a matrix and they are categorized by their manufacturing technology. The most common technologies are Charged Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) sensors [6, p. 204].

CCD sensors have been dominating the market until the late 1990s, but a more cost-effective production and various performance advantages have led to a reversion of the market situation. Generally, all solid-state sensors operate following the same principle [6, p. 205]:

1. Light in the form of photons generates free electrons in the light-sensitive part of the solid-state crystal sensor.
2. The generated electrons are collected over a defined time period, that is the exposure time.
3. The acquired charge is transferred pixel-wise to the converter stage in the CCD sensor.
4. At last, the accumulated charge is converted into a proportional voltage that is subsequently converted into a numerical value by an analogue-digital (AD) converter.

Both sensor types exhibit various advantages and disadvantages due to the complex mechanisms that take place during image acquisition. Detailed information about the sensor's mannerisms can be found in [6]–[8].

The numerical values that were converted by the AD converter are digitally stored in a matrix. The values of the image matrix correspond to the brightness, or more specifically the intensity of light (the number of photons), that fell onto the pixels. An example is given in Figure 2.2.



40	0	170	20
180	80	10	150
70	140	255	200
20	200	80	160

Figure 2.2: Visualization of an image matrix with values from 0 to 255

Each cell of the matrix can have a value between 0 and 255. For every value there is distinct shade of grey, with the whole scale reaching from total black to total white. A higher value indicates a whiter cell, respectively a pixel which was exposed to a higher intensity of light. The range of 256 different grey values was not chosen by accident, this will be discussed in detail in chapter 2.2.4.

2.2.3. Colour Sensors

Especially the mainstream consumer market depends on a technical feature, that provides that users can take images not only in black and white, but in colour. While the brightness of grey level images is easily explained by the physical quantity of intensity and can therefore also be used with ultraviolet or infrared light, colour is inseparably linked to human perception [6, p. 211].

Unlike bats or dolphins who are sorely sensitive to ultrasonic, most organisms use a chemical reaction in their eyes to convert electromagnetic waves into visual information. Humans see, because the brain is able to obtain visual information via the conversion of light-sensitive proteins like rhodopsin. Many organisms however, are unable to distinguish between different wavelengths of the electromagnetic spectrum, only seeing different shades of grey as the example in Figure 2.2. To be able to see colour, organisms have evolved in a way, where they use different kinds of proteins to interact with different wavelengths of light [9].

To the human eye, the visible spectrum, thus the wavelengths of the electromagnetic spectrum that interact with the proteins of the retina, has an approximate range of 380 to 780 nm. These colour-sensitive photo receptor proteins are divided into three different types, each having its own sensitivity for distinct wavelengths. The different receptors are called cones and each cone reacts, according to its name, specifically strong to a certain wavelength [6, p. 211]:

- L-cones, for long-wave, red light
- M-cones, for the middle spectrum
- S-cones, for short-wave, blue light.

As shown in Figure 2.3, the human eye is the most sensitive to light in the range of red and green light.

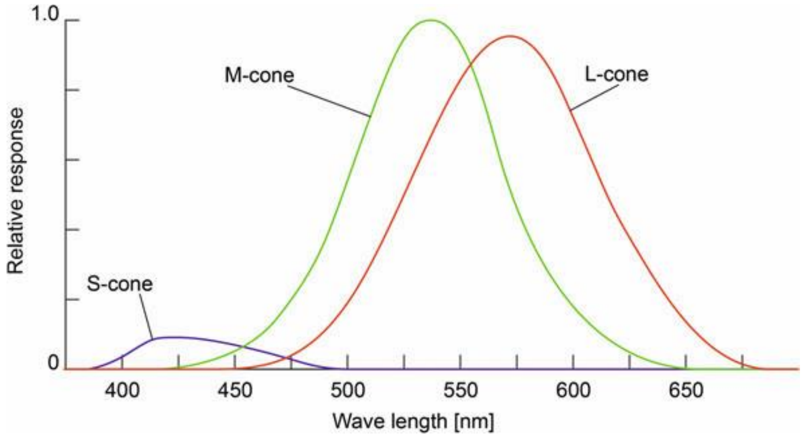


Figure 2.3: Spectral sensitivity of cones in the human eye [6, p. 212]

Since each pixel can only absorb a brightness value that is proportional to its exposure to photon intensity, different methods have been developed to image colour in machines. In general, the two main types of hardware arrangements of cameras that are commonly used to capture colour images with a CCD or a CMOS sensor, are three-chip and one-chip solutions. Three-chip colour cameras use a beam splitter to guide the light according to colour onto one of the three solid-state sensors. A more common approach however, is a one-chip colour camera using a colour filter array in front of the array, that only permits the corresponding colour to reach the defined pixels [6, p. 261]. Such an array is called Bayer filter or Bayer array, named after its developer Bryce E. Bayer, who worked in 1976 at the Eastman Kodak Company in Rochester, NY, USA. The filter array is directly applied to the sensor, making it possible to interpolate the complete information between the pixels, since each pixel receives the brightness of one colour channel [6, p. 212]. Figure 2.4 shows the schematic drawing of such an array in front of the actual camera sensor.

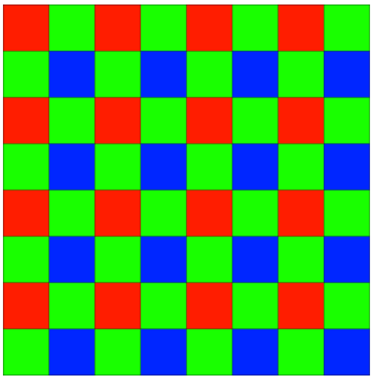


Figure 2.4: Schematic figure of a Bayer pattern (according to [6, p. 213]*)*

In conclusion, coloured pictures veritably consist of three pictures, each representing the intensity of a certain part of the electromagnetic spectrum. To clarify this, the picture of Figure 2.5 was split into its red, green and blue picture (Figure 2.6), each representing the intensity of its corresponding colour in a grey scale picture. Keeping in mind the conventions of hyperspectral imaging, the red, green and blue image will from now on be called red, green and blue layer.

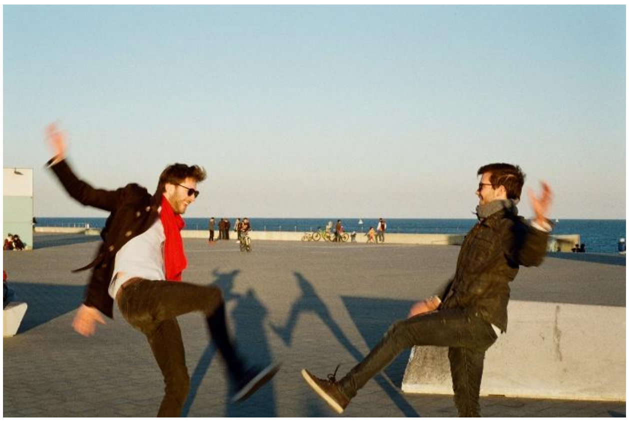


Figure 2.5: Coloured image

When focussing on the red scarf, it is easy to understand, that the left image of Figure 2.6, which is the image obtained by the pixels receiving long-wave red light, exhibit a high intensity of radiation in that area, whereas in the other picture that same area is quite dark, indicating a low intensity of radiation. Furthermore, one can note, that black and white are not colours but rather an equal mixture of each wavelengths, clarified by the fact that the white shirt is equally bright, while the black pants are equally dark in all three pictures.

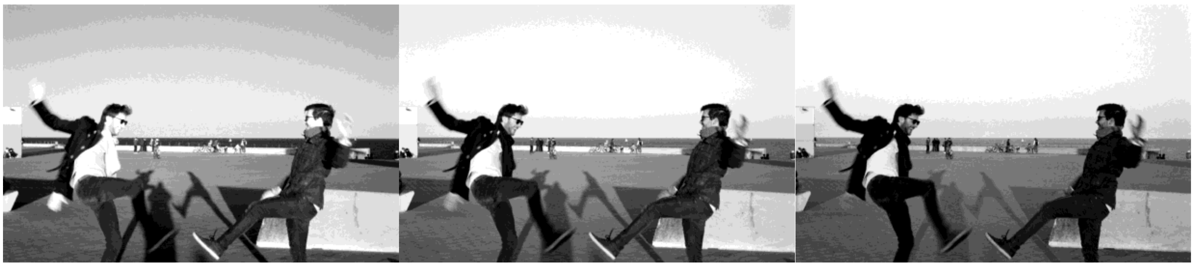


Figure 2.6: Left: red image; middle: green image; right: blue image

2.2.4. Bit Depth

In chapter 2.2.2 and 2.2.3 we learned that the intensity of light is converted and digitally saved as a matrix of values, i.e. an image. As a consequence of the computer's disability to handle analogue values, the obtained light intensity has to be discretized into said image matrix. This leads to the concept of bit depth, which describes the possible grey levels or numeric values a camera sensor can issue. Since computers process information in bits, the same concept is applied to each pixel of an image sensor, where the range of possible values is defined by the combination of bits. By combining several bits, more detailed information can be displayed. Following the mathematical logic of combining zeros and one, shown in Equation 1, the number of possible grey levels is two raised to the power of the bit depth.

$$Range\ of\ grey\ scale = 2^{Bit\ depth} \qquad \text{(Eq. 1)}$$

This is clarified in Table 2.1, which shows the range of possible values depending on the bit depth of the camera sensor.

Table 2.1: Bit depth and corresponding grey scale

Bit Depth	1	2	3	4	5	8	12
Possible Grey Values	2	4	8	16	32	256	4096

Naturally, a greater bit depth corresponds to more information, i.e. better image quality. An example is given in Figure 2.7 and Figure 2.8, where the image matrix of a red image was discretized to the number of possible values according to the related bit depth. Figure 2.7 shows bit depths of 1,2 and 3. With a bit depth of 3 and therefore 8 possible grey values, it is already possible to adequately image complex structures like faces and transparent objects like drinking glasses.

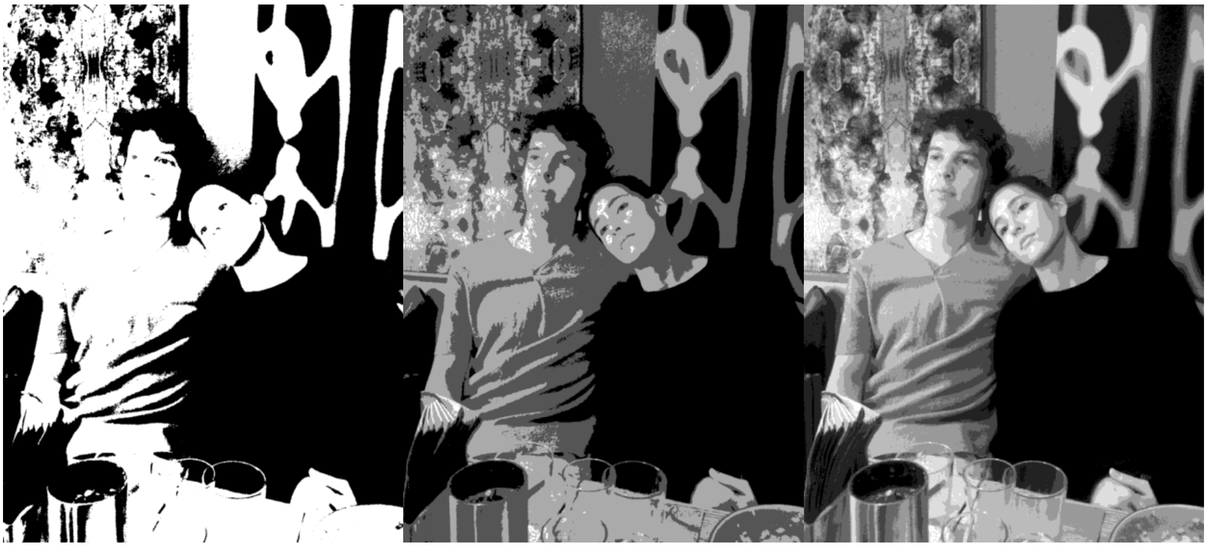


Figure 2.7: Left: bit depth of 1; middle: bit depth of 2; right: bit depth of 3

Figure 2.8 gives a hint, that even with a camera bit depth of 5 and therefore 32 possible values, great image quality is achieved. Most consumer camera sensors however use a minimum bit depth of 8 to accurately image fine texture or colour differences. This can be observed when one closely examines the girl's pullover and its pleats. The importance of bit depth becomes greater, when objects similar in colour or brightness are imaged.

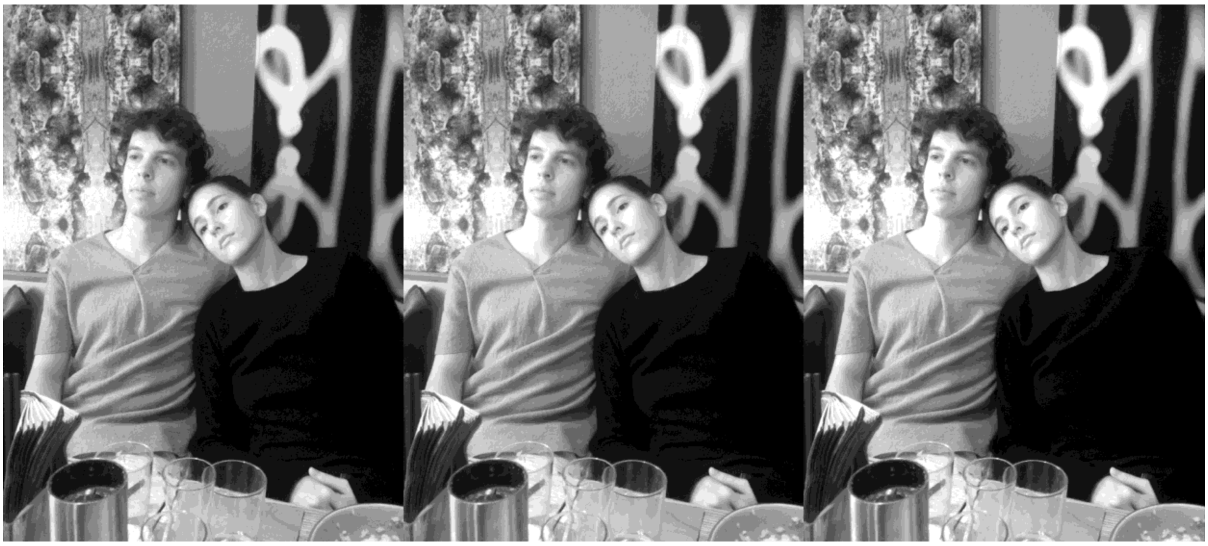


Figure 2.8: Left: bit depth of 4; middle: bit depth of 5; right: bit depth of 8

2.2.5. RGB and HSI Colour Space

Earlier, the concept of colour filter arrays was introduced to understand how coloured pictures are obtained with solid-state sensor cameras. Hence, by exploiting this knowledge one can conclude, that by combining the colours red, green and blue to different extents, it is possible to display the colours we see in reality quite accurately (see Figure 2.9; left).

Using the RGB system ensures that there is no distortion introduced to the initial information. However, one also must be aware of disadvantages. First, representing colours using a mixture of red, green and blue is far from the human concept of colour. Due to the mixing of only three components, colour features are highly correlated and evaluating the similarity of two colours from their distance in the RGB space becomes impossible [10]. Thus, RGB colour space system is highly appreciated for hardware-oriented applications, for colour processing however, the HSI colour space offers various advantages. HSI is based on how humans perceive colour and thus more suitable for describing and interpreting colours. H, S and I represent hue, saturation and intensity, respectively [11].

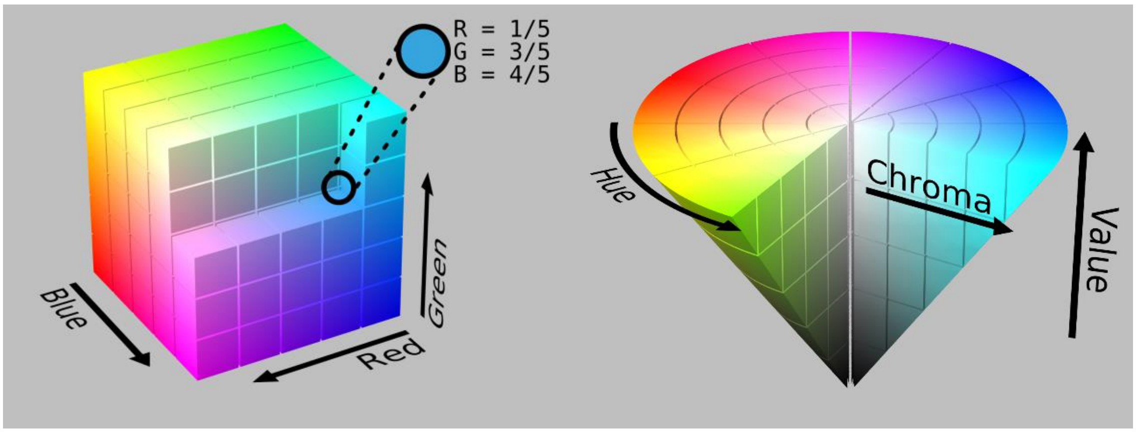


Figure 2.9: Left: RGB cube [12]*; right: HSI cone* [13]

In the HSI system, Hue represents the colour feature, having the property of being relatively unaffected by shadow caused by the light source. In Figure 2.9 (right), hue shows the localization of basic colours when stepping in rotational direction of the colour space. The purity of hue is measured in saturation, i.e. the mixing of pure white to the hue component, resulting in a lower saturation. In Figure 2.9 (right) chroma indicates the saturation. The intensity dimension, or value in Figure 2.9 (right), describes the mixing of the colour with black or white [10].

There are several different ways to perform RGB to HSI conversion, an example [10] is given below. First, the RGB values are converted into the YC_1C_2 tristimulus values (Equation 2).

$$\begin{bmatrix} y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{(Eq. 2)}$$

Then, the tristimulus values are transformed into HSI coordinates by means of Equations 3-5.

$$I = Y \qquad \text{(Eq. 3)}$$

$$S = \sqrt{c_1^2 + c_2^2} \qquad \text{(Eq. 4)}$$

$$\begin{aligned} &If\ c_2 \geq 0\ then\ H = cos^{-1}\left(\frac{c_2}{S}\right) \\ &Else \qquad H = 2\pi - cos^{-1}\left(\frac{c_2}{S}\right) \end{aligned} \qquad \text{(Eq. 5)}$$

2.2.6. Intensity Distribution

The image, which is now regarded as an image matrix, with each cell representing a light intensity of the corresponding pixels, can be mathematically analysed by examining all the occurring values. When the number of every observed intensity from 0 to 255 is drawn into a diagram, a distribution of intensities is obtained. This can be done for every layer of a coloured image. Figure 2.10 (top left) shows a coloured sample image, Figure 2.10 (top middle) the green layer and Figure 2.10 (top right) the blue layer of said image.

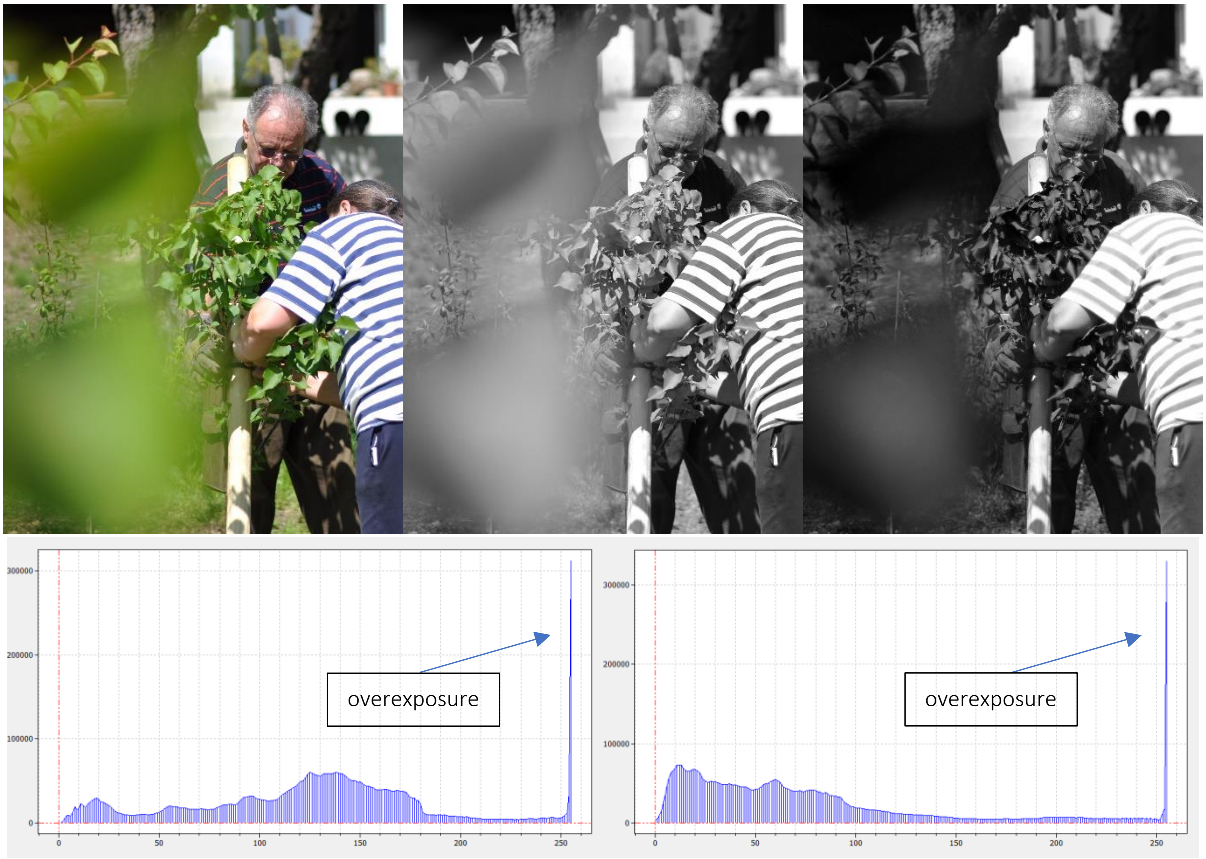


Figure 2.10: Top left: coloured sample image for intensity distribution; top middle: green image; top right: blue image; bottom left: intensity distribution of green image; bottom right: intensity distribution of blue image

Figure 2.10 (bottom left and right) shows the intensity distribution analysis of both green and blue image layer. When comparing both distributions, the intensity distribution of the green layer reflects what we see in the colour picture, namely a higher number of high intensities. This is not surprising, as the coloured image exhibits a substantial number of green pixels. Moreover, the high number of intensities around 255 indicate that many pixels where overexposed to light, i.e. received too much light so that their returning value was the maximum intensity. Usually, such a scenario is avoided in photography, due to the difficulty of post-processing an overexposed image.

2.2.7. Thresholding

In classical defect detection algorithms of machine vision systems, an image is converted into a black and white picture with no grey scale values in between. Such an image would then be stacked over another image and afterwards compared if all black and white regions match.

The process of converting an image into a B&W picture using a threshold is a perfect example to understand how thresholding or so-called binary segmentation works. The most frequently used method to obtain a B&W picture is by defining a single value as a threshold and comparing if the intensity of a pixel lies above or beneath said threshold. There are global and local threshold methods, with local thresholds usually being more efficient [6, p. 84]. An example is given in Figure 2.11, where the fact that the green leaf lacks blue light intensity was used to separate it from the background.

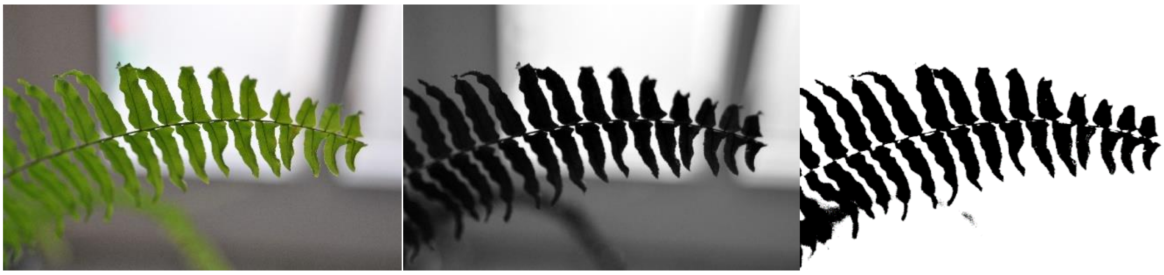


Figure 2.11: Binary image segmentation using a threshold; left: original image; middle: blue layer; right: segmented image

Thresholds can also be seen as a limit value of the histogram, i.e. the intensity distribution (e.g. in Figure 2.10 (bottom left and right)), where intensities above or below the threshold signalize a difference to the other pixels. When smart image post-processing algorithms have made a defect visible, because of its difference in intensity to surrounding pixels, that knowledge can be exploited by thresholding the image, marking a defected area as white, while non-defected areas remain black, which is the same procedure as in Figure 2.11, the only difference being that the defect lights up in such a segmented image.

2.2.8. Masking Pixels

Another key element for the defect detection algorithms used in this work, was the application of pixel masks. Pixel masks are Boolean matrices where each cell is filled with a *true* or *false* value. When the image matrix of Figure 2.11 (right) is processed via thresholding and every black pixel inhibits the information *true* and every white pixel the information *false,* the Boolean image matrix can later be used to hide and omit either the *true* or the *false* pixels from post-processing. From now on, such a matrix will be called *Pixel Mask*, because of its property to mask and thus leave out certain pixels from processing using an algorithm.

2.2.9. Linear Filters

This chapter presents condensed knowledge about the principle of linear filters and their application in image processing. Jähne [14] and Demant [6] provide detailed information on formula and theory, here however the main focus lies on understanding the basic principle of matrix operations in images.

Linear filters are local image operations, where the grey value (GV) of a pixel is determined by a group of pixels in the source image [6, p. 39]. That means, that the GV of a local group of pixels, called *neighbourhood,* are subject to a mathematical operation with the filter kernel, depicted in Figure 2.12.

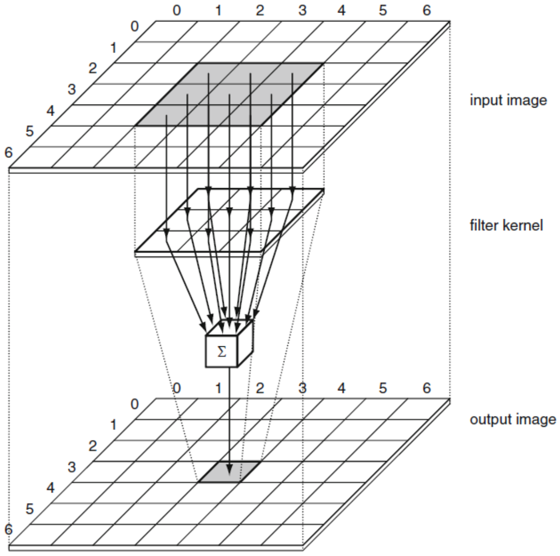


Figure 2.12: Principle of a linear filter [6, p. 41]

When the kernel *h* is applied to a 3x3 matrix of GVs, each source GV is multiplied with the weight of 1, then all GVs are summed up and divided by 9.

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The new grey value, which is in fact just an average of the 9 source GVs, is written in a defined cell of a new image matrix. This operation is repeated for all pixels of the image. Consequently, such a filter is a very basic smoothing filter. Figure 2.13 gives an example of a smoothing filter.

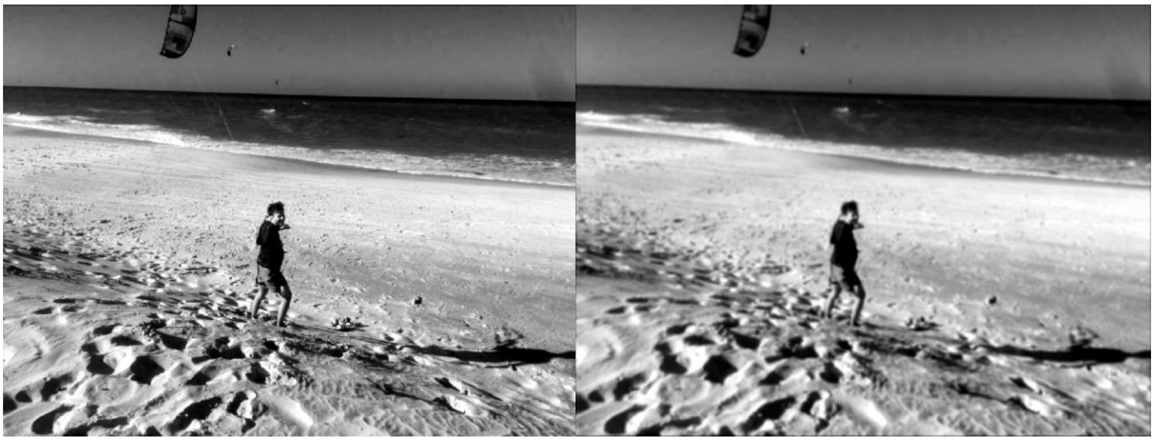


Figure 2.13: Smoothing filter; left: original image; right: smoothed image

Depending on the filter weights, very different effects can be achieved. There are smoothing, edge and median filters, as well as morphological and non-linear filters [6]. One filter that will be used in this thesis is the Sobel filter. This classical edge filter successively applies the two kernels h_x and h_y to the source image.

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The application of the filter emphasizes edges in both x- and y-direction, highlighted in Figure 2.14. Clearly, areas of similar intensity remain dark in the processed image, while areas with steep intensity gradients are highlighted, visible on the edges of the scarf or the hood. Further information on edge detection was summarized by Gao [15].

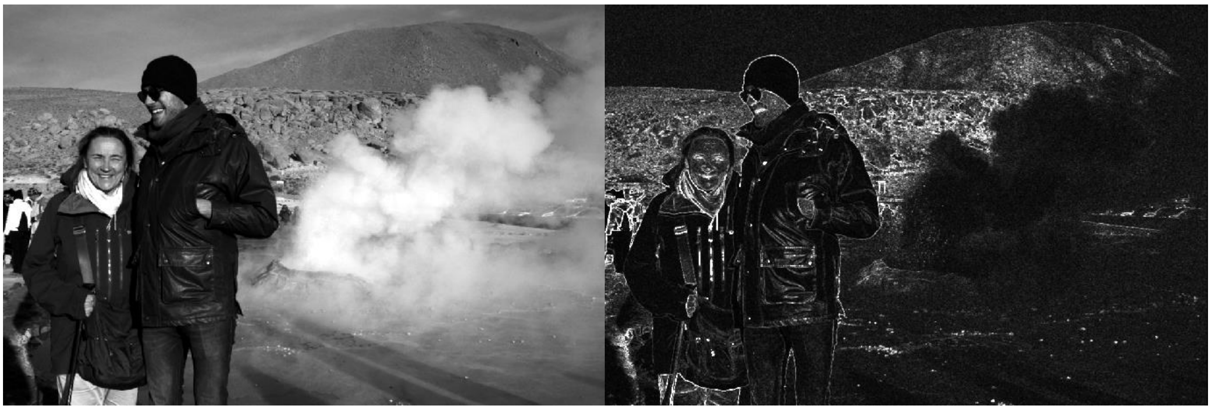


Figure 2.14: Sobel filter; left: original image; right: processed image

In the same way, a spatial percentile filter can be applied to the image matrix. It also serves as a smoothing filter, but in a different way. Here, a user-defined percentile of the *neighbourhood* pixels is determined. Accordingly, the processed image will be free of extreme values.

2.3. Statistical Tests

2.3.1. Introduction

Many machine vision algorithms for industrial applications are kept simple for reasons of applicability and robustness. There is a myriad of different approaches to MV related problems, all having in common to perform some kind of pixel processing. The most classical method is thresholding, which was previously explained in chapter 2.2.7. Template matching is classically used to check the shape of objects. There, two B&W images are directly compared by stacking them onto each other and calculating if there is a difference in area. Finally edge detection is the method, where various edge filters are applied and the resulting image is again compared to a template image [1], [2], [6], [14].

A rather unconventional method of defect detection, however, would be using a statistical test to compare the intensity distribution of a defected object with the intensity distribution of a non-defected object. Such a statistical test that is commonly used to compare distributions, is the Chi Square Test. This chapter focusses on the theoretical background of this procedure.

In defect detection algorithms, there is an utter need for a systematic decision process that takes a variation of one sample to another into account. The decision process has to be able to reflect the risk of making incorrect decisions [16, p. 189].

2.3.2. Hypothesis Testing

The formal procedure of applying statistical concepts and measures for decision making is called hypothesis testing. The following steps are used to perform a statistical analysis of a hypothesis [16, p. 209]:

1. Formulate two hypotheses.
2. Select the appropriate statistical model that identifies the test statistic.
3. Specify the level of significance, which is a measure of risk.
4. Collect a sample of data and compute an estimate of the test statistic.
5. Define the region of rejection for the test statistic.
6. Select the appropriate hypothesis.

Ayyub et al. [16, pp. 210–214] recapitulate the six steps of hypothesis testing in detail, the following paragraphs condense their explanations:

Step 1: Formulation of Hypotheses

The first step of every statistical test is the formulation of two hypotheses for testing. Hypotheses are statements that indicate that a parameter has some specific value or in our case, the random variables have some specific distribution. The hypotheses should be formulated to indicate the absence or presence of differences.

The first hypothesis is called *null hypothesis*, denoted by H_0. It is formulated in a way, where it declares equality, i.e. a difference does not exist. The second hypothesis is called *alternative hypothesis*, denoted as H_1 or H_A. It declares inequality, i.e. a difference does exist. Both hypotheses should be expressed both grammatically and in mathematical terms and more importantly, they should mutually exclude each other in their conditions, as shown in Equation 6 and Equation 7.

$$H_0 \cup \, H_1 = \{P\} \qquad \text{(Eq. 6)}$$

$$H_0 \cap \, H_0 = \{\} \qquad \text{(Eq. 7)}$$

Thus, when H_0 is rejected, H_A must be accepted. Some users may seek that their conducted experiments lead to a difference, expressing H_0 in a way where it suggests an existing difference. This should be avoided, the null hypothesis must be a statement of equality, not inequality.

Step 2: The Test Statistic and Its Sampling Distribution

H_A of step 1 indicates a difference between specified populations or parameters. This difference is tested using a test statistic, which reflects the difference suggested by the alternative hypothesis. The computed value of a test statistic varies for every sample and thus is a random variable and has a sampling distribution. Hypothesis tests should be based on a theoretical model. The model defines the distribution function of the test statistic, which can be used to make probability statements about the computed values.

Step 3: The Level of Significance

After formulating both hypothesis and selecting an adequate test hypothesis, a probabilistic framework for accepting or rejecting H_0, i.e. making a decision, is needed. It should reflect the chance variation that is expected in a sample of data. Such a chance variation is referred to as sampling variation. The website of the New Zealand Ministry of Education [17] explains more detailed:

> “Suppose a sample is taken and a sample statistic, such as a sample mean, is calculated. If a second sample of the same size is taken from the same population, it is almost certain that the sample mean calculated from this sample will be different from that calculated from the first sample. If further sample means are calculated, by repeatedly taking samples of the same size from the same population, then the differences in these sample means illustrate sampling variation.”

When a decision is made, there are two possible situations and the potential decision may lead to a correct or an incorrect outcome of the test. Table 2.2 depicts all the available scenarios.

Table 2.2: Decision table for hypothesis testing [16, p. 211]

	Situation	
Decision	**H_0 is true**	**H_0 is not true**
Accept H_0	Correct decision	Incorrect decision: type II error
Reject H_0	Incorrect decision: type I error	Correct decision

The decision table suggests two types of error:

1. Type I error: H_0 is rejected, even though H_0 is true.
2. Type II error: H_0 is not rejected, even though H_0 is false.

Both types are not independent from each other. The probability of making a type I error is denoted by α. The probability of a type II error is denoted by β. Since the possible incorrect decisions are dependent from each other, setting the level of significance very small, α becomes smaller but at the same time, β will increase drastically. Hence, the level of significance should be selected after extensively investigating the effects of a decision. Nonetheless, α is often chosen due to convention and availability of statistical tables. Most frequently α is selected as 0,05 or 0,01, representing 5% or 1% probability of rejecting H_0, when it is in fact, correct.

Step 4: Data Analysis

When the level of significance is set, the sample size required to meet any rejection criteria is set. Depending on the statistical test and knowledge of the population's variance, different formulas exist for processing its data [16, pp. 232–235].

Step 5: The Region of Rejection

When the possible values for the test statistic are put in a diagram, one finds two main regions. Figure 2.15 shows that those are called region of acceptance and region of rejection. The region of acceptance contains those values when the null hypothesis is true. When the null hypothesis is true, but a value is unlikely to occur, that value lies in the region of rejection. There is a critical value that separates both regions. It depends on the statement of the alternative hypothesis, the distribution of the test statistic, the level of significance and the characteristic of the data. Depending on the question, a test may be either one- or two-sided.

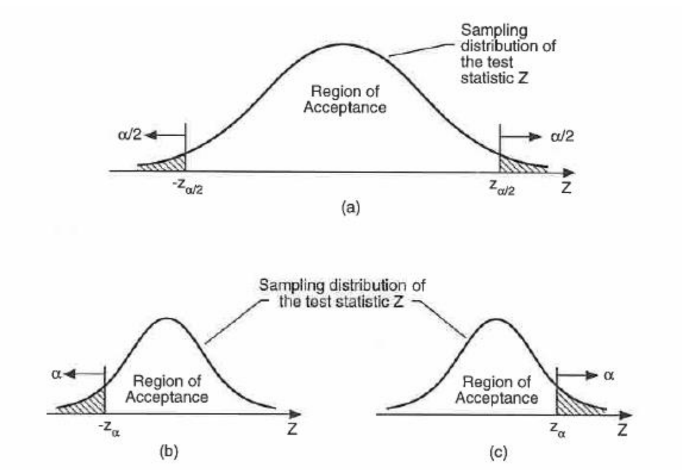


Figure 2.15: Region of acceptance and rejection [16]

Step 6: Select the Appropriate Hypothesis

The decision whether the null hypothesis is rejected depends on the comparison of the computed test statistic to the critical value. When the value lies in the region of rejection, the null hypothesis is rejected, a difference of its grammatically and mathematically expressed statement was found. Two possibilities lead to a rejection: First, the computed value lies purely by incident in the region of rejection, earlier explained as type I error. Secondly, the extreme value of the test statistic occurs, because the null hypothesis is false.

2.3.3. The p-Value

An additional approach to whether the null hypothesis is rejected would be the application of the p-value. Instead of using the comparison of the computed test statistic and the critical value one might use a value of probability, the so-called p-value. When H_0 is true, the p-value of the null hypothesis corresponds to the probability of computing an observed or more extreme value of the test statistic. The meaning of extreme depends on the alternative hypothesis. As a convention, the following scheme has become most common for the interpretation of the p-value [18, p. 42]:

- < 0,01 → very strong objection against H_0
- 0,01 – 0,05 → strong objection against H_0
- 0,05 -0,10 → weak objection against H_0
- > 0,10 → no objection against H_0

Later in this thesis the rejection of the null hypothesis will be handled using the p-value. Two distributions will be compared of being equal using the Chi Square (denoted as χ^2) test and p-values with a lower value than the level of significance will result in a rejection of the null hypothesis.

2.3.4. χ^2 Test

When two empirical distributions should be compared, whether they match or differ from each other, the easiest way would be comparing both distributions visually. Nonetheless, the aim of using machine vision systems is to hand tasks over to machines which can handle them with higher speed and crucial accuracy. Therefore, two empirical distributions of intensity distributions obtained from an image matrix taken by the image sensor, will later be compared using a χ^2 test.

Lohninger [19] recapitulates the theoretical background, a short overview is given in the following passages:

The χ^2 test compares an empirical with theoretical distribution that is a parametric distribution. A well-known problem is that the evaluation of parametric distribution functions results in probabilities and not in frequencies [19]. When the distributions are compared, the expected frequencies have to be estimated by multiplying the theoretical probabilities by the number of samples.

The probability of the variable falling into a bin [a_i, a_{i+1}] is the difference of the probabilities of x being less than the bin boundaries a_i and a_{i+1}, shown in Equation 7 and graphically underlined in Figure 2.16:

$$Prob\ (a_i < x < a_{i+1}) = Prob\ (x < a_{i+1}) - Prob\ (x < \ a_i) \qquad \text{(Eq. 7)}$$

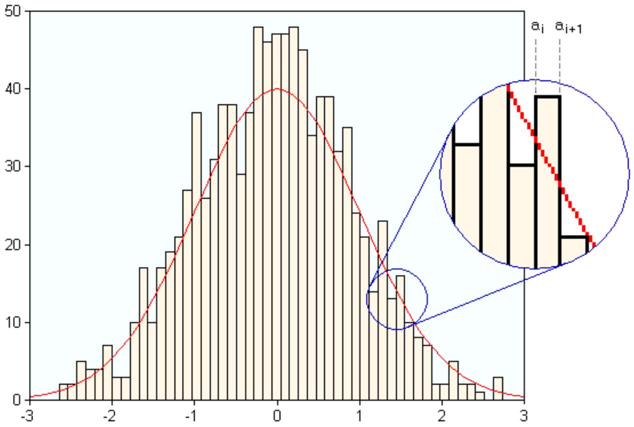


Figure 2.16: χ² test of distributions [19]

The next step is calculating the difference of each bin between the empirical and the theoretical distribution, which is then squared and divided by the expected frequencies. The sum of these relative or weighted squared differences is the Chi Square test statistic, shown in Equation 8. The corresponding null hypothesis of the test is that both distributions are the same and that the differences between them are due to random errors.

$$\chi^2 = \sum_{i=1}^{k} (F_i - E_i)^2 \;/\; E_i \qquad \text{(Eq. 8)}$$

F_i … empirical frequency in bin i
E_i … theoretical frequency in bin i
k … number of bins

Table 2.3 gives an overview over the details of the χ^2 test.

Table 2.3: χ² test overview

Hypothesis	H_0 : distributions are equal H_1 : distributions differ
Test statistic	See Equation 8
Degrees of freedom	k-3
Rejection	Reject H_0 if $X^2 > X^2_{\alpha,\,k\text{-}3}$

3. Quality Monitoring of Machine Vision Systems

Modern production facilities are using automated image-based quality control of their produced goods. There is however little open source knowledge about how to control the quality control system itself. Therefore, a part of this thesis concentrates on the question how to ensure the quality of the taken images and more important on a how a transnationally operating company can remotely detect which part of a machine vision system is malfunctioning. This is especially important since travelling to a production site and determine the machine vision system's problem is genuinely time consuming. Idle time on production sites is usually causing production delays that could lead to immense costs in case of not meeting the customers' production deadlines.

Hence, this work proposes a procedure where locally available templates are used to specify the defected part of a machine vision system. Ideally, a local worker would place several of these templates on the conveyor belt, respectively at the focal plane of the part that is usually inspected. Then, the machine vision system should obtain images that are then sent to the company's skilled expert on machine vision quality control, who then carries out various algorithms and visual inspections of those images to determine the reason for the malfunction. This procedure should be more time efficient than sending the company's expert around the world and more cost efficient than hiring several experts who are able to address the same problem.

Furthermore, a routine-based quality control of the machine vision system should enable the company to improve on when it is necessary to change the MV system's body parts and thus improve the organisation of ordering and changing spare parts to avoid any production stops in inconvenient periods.

This chapter lists a variety of possible malfunctions of a MV system and how to detect those malfunctions using only a small number of templates.

3.1. Illumination Related Defects

The key component to a stable machine vision system is the source of illumination. Demant et al. [6] as well as Martin [20] show that there are many approaches to address different kinds of illumination problems. For industry applications, however, it is of most importance to monitor whether the illumination system fails to deliver the intended performance and when the light source is due for maintenance.

3.1.1. Degradation of Light Source

To ensure the stability of an algorithm in machine vision systems, one must focus on using a light source that is not changing in its performance power. Varying intensity of light emittance due to production errors or degradation of the light source might lead to a failure of the algorithms, especially, when a statistical approach of comparing intensity distributions is chosen to differentiate between defected objects. Figure 3.1 shows the red layer intensity distributions of images of a white blank sheet of paper with decreasing power of the light source. A commercially available ring flashlight unit was set to different lighting power in each photo.

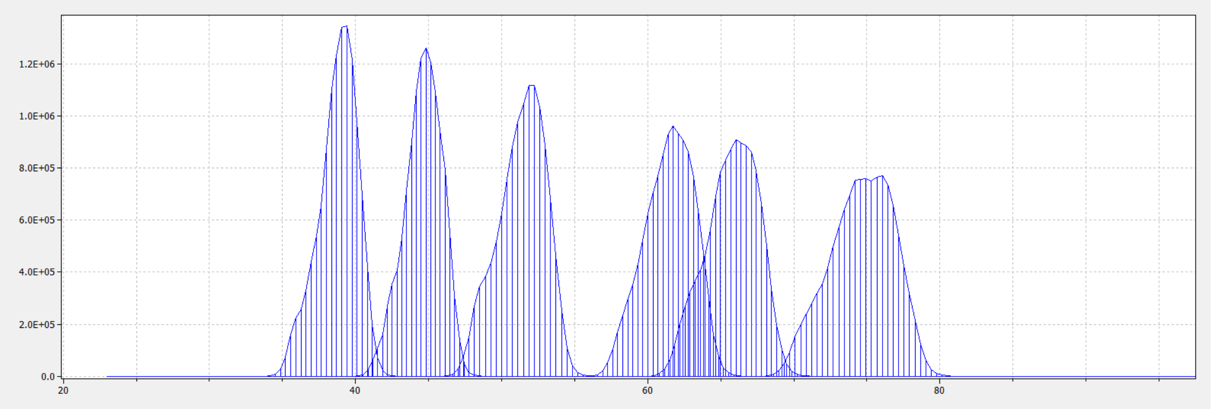


Figure 3.1: Intensity distributions of a white blank sheet with varying power of the light source

One can easily see that with differing intensity of the flashlight, the intensity distributions shift. A quality monitoring system might exploit that knowledge by regularly acquiring images of a white blank page, denoted as *Template W*, and save the obtained intensity distributions for a later on comparison with newer intensity distributions. In this way, the intensity distributions can be added to a timeline of distributions, indicating when the light source has degraded to an extent, where it is due to replacement.

However, a possible drawback has to be taken into account: *Template W* should have a non-reflective surface or should be tested with low light intensity. Reflections of the template might lead to different outcomes when the camera is not mounted on the square of the examined object. This can be cross-checked by determining if the camera is in a 90° angle, described in chapter 3.2.1 for mount related defects.

3.1.2. Inhomogeneous Illumination

The previous chapter focused on the general observation of the performance of light sources. Most illumination systems however contain not only one light source, but many built into a single casing. Martin [20] and Demant [6] explain in detail advantages and disadvantages of different types of illumination systems, concluding with LED being the most common light source in industrial applications, independent from the type of casing. Despite their high life expectancy compared to other light sources, they are not immune to sudden failure. When a casing consists of many LEDs, it is of most importance to ensure that all diodes are working. Single malfunctioning diodes might result in inhomogeneous illumination of the examined object, which might cause threshold or intensity distribution-based algorithms for defect detection to fail. Thus, *Template A*, as seen in Figure 3.2, is introduced.

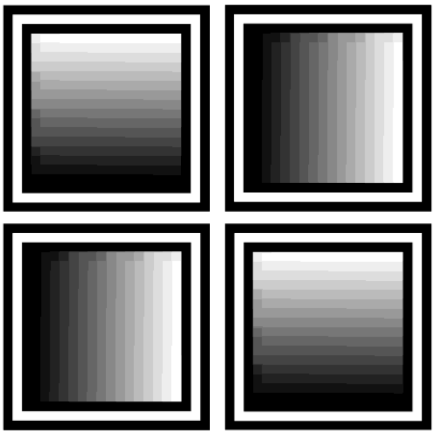


Figure 3.2: Template A for black and white sensors

Figure 3.2 gives an impression of the template's design, which in any case can and should be adapted to meet the camera's performance. The shown template consists of four similar squares of grey value patterns. Each square consists of 256 grey values, ascending from 1 to 256. The grey value patterns are surrounded by a black, white and another black frame. The main idea of such a template is that, under perfect illumination conditions, all patterns would show the same intensity distribution, regardless of their position or rotation on the template.

Ideally, the intensity distribution of one pattern should cover a broad range of intensities. However, the intensity distribution will never have a range from the minimum to the maximum possible intensity. This would only be the case if certain pixels would be completely saturated, while others would receive no light at all. Therefore, the 256 grey values in the pattern may be reduced to an extent that is satisfying for the system's setup.

Patterns next to each other are rotated to be able to obtain more information of compared patterns. The number and size of the patterns should be fitted to the camera's resolution and distance to the examined object.

The surrounding frames serve for the ability to individually detect each pattern. By using a post processing algorithm like a flood fill algorithm, every pixel surrounding the black frame that lies within a certain intensity range will be set to zero. Then, a threshold can be applied, detecting every pattern individually, because any pixel of any pattern, except for the ones previously set to zero, will have an intensity higher than zero. This procedure is shown in Figure 3.3, where *Template A* was printed out using a consumer office printer and then photographed.

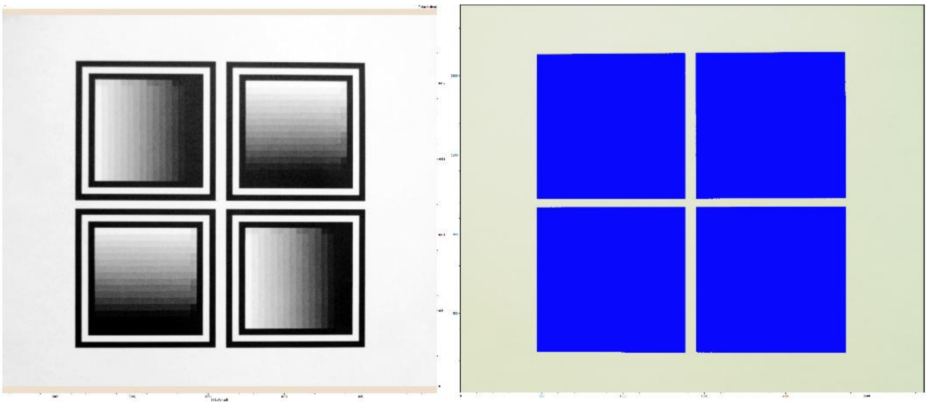


Figure 3.3: Left: photo of Template A; right: detection of individual squares

The right picture indicates that after performing the algorithm, only the distinct areas of blue pixels are taken into account for further calculations, namely the statistical analysis of each square compared to the others. The Nassi-Shneiderman diagram of the algorithm is shown in Figure 3.4.

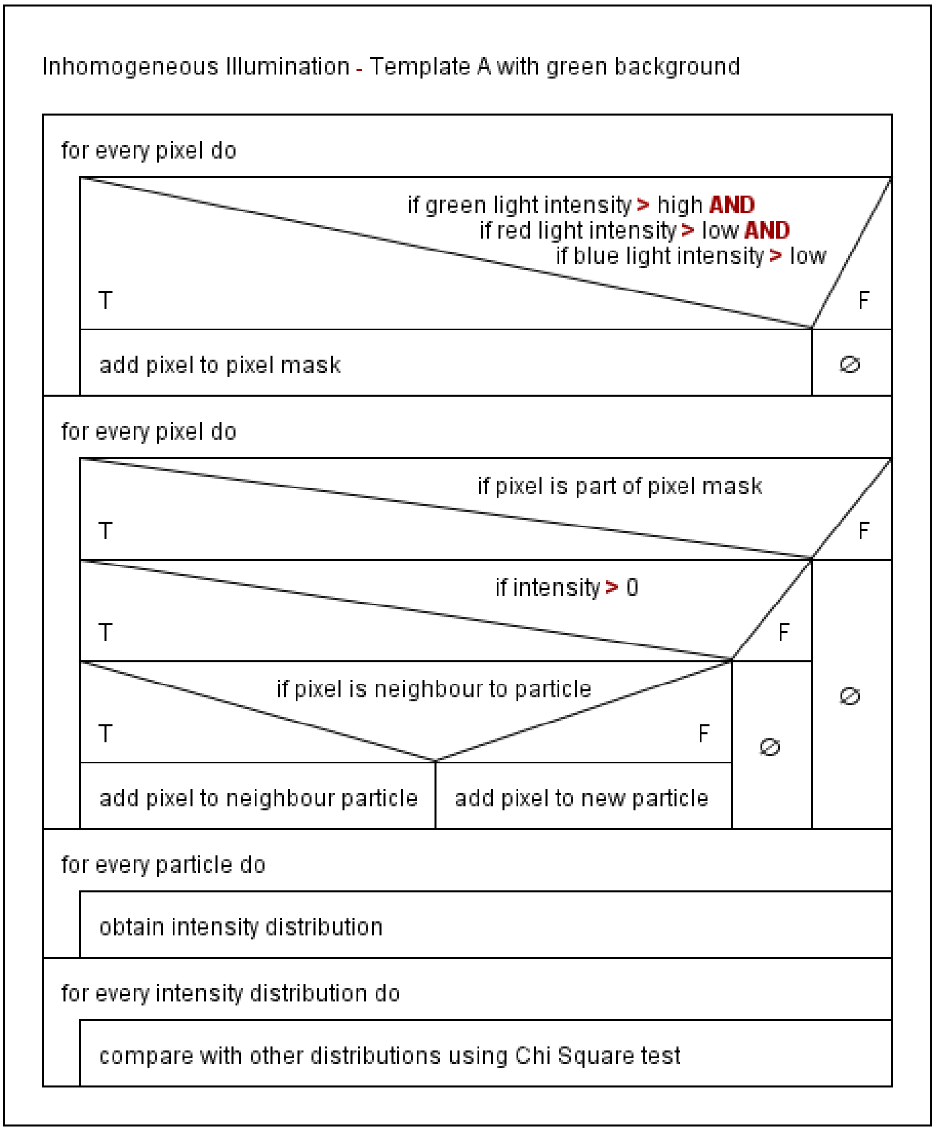

Inhomogeneous Illumination - Template A with green background
for every pixel do
if green light intensity > high AND
if red light intensity > low AND
if blue light intensity > low
T
F
add pixel to pixel mask
∅
for every pixel do
if pixel is part of pixel mask
T
F
if intensity > 0
T
F
if pixel is neighbour to particle
T
F
add pixel to neighbour particle
add pixel to new particle
∅
∅
for every particle do
obtain intensity distribution
for every intensity distribution do
compare with other distributions using Chi Square test

Figure 3.4: Nassi-Shneiderman diagram of the Inhomogeneous Illumination algorithm using Template A with green background

For colour sensors, a second approach to detect each pattern individually is using a coloured background for differentiation. This variation of *Template A* is shown in Figure 3.5.

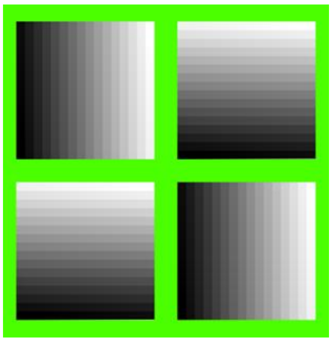


Figure 3.5: Template A for colour sensors

When the background is painted in colour, the fact that the grey values have similar intensity in each colour channel can be exploited to distinguish them from the background. This is shown in Figure 3.6 and Figure 3.7, where the intensity of the corresponding colour channel at the marked pixel is drawn in a graph.

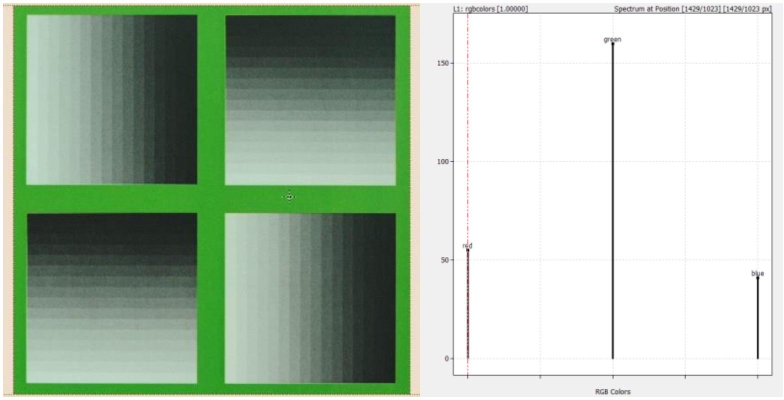


Figure 3.6: Left: photo of Template A and marked pixel within the green background; right: RGB intensities at marked pixel in the green background

The green background has a high intensity of green light in each pixel, while red and blue intensities are significantly lower. Using this knowledge, the algorithm is programmed to again create a pixel mask that only detects the grey value patterns, as previously shown in Figure 3.3 (right).

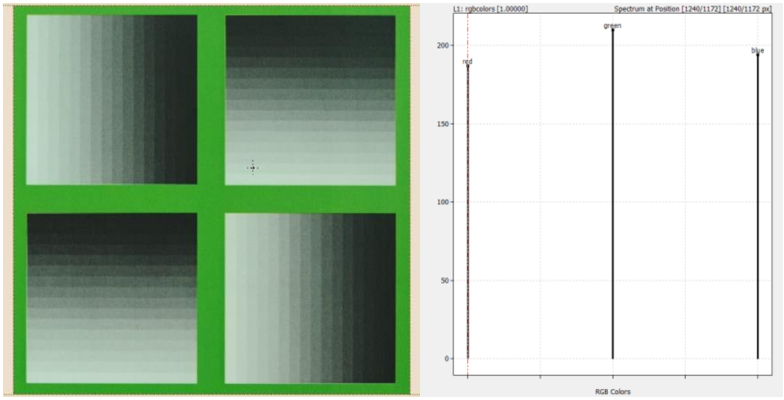


Figure 3.7: Left: photo of Template A and marked pixel within the grey value pattern; right: RGB intensities at marked pixel in the grey value pattern

The next part of the algorithm is the comparison of each pattern to the others. Before, one pattern is artificially brightened to mark a clear difference to the others and simulate an illumination defect. Then, each pattern is analysed on its intensity distribution. Both steps are depicted in Figure 3.8.

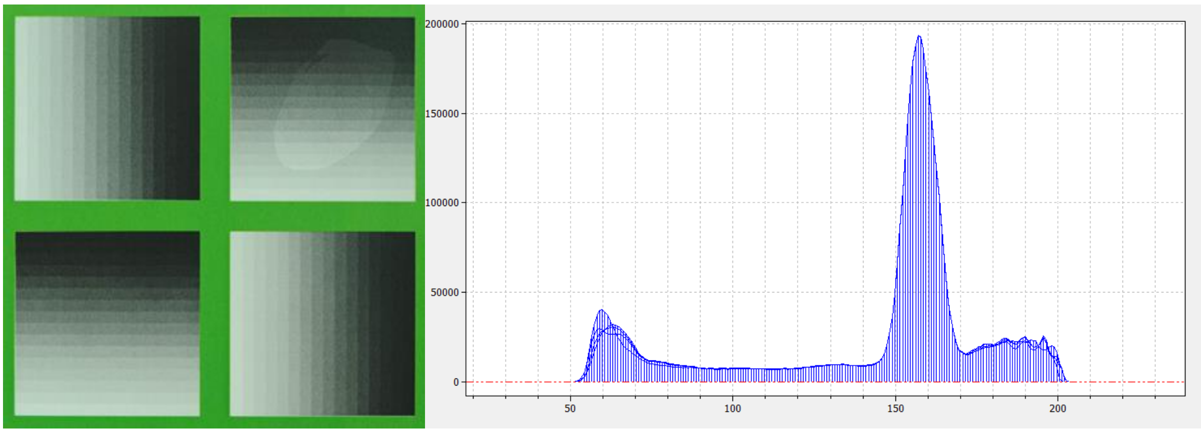


Figure 3.8: Left: photo of Template A with upper right pattern artificially brightened; right: intensity distribution of all four patterns

Afterwards, the intensity distributions were scaled in a way, where every frequency bin was divided by a constant factor. This results in fewer counts for each frequency bin, the shape of the distribution, however, stays the same. The scaling is necessary, because the high number of counts lead to the χ^2 test being insensitive. When all the bins have a high number of counts and thus the total sum of counts is high, the differences between the compared distributions are weighing less. When the number of total counts is low and thus the number of counts per bin is low, the differences between the compared distribution are weighing more.

Table 3.1 shows the results when the intensity distribution of each pattern of Figure 3.8 (left) is compared to the others. The test outputs the test statistic χ^2 and a p-value. Their meaning was explained in chapter 2.3. High values of the test statistic and low values of p indicate that the distributions differ. The level of significance was chosen to be 0,05. This means, that if the p-value is lower than 0,05, the null hypothesis is rejected and the distributions are not the same. Table 3.1 displays that the upper right pattern of Figure 3.8 is always recognised to be different to the others. For multiple comparisons, the Bonferroni correction is necessary in order to maintain the level of significance.

Table 3.1: Results of χ^2 test of Template A with coloured background

Pattern	Top left		Bottom left		Top right	
	χ^2	p-Value	χ^2	p-Value	χ^2	p-Value
Bottom left	192	0,35				
Top right	226	0,03	271	0,00		
Bottom right	211	0,09	60	1,00	227	0,02

Figure 3.9 (top) shows the scaled distributions of the top left and bottom left pattern being quite similar. Figure 3.9 (bottom) displays the scaled distributions of the bottom left and top right pattern differing in the grey value region of 40 to 60.

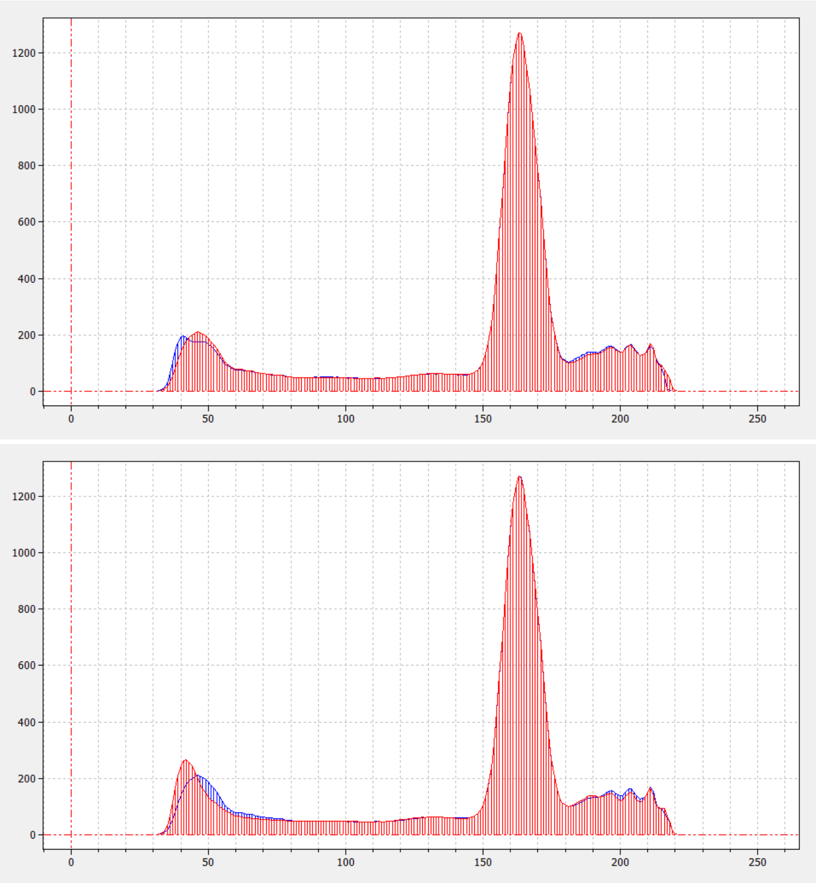


*Figure 3.9: Top: distributions of top left and bottom left pattern;
bottom: distributions of bottom left and top right pattern*

These results were only achieved after scaling the distributions properly and the rather low values of the top left pattern compared to the bottom left and bottom right pattern indicate potential for improvement. It is, however, considering the low budget equipment that was used, a satisfactory outcome of the aim to show that differences in illumination can be determined by using a simple pattern and a consumer camera.

Additionally, it is stated, that the comparison of one pattern to the others should follow a strategic thinking. One might program the algorithm to compare one pattern to all the others, this however will lead to extensive computing time, especially when a high number of patterns is used in one template. More patterns in one template, however, will lead to more accurate testing of uniform illumination. One possible strategy would be comparing only neighbouring patterns or divide the template in regions, where a randomly selected pattern is compared to one of a different region. Figure 3.10 shows the Nassi-Shneiderman diagram of the algorithm.

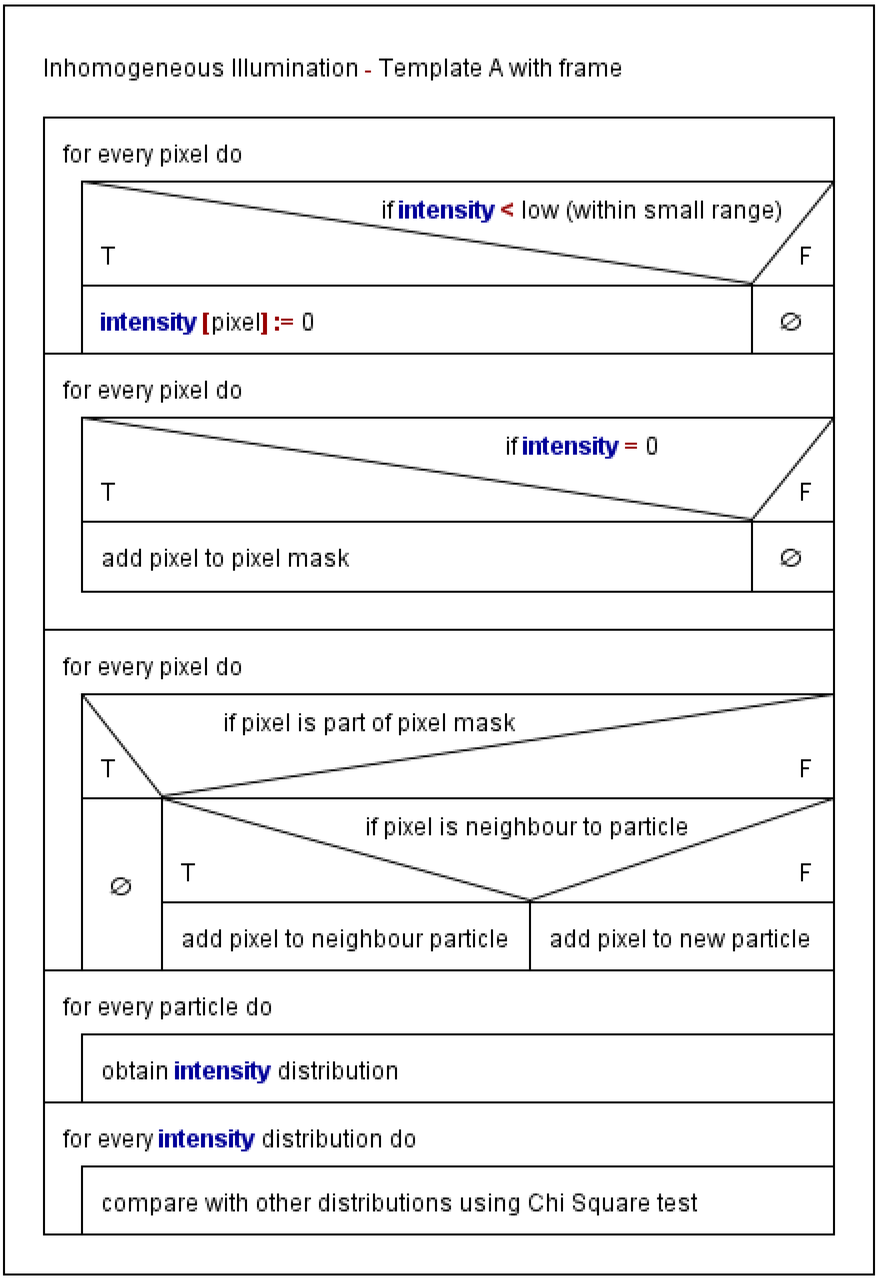

Inhomogeneous Illumination - Template A with frame
for every pixel do
if intensity < low (within small range)
T
F
intensity [pixel] := 0
∅
for every pixel do
if intensity = 0
T
F
add pixel to pixel mask
∅
for every pixel do
if pixel is part of pixel mask
T
F
∅
if pixel is neighbour to particle
T
F
add pixel to neighbour particle
add pixel to new particle
for every particle do
obtain intensity distribution
for every intensity distribution do
compare with other distributions using Chi Square test

Figure 3.10: Nassi-Shneiderman diagram of the Inhomogeneous Illumination algorithm using Template A with frame

At last, an important detail is discussed. How is it possible to differentiate between one or several diodes malfunctioning or the pixels of the camera being less sensitive or broken? Figure 3.11 depicts a modification of *Template A* with 8x8 grey value patterns. The red stripes indicate broken arrays of sensor pixels, while the red circle represents one or several broken light sources.

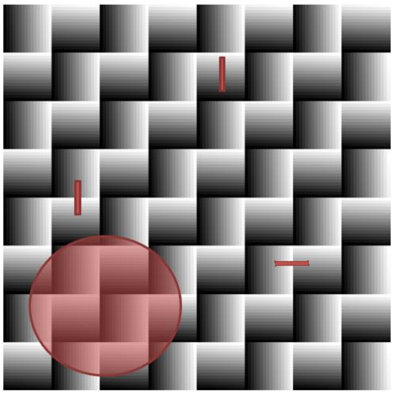


Figure 3.11: Schematic differentiation of the defected light source and the defected pixels

First, light sources are more prone to degradation, due to the higher thermal stress they are set out to. Furthermore, it is less likely that all pixels fail at the same time, it is more probable that certain arrays or distinct pixels will break. When certain pixels fail, this will be noted, because only in the distinct pattern where the failing pixels are located, a difference of the intensity distribution is noted. When diodes of the light source are malfunctioning, a difference of several patterns to the others will be noted, because usually the light source is emitting light in a cone or even more broadly distributed in a diffuse form. Figure 3.12 depicts the malfunctioning of a single diode. The pattern located under the defected light source receive less light and will thus be recognized by the algorithm.

In the unlikely case of all sensor pixels degrading at the same time to the same extent, one first might to drive the light source with more power. When this fails to reach the same intensity distributions for all patterns as when they were mounted, the light source has degraded and must be changed.

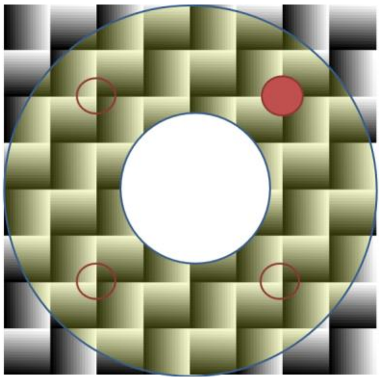


Figure 3.12: Schematic failure of one diode in a 4-diode illumination case

3.2. Mount Related Defects

When a MV system is first installed, a key challenge is to check whether the one or usually the several cameras are mounted in such a way that all of them are focussing the desired object in the right angle. Once the MV system is operating, the angle of the cameras might change due to vibrations at or around the production machine that are strong enough to loosen the mounting screws of the cameras. This then would lead to blurred images because the originally vibration isolated camera is starting to shake as well. A worse outcome of crooked mounted cameras is that they have a different perspective of the object than intended. This might cause the defect detection algorithm of the system to fail, because the spatial resolution of the analysed object changes, as depicted in Figure 3.13. When viewed from a central position, a rectangular object with the same side length will appear as a square. However, when the same object is recognised from a different angle, it is imaged as a trapezoid, because the distance to the corner points has changed.

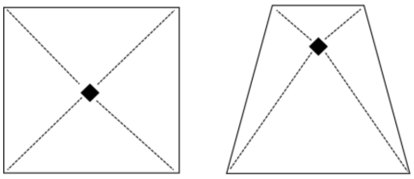


Figure 3.13: Perception of geometry due to different angles

3.2.1. Crooked Mounting

Ideally, a machine vision system's camera should be mounted in a central position to the detected object. This is of course, especially when using more than one camera, a rather difficult task. To check whether the camera is mounted in a 90° angle to the examined object, again *Template A* might be used. Figure 3.14 (left) shows a photograph of *Template A* shot in a crooked position. Figure 3.14 (right) depicts the detection of the individual patterns using the same algorithm as in chapter 3.1.2.

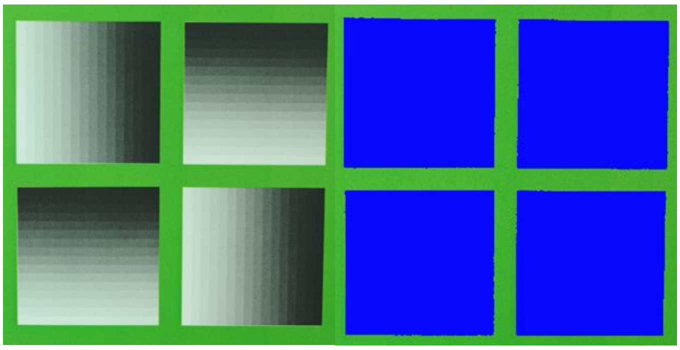


Figure 3.14: Left: Template A photographed from slightly above the central position; right: detection of the individual squares

Now, the size of the patterns is examined to check whether they all face the same distance from the camera. If the camera was mounted in the very middle of the template, all patterns would be detected with a similar amount of pixels. If, however, the perspective of the camera was changed, this would result in a difference of overall detected pixels between the patterns. This is concluded in Table 3.2.

Table 3.2: Pixel size of crooked perspective photograph of Template A

Pattern	Top left	Top right	Bottom left	Bottom right
Size [px]	694 964	688 925	662 777	659 871

Since the image was taken from a slightly higher angle than the central position, the top patterns have a smaller distance to the lens, making them appear bigger than if they were photographed from the central position. The bottom patterns have a wider distance, resulting in a smaller appearance and less counted pixels. In conclusion, the difference between the top and the bottom patterns is about 30 000 pixels.

This method offers a great outlook to be developed into a tool, that is not only able to state that there is crooked mounting, but how much the degrees the camera is off to the central position. The Nassi-Shneiderman diagram of the algorithm is shown in Figure 3.15.

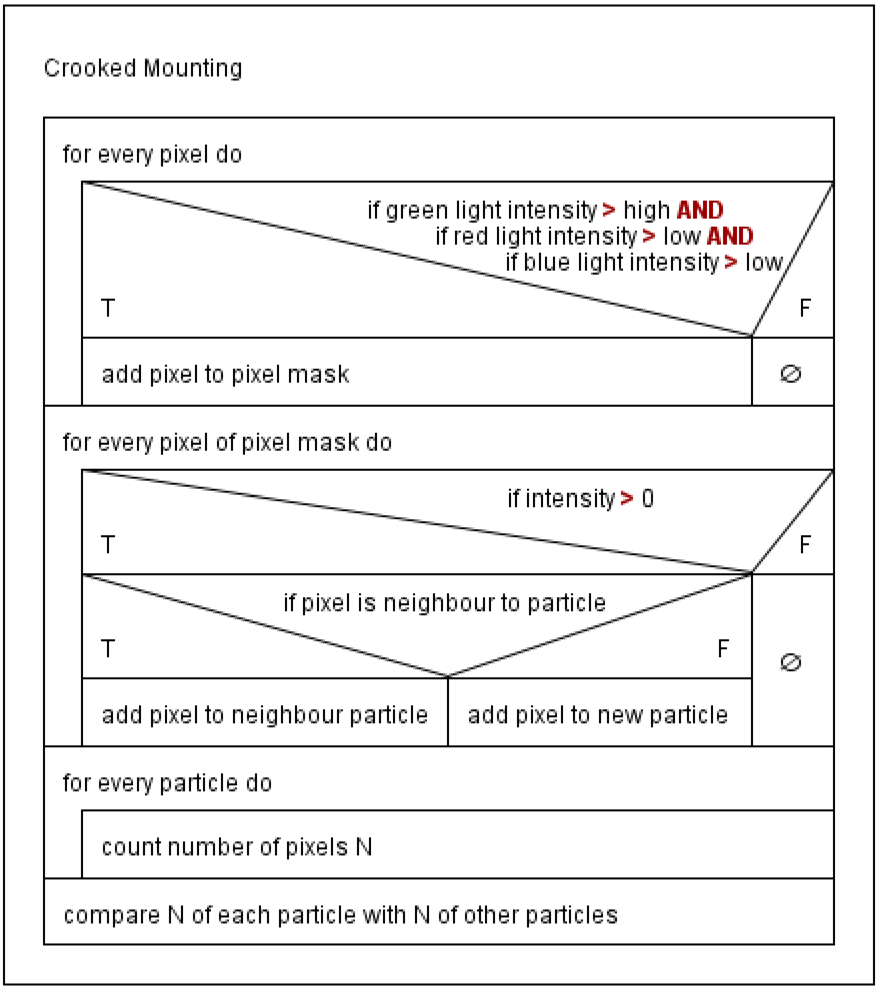

Crooked Mounting
for every pixel do
if green light intensity > high AND
if red light intensity > low AND
if blue light intensity > low
T
F
add pixel to pixel mask
∅
for every pixel of pixel mask do
if intensity > 0
T
F
if pixel is neighbour to particle
T
F
add pixel to neighbour particle
add pixel to new particle
∅
for every particle do
count number of pixels N
compare N of each particle with N of other particles

Figure 3.15: Nassi-Shneiderman diagram of the Crooked Mounting algorithm

3.3. Sensor Related Defects

The two key hardware components of a machine vision system are the camera sensor and the illumination. While illumination is prone to failure due to mechanical and thermal stress of operation, the sensor might cause quite different problems. These are examined in the following chapter.

3.3.1. Pixel Degradation and Breakdown

In chapter 2.2.2, the theoretical background of camera sensors was explained. There might occur two defects, when closely examining the sensor. First, individual or several pixels are subject to complete failure. Electronic or manufacturing failure might cause individual or an array of pixels not to transmit any signal, resulting in minimum or maximum values, when reading out the electronic image. According to G. Klinglmayr [21], broken pixels are recognised, when a new sensor is installed. The problem of degrading light is more extensive than sudden failure of pixels.

Nonetheless, the defects are described as follows: First, there are so many broken pixels that the image quality is poor, which would be easily visible when doing visual inspection of an image. When there are too few broken pixels to visually recognise the defects, this might not cause an algorithm to fail. Still, a minor possibility of failure remains. It is thus worthy of inspection. At last, the least likely, but at the same time the most delicate complication would be the degradation of the light-sensitive elements or the electronics behind them. Depending on the manufacturing quality of sensors, terminal capacitance of transistors or other material changes might occur, leading to a degradation of pixel sensitivity. While the total failure of several pixels in relation to the total amount of pixels might have very little to no effects on defect detection algorithms, a change of sensitivity leads to the same problems as previously described in chapter 1, namely a change or constant offset of the obtained intensity distribution. For a deeper understanding of pixel defects, Geradts et al. [22] summarize the subject in detail.

The number of broken pixels can be encountered by applying the following algorithm to a white blank page, *Template One*. In order to obtain optimal results, over- or underexposure in the obtained image must be avoided. The Nassi-Shneiderman diagram of the algorithm is depicted in Figure 3.16.

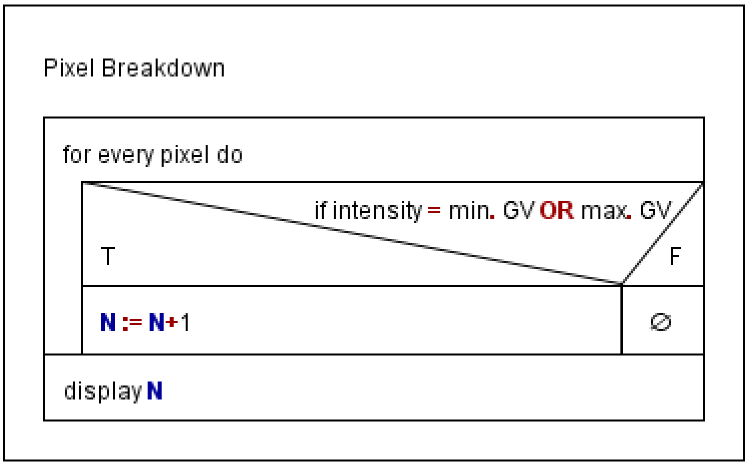


Figure 3.16: Nassi-Shneiderman diagram of the Pixel Breakdown algorithm

Pixel degradation should be supervised by doing a regular check on the intensity distribution of *Template One*. The periodically obtained distributions can be cross-checked to earlier distributions either visually or using the χ^2 test.

3.3.2. Dark Current and Temperature Related Pixel Sensitivity

One of the easiest quality monitoring measurements is the periodical acquisition of a dark current or dark noise image. For this, the lens of the camera has to be covered, prohibiting any light to fall onto the sensor array. The image that is then registered consists only of the dark current. The current that is registered is composed of the thermo-induced movement of atoms and extra-terrestrial radiation the pixels are exposed to. On the one hand the current is random and should stay constant over time. However, when stronger noise is detected after a certain time span, this may indicate some kind of alteration of the sensor’s performance. On the other hand, the movement of electrons always correlates to the temperature of a material. Thus, also the photo-sensitive components, as well as the electronic hardware should issue different magnitudes of dark current at different temperatures.

This was revised with a simple experiment. The camera was kept in a room of 27°C and several images were taken using different aperture and shutter times. The same was done after keeping the camera in a fridge with 6°C temperature. For every temperature, the aperture was set and five photos were taken for each shutter time. A detailed overview for the used settings is shown in Table 3.3. The temperature was measured using a standard lab thermometer.

Table 3.3: Design of experiment for dark current investigation

Shutter Time	60	200	500
Aperture	2	4.5	

The data that was generated leads to the following conclusion: at 27°C the dark current is higher than at 6°C, showing a few hundred counts more at dark intensities. Regarding the total number of pixels, however, the registered noise is diminutively low. All the distributions show similar results, thus only a small selection of distributions was put in the appendix, respectively in Table A.1 and Table A.2. A summary of the obtained values is shown in Table 3.4. In the table only the first 10 intensities, i.e. the ten darkest GV of the intensity distribution, are displayed, because the recording of brighter values was thoroughly low.

Table 3.4: Summary of dark current data

Temp. [°C]	6		27		27	
Shutter Time	60		60		200	
Intensity	x̄	s	x̄	s	x̄	s
1	12205069	0	12204808	61	12204796	45
2	6	0	141	18	197	44
3	6	0	63	43	34	5
4	4	0	21	9	16	1
5	4	0	20	10	15	1
6	0	0	7	1	10	1
7	0	0	3	1	5	1
8	0	0	4	3	3	1
9	0	0	3	2	3	1
10	0	0	5	3	3	2

Naturally, the goal of the experiment was to show that the dark current affects the image acquisition of the sensor. However, the numbers indicate that the effect, regarding the resolution of the sensor, is so little, that a strong influence on the image quality is very unlikely. Nonetheless, the monitoring of ambient temperatures might be interesting for production sites that are subject to grave temperature fluctuations.

3.3.3. Pixel Overdrive and Electronic Defects

Another problem of the sensor and its electronics is an overdriven output. If, for whatever reason, a pixel registers a higher intensity value than possible considering the bit depth of the sensor, a simple algorithm is able to find such a defect. The algorithm can be run with any taken image. Figure 3.17 shows the Nassi-Shneiderman diagram of the algorithm.

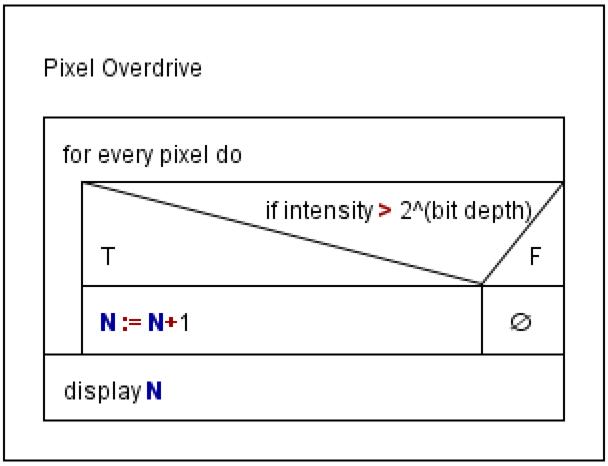


Figure 3.17: Nassi-Shneiderman diagram of the Pixel Overdrive algorithm

3.3.4. Resolution

The sensor's resolution is not subject to changes, nonetheless, it may be checked using commercially available test charts. Figure 3.18 depicts such a test chart, with which it is possible the determine the resolution of the camera. Koren [23] explains the concept of image sharpness in detail, while Jähne [14] explicitly summarizes resolution theory.

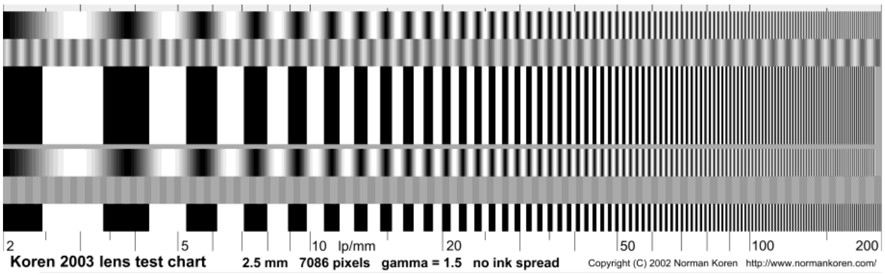

2
5
10 lp/mm
20
50
100
200
Koren 2003 lens test chart
2.5 mm 7086 pixels gamma = 1.5 no ink spread
Copyright (C) 2002 Norman Koren http://www.normankoren.com/

Figure 3.18: Koren 2003 Lens Test Chart [24]

3.3.5. Exposure Time

A hardware feature that is well worth investigating is the camera's exposure time. Being crucial for the image brightness, the exposure time has to be kept constant to avoid falsely detecting the sensor's sensitivity or the illumination systems as malfunctioning parts, when the intensity distribution of *Template One* is changing over time. There are various shutter types [6], all of them face mechanical stress when in use. Therefore, the performance of the mechanism has to be monitored. Due to their mobility, consumer cameras are easy to handle and checked on their exposure times building different setups around them. In industry applications, however, detaching the camera from its setup is vividly avoided. Thus, the design of a *periodically light emitting box*, denoted as *PLEB* is introduced. It might be easily manufactured using basic electronic engineering and programming skills.

The *PLEB* consists of several light emitting diodes and is computer-controlled. LEDs can be addressed with great precision, thus being perfect for monitoring a shutter that is open for only a few milliseconds. The *PLEB*'s diodes are built in in a simple array. Using an adequately programmed software, they are controlled in way, where two LEDs flash at the same time. After a user-defined time span, the next two LEDs flash at same time, the left diode being the right diode from the previous flash. This is depicted in Figure 3.19. In this way, two flashing diodes are "running" over the device.

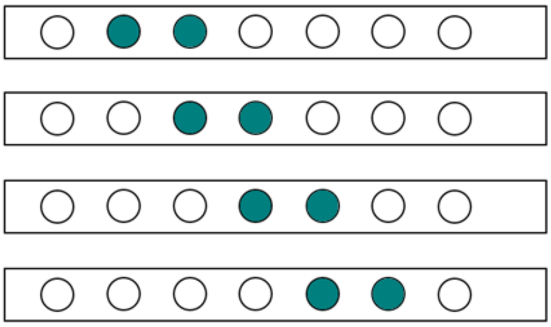


Figure 3.19: Light emitting scheme of a PLEB

When a user-defined time span is set, the same time should be set as the shutter time in the camera setup. Thus, when an image is taken, a precise shutter would lead to images in which there are always just two LEDs flashing. If there are four or even six lights flashing, the shutter opens and closes too slow, making the image brighter than intended. In conclusion, the *PLEB* can be used to check whether the shutter in fact opens for the time span it was set to.

3.4. Lens Related Defects

Regarding the optical system of the camera, certain features of the lens might lead to a malfunctioning of the machine vision system. However, lens related issues are less to be seen as a defect, they rather limit the system’s performance due to bad quality of hardware components or weakly adjusted settings.

3.4.1. Defocussing

A weakly adjusted focus has always been a nightmare for every hobby photographer. In industry, the problem of an incorrectly set focus is not a spontaneous phenomenon, it’s rather a matter of adjustment when the machine vision system is implemented. Nonetheless, it deserves mentioning that there are commercially available test templates. Visual interpretation of photos of those templates can be used for an optimal focus setting of the machine vision system. An example for determining focus is shown in Figure 3.20.

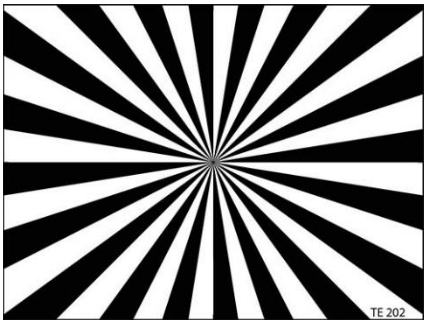


Figure 3.20: Image Engineering’s TE202 test template [25]

3.4.2. Geometrical Distortion

A tremendously important feature of lenses is their liability for the distortion of an object’s shape. Weak design or mechanical and optical constraints deteriorate the lens’ pathway of light to the sensor, leading to a distortion of the imaged shapes. There are two possible distortions: the barrel distortion and the pincushion distortion. Both types are shown in Figure 3.21.

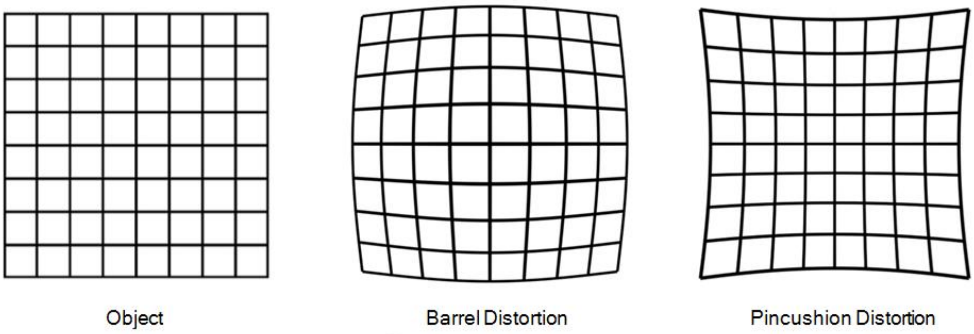


Figure 3.21: Visualisation of geometrical distortion [26]

Geometric distortion can be detected visually by using a grid template like in Figure 3.21. However, to eliminate individual human perception and to introduce a standardized detection procedure, an algorithm using *Template A* is introduced. The algorithm is similar to the one used for detection of crooked mounting. The main idea is to adapt *Template A* so that it is similar to a grid shown in Figure 3.21. To avoid lens distortion, the user should use telecentric lenses as they usually show minimum distortion.

4. Defect Detection in Polymer Products

Following Jähne [14], Demant [6], Snyder [1] and relevant publications [27]–[29] in the scientific imaging community, it is important to emphasize that machine vision is a powerful tool for various applications due to its versatility, but at the same time exhibits sundry difficulties. The task of making our perceptible world visible to a machine is a difficulty on its own. However, the idea to configure a machine vision system in a way, where it not only acquires images, but finds abnormalities in the latter has been subject of investigation and development since the birth of computer and machine vision.

The previously mentioned sources explain a variety of machine vision related problems occurring in industry and the corresponding approaches of how to solve them. In contrast to those standard approaches, new algorithms and general solutions have to be investigated to be able to face more delicate defect detection problems. Chapter 4 contains the examination of two unusual, yet highly interesting defects on objects with a hard-to-handle surface. The programming and application of the algorithms was done using the software ImageLab [30].

4.1. Defect Detection of Transparent Circular Shaped Lids

Regular machine vision applications process images of opaque objects to identify certain kinds of defects or irregularities. Admittedly, detecting defects in a both transparent and reflective material might be grasped as an idea of lunacy, this avant-garde thinking, however, is one of the main entitlements of science and thus an idea worth investigating.

4.1.1. Image Acquisition

The experimental setup that was used was successively adjusted and consisted in its final stage of the following parts, represented in Figure 4.1:

- Nikon D90 digital single-lens reflex camera with a 23,6 mm x 15,8 mm CMOS sensor and a resolution of 4288 x 2848 pixels (12,3 megapixels)
- Nikorr 50 mm (fixed focal length) lens
- Low-end LED ring-flashlight powered by 4 AA batteries
- Remote trigger control
- Tripod
- Physical enclosure

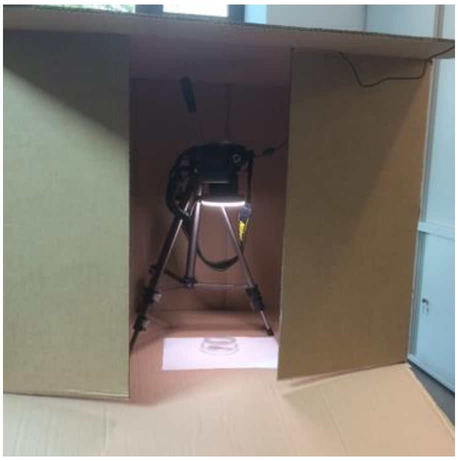


Figure 4.1: Experimental setup for defect detection of transparent circular shaped lids

The investigated lids were placed on a white sheet with marks to ensure similar positioning when the lids were exchanged. A spirit level was used to adjust the tripod perpendicular to the examined lid. In order to ensure constant light conditions, the cardboard box serving as a physical enclosure against ambient light was shut. The flashlight was operated in continuous mode and finally a photo was taken by manually triggering the remote control. A wide variety of different settings were tested. The most prosperous that was chosen at last and with which all images were taken is shown in Table 4.1.

Table 4.1: Settings for image acquisition of transparent circular shaped lids

Shutter time	Aperture	ISO	File format
1/500 sec	4.5	200	RAW

The images were taken in RAW format, which means that no camera-internal post-processing was done. Afterwards, the RAW-files were converted into TIFF-files using IrfanView. The conversion was done without any alteration or compression, maintaining the originally registered intensity values of the sensor. Finally, the images were imported into ImageLab, where the post-processing defect detection algorithm was applied.

In total, exactly 100 different lids were photographed. 67 of those lids were non-defected (NDL), while 37 exhibited different types of defects. Figure 4.2 depicts three images of non-defected lids. Incidentally, it is noted that the 37 lids represent the entirety of defected lids (DL) that were available for testing.

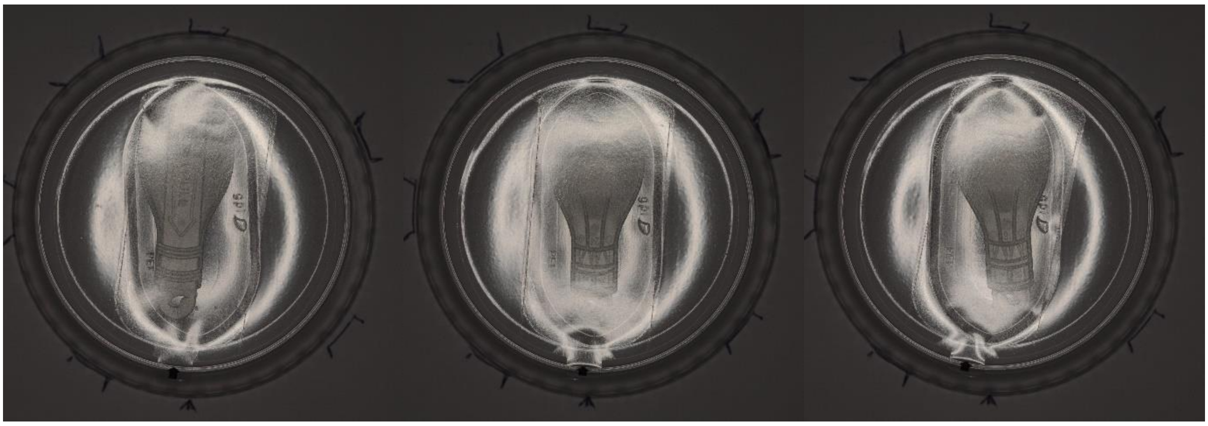


Figure 4.2: Non-defected lids; left: NDL sample 2; middle: NDL sample 3; right: NDL sample 4

4.1.2. Types of Defects

The examined object is a transparent, circular lid with an elongated indentation, in which a spoon is placed. The spoon's palm is randomly either facing the top or the bottom. Additionally, the indentation is slightly wider than the spoon's width, allowing the spoon carrying out minor rotations. On top, a foil is neatly applied, ensuring a contamination-free storage and transport of the spoon.

The lids exhibited three different types of defects that were categorized as follows:

I. Crinkled foil
II. Displaced spoon
III. Displaced foil

Defect Type I and Type II, shown in Figure 4.3 and Figure 4.4, share the same feature that accounts for their removal in the production process. A foil that does not stick neatly on the surface of the lid exposes the spoon to contamination. Defect Type III, shown in Figure 4.5, might detain the user from uncasing the spoon.

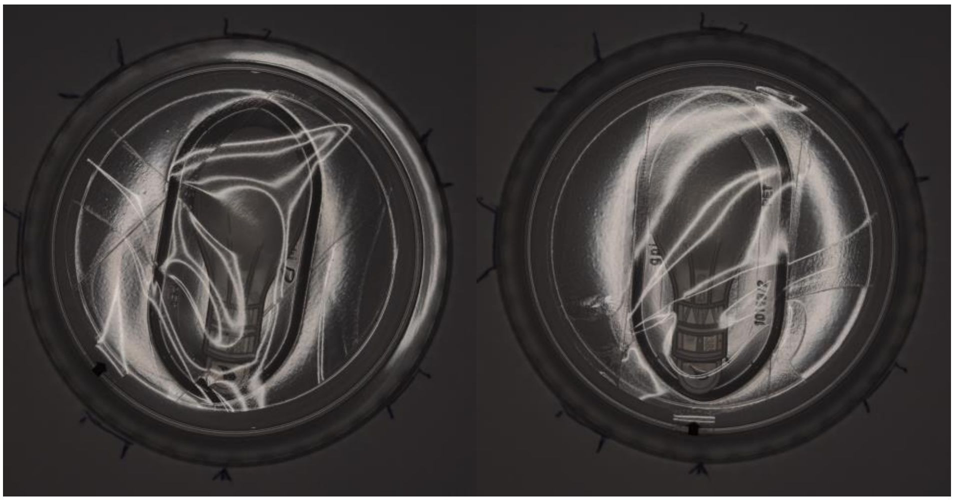


Figure 4.3: Type I defected lids; left: DL sample 9; right DL sample 5

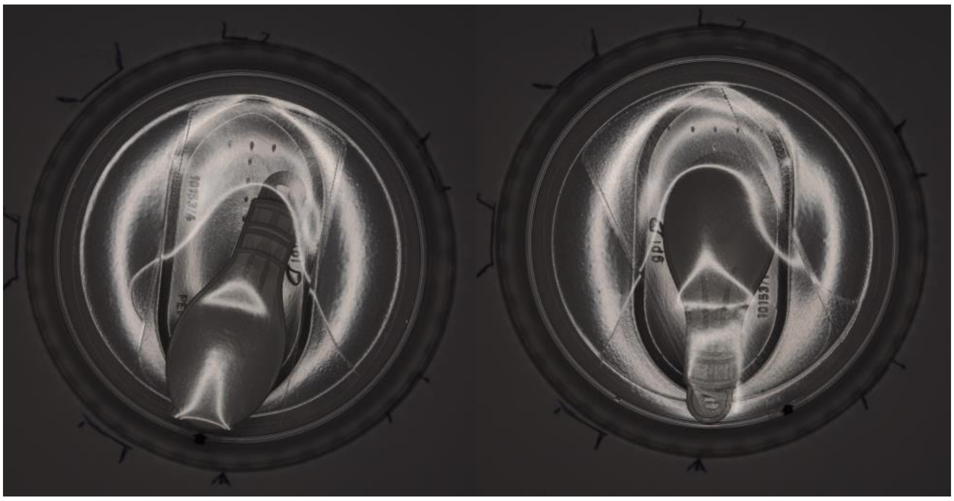


Figure 4.4: Type II defected lids; left: DL sample 31; right: DL sample 34

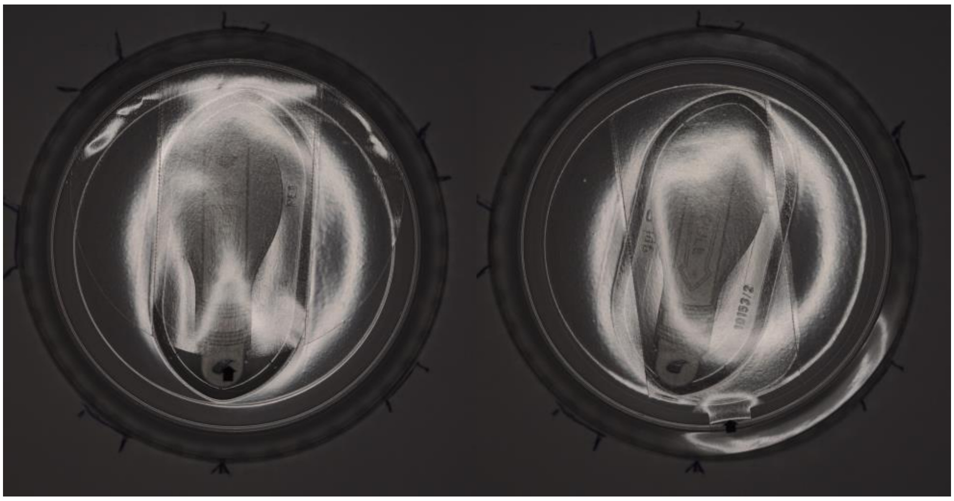


Figure 4.5: Type III defected lids; left: DL sample 30; right: DL sample 25

4.1.3. Algorithm for Defect Detection

The foregoing explanations make it obvious, that finding defects in transparent and reflective surfaces inhibits many difficulties. For this study, several approaches were tested to detect the difference between non-defected lids (NDLs) and defected lids (DLs). However, most of them performed very poorly. Due to the complicated structure, template matching was not an option. Classic thresholding failed, also because the lid's structure does not provide constant conditions for an algorithm to detect irregularities. These options do not even keep in mind the lighting difficulties caused by the surface.

For all the stated reasons, a completely new and unusual approach was chosen to be implemented in a machine vision defect detection algorithm. The main idea was creating a rotation-invariant algorithm, that is able to distinguish between NDLs and DLs lids by using a simple statistical test. Therefore, the hardware (Figure 4.1) was set up in a way, so that the obtained images were always taken from a 90° angle. In this way, one would always see a similar amount of light reflected, when the lid is not defected. For DLs, however, a crinkled foil or misplaced spoon would severely change the reflection of the flashlight. This change is easily visible to the human eye, but also a computer can be trained to recognize such a difference. When an algorithm only analyses the intensity distribution of the lid itself, ignoring the surrounding background, a clear difference between NDLs and DLs can be noted. This difference of the distributions is recognised by the χ^2 test. The algorithm consists of the following four main steps:

1) Create a circular-shaped mask
2) Analyse intensity distribution of the circular shaped mask
3) Scale and shift distributions
4) (Calculate a mean distribution from non-defected lids)
5) Compare distributions using the χ^2 test

The first step of creating a mask is necessary to exclude the background from being processed for both precision and computing time reasons. The mask is created by firstly applying a Sobel filter and then using a Hough Circle detection. The Sobel filter is a typical edge detection algorithm, that highlights stark spatial differences of intensity. A detailed explanation of its functionality is given in chapter 2.2.9. In this way, the circular edge of the lid is highlighted, resulting in a faster calculation of the Hough Circle detection, that searches for such conspicuous circles within a certain threshold. The detected circle is then used to create a mask, ignoring pixels outside of the lid's circle from further processing.

Afterwards, all the pixels within the circle are analysed on their intensity, by simply adding every occurring value in the corresponding intensity bin. Regarding the bit depth of the camera, the desired precision and the low time availability that was aimed for, the values were put in 256 bins.

The third step was necessary for two reasons. First, all the obtained distributions had to be shifted in a way, where all the distributions would show the first occurring value at the same bin. Visually speaking, this was an adaption to change the brightness of the picture. Even though not desired, this step was vital for further calculations, because the battery-driven low-quality flashlight emitted less light on the lids that were photographed at last, than on the ones photographed in the beginning, resulting in shifted distributions. The second reason is the precision of the χ^2 test. The sensitivity of the test has to be adjusted by dividing the counts of every bin with the same number. A detailed explanation for this procedure is given in the chapter on Inhomogeneous Illumination 3.1.2.

Finally, a mean of all the NDL distributions was calculated. In further tests this was optimized to use only 15 distributions, obtaining the same results as with using all 63 NDL distributions. The 15 distributions that were used for the calculation of a mean distribution were cross-checked by comparing every single one of them to a mean of the other ones to ensure equality of the distributions.

The mean distributions calculated by ten NDL distributions was then subsequently compared to the DL distributions using the χ^2 test. A threshold for the p-value was set, marking distributions with a p-value under 0,05 as defected lids. The results are explained in detail in the following chapter 4.1.4. A detailed overview for the algorithm is given in the following section.

Pseudo-code for Defect Detection of Circular Shaped Lids

This pseudo-code is a simplified version of the code that was used for defect detection. The functions in the top section contain the main procedures. The main code with the sequence of the functions can be found in the bottom section.

```
function SetImageMatrix (ImageMatrix : TDouble2DArray; Mask : TBool2DArray);
begin
ImageMatrix := GetCubeLayer (RedLayer);
for every pixel do
  if pixel is part of Mask then
    set intensity to 0;
end;
```

```
function SobelFilter (ImageMatrix : TDouble2DArray);
begin
Apply SobelFilter to ImageMatrix;
end;
```

```
function CreateCircularShapedMask (Image : TDouble2DArray; LidRadius: integer;
                                   Mask : TBool2DArray);
var
HoughImage, CircleParameters : TInt2DArray;
begin
HoughCircleTransformation (Image, LidRadius, HoughImage);
DetectHoughCircleCenters (HoughImage, CircleParameters);
for every pixel do
  if pixel is not part of CircleParameters then
    add pixel to Mask;
end;
```

```
function AnalyseFrequencyDistribution (Image : TDouble2DArray;
                                       var FreqDist : TIntArray);
begin
for every pixel do
  FreqDist [intensity] := FreqDist [intensity] + 1;
end;
```

```
function ScaleIntensityDistribution (FreqDist : TIntArray;
                                     ScalingFactor : integer);
for intensity := 0 to 255 do
  begin
  FreqDist [intensity] := FreqDist [intensity] div ScalingFactor;
  end;
```

```
function SetShiftProperties (FreqDist : TIntArray;
                             var LowValueBin : integer);
begin
for intensity := 1 to 255 do
  begin
  if (FreqDist [intensity] <> 0) and (FreqDist [intensity] < LowValueBin) then
    FreqDist := LowValueBin;
  end;
end;
```

```
function ShiftIntensityDistribution (FreqDist : TIntArray; LowValueBin : integer);
begin
for intensity := 0 to 255 do
  begin
  if (FreqDist [intensity] <> 0) and (intensity > LowValueBin) then
    begin
    ShiftValue := intensity - LowValueBin;
    stop loop if ShiftValue <> 0;
    end;
  end;
if ShiftValue <> 0 then
  for intensity := 0 to 255 do
    begin
    FreqDist [intensity-ShiftValue] := FreqDist [intensity];
    end;
end;
```

```
function CalculateMeanIntensityDistribution (NumberOfNonDefectedSamples,
                                             pValueThreshold : integer;
                                             var MeanDist : TIntArray);
var
  pThres                    : bool;
  TestArray, Resultvector   : TIntArray;
  TestMatrix                : TDouble2DArray;
begin
while pThres = true do
  for SampleNumber := 1 to NumberOfNonDefectedSamples do
    begin
    TestArray := FreqDist (SampleNumber);
    for FillTestMatrix := 1 to NumberOfNonDefectedSamples do
      begin
      if (FillTestMatrix <> SampleNumber) and (FillTestMatrix <> DeletedArray) then
        begin
        TestMatrix [FillTestMatrix] := FreqDist (SampleNumber);
        end;
      MeanArray := CalculateMeanVector (TestMatrix)
      ResultVector [SampleNumber] := ChiSquareTest (TestArray, MeanArray);
      end;
    SortResultVector;
    if ResultVector [max] < pValueThreshold
      then begin
           pThres := true;
           end
      else begin
           DeleteResultvector [max];
           end;
    end;
    MeanDist := CalculateMeanVector (ResultVectorMatrix);
end;
```

```
//Main Code
var
  LidRadius, ScalingFactor, LowValueBin, pValue : integer;
  ImageMatrix                                   : TDouble2DArray;
  Mask                                          : TBool2DArray;
  FreqDist, MeandDist, TestArray                : TIntArray;
begin
Set LidRadius;
Set ScalingFactor;
Set pValueThreshold;
for i := 1 to NumberOfSamples do
  begin
  SetImageMatrix (ImageMatrix, Mask);
  SobelFilter (ImageMatrix);
  CreateCircularShapedMask (ImageMatrix, Mask);
  SetImageMatrix (ImageMatrix, Mask);
  AnalyseFrequencyDistribution (ImageMatrix, FreqDist(SampleNumber));
  end;
for SampleNumber := 1 to NumberOfSamples do
  begin
  FreqDist := FreqDist (SampleNumber);
  ScaleIntensityDistribution (FreqDist, ScalingFactor);
  end;
for SampleNumber := 1 to NumberOfNonDefectedSamples do
  begin
  SetShiftProperties (FreqDist, LowValueBin);
  end;
for SampleNumber := 1 to NumberOfSamples do
  begin
  FreqDist := FreqDist (SampleNumber);
  ShiftIntensityDistribution (FreqDist, LowValueBin);
  end;
CalculateMeanIntensityDistribution (NumberOfNonDefectedSamples,
                                    pValueThreshold, MeanDist);
for SampleNumber := 1 to NumberOfSamples do
  begin
  TestArray := FreqDist (SampleNumber);
  pValue := ChiSquareTest (TestArray, MeanDist);
  WritePValueToTxtFile;
  end;
end.
```

4.1.4. Results

After applying the algorithm, a χ^2 value and a p-value was obtained for every comparison of a DL distribution with the mean distribution. The results are shown in Table 4.2.

Table 4.2: Results of χ^2 test: mean distribution compared to DL

Sample	χ^2 Value	p-Value		Sample	χ^2 Value	p-Value
1	50,69	1,00		20	1991,66	0,00
2	208,76	0,03		21	2313,11	0,00
3	162,85	0,74		22	2456,08	0,00
4	1415,09	0,00		23	2401,64	0,00
5	859,97	0,00		24	1929,98	0,00
6	2063,09	0,00		25	283,59	0,00
7	2375,36	0,00		26	1894,14	0,00
8	2334,67	0,00		27	119,50	0,99
9	2365,76	0,00		28	1059,24	0,00
10	2034,41	0,00		29	2029,45	0,00
11	3123,31	0,00		30	1850,72	0,00
12	657,91	0,00		31	1646,32	0,00
13	1145,89	0,00		32	1958,93	0,00
14	2053,55	0,00		33	1324,09	0,00
15	1408,07	0,00		34	1220,37	0,00
16	1030,58	0,00		35	1405,53	0,00
17	2865,61	0,00		36	1195,48	0,00
18	2222,91	0,00		37	979,12	0,00
19	1575,09	0,00				

After careful examination of the results, it is revealed that the algorithm detects 34 of the 37 DL as defected. This corresponds to a type II error percentage of 8,12%. The three lids that were not detected are shown in Figure 4.6. The reason for this rather contradictory result is still not entirely clear. One possible explanation is that random reflections account for a resemblance of the distribution of NDL and thus forbid detection using the presented algorithm. The images of all DL can be found in the appendix on pages 74-76.

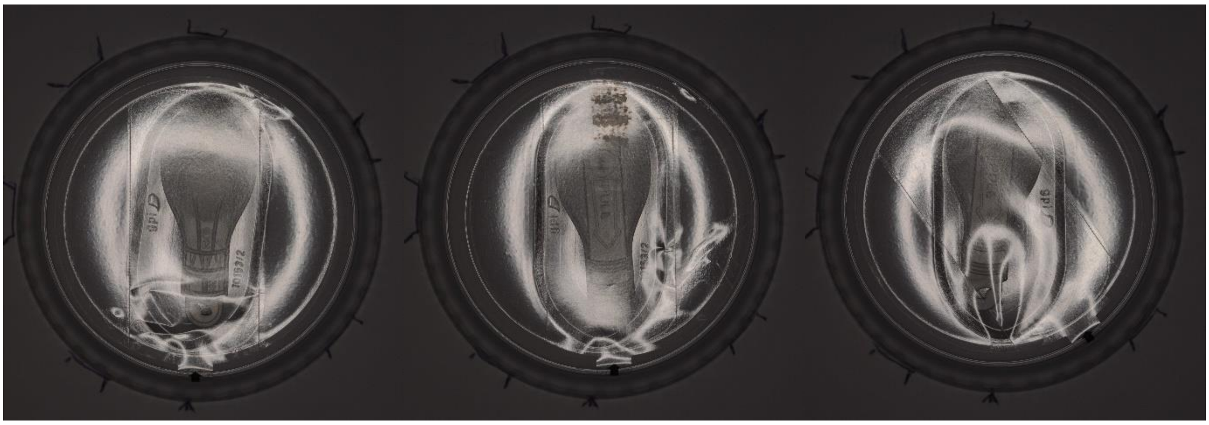


Figure 4.6: Not detected DL: left: DL sample 1; middle: DL sample 3; right: DL sample 27

The first question one might ask, is how to improve the quality of the algorithm. When using all 63 NDL for the calculation of the mean distribution, the p-value of sample 3 is lowered to 0,22. The p-values of the samples 1 and 3 however stay the same, meaning that increasing the samples taken for the NDL mean distribution not necessarily improves the selection process.

Another important fact to keep in mind, is the scaling of the distributions. Lowering the total number of counts per distributions indeed increases the precision of the statistical test, however, the three non-detected DL are still not recognized to be defected. Nevertheless, NDLs, such as sample 39 and 43 are then detected as defected, resulting in an augmented waste rate, because NDL are falsely recognised as defected. Thus, surely the easiest way to generate better results is to invest in adequate hardware, as these tests were conducted with rather low-quality equipment.

Additionally, all the NDL distributions that were not used for the mean distribution, were tested as well. These results are shown in Table 4.3, clarifying that no NDL was falsely detected to be defected.

Table 4.3: Results of χ^2 test: mean distribution compared to NDL

Sample	χ^2 Value	p-Value
16	16,69	1,00
17	27,88	1,00
18	77,05	1,00
19	51,29	1,00
20	48,53	1,00
21	49,88	1,00
22	41,92	1,00
23	26,74	1,00
24	103,29	1,00
25	56,19	1,00
26	75,42	1,00
27	95,31	1,00
28	69,99	1,00
29	112,54	0,99
30	81,42	1,00
31	92,07	1,00
32	123,39	1,00
33	95,43	1,00
34	138,15	0,98
35	92,04	1,00
36	68,55	1,00
37	101,57	0,99
38	79,63	1,00
39	170,07	0,53
40	73,51	1,00
41	87,09	1,00
42	29,32	1,00
43	202,09	0,06
44	73,14	1,00
45	98,81	1,00
46	91,94	1,00
47	80,35	1,00
48	66,44	1,00
49	149,08	0,87
50	77,66	1,00
51	61,52	1,00
52	135,29	0,98
53	70,99	1,00
54	27,69	1,00
55	99,04	1,00
56	90,70	1,00
57	191,55	0,19
58	128,38	0,99
59	88,59	1,00
60	99,39	1,00
61	118,43	0,99
62	132,81	1,00
63	109,99	0,99

The computing time for the analysis of one image was roughly 35 seconds. Due to fast production rates of packaging products, processing time should be less than half a second.

Merely by reducing the image's resolution by half, the computing time was reduced to 18 seconds. As the calculation was done using a regular consumer notebook, the time required for the calculation can certainly be cut drastically by optimizing both the hardware and vital software procedures.

In conclusion, it is emphasized that the power of the selection process heavily depends on the required quality demands and thus the chosen settings. Naturally, a high rate of correctly detected defects corresponds to a higher rate of falsely sorted out NDL, resulting in a higher overall waste rate. All the same, this is rather an economical question than a technical one. Computing time was not satisfactory, nevertheless, optimization potential strongly suggests that the chosen approach offers compelling benefits.

4.2. Defect Detection of Opaque Parts

There are few lines of business in the producing sector, where polymer materials do not play an important role in manufacturing. When glossy polymers are applied in products, they usually get covered by an extra layer of polymer foil to prevent scratches or bumps that damage the product and resultingly lower the visual experience and sensation for the consumer. Other products, such as medical supplies, require flawless surfaces for hygienical reasons. A primary and challenging problem is the assurance of producing flawless, reflective and glossy polymer surfaces. One has to keep in mind that these products are also subject to receiving surface defects during production. Thus, quality control naturally urges machine vision technologies to find such defects.

4.1.1. Image Acquisition

The algorithm that is subsequently introduced, heavily depends on uniform illumination over the whole area of the parts. The defects of the parts are only visible, both for the eye and the camera, when both the viewer and the light source are located in a certain angle to the object. Accordingly, choosing the hardware and setting up the latter is an elaborate task.

Ideally, the sample would be imaged using a scanner, so that every area of the part receives the same amount of illumination. The constant close distance of the scanner's light source to a flat part would ensure even illumination. Since this is a rather impractical solution for high velocity industry applications, the more difficult approach of using a camera setup was chosen. After various experiments, illumination was identified as the most challenging difficulty.

The best results were obtained using a setup that contained the following parts, its arrangement shown in Figure 4.7:

- Nikon D90 digital single-lens reflex camera with a 23,6 mm x 15,8 mm CMOS sensor and a resolution of 4288 x 2848 pixels (12,3 megapixels)
- Nikorr 50 mm (fixed focal length) lens
- High-end Profoto B1X flashlight
- Diffusor
- Nikon SB 900 flashlight
- Remote trigger control
- Riser for camera mounting

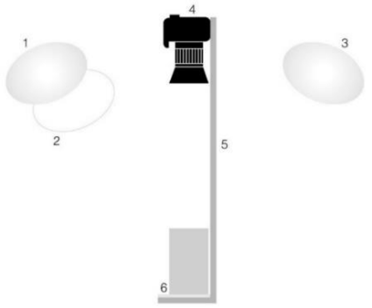


Figure 4.7: Schematic drawing of experimental setup for defect detection of the opaque parts
1: Profoto B1X flashlight; 2: diffusor; 3: Nikon SB 900 flashlight;
4: Nikon D90 with Nikorr 50mm; 5: Riser for camera mounting; 6: investigated sample

The investigated parts were placed on a white sheet, in order to obtain an even and easy-to-handle background. The camera was mounted above the parts and focused on the area of interest. Flashlights were placed on both sides of the sample. In order to achieve a more uniform illumination, a diffusor was put in front of the high-performance flashlight. The latter was triggered using the camera, the consumer flashlight was triggered in slave mode, a setting where the flash is released when detecting light from another flashlight, in this case the high-performance flashlight. A wide variety of flash settings were tested to achieve the most uniform illumination possible. All images were taken with the camera settings shown in Table 4.4.

Table 4.4: Settings for image acquisition of opaque parts

Shutter time	Aperture	ISO	File format
1/200 sec	10	200	RAW

Similar to the lids, the images were taken in RAW format and afterwards converted into TIFF-files using IrfanView. Eventually, the images were imported into ImageLab, where the post-processing defect detection algorithm was applied.

4.1.2. Types of Defects

The key to understanding surface defects of reflective objects lies in basic physics knowledge. Figure 4.8 depicts specular reflection of flat surfaces and diffuse reflection of rough surfaces. Generally, the angle of reflection is the same as the angle of incidence. Glossy polymers reflect the light as shown in Figure 4.8 (left), giving them their mirror-like properties. If the surface is damaged for whatever reason during production, the reflection becomes diffuse, as depicted in Figure 4.8 (right). Therefore, to be able to recognize such defects, one must look at the object in the right angle. This phenomenon can easily be tested using any glossy surface and turning it under fixed spot light. Depending on the angle of the viewer and light source, more or less details of the surface will be visible.

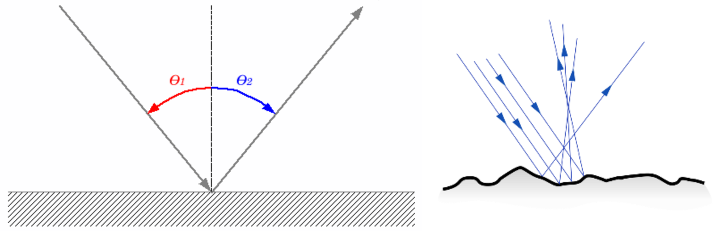


Figure 4.8: Left: reflection of light on flat surfaces; right: diffuse reflection [31] *of light on rough surface*

Mechanical forming processes of thermoplastic polymers, such as injection moulding or deep drawing, always depend on an adequately set temperature to process the polymers without defects. If that is not the case, scratch-like defects are caused. The latter are denoted as *silver stains* and highlighted in Figure 4.9 and Figure 4.10.

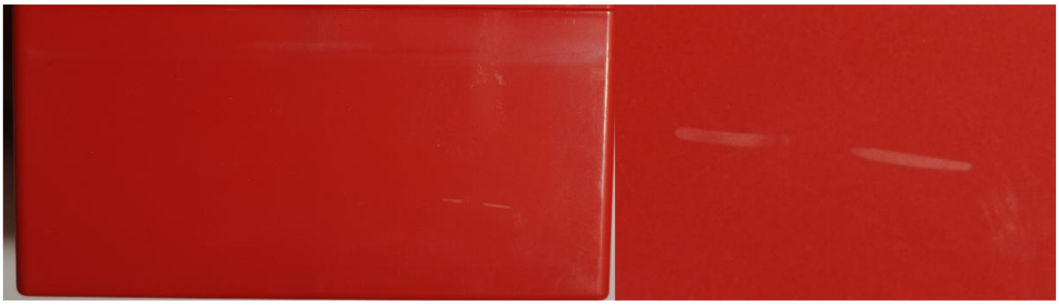


Figure 4.9: Left: red part with silver stain; right: zoomed in on silver stain

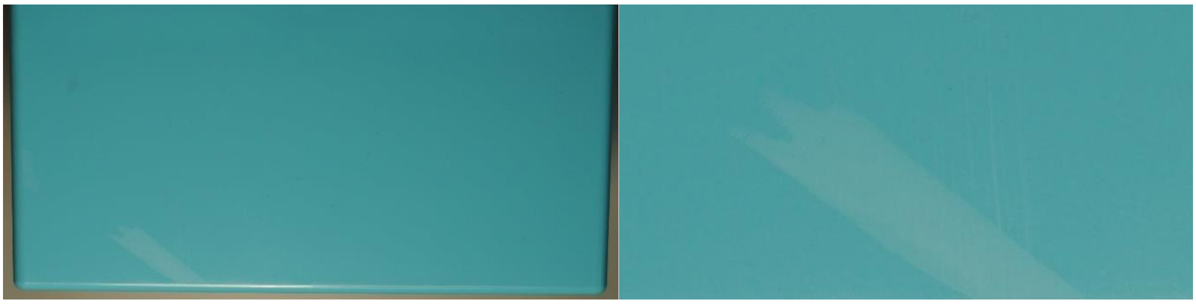


Figure 4.10: Left: cyan part with silver stain; right: zoomed in on silver stain

Aggravatingly, the silver stains should be detected on parts of different colours. Figure 4.11 shows the given samples, which are blue, cyan, pink and red parts.

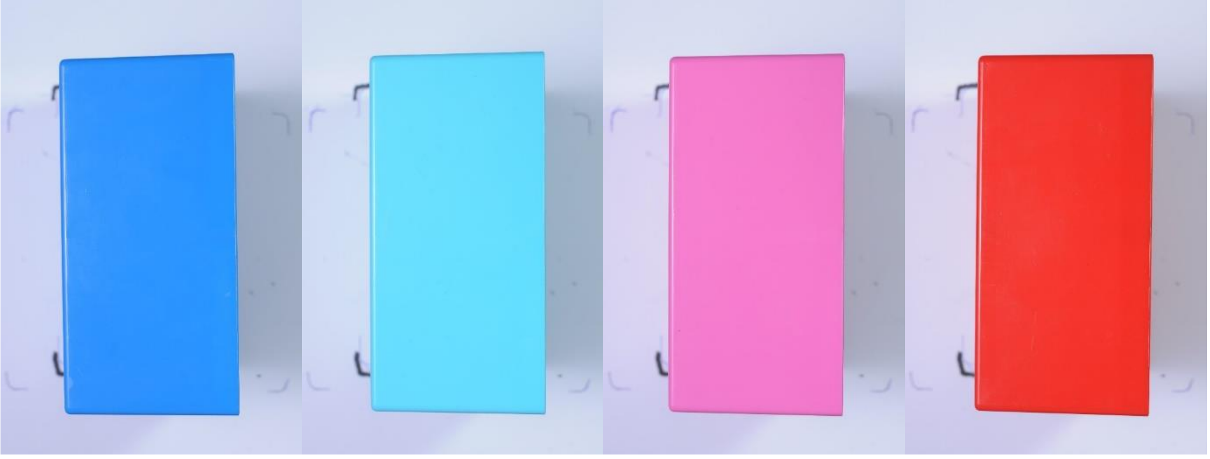


Figure 4.11: Left: blue sample part; middle left: cyan sample part; middle right: pink sample part; right: red sample part

Although there are several defects that are likely to occur, this work focusses on the detection of silver stains. Figure 4.12 depicts the red, green and blue layer of the coloured image of the cyan sample part, previously shown in Figure 4.10.

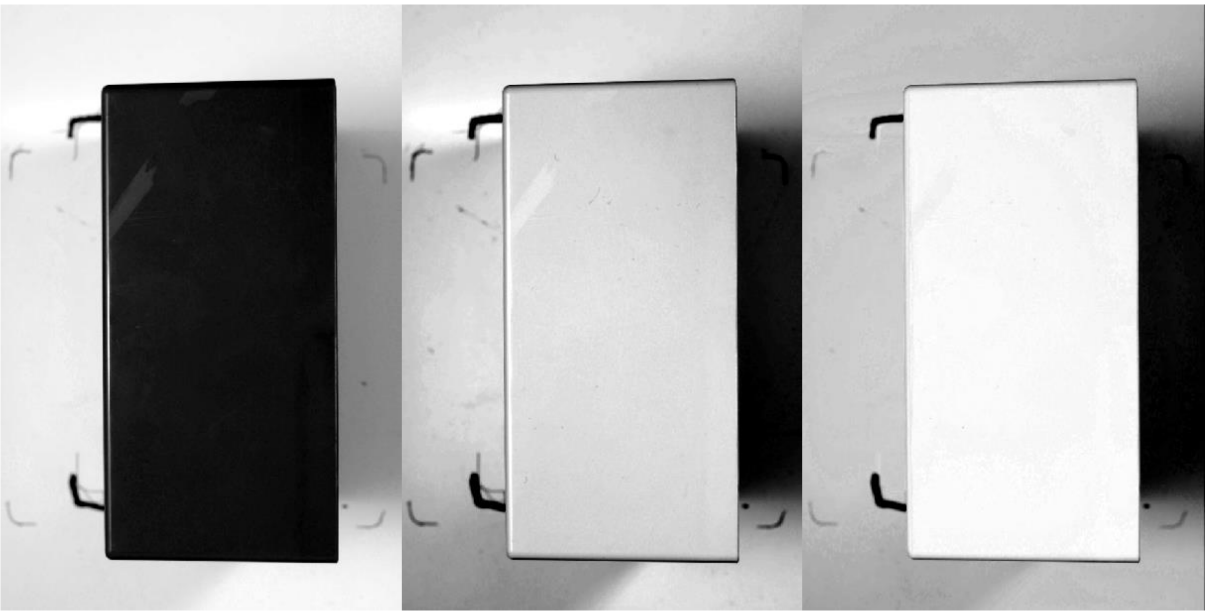


Figure 4.12: Left: red layer of cyan sample part; middle: green layer of cyan sample part; right: blue layer of cyan sample part

As a first approach, Figure 4.12 indicates that the detection of a silver stain defect heavily depends on the feature of taking coloured images. This is due to the fact, that when the coloured image is split into its three layers, depending on the colour of the part, the defect is visible on a different layer. To clarify said observation, Figure 4.13 shows the three layers of a red sample part, with the defect visible only in the green and blue layer, where the light intensities are naturally low. Concurrently, the silver stain in the cyan part is best visible in the red layer. Hence, one must take advantage of the image that is the complementary colour of the actual part colour.

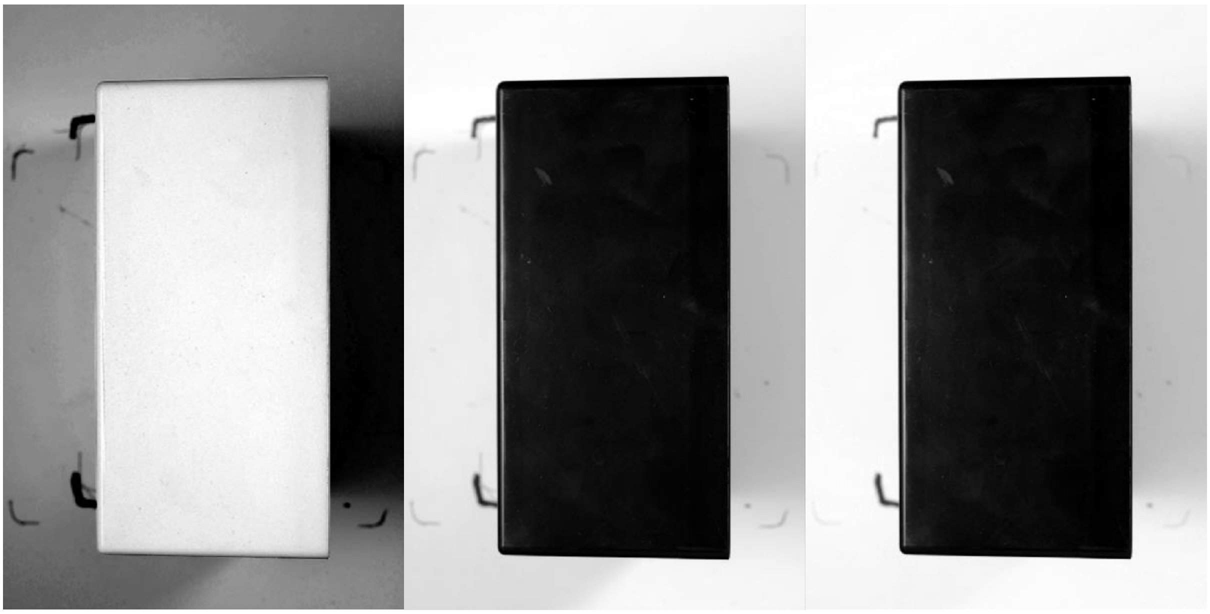


Figure 4.13: Left: red layer of red sample part; middle: green layer of red sample part; right: blue layer of red sample part

Unfortunately, simply choosing the right layer for a given part colour and applying an intensity threshold is not the solution to build a stable algorithm. Partly, this is due to the rather low differences of the intensity of defected pixels compared to the background. Figure 4.14 may indeed suggest, that the pixels, where the defect is located, show a higher intensity than non-defected pixels.

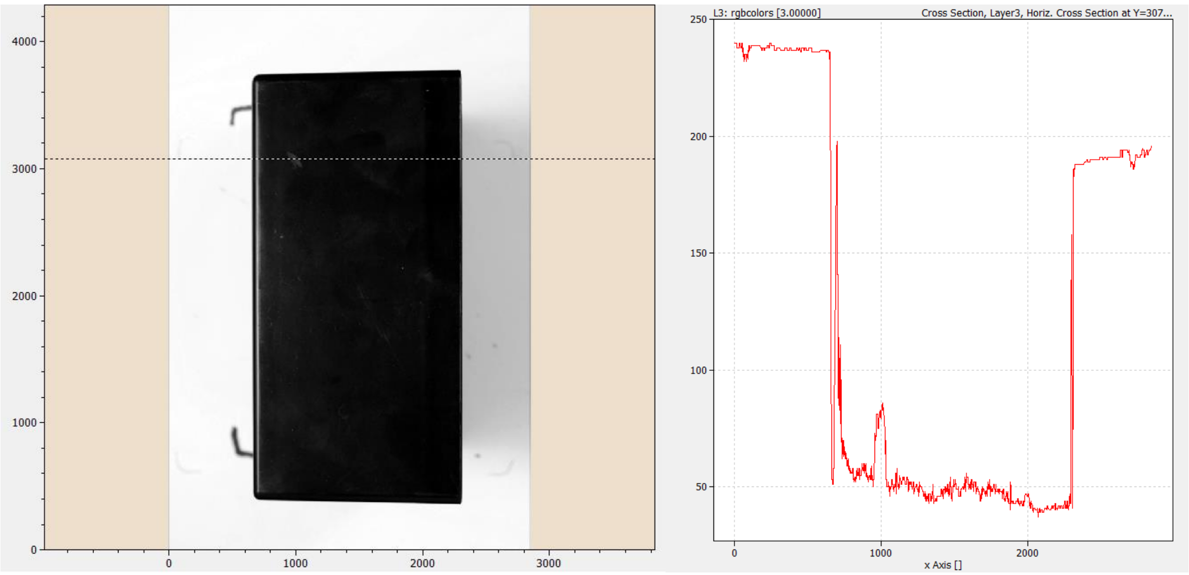


Figure 4.14: Left: green layer of red sample part with intensity cross section; right: intensity distribution of cross section

Nonetheless, the assumption of applying a threshold within the area of the part and subsequently finding defects has proven to be unsuccessful. The reason can be found in the baseline of the non-defected area of Figure 4.14. Even with a high-performance flash light, it was quite difficult to accomplish roughly the same power of illumination over the whole area of the part. Figure 4.15 shows the 3D surface plot of the part from Figure 4.14. It clarifies that the most insidious problems of finding silver stain defects lies within the proper illumination setup. To be able to distinguish defected areas from non-defected areas, the illumination should be so consistent, that the 3D surface plot shows a flat surface. Hence, a defected area, qualified by an incline or decrease of intensity, would light up in such a plot.

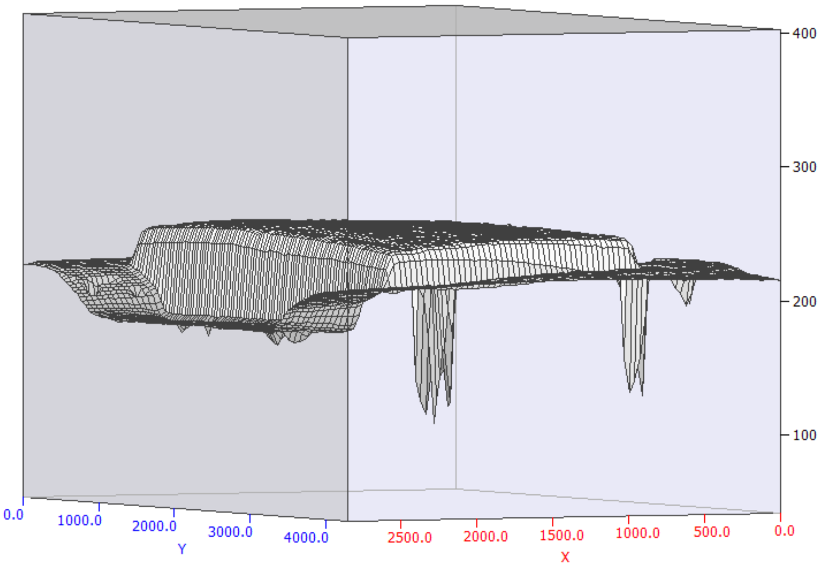


Figure 4.15: 3D surface plot of the blue layer of the red sample part

Another main reason for a consistent illumination is the overall size of a defected area. Parts that exhibit large defected areas, as the previously shown cyan sample part from Figure 4.12, may even be correctly detected by simple thresholding. Such a large area defect is depicted in a 3D surface plot in Figure 4.16. However, the detection of smaller silver stains has proven to be unsuccessful for reasons that were stated previously.

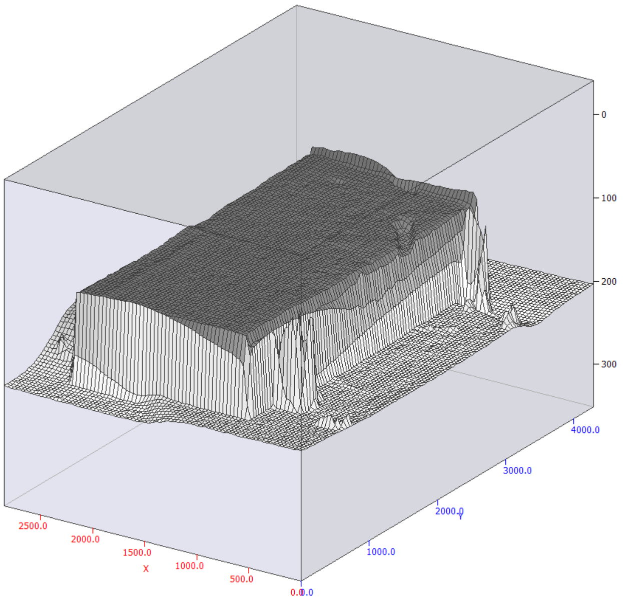


Figure 4.16: 3D surface plot of the red layer of the cyan sample part

In conclusion, the approach of thresholding the 3D plot was adjusted and developed to an extent, where it was possible to also find small area defects. The algorithm is presented in the following chapter.

4.1.3. Algorithm for Defect Detection

The algorithm for finding defects resembling scratches, such as the previously described silver stains, consists of seven main parts:

1) Create a background mask
2) Trim the image using the mask
3) Convert the RGB image into an HSI image
4) Model the saturation surface and subtract the result from the original image
5) Apply percentile filter
6) Dichotomize image
7) Apply threshold and detect defects

First, for each part colour the relevant *counter layer* with the lowest intensities was taken to distinguish the area of the part from the background. The layer where the defects were best visible is denoted as *counter layer*. Table 4.5 gives an overview for which part colour which layer was used.

Table 4.5: Image layers used for creating background mask of opaque parts

Part Colour	Red	Blue	Pink	Cyan
Counter Layer	Blue	Red	Red / Green	Blue / Red

The colours pink and cyan require the combination of two layers to achieve better mask quality. This is due to need of combining colours when working in the RGB colour space. The algorithm however is explained using the cyan part, previously shown in Figure 4.10 and Figure 4.12. Thus, Figure 4.17 shows the same image as Figure 4.12 (left), but with the background intensity set to zero using the background mask. The part itself appears to be brighter, this is due to shifting the colour scale for reasons of better visibility of the difference between background and colour.

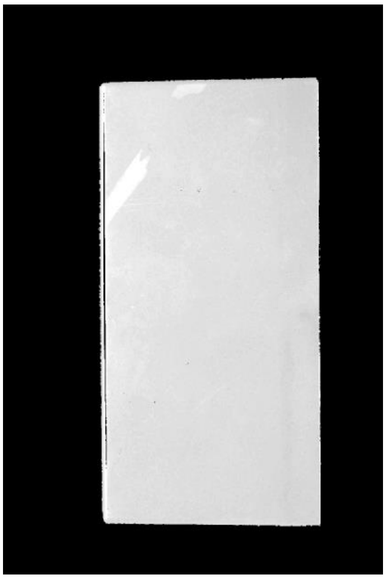


Figure 4.17: Red layer of the cyan part with applied background mask

Then, every pixel that is not hidden because of the background mask is detected as the part's surface. This is done using ImageLab's *Particle Detection* tool, which is ultimately a thresholding tool that unites neighbouring pixels that lie above or beneath a threshold to a so-called *particle*. The obtained *particle* naturally represents the part, because the background mask only hides the pixels surrounding the part. As every *particle* in ImageLab contains various properties, such as the centre, length and width of the *particle*, this information is used to trim the image. A rectangular area is cut out of the image, resulting in a smaller image that does only contain the area of the part. Naturally, this approach is rather unsatisfying, especially when the parts are not photographed in the same position. In a worst-case scenario, a rotated part would only be grasped partly. Nevertheless, it is assumed that for industrial applications the parts are imaged in the same position using a robot arm. In theory, the algorithm could be designed rotation-invariant, by using the mask to omit background pixels from further processing. However, this step was avoided to reduce computing time.

Figure 4.18 depicts the red, green and blue layer of the trimmed image with a cross-section line and the corresponding intensity distribution along the cross-section.

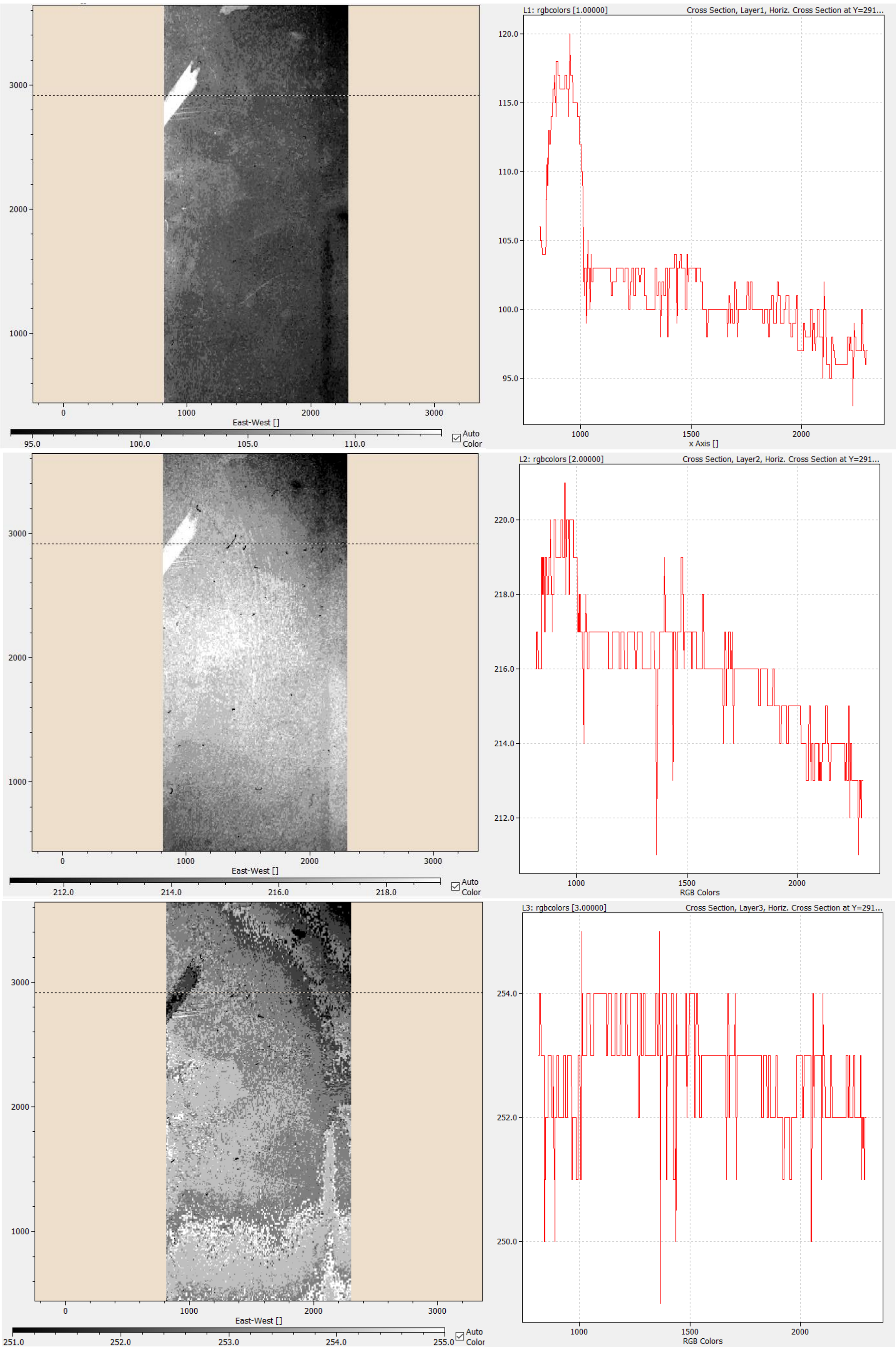


Figure 4.18: Top left: red layer with cross-section; top right: intensity distribution of the cross-section middle left: green layer with cross-section; middle right: intensity distribution of the cross-section bottom left: blue layer with cross-section; bottom right: intensity distribution of the cross-section

The cross-section intensity distributions of the different layers clarify, why the *counter layer* of the corresponding part colour was investigated for finding silver stains. Relatively, there the silver stains show a greater difference to the baseline of intensities than at the part's colour layer, where the peak lies within the baseline noise. However, thresholding such an image has not proven to be successful for lighter or smaller defects. Thus, for a more stable and powerful detection, post-processing was expanded.

The enhancement of the defect detection was achieved by handling the obtained image in a different colour space. Thus, the image consisting of the three RGB layers was converted into an HSI image. The latter consists also of three layers, shown in Figure 4.19, Figure 4.20 and Figure 4.21. These three images represent the hue, saturation and intensity values. A detailed explanation for the conversion can be found in Chapter 2.2.5, starting on page 8.

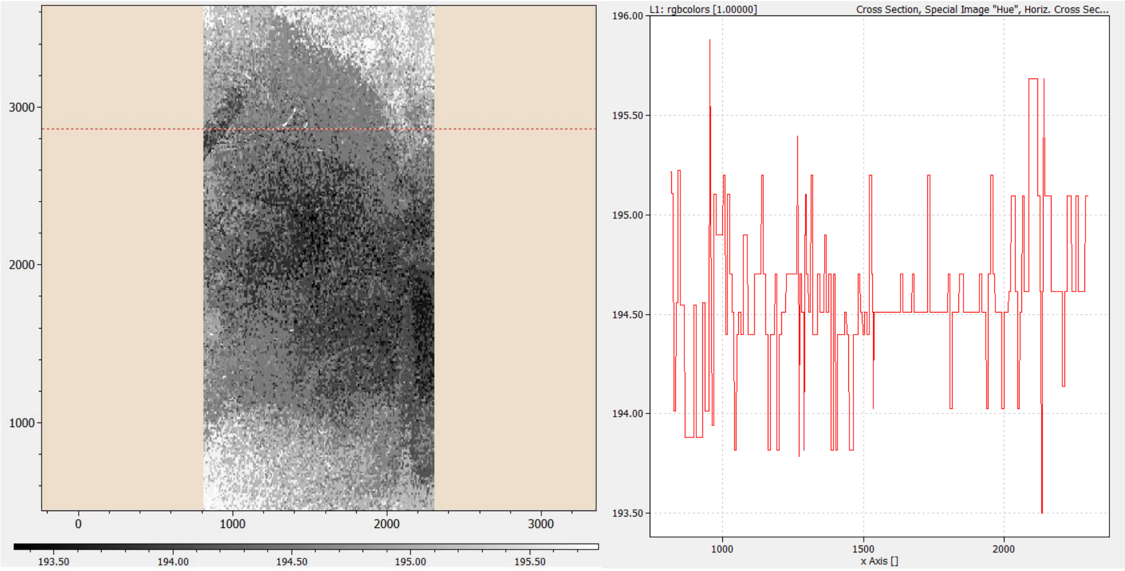


Figure 4.19: Left: hue image with cross-section; right: intensity distribution of the cross-section

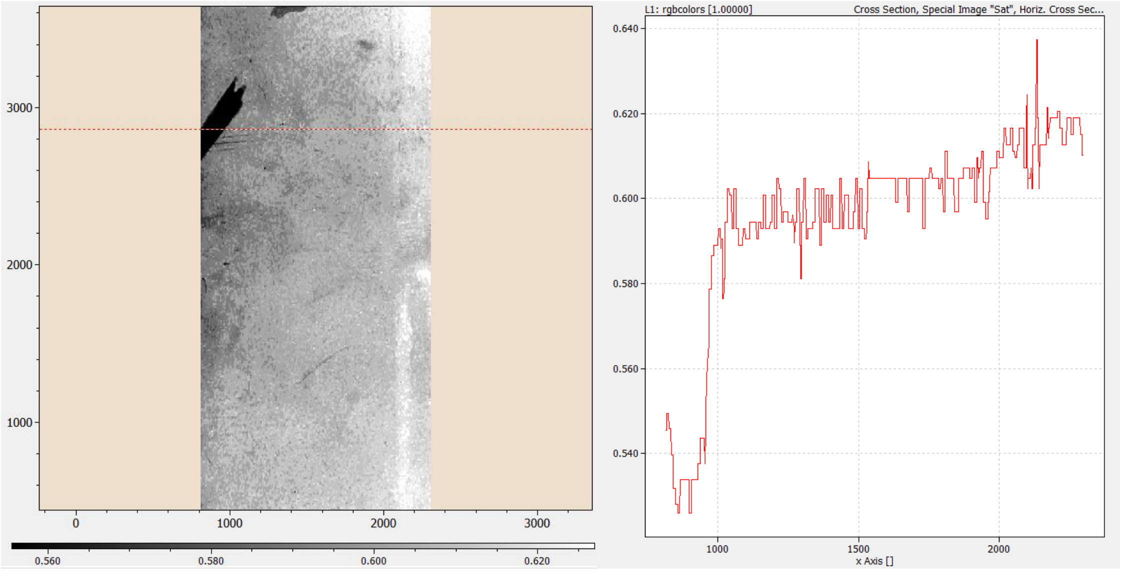


Figure 4.20: Left: saturation image with cross-section; right: intensity distribution of the cross-section

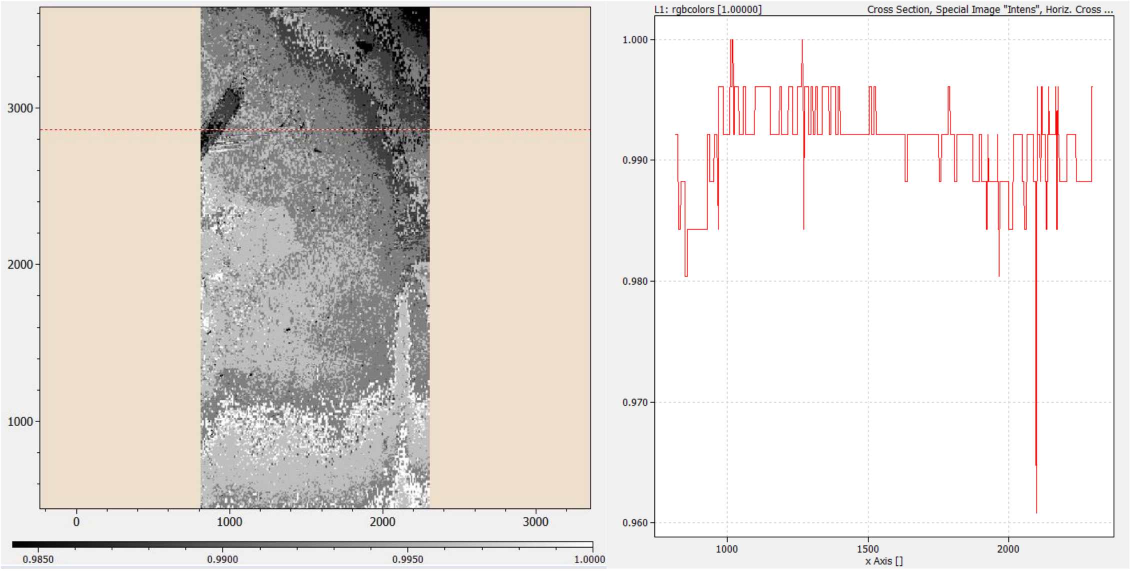


Figure 4.21: Left: intensity/value image with cross-section; right: intensity distribution of the cross-section

After close examination, Figure 4.20 depicts that the saturation image in HSI colour space promotes a quite clear image of defected areas of the part. The hue image representing the colour sphere and the value image representing the mixing with black or white, naturally do not offer any information with respect to the investigated defect and will thus be neglected in further processing.

The saturation image indeed may be suited for thresholding, nonetheless, when looking both at the image and cross-section distribution of Figure 4.20, one finds that illumination is not equal on all areas of the part. Naturally, a single spot light source emits a cone of illumination, which is visible in Figure 4.21. As the central point beyond the light source is closest, it receives the highest light intensity because it has to pass the fewest air molecules, that absorb small amounts of the radiation. Therefore, it is not surprising that even with the high-performance flashlight uneven illumination is imaged. This drawback might be reduced by using a light source with more lamps, however, due to difficulty of setting up the flashlights in a suitable geometry to the camera and investigated part, it is more than likely that this issue can never be fully resolved.

Consequently, a stable algorithm depends on the minimization of irregularities in illumination caused by the setup. When all the intensities of the image are drawn into a three-dimensional picture, they can be understood as a *base surface*, depicted in Figure 4.22.

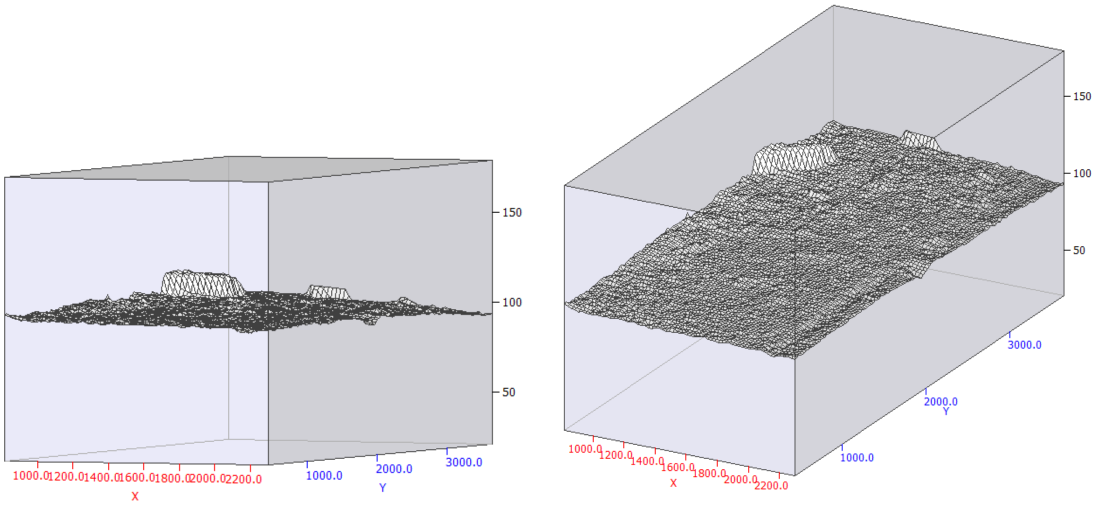


Figure 4.22: 3D surface plot of the saturation image from different angles

Hence, in the next step this *base surface* is modelled and then subtracted from the original image, enhancing the contrast between defect and surface. The modelling was done with a built-in function of ImageLab, which estimates a regression surface by applying a regular raster of pivot points over the whole parabolic or hyperbolic surface, suggested in Figure 4.23. Naturally, the number of set pivot points is a trade-off between several effects. The higher the number of pivot points, the better the mapping of the surface. However, simultaneously computing time will increase and more importantly, the risk of overfitting and thus hiding small defects as a part of the estimated surface is augmenting all the same. Hence, a suitable raster width has to be determined experimentally, with regard to the camera's resolution and the minimal size of defects that should be detected.

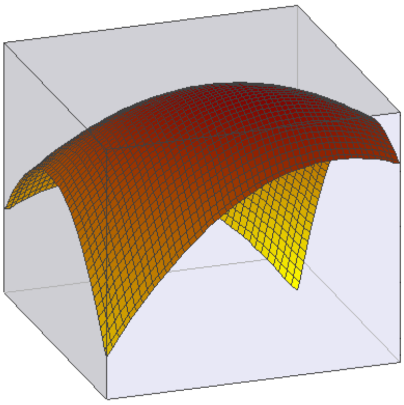


Figure 4.23: Regular raster of pivot points over a parabolic/hyperbolic surface [30]

In this study, the number of 1000 pivot points was set to calculate the estimated paraboloid, shown in Figure 4.24. The image appositely clarifies the uneven illumination over the area of the investigated object.

The next step consisted of calculating the difference of intensities of the saturation image (Figure 4.20) and the estimated paraboloid (Figure 4.24). The result is shown in Figure 4.25.

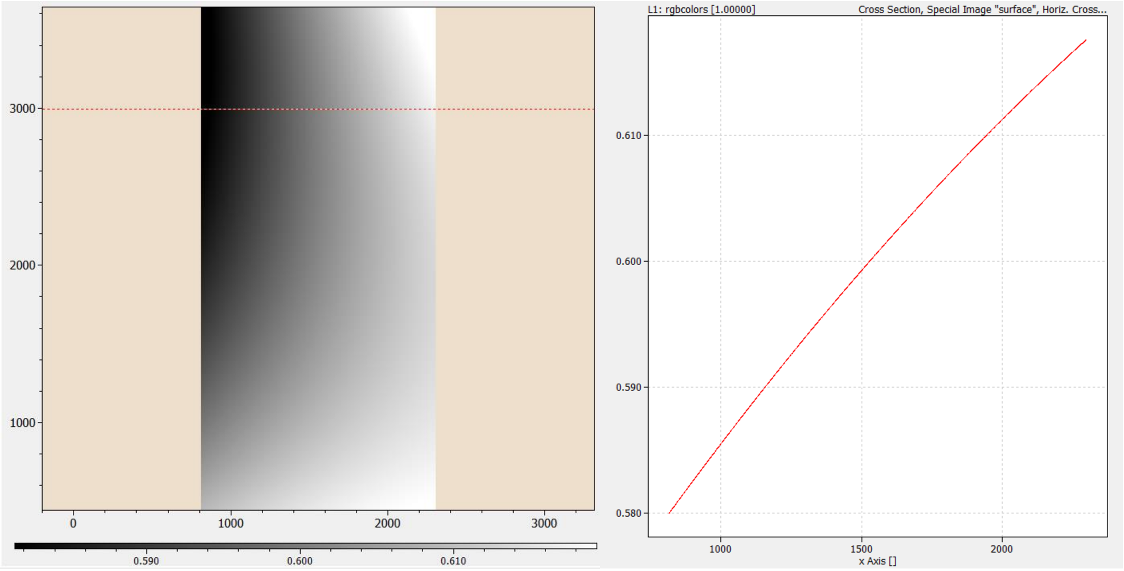


Figure 4.24: Left: estimated paraboloidal surface with a cross-section of the trimmed image; right: intensity distribution of the cross-section

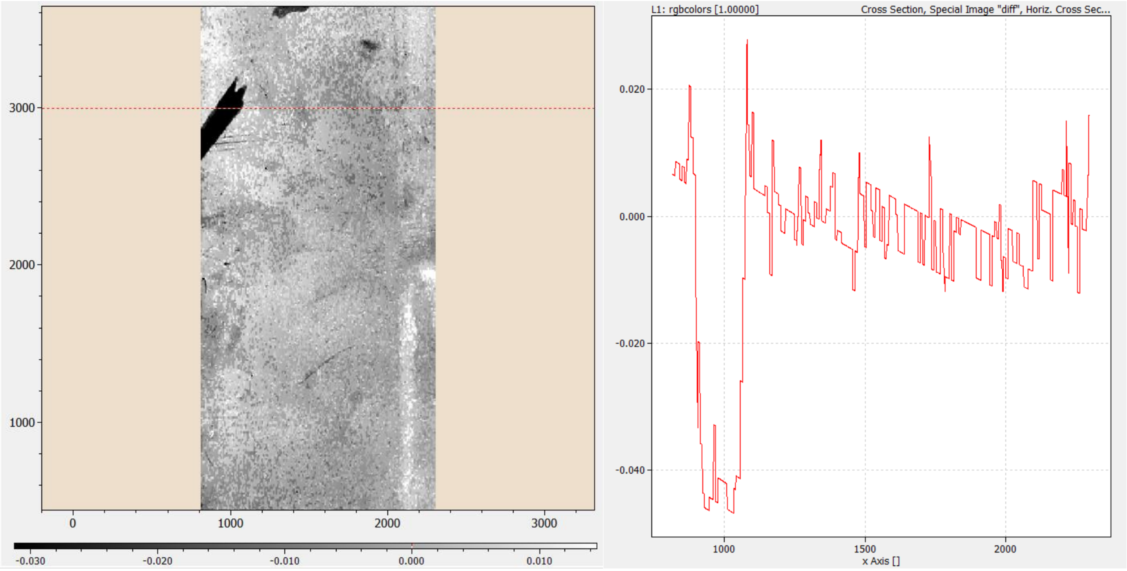


Figure 4.25: Left: difference of the saturation image and the estimated surface image with the cross-section; right: intensity distribution of the cross-section

Close examination of Figure 4.25 offers insight that the previously tilted surface and ascending cross-section were manipulated to be practically levelled out. As a result, the contrast between defect and noise of non-defected areas was improved.

In order to neglect possible intensity spikes and for soft surface smoothing reasons, a percentile filter was applied. The schematic procedure of applying such a kernel filter is shown in Figure 4.26. Detailed information about linear filters is given in chapter 2.2.9. Here, it is important to understand that the new value of the schematic blue matrix in Figure 4.26 is the chosen percentile of said matrix. During the operation extreme values are eliminated.

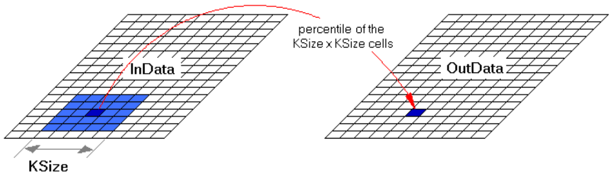


Figure 4.26: Schematic process of applying a spatial percentile filter [30]

After application of the percentile filter, the processed image, depicted in Figure 4.27, is obtained. Compared to the initial, unprocessed red layer image, it offers a better signal-to-noise ratio.

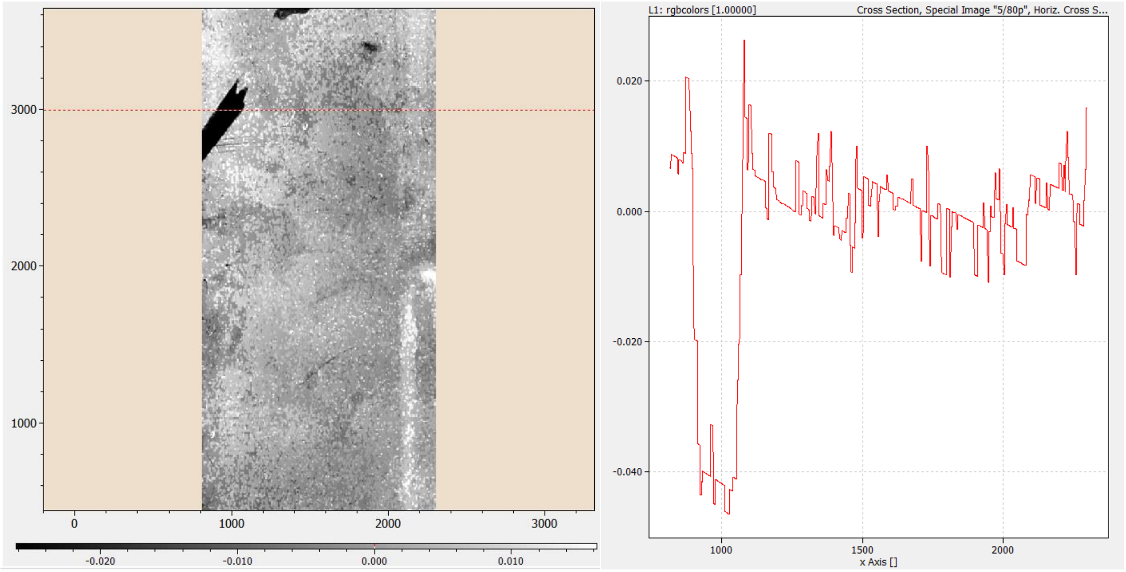


Figure 4.27: Left: contrast-enhanced image with applied percentile filter with cross-section; right: intensity distribution of the cross-section

Finally, the image of Figure 4.27 is ready for detecting the defects via thresholding. The last part of the algorithm consists of dichotomizing the image matrix. In this procedure, a threshold is applied. Depending on whether the value of a pixel lies below or above said threshold, a zero or one value is assigned to the prompted pixel. In this way, a 1-bit black and white picture, as shown in Figure 4.28, is obtained.

Figure 4.28: Dichotomized post-processed image of the cyan part

Afterwards, all connected pixels that credit a one value are detected as a *particle* using the *Particle Detection* tool. As previously explained, *Particle Detection* refers to a thresholding tool that unites neighbouring pixels that lie above or beneath a threshold to a so-called *particle*. In fact, the dichotomizing of the image is not necessary for a succeeding *particle* detection, especially, because the algorithm behind both processes is the same. However, for reasons of easy visibility and fast calculation, this method was chosen to highlight and conveniently visualize the defect detection for the user.

In this manner, a list of *particles* is obtained, where every *particle* exhibits information such as size, length, width and aspect ratio. By exploiting this information, it is possible to distinguish between small size *particles*, that are not actual defects but rather singular pixels with intensity spikes, and larger, actual defects. Furthermore, cut-offs for the various properties can be introduced, avoiding artefacts to be detected as defects. When introducing a reasonable cut-off criterion for the *particle* size, in this case a minimum size of 1000 pixels per *particle*, the remaining *particles* are then identified as defects. They can be added to an image mask, which is applied to the original photograph, depicted in Figure 4.29. Hence, the user receives visual feedback of the found defects.

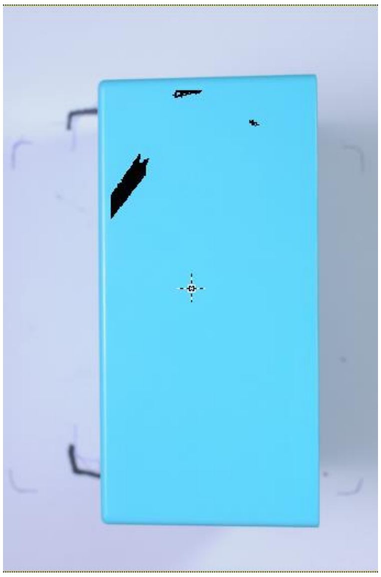


Figure 4.29: Marked defects on the original image of the cyan part

A detailed overview of the algorithm is given in the following section:

Pseudo-algorithm for Defect Detection of Opaque Parts

This pseudo-code is a simplified version of the code that was used for defect detection. The functions in the top section contain the main procedures. The main code with the sequence of the functions can be found in the bottom section.

```
function SetRedPixelMask;
begin
for every pixel do
      if blue light intensity < low then
        add pixel to mask;
end;
```

```
function SetBluePixelMask;
begin
for every pixel do
      if red light intensity < low then
        add pixel to mask;
end;
```

```
function SetPinkPixelMask;
begin
for every pixel do
      if red and green light intensity > high then
        add pixel to mask;
end;
```

```
function SetCyanPixelMask;
begin
for every pixel do
      if red and blue light intensity > high then
        add pixel to mask;
end;
```

```
function TrimImage;
begin
for every pixel do
      if intensity > 0 and not part of mask do
        detect as part of particle;
Trim image in x-dimension from (PartCenter-PartWidth) to (PartCenter+PartWidth);
Trim image in y-dimension from (PartCenter-PartLength) to (PartCenter+PartLength);
end;
```

```
function ConvertRGBToHSIImage (var SaturationImage : TDouble2DArray);
var
  RedLayer, BlueLayer, GreenLayer : TDouble2DArray;
  Hue, Saturation, intensity      : TDouble2DArray;
begin
RedLayer := GetCubeLayer (1);
BlueLayer := GetCubeLayer (2);
GreenLayer := GetCubeLayer (3);
ConvertRGBToHSIImage (RedLayer, BlueLayer, GreenLayer, Hue, Saturation, Intensity);
SaturationImage := Saturation;
end;
```

```
function EstimateParaboloidicSurface (SaturationImage : TDouble2DArray;
                                      var ParSurface : TDouble2DArray);

var
  PivotPoints : integer;
begin
PivotPoints := 1000;
ParSurface := EstimateParaboloid (SaturationImage, PivotPoints);
end;
```

```
function PostProcessing (DichotomizeThreshold : integer;
                         SaturationImage, ParSurface : TDouble2DArray
                         var ProcessedImage : TDouble2DArray);
var
  LevelledImage, FilteredImage : TDouble2DArray;
  KernelSize, Percentile       : integer;
begin
LevelledImage := MatrixSum (SaturationImage, -ParSurface);
KernelSize := 5;
Percentile := 80;
FilteredImage := SpatialPercentileFilter (KernelSize, Percentile, LevelledImage);
ProcessedImage := DichotomizeMatrix (DichotomizeThreshold, LevelledImage);
end;
```

```
function FindDefects (ImageMatrix : TDouble2DArray;
                       DichotomizeThreshold, DefectSize : integer);
begin
for every pixel of ImageMatrix do
      if intensity < DichotomizeThreshold and
        pixel is neighbour of previous pixel
          then begin
              add pixel to particle;
              end
          else begin
              add pixel to new particle;
              end;
end;
```

```
//Main Code
var
  SaturationImage, ParSurface       : TDouble2DArray;
  DichotomizeThreshold, DefectSize  : integer;
  ProcessedImage                    : TDouble2DArray;
```

```
begin
Set DichotomizeThreshold;
Set DefectSize;
case AskPartColour of
  1 : SetRedPixelMask;
  2 : SetBluePixelMask;
  3 : SetPinkPixelMask;
  4 : SetCyanPixelMask;
end;
TrimImage;
ConvertRGBToHSIImage (SaturationImage);
EstimateParaboloidicSurface (SaturationImage, ParSurface);
PostProcessing (DichotomizeThreshold, SaturationImage, ParSurface, ProcessedImage);
FindDefects (ProcessedImage, DichotomizeThreshold, DefectSize);
end.
```

4.1.4. Results

The detection of silver stains was tested on opaque parts of four different colours, namely red, blue, pink and cyan. For every colour, the part's sidewall that contained a conspicuous silver stain was imaged and processed with the defect detection algorithm. The number of investigated samples was limited to one part per colour, since resources were scarce and imaging the parts with professional photographer's equipment was done under high time pressure due to cost minimization. Furthermore, the scope of the study was confined to a preliminary scale, were the main aim was to show that such defects can be found in hard-to-handle surfaces.

The following figures show the results of the application of the algorithm. All figures contain one of the RGB layers in which the defect is easiest visible, the processed image where the found defects are shown after dichotomization and an image-stack of the original RGB with the size excluded defects. An image-stack is the combination of two or more images into one visualisation. It is very much like the fusion of different photograph layers in common image editing programs. In our case it is possible to stack the artificial processed image that contains the found defects on top of the original RGB image. Before the dichotomized image is stacked, a *particle* boundary of a minimum pixel size of 1000 per defect was set. In this way, small *particles* that are not actual defects were excluded from the defect detection process. This was a simple measure to enhance the results, no selection was done regarding length, width or aspect ratio. As an introduction to the results, Figure 4.30 shows the familiar cyan part sample that was used to explain the algorithm in the previous chapter. The three silver stains were fully detected, there was no false-positive detection of unsuspicious areas.

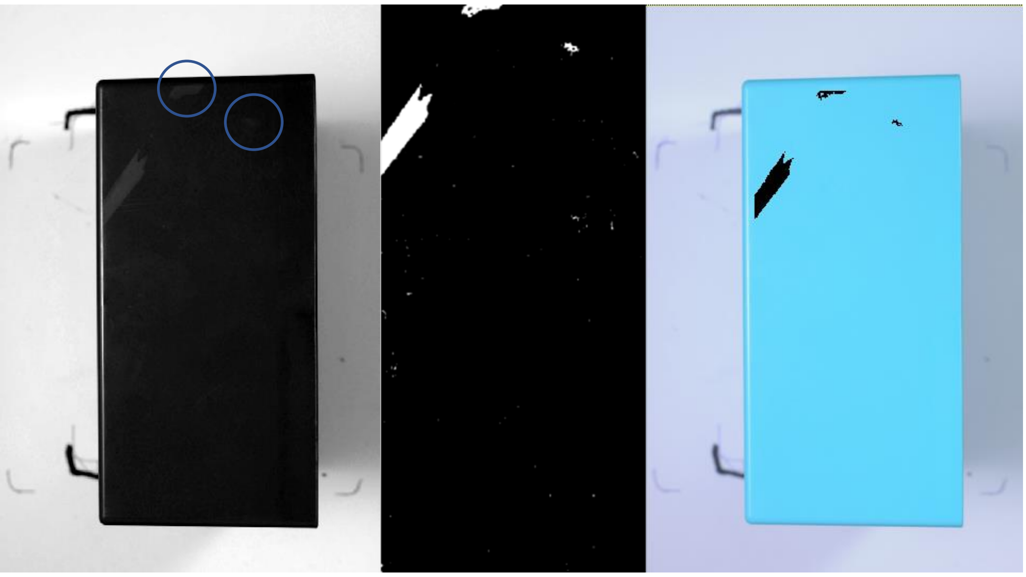


Figure 4.30: Cyan sample 1: left: red layer of the defected cyan part (original image); middle: dichotomized image of the detected defects; right: image-stack of the size-relevant defects and the original RGB photograph

Figure 4.31 shows a different cyan part sample. The two silver stripes next to the edge on the right were both detected. The small and thin silver stain, top right to the detected scratch on the left side was not detected. One can see that it was originally detected in the dichotomized image, but then eliminated from the defect list. This is due to its small size. In order to lower the risk of false-positive detection of defects, the size limit for defects must not be set too low. Here, another power of the algorithm is revealed. It does not only find silver stains, but similar defects like the big scratch on the left.

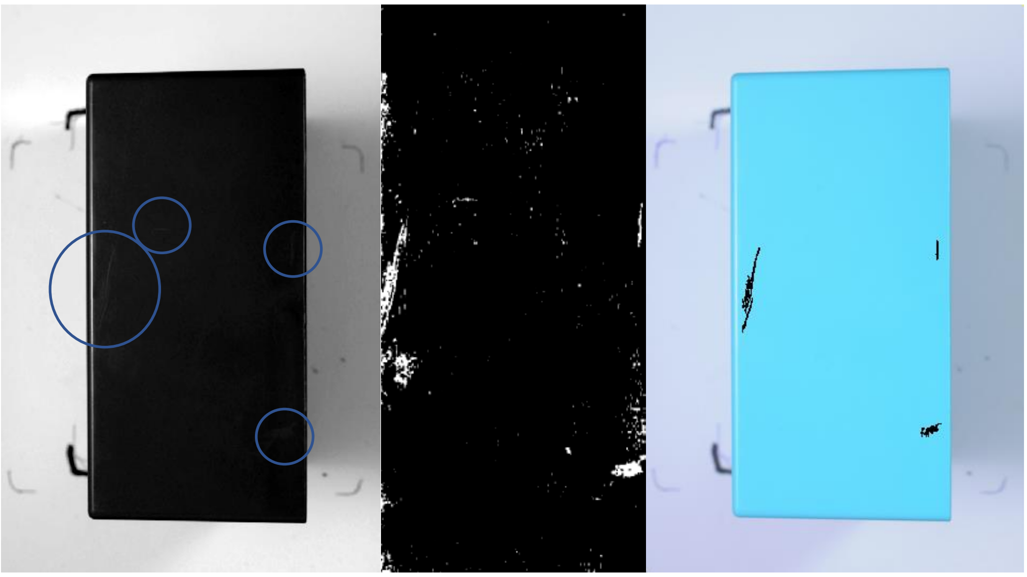


Figure 4.31: Cyan sample 2: left: red layer of the original RGB image; middle: dichotomized image of the detected defects; right: image-stack of the size-relevant defects and the original RGB photograph

After the analysis of the red part in Figure 4.32, another difficulty arose. In the previous example the found defects were decimated regarding their size. However, the detection can be adjusted before by modifying the sensitivity of the dichotomization. Figure 4.32 (middle) shows the analysis of the red part with the same threshold that was used for the cyan parts. It is obvious, that the detection of false-positive areas is quite extensive.

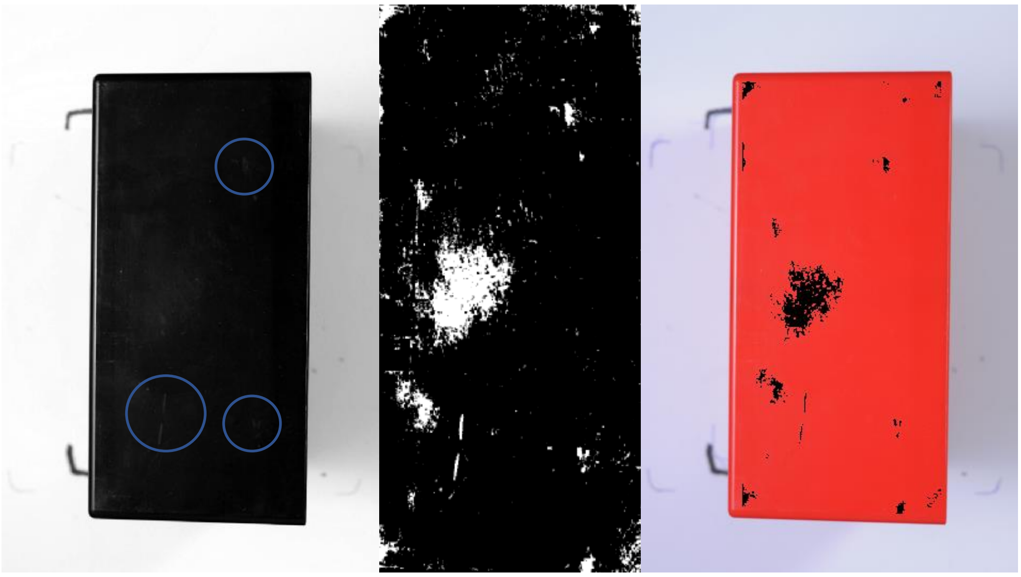


Figure 4.32: Red sample: left: blue layer of the original RGB image; middle: dichotomized image of the detected defects; right: image-stack of the size-relevant defects and the original RGB photograph

This behaviour leads to two conclusions. First, for the correct analysis of the defects of interest, the threshold must be experimentally examined for each part colour separately. Secondly, it is suspected that there is a relation between the part colour and the nature of the image sensor, which affects the sensitivity of the analysis. Since the CMOS sensor uses a Bayer filter (see 2.2.3 Colour Sensors) for the differentiation of the wavelengths, the colours respectively, the questions arises if the sensor-pure colours like red, green or blue parts deliver a different detection precision than colours that only can be represented as a mixture of the latter. Cyan is the mixture of blue and green and is thus differently processed when transformed into HSI colour space (see 2.2.5 RGB and HSI Colour Space). However, due to lack of data, namely more part images, this relation could not be tested thoroughly, but remains an objective for further studies.

In order to illustrate the necessity of proper thresholding, Figure 4.33 depicts the percentile image, on which the dichotomization is applied. There the contrast suggests that it is possible to detect the defects of interest quite easily. Unlike in Figure 4.32, where the threshold was set too low, which resulted in false-positive detection, it was set high enough so that the detection of the four main silver stripes was possible.

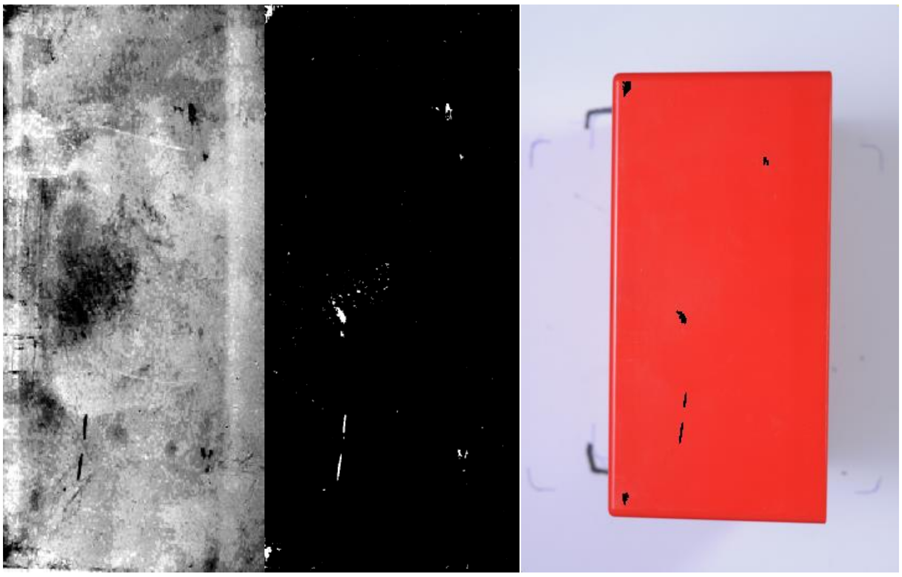


Figure 4.33: Red sample: left: percentile image before the application of the particle detection; middle: dichotomized image of the detected defects; right: image-stack of size-relevant defects and the original RGB photograph

Figure 4.34 and Figure 4.35 clarify why the thresholding has to be adjusted for each colour separately. Both figures contain the percentile image with a cross-section line and the intensity distribution of the cross-section. If one focuses on the intensity scale, it becomes obvious that the intensity drop for the defect in the red part is higher than the one in the cyan part. Thus, the threshold can be set higher (relative to the intensity baseline), which results in detection of fewer false-positive defects. In fact, this relation can be seen as a classical signal-to-noise relation. Ideally, the surface would be photographed and processed in a way, where areas without defects have intensities closely scattering around zero and defects have intensities well outside this *base surface* noise. It is therefore important to optimally photograph and process the part's surface, so that this variance, which can be seen as noise, is kept as low as possible. In conclusion, the MV setup and especially the lighting should strive for lowering the noise of non-defected areas. This will improve the signal-to-noise ratio. In this way, the defects that are characterised by a clear intensity drop, can be detected with a higher sensitivity. Thus, small-area defect detection will improve, because silver stains or scratches that are ultimately just areas of lower saturation in the HSI space, will stand out more clearly.

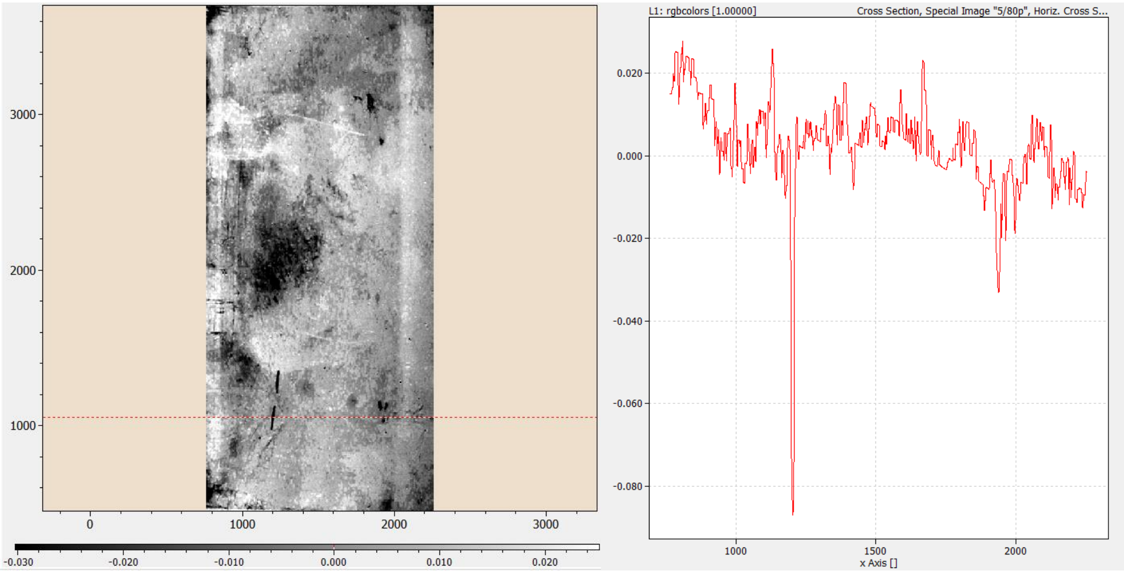


Figure 4.34: Red sample: left: percentile image with the intensity cross section; right: intensity distribution of the cross section

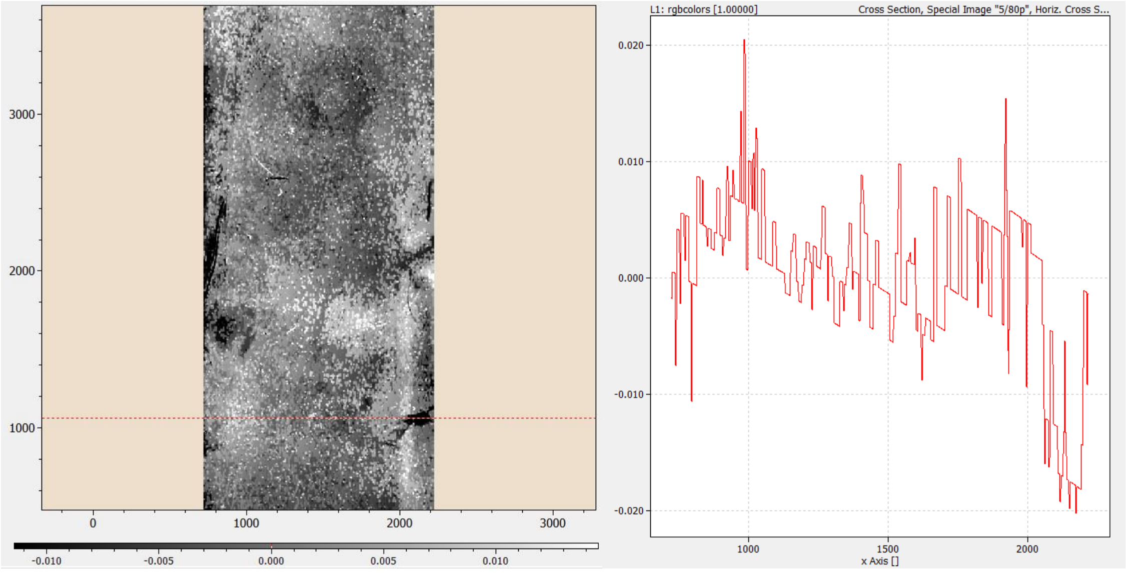


Figure 4.35: Cyan sample: left: percentile image with the intensity cross section; right: intensity distribution of the cross section

The sampled blue part exhibited a different type of problem. Again, it is assumed that the blue pixel array is the reason for a more detailed image of the nuances of the blue surface. Principally, high sensitivity is a main goal when setting up the MV system. In this case, however, it complicates the thresholding and size exclusion process. Because the algorithm is so sensitive, it detects not only the true silver stains in the top-right area, but also the brighter area in the middle-left area of the part, as seen in Figure 4.36. The middle-left area is presumably brighter due to the production process or stretching of the part's wall during transport, which results in an alteration of the surface. This effect is often observed when coloured polymers are bent.

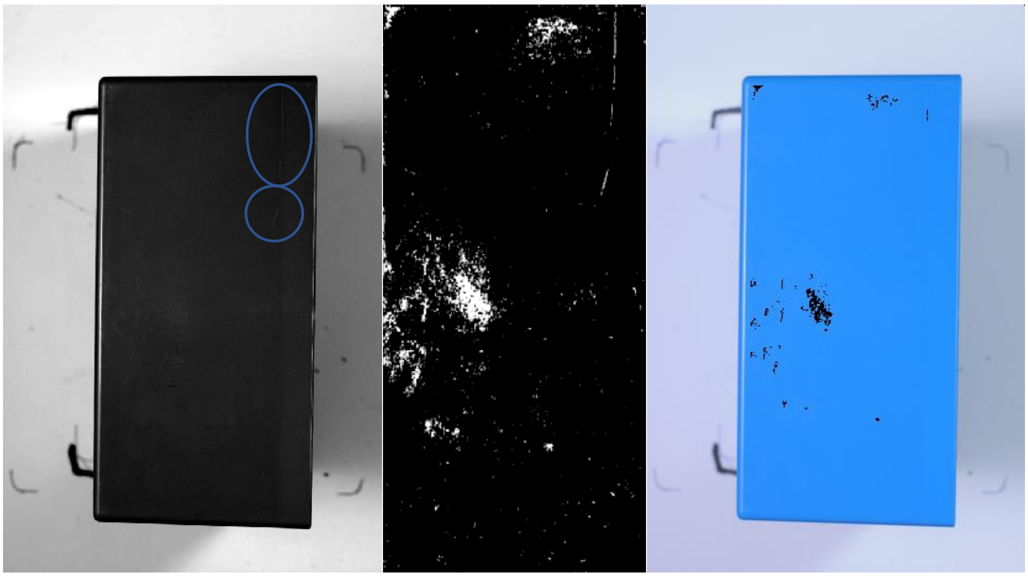


Figure 4.36: Blue sample: left: green layer of the defected cyan part (original image);
middle: dichotomized image of the detected defects;
right: image-stack of the size-relevant defects and the original RGB photograph

Unfortunately, the problem at hand can neither be resolved by an optimization of the threshold, nor by adjusting the boundaries of the size exclusion of the found defects. When the threshold is set lower, so that less *particles* are detected, the actual defects in the top-right area become so thin and small, that they vanish when a characterization depending on size is done. Thus, they cannot be distinguished from the bulged area on the middle-left, as seen in Figure 4.37.

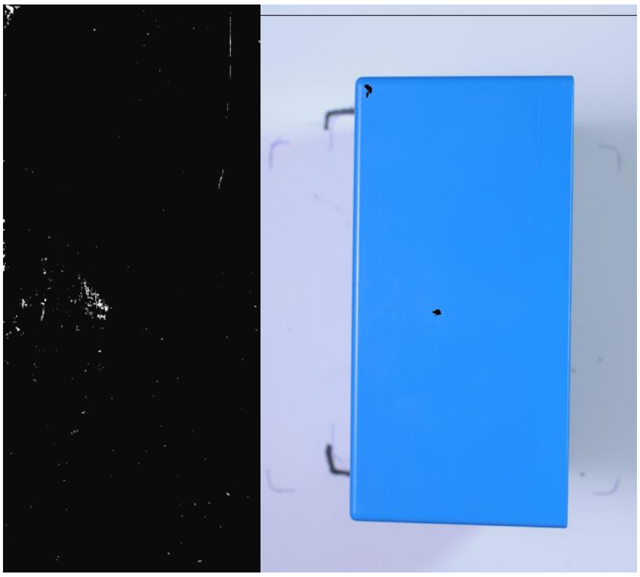


Figure 4.37: Blue sample: left: dichotomized image of the detected defects;
right: image-stack of the size-relevant defects and the original RGB photograph

The indicated issue can again be explained by examining the intensity distribution of the cross-section of the percentile image, shown in Figure 4.38. The polymer in the bulged area was altered to such an extent, that it shows an even higher drop of intensity than the silver stain in the top-right area. Hence, when such surface alteration is accepted, but silver stains are not, there is no possibility to detect silver stains with the suggested algorithm. However, if bulged areas are considered to be defects, the used setup and algorithm are certainly powerful and sensitive enough to detect them.

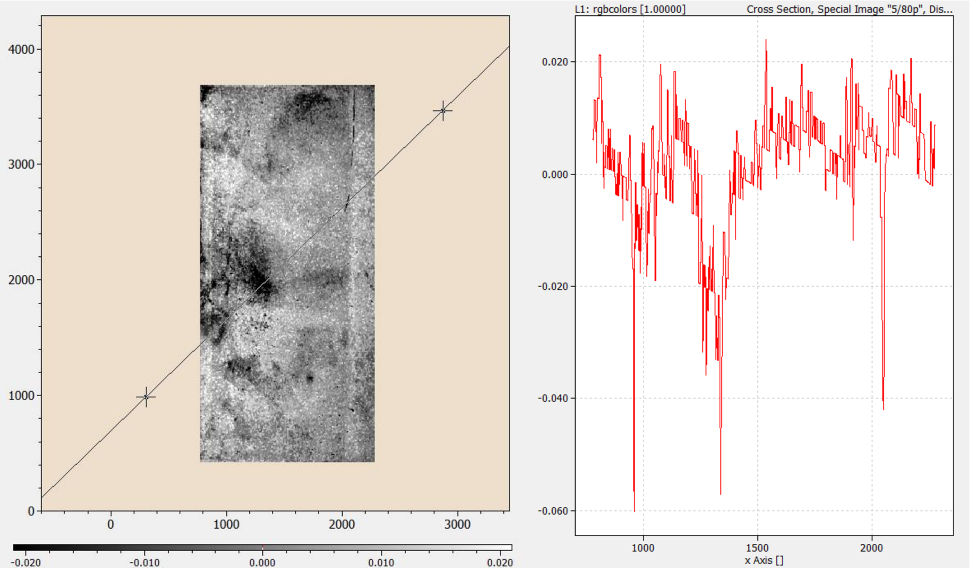


Figure 4.38: Blue sample: left: percentile image with the intensity cross section; right: intensity distribution of the cross section

Finally, the analysis of the pink part brought forth satisfactory results. Figure 4.39 illustrates that both the silver stain in the top-right and bottom-right, as well as the scratch in the middle area were found. However, due to positioning of illumination and camera, a strong reflection can be seen on the left and bottom edge of the part. This reflection is naturally detected as a defect, since it is significantly brighter than the regular surface. Such false-positive detections of side reflections might be reduced by hardware adaptions: if not by repositioning of the illumination system, then by adjusting the trim of the algorithm. Assuming that the parts will be photographed at constant conditions, because a robot arm is locating them with a minimum of variation at the same place, the image trim, which is part two of the algorithm, can be set in a way where it just cuts off reflections at the edges of the part, omitting them in the analysis.

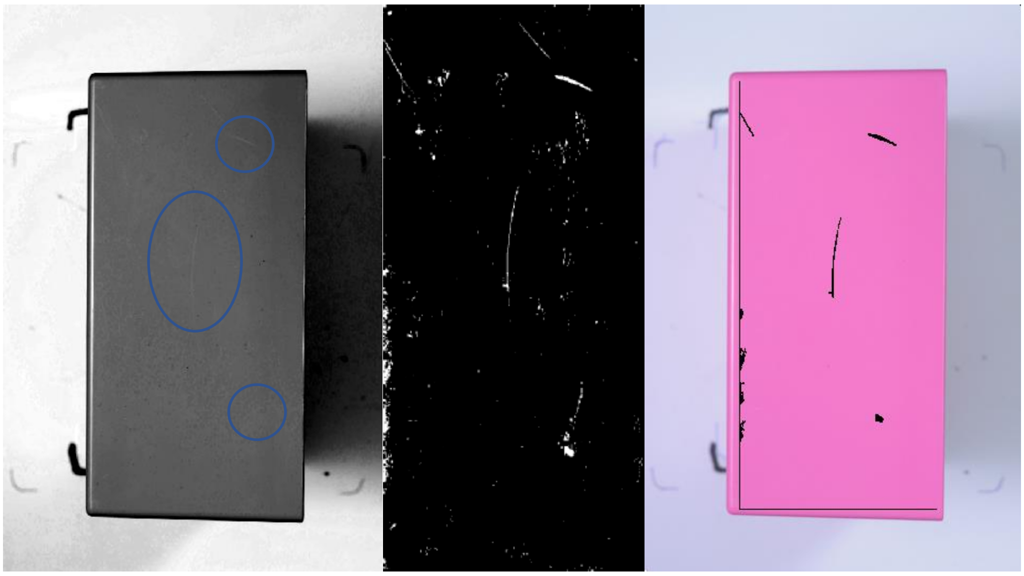


Figure 4.39: Pink sample: left: blue layer of defected cyan part (original image);
middle: dichotomized image of detected defects; right: image-stack of size-relevant defects and original RGB photograph

In conclusion, it is highlighted that the crucial hardware component for satisfying results is certainly a well-designed illumination system. It will avoid further problems that occur when thresholding is chosen as the method to detect defects. Hence, the accomplishment of the most uniform illumination possible has to be the key objective when setting up the system.

Moreover, computation time of the algorithm amounted to roughly 15 seconds. Considering the slower production rates of such parts, the duration of processing is more than satisfactory. Especially because computation time can be optimized by using professional hardware and enhancing the algorithm. For instance, when a robot arm is used for positioning the parts, the calculation of an image mask, before necessary to trim the photograph, becomes obsolete. The area of interest can directly be cut out of the original photograph, moving along with a conversion of the trimmed image into an HSI image, reducing computing time.

Sensitivity is an issue regarding the colour of the parts. Further studies have to be carried out to investigate the relation between part colour and type of colour filter that is used in the camera. Regarding the power of the algorithm, more samples with well-defined defects need to be photographed to be able to determine limits and possibilities of the method to full extent.

Conclusion

This study is the first step towards establishing an academic point of view regarding the quality monitoring of machine vision systems. An innovative solution was presented to detect several issues concerning MV hardware. This was done by creating and photographing a test template. The obtained images were processed with algorithms, that indicate the nature and graveness of the malfunction.

The author's method supplies knowledge to develop a remote-controlled system, which is able to monitor the degradation and homogeneity of the illumination, the correct mounting of the camera system, the pixel degradation and defects, as well as the geometrical distortion of lenses. Additionally, pseudo-algorithms are supplied to build such a system. The academic charm of the test template is the power of handling several issues with one image and simply applying a series of rapid algorithms to detect the problem's nature. The algorithms provide statistical analysis of the images in order to avoid biased decisions from human experts. Certainly, a routine control of the parameters is suggested to monitor quality requirements of the hardware. The test template consists of a user-defined and thus flexible number of squares filled with a range of grey values. It can be adapted to the sensor's resolution and the lenses focal length.

Moreover, an idea was given on how to build a device, that allows the measurement of exposure times without a demounting of the camera. The research on dark current suggests that effects of ambient temperature changes on the quality of the taken images are minor.

The second part of the thesis relates to the investigation of detecting defects in two objects with challenging surfaces. In the first case, a setup was introduced to standardize the imaging of circular, transparent, polymer lids, which contain a spoon and are covered by a transparent foil. The objective of distinguishing non-defected from defected lids, where the foil was crinkled and thus not sealing correctly, was done by introducing several pre-processing steps, before applying a statistical test on the data. The main pre-processing steps consisted of finding the circular object using Hough circle detection, analysing and scaling its intensity distribution, before comparing the final distribution with an averaged distribution of non-defected lids using a χ^2 test. Out of 37 defected samples, 91.88 % were detected correctly. Accordingly, correct scaling of the data is certainly a vital part to increase the algorithm's performance. The restricted use of resources and time omitted further optimizations, however, it is emphasized that both setup and algorithm, as well as computation time have potential of performance enhancement.

In the second case, walls of opaque parts were examined on production-related defects called *silver stains* and scratches. An innovative algorithm was presented to detect surface defects, as traditional thresholding will fail dealing with such challenging surfaces. The devised methodology consisted of locating the part, converting the RGB into an HSI image, modelling of the saturation surface with subsequent subtraction, smoothing the obtained image using a percentile filter and finally applying a suitable threshold to highlight the defects. Final characterization was done due to defect size.

Contrary to expectations, the most crucial part was setting up proper illumination. Thoroughly uniform illumination is still an open problem and subject to optimization. The algorithm itself has proven to be effective on both silver stains and scratches exceeding a certain size limit, however, distinction between the latter and altered areas, that were brighter due to bulging, was not possible. When bulged areas are considered to be defects, the methodology can be used to find them as such. Given the small sample size, the results have to be regarded with care, nonetheless the discoveries promise a bright outlook for defect detection in opaque, reflective surfaces.

In addition, the findings suggest a relation between the colour filter array and the part colour, however, these results are still subject to further investigations. Computing time of the algorithm was decent and has potential for optimization. Overall, elaborate testing of the methodology requires supplementary data collection.

All these results provide new insights and approaches for defect detection of hard-to-handle surfaces. In the view of the author, development of new hardware ensures that machine vision technologies and their application for quality control face exciting years.

References

[1] W. E. Snyder and H. Qi, *Machine vision*. New York: Cambridge University Press, 2010.

[2] C. Steger, M. Ulrich, and C. Wiedemann, *Machine vision algorithms and applications*, First Edit., vol. 46, no. 02. Wiley-VCH-Verlag, 2013.

[3] “Modern Photography/The Camera,” 2018. [Online]. Available: https://en.wikibooks.org/wiki/Modern_Photography/The_camera. [Accessed: 25-Feb-2019].

[4] N. Mansurov, “Understanding ISO, Shutter Speed and Aperture – A Beginner’s Guide,” 2018. [Online]. Available: https://photographylife.com/iso-shutter-speed-and-aperture-for-beginners. [Accessed: 25-Feb-2019].

[5] L. Bountouri, “Digitization,” *Arch. Digit. Age*, pp. 29–36, 2017.

[6] C. Demant, B. Streicher-Abel, and C. Garnica, *Industrial Image Processing. visual quality control in manufacturing*, 2nd Edition. Heidelberg: Springer, 2013.

[7] R. Hain, C. J. Köhler, and C. Tropea, “Comparison of CCD, CMOS and intensified cameras,” *Exp. Fluids*, vol. 42, no. 3, pp. 403–411, 2007.

[8] P. G. R. Inc., “How to Evaluate Camera Sensitivity,” *Point Grey White Paper Series*, 2015. [Online]. Available: https://www.ptgrey.com/white-paper/id/10912.

[9] T. Goldsmith, “Optimization, Constraint and History in the Evolution of Eyes,” *Q. Rev. Biol.*, vol. 65, no. 3, pp. 281–322, 2008.

[10] T. Carron and P. Lambert, “Color edge detector using jointly hue, saturation and intensity,” *Proc. - Int. Conf. Image Process. ICIP*, vol. 3, pp. 977–981, 1994.

[11] F. Kong and Y. Peng, “Color Image Watermarking Algorithm Based On HSI Color Space,” in *2nd International Conference on Industrial and Information Systems*, 2010, pp. 464–467.

[12] SharkD, “RGB Cube Show lowgamma cutout b.png,” *22 March 2010, 08:45*, 2010. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/8/83/RGB_Cube_Show_lowgamma_cutout_b.png. [Accessed: 02-Aug-2018].

[13] J. Rus, “HSV color solid cone chroma gray.png,” *22. March 2010*. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/0/00/HSV_color_solid_cone_chroma_gray.png. [Accessed: 02-Aug-2018].

[14] B. Jähne, *Digitale Bildverarbeitung*, 6. Edition. Heidelberg: Springer, 2005.

[15] W. Gao, L. Yang, X. Zhang, and H. Liu, “An improved Sobel edge detection,” *Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010*, vol. 5, pp. 67–71, 2010.

[16] B. M. Ayyub and R. H. McCuen, *Probability, Statistics, & Reliability for Engineers*, First Edit. CRC Press, 1997.

[17] New Zealand Ministry of Education, “Sampling Variation.” [Online]. Available: https://nzmaths.co.nz/category/glossary/sampling-variation. [Accessed: 08-Aug-2018].

[18] J. Abfalter, “Statistik Gurker WS 2014 / 2015 Zusammenfassung,” 2015. [Online]. Available: https://vowi.fsinf.at/images/6/60/TU_Wien-Statistik_und_Wahrscheinlichkeitstheorie_VO_%28Gurker%29_-_Zusammenfassung_WS_2014.pdf. [Accessed: 15-Aug-2018].

[19] H. Lohninger, “Fundamentals of Statistics,” 2012. [Online]. Available: http://www.statistics4u.com/fundstat_eng/cc_test_chi-square.html. [Accessed: 08-Aug-2018].

[20] D. Martin, “Practical Guide to Machine Vision Lighting,” *Advanced Illumination*, 2012. [Online]. Available: http://www.ni.com/white-paper/6901/en/.

[21] G. Klinglmayr, “Personal Communication, 12.07.2018,” 2018.

[22] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, “Methods for identification of images acquired with digital cameras,” *Enabling Technol. Law Enforc. Secur.*, vol. 4232, no. February 2002, pp. 505–512, 2000.

[23] N. Koren, “Understanding Image Sharpness.” [Online]. Available: http://www.normankoren.com/Tutorials/MTF5.html. [Accessed: 30-Aug-2018].

[24] N. Koren, “Koren 2003 Lens Test Chart,” 2003. [Online]. Available: http://www.normankoren.com/Tutorials/Lenstarg_25_7086p_15g_0is.png. [Accessed: 30-Aug-2018].

[25] I. Engineering, “TE 202 Lens Chart.” [Online]. Available: https://www.image-engineering.de/products/charts/all/524-te202. [Accessed: 03-Sep-2018].

[26] “Geometrical Distortion.” [Online]. Available: https://learn.foundry.com/modo/902/content/help/pages/rendering/camera_matching.html. [Accessed: 05-May-2018].

[27] G. Cao, S. Ruan, Y. Peng, S. Huang, and N. Kwok, “Large-Complex-Surface Defect Detection by Hybrid Gradient Threshold Segmentation and Image Registration,” *IEEE Access*, vol. 6, pp. 36235–36246, 2018.

[28] N. M. Zaitoun and M. J. Aqel, “Survey on Image Segmentation Techniques,” *Procedia Comput. Sci.*, vol. 65, no. Iccmit, pp. 797–806, 2015.

[29] J. Bandouch, “Grundlagen: Bildverarbeitung / Objekterkennung,” 2006. [Online]. Available: https://www.cosy.sbg.ac.at/~uhl/imgProcess.pdf.

[30] H. Lohninger, “ImageLab.” Epina GmbH, Retz, Austria, 2018.

[31] M. Reis, “Diffuse Reflection,” *27.10.2005*. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/6/6e/Difracao.svg. [Accessed: 20-Sep-2018].

Appendix

Table A.1: Section of intensity distributions of the dark current pictures

Temp. [°C]	6	6	6	27	27	27
Shutter Time	60	60	500	60	60	60
Aperture	2	4.5	4.5	2	2	2
Intensity	Red	Red	Red	Red	Green	Blue
1	12205069	12205069	12205069	12204837	12204814	12204701
2	6	6	6	143	162	113
3	6	6	6	46	52	139
4	4	4	4	15	19	37
5	4	4	4	17	18	37
6	0	0	0	7	6	9
7	0	0	0	4	2	1
8	0	0	0	2	2	9
9	0	0	0	2	2	1
10	0	0	0	6	4	9
11	0	0	0	3	2	3
12	0	0	0	1	1	5
13	0	0	0	0	0	2
14	0	0	0	1	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	2
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	2
21	0	0	0	0	0	0
22	0	0	0	0	0	1
23	0	0	0	1	1	2
24	0	0	0	1	1	2
25	0	0	0	0	0	0
26	0	0	0	1	1	1
27	0	0	0	1	1	1
28	0	0	0	0	0	1
29	0	0	0	0	0	1
30	0	0	0	0	0	2
31	0	0	0	0	0	0
32	0	0	0	0	0	0
33	0	0	0	0	0	0
34	0	0	0	0	0	1
35	0	0	0	0	0	0
36	0	0	0	0	0	0
37	0	0	0	0	0	1
38	0	0	0	0	0	0

Table A.2: Section of intensity distributions of the dark current pictures

Temp. [°C]	27	27	27	27	27	27
Shutter Time	60	60	200	200	200	500
Aperture	2	4.5	2	2	4.5	2
Intensity	Red	Red	Red	Red	Red	Red
1	12204834	12204852	12204774	12204752	12204820	12204837
2	148	137	220	240	169	160
3	44	32	30	38	37	29
4	14	21	16	16	15	17
5	15	12	15	15	15	16
6	8	7	12	10	10	8
7	2	4	5	5	6	4
8	3	4	3	3	2	3
9	5	4	2	2	3	3
10	1	4	3	1	3	5
11	3	2	2	1	3	1
12	1	0	1	1	0	0
13	1	0	0	0	0	0
14	1	0	1	0	1	1
15	1	1	0	0	0	0
16	2	1	0	0	1	0
17	0	3	0	0	0	1
18	0	0	0	0	0	0
19	0	1	0	0	1	1
20	2	1	2	2	0	1
21	0	0	0	0	1	0
22	1	0	1	1	0	0
23	1	0	0	0	0	0
24	1	1	1	1	0	0
25	0	1	0	0	1	1
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0
32	0	0	0	0	0	0
33	0	0	0	0	0	0
34	0	0	0	0	0	0
35	0	1	0	0	0	1
36	0	0	0	0	1	0
37	0	0	0	0	0	0
38	0	0	0	0	0	0

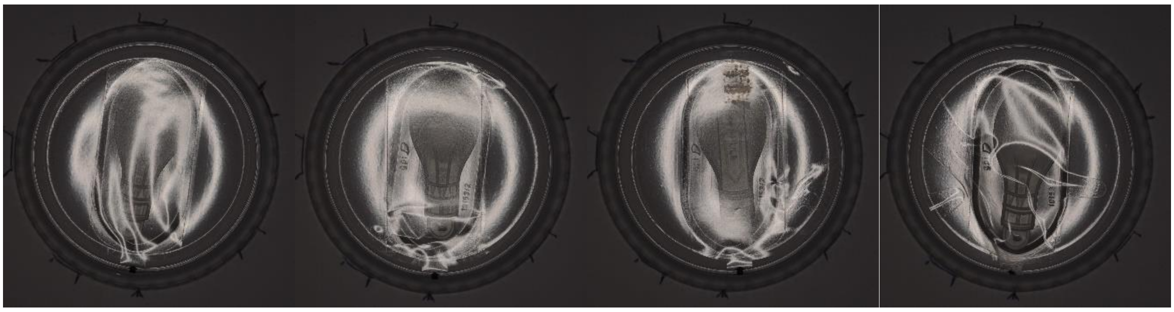


Figure A.1: Left: DL sample 1; middle left: DL sample 2; middle right: DL sample 3; right: DL sample 4

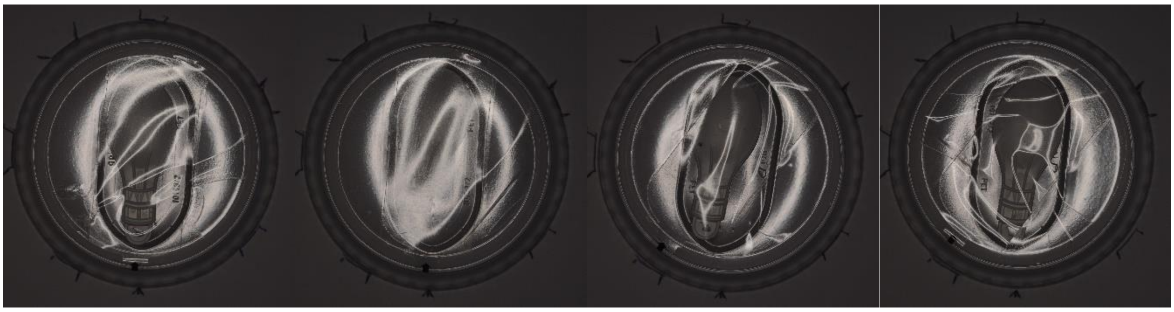


Figure A.2: Left: DL sample 5; middle left: DL sample 6; middle right: DL sample 7; right: DL sample 8

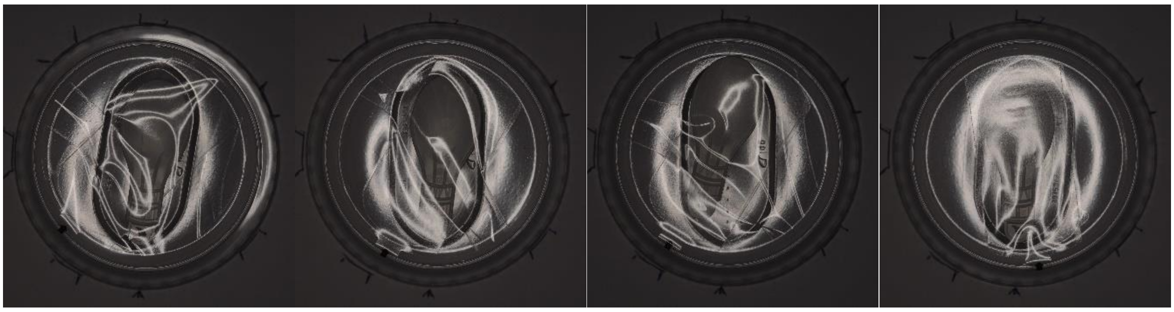


Figure A.3: Left: DL sample 9; middle left: DL sample 10; middle right: DL sample 11; right: DL sample 12

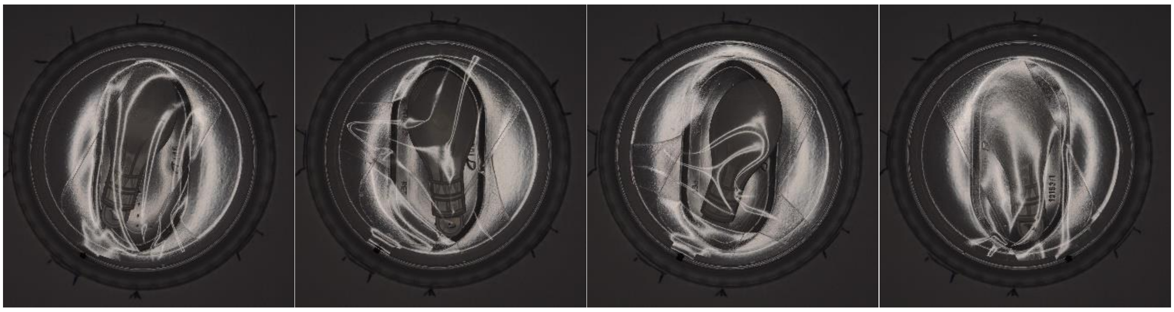


Figure A.4: Left: DL sample 13; middle left: DL sample 14; middle right: DL sample 15; right: DL sample 16

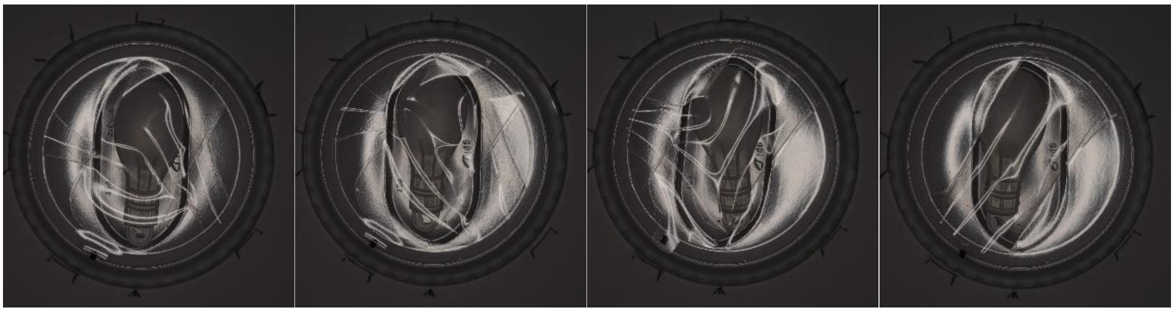


Figure A.5: Left: DL sample 17; middle left: DL sample 18; middle right: DL sample 19; right: DL sample 20

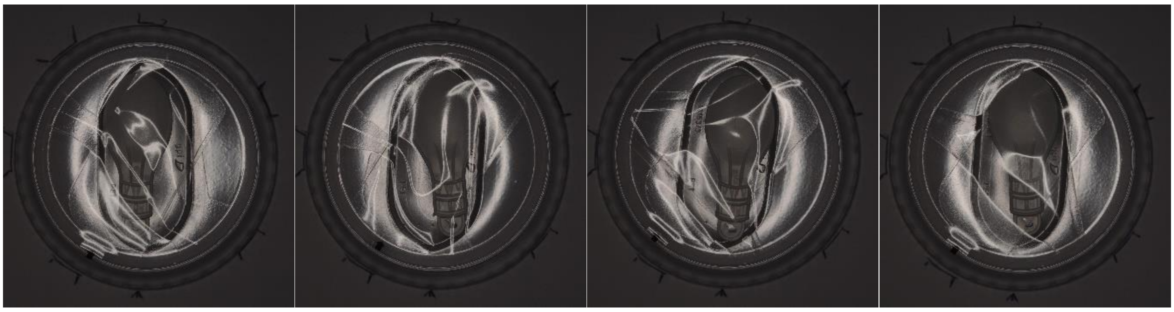


Figure A.6: Left: DL sample 21; middle left: DL sample 22; middle right: DL sample 23; right: DL sample 24

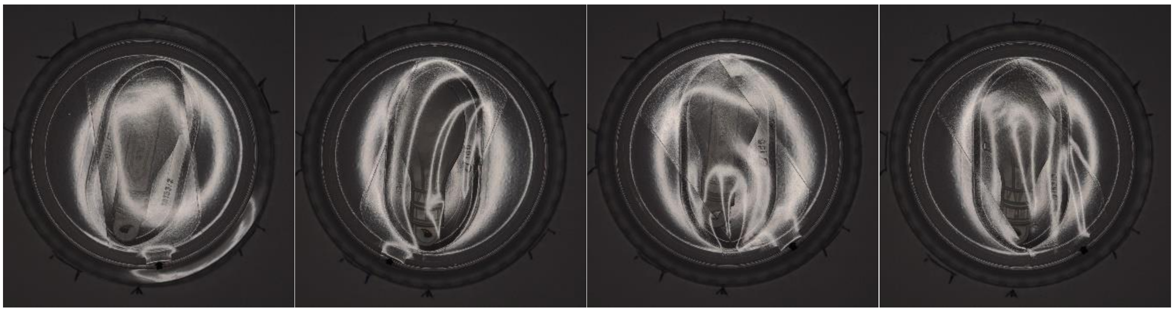


Figure A.7: Left: DL sample 25; middle left: DL sample 26; middle right: DL sample 27; right: DL sample 28

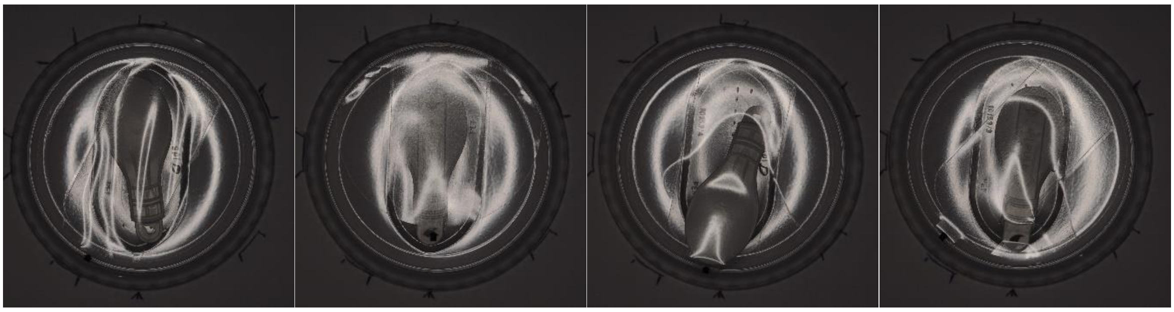


Figure A.8: Left: DL sample 29; middle left: DL sample 30; middle right: DL sample 31; right: DL sample 32

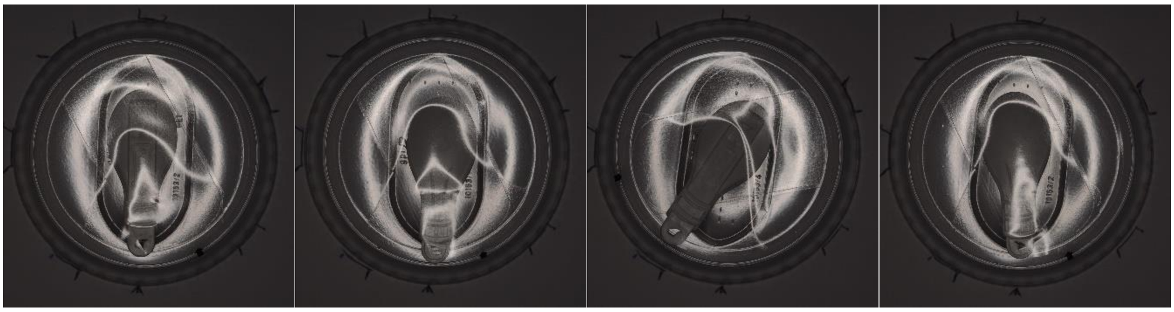


Figure A.9: Left: DL sample 33; middle left: DL sample 34; middle right: DL sample 35; right: DL sample 36

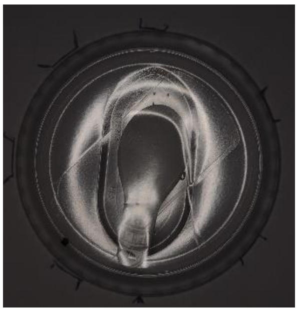


Figure A.10: DL sample 37