Fully Automated Repair of Surface Flaws using an Artificial Vision Guided Robotic Grinder

Doctoral Thesis by

Norbert Koller

Institute for Automation
University of Leoben
Leoben, Austria

May, 2007

Supervisors:

O.Univ.Prof. Dipl.-Ing. Dr.techn. Paul O'Leary B.A., B.A.I., M.E.E.
University of Leoben, Leoben, Austria

Prof. Paul Zsombor-Murray B.Eng., M.Eng., Ph.D., ing., FCSME
McGill University, Montreal, Canada

Abstract

This thesis presents a new concept for the fully automated in-line repair of surface flaws on steel billets using an artificial vision guided robotic grinder. The material entering this flaw repair system has had the flawed regions pre-marked with colored paint patches. In the following, a machine vision system is presented that detects and dimensions these color markings to guide an industrial robot, which then grinds out the defects. There are three main modes of operation:

Real-time color detection: A color camera observes the surface of the steel bar during its transport, with the aim of detecting the colored patches that mark a defect to be processed. Thereby, the sequencing of several color image processing algorithms, i.e. noise suppression, white balance correction, Bayer demosaicing, color modeling, thresholding and morphological analysis, yields a logical sequence of pipelined decision algorithms that are used to reliably determine, in real-time, a bounding box or boxes delimiting the region(s) to be ground.

3D modeling: Following the detection of a color marking, the geometry of the steel bar is measured using two plane-of-light scanners. The geometric measurements at both ends of the grinding region combined with the bounding box from the color detection deliver the necessary information to construct a three-dimensional model for that portion of the bar, which needs to be ground. In the process, metric coordinates are obtained by means of projective geometry and algebraic as well as geometric least squares fitting techniques are applied to generate the three-dimensional model.

Calibration: A calibration apparatus and procedure are used to individually calibrate the coordinates of the machine vision system (two plane-of-light scanners and the color detection camera) and to register them with the coordinate system of the robot. This ensures a single, consistent and accurate coordinate frame for the optical measurements and the mechanical processing. A direct linear transformation (DLT) algorithm is used to determine an initial estimate for a nonlinear iterative Gauss-Newton optimization of the calibration parameters.

Zusammenfassung

Diese Arbeit präsentiert ein neues Konzept für eine voll automatisierte Anlage zum Ausschleifen von Oberflächendefekten auf Stahlknüppel mittels eines optisch geführten Schleifroboters. Die zu bearbeitenden Fehlstellen werden in einem dieser Anlage vorgeschalteten Prüfverfahren lokalisiert und farblich gekennzeichnet. Das Ziel dieser Arbeit ist die Entwicklung eines Bildverarbeitungssystems, welches diese Farbmarkierungen detektiert, dimensioniert und die ermittelten Koordinaten an einen Schleifroboter übermittelt, welcher im Anschluss die Oberflächenfehler ausschleift. Die dafür erforderlichen Grundlagen können in drei Teilgebiete unterteilt werden:

Erkennung von Farbmarkierungen in Echtzeit: Während des Transports eines Knüppels wird dessen Oberfläche von einer Farbkamera, mit dem Ziel der zuverlässigen Detektion der farblich markierten Fehlstellen, überwacht. Durch die Anwendung mehrerer Bildverarbeitungsalgorithmen auf die aufgenommenen Bilder erhält man als Ergebnis die Koordinaten von Rechtecken, welche die Farbmarkierungen umgrenzen und somit den zu schleifenden Bereich definieren. Die dabei verwendeten Verfahren sind echtzeitfähige Algorithmen zur Rauschunterdrückung, Weißabgleich, Bayer-Farbkonvertierung, Farbmodellanpassung, Farberkennung und morphologische Methoden zur Filterung und Kennzeichnung der Fehlstellen.

Berechnung von 3D Konturen der markierten Fehlstellen: Nach der Detektion einer oder mehrerer Farbmarkierung(en) wird der Querschnitt des Knüppels an beiden Enden des Schleifbereichs mit je einem Lichtschnittsensor vermessen. Basierend auf den Resultaten der Farberkennung und der Geometrievermessung errechnet die Bildverarbeitung dreidimensionale Konturen der markierten Fehlstellen. Dabei werden die gemessenen Datenpunkte mittels projektiver Geometrie in ein metrisches Koordinatensystem umgerechnet. Unter Verwendung algebraischer und geometrischer Modelle zur Approximation der Datenpunkte werden die gesuchten Konturen ermittelt.

Kalibration: Ein spezieller Kalibrationsaufbau und ein neues Kalibrationsverfahren ermöglichen sowohl die individuelle Kalibration der einzelnen Komponenten der Bildverarbeitung (zwei Lichtschnittsensoren und eine Farbkamera) als auch deren räumliche Korrelation zum Koordinatensystem des Schleifroboters. Das zur Kalibration verwendete Verfahren zur Berechnung der Projektivitäten basiert auf der Methode der Direkten Linear Transformation (DLT) und einer anschließenden nichtlinearen Gauss-Newton Optimierung der ermittelten Parameter.

I hereby declare, that myself composed this thesis and that all work contained therein is my own, except where stated.

Norbert Koller

Acknowledgements

First of all, I would like to thank my supervisor Paul O'Leary, who inspired this work from its onset. I am deeply grateful for his help, the fruitful discussions and beneficial suggestions over the last several years. I could not imagine having had a better advisor and mentor for my PhD.

Many thanks go also to my co-examiner Paul Zsombor-Murray for his valuable comments and the support which helped a lot to finish this thesis. Moreover, I am deeply indebted to him for his prompt decision to examine my thesis, although his homeland is far away from Austria.

During the past years, my colleagues at HotVision Research GmbH and the Institute for Automation have been a great support in all parts of my work. They gave me useful hints and were co-operating even outside the coffee breaks. Therefore, I am especially grateful to Ewald Fauster, Matthew Harker, Ronald Ofner, Beate Oswald-Tranta, Gerold Probst, Gerhard Rath, Ingo Reindl, Peter Schalk, Gernot Wally and last but not least, our secretary Doris Widek.

I also acknowledge the support I received from my former colleagues Christian Sallinger and Mark Tratnig. They always lent a hard and gave constructive feedback when I called upon them.

I would like to express gratitude to *Deutsche Edelstahlwerke GmbH* in *Witten/Germany* and *Mittal Steel Ruhrort GmbH* in *Duisburg/Germany* for financially supporting this project. Further, I want to thank *Carat robotic innovation GmbH* in *Dortmund/Germany* for the setup and the programming of the grinding robot.

Special thanks go to my friends for their unlimited support when it was needed. I am thankful for their company and the endless discussions while delicate enjoying food and a cold beer or two.

Finally, I want to express my deepest gratitude to my family. I am thankful to my parents Gertrud and Otto for their loving support, their understanding over the years and I am proud of the things they taught me. Furthermore, I would like to show grateful acknowledgments to my big sister Astrid and to my two elder brothers Otto and Roland, who all gave me enough hard times to make me manage in the outside world. Additionally, they extended the emotional support I needed to complete this work.

Outline of the Thesis

Part I:

Part I introduces the principle of operation of the image processing system used to guide an industrial grinding robot.

Chapter 1 - This chapter presents the motivation for this work, explains the necessity to inspect and repair surface flaws on steel billets in a steel production plant and specifies required design parameters for this application.

Chapter 2 - The system concept of the fully automated grinding facility is introduced in this chapter. In the first half, the individual components of the plant are presented to focus on the specification of the main parts. These are: the optical servoing unit, which includes the color camera and two plane-of-light scanners and the industrial 6R robot with a belt grinding unit on its end-effector. The second half of this chapter describes the operating sequence of the fully automated process, wherein the optical servoing unit is the supervising controller.

Part II:

Part II provides the essential theoretical and practical background that is necessary to implement the system concept. The proposed methods are designed to achieve operational reliability in the inspection of steel billets on the one hand, and the guidance of the grinding robot on the other.

Chapter 3 - This chapter deals with the detection of color markings in images, indicating the presence of a crack. At the beginning, an introduction to the theory of how light is reflected by rough surfaces is presented. Taking this knowledge into account, several color image processing algorithms are selected to be applied in sequence to perform the color detection. Starting with signal preprocessing, carried out via a lookup-table to suppress the sensor's noise floor, white balance correlation is used to attain true color for every color component. In addition, a demosaicing algorithm, that is applied to the preprocessed camera's color filter array, leads to RGB[1] information for every pixel. RGB to HSL[1] color modeling is used to increase the stability of the color detection even under varying illumination and changing surface reflectance properties in the observed scene. Finally, the literal color detection is performed by means of thresholding and morpholog-

---

[1] See page x.

ical analysis that lead to the pixel-coordinates of a box that surrounds the flaw marking.

Chapter 4 - This chapter describes the artificial vision based measurement of steel billet cross section using light sectioning. In the process, the specimen is intersected by a plane-of-light and the resulting intersection curve is acquired by a monochrome digital camera in an arbitrary pose, displaced from the natural billet frame. Furthermore, an accurate estimate of the position of the laser line is extracted from the image by appropriate image processing, e.g. center of gravity calculation, and is subsequently mapped to the real-world billet frame to obtain coordinate data that represents an unsmeared and rectified cross section image. At this point, two proven mathematical models are selected to determine the geometry of two types of billet. Circular cross sections are modeled with a fitting algorithm for circles and billets with square cross sections and corner radii are fitted using a combination of an orthogonal line fitting algorithm and a special fitting algorithm for circles defined by two intersecting tangents. In all cases, algebraic as well as geometric solutions are derived and their precision is systematically estimated.

Chapter 5 - The aim of this chapter is to combine color detection (Chapter 3) with cross section measurement (Chapter 4) to map the color patch that represents a billet surface flaw. In so doing, two mathematical models are presented: Firstly, collineations are used to map coordinates between two projective planes and finally, their three-dimensional coordinate transformation. These techniques are subsequently applied to the pixel-coordinates of both results to achieve the undistorted and registered metric-coordinates of the color patch bounding box and the measured points from light sectioning. In addition, the cross section is fitted and used to create a three-dimensional model of the billet surface, upon which the bounding box is projected to create the required 3D patch. Finally, the quantitative patch topography is determined and supplied to the industrial 6R robot that carries out the grinding process.

Chapter 6 - The topic of this chapter is system calibration. Initially a novel calibration concept is introduced. Then the mathematical principles pertinent to this calibration method are laid out. This includes Gauss-Newton approximation and fitting routines for ellipses and planes. Furthermore, the application of these techniques is explained in detail in the additional sections, where the calibration of all the individual parts of the grinding facility is presented. The application of the direct linear transformation algorithm for the determination of the collinear relationships and their subsequent nonlinear approximation are pointed out. This yields collineation matrices and the distinct coordinate frames for all components. Finally, this chapter is concluded with a description of how these individual frames are combined in a common coordinate system.

Part III:

Part III presents the results of operational testing to verify that the proposed approach performs satisfactorily as implemented in a fully automated optical servoing system for industrial surface machining.

Chapter 7 - This chapter presents results obtained from five machining facilities of similar design, which were successfully started up in the production line of two different steel production plants. Results are given for the industrial implementation, the color detection and the measurement of the cross section. Moreover, the outcome of the machining is provided by comparing the material surface before and after the grinding process. This shows that the performance of the system is satisfactory, since the robot reshapes the bar surface accurately. All this was done with both types of billet, i.e. bars with circular as well as those with square cross section.

Chapter 8 - This chapter summarizes conclusions drawn for this work and outlines promising avenues of future research in this vein.

Contribution of this Work

In the author's opinion, this thesis contains the following content that represents original work:

- Chapter 3 presents an efficient combination of common color processing techniques to obtain a fast and reliable color detection. These algorithms are fine tuned to perform this task for color markings on steel billet surfaces in real-time. Moreover, the described method is a powerful tool which fulfills the intention to reproduce the original color information with high fidelity and enables a reliable color detection even under apparent color variations due to non Lambertian scattering properties of the observed scene.

- An algebraic as well as a geometric fitting model for the calculation of circles that are defined by two intersecting tangents are given in the second half of Chapter 4. The introduced algorithms allow the precise calculation of corner-radii and their center points on steel billets, which are determined by a square shape with rounded corners. In so doing, results from the more accurate iterative geometric approach are achieved by using the algebraic solution as starting point.

- The combination of three image processing systems, consisting of a color camera and two plane-of-light scanners, to guide an industrial 6R grinding robot is presented in Chapter 5. Based on the measurement results from the plane-of-light scanners, a three-dimensional model of the steel billet is calculated. In addition, color detection makes it possible to obtain the coordinates of the color patch bounding box and these are projected upon the surface. This yields the required patch topography that represents the billet surface to be ground.

- A new concept has been implemented which enables the calibration of the three image processing systems and the industrial robot so as to have one common coordinate system (see Chapter 6). Moreover, the proposed concept is based on a further development of the determination of homographies by using direct linear transformation (DLT). The DLT algorithm requires the coordinates of at least four rigid calibration points in general position with respect to their projective planes to determine the homography. In metric vision, the real-world coordinates are commonly known and the pixel-coordinates are determined by means of adequate fitting algorithms. Because of perspective distortions, the acquired pixel coordinates normally do not exactly match the true position of the calibration points. The resulting homography is prone to systematic errors. An original way to overcome these errors is introduced. This involves the insertion of a non-linear homography approximation subsequent to the initial DLT algorithmic step.

- Chapter 2 presents the industrial concept of a fully automated in-line repair of surface flaws on steel billets using an artificial vision guided robotic grinder, which seems to be unique in the steel industry. The uniqueness is characterized by the following fully automated features:

    - The reliable detection of color markings indicating the presence of a crack;
    - The combination of the results, derived from three independent image processing systems, for the guidance of a grinding robot;
    - Coordinated motion of the material conveyor transporting circular as well as square billets.

  The first facility was implemented and commissioned successfully in the production line of a steel production plant in 2005 and has operated trouble free since then.

List of Abbreviations

3D	Three-Dimensional
6R	Six Revolute Joints
CCD	Charged Coupled Device
CFA	Color Filter Array
CMYG	Cyan-Magenta-Yellow-Green
COG	Center of Gravity
CPU	Central Processing Unit
CS	Coordinate System
DLT	Direct Linear Transformation
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HSL	Hue-Saturation-Luminescence
I/O-port	Input/Output port
IP	International Protection, Ingress Protection
KW	Kilowatt
LUT	Look-Up Table
PCL	Programmable Logic Controller
RGB	Red-Green-Blue
SVD	Singular Value Decomposition

Contents

Part I

Principle of Operation

Chapter 1

Introduction

The hot rolling of steel bars leads inevitably to flaws on or near the surface of the material. While the size of these surface defects is small on the semi-finished steel-mill product, they may extend and penetrate deep into the material when processed downstream. As a consequence, these flaws can seriously deteriorate the quality of final products, e.g. springs, bearings, crank shafts and gears, to name a few. They become the source of high maintenance and repair costs. Furthermore, if these defects establish failure prone stress concentration in critical parts, the flaws become life threatening safety hazards. Therefore, it is important to remove flaws at the semi-finished product stage to avoid more costly problems with downstream products.

The detection of the surface cracks is commonly accomplished by a Thermomatic inspection facility [1, 2]. These defects are then removed by grinding but, unlike inspection, the time required to grind flaws is variable. This disrupts continuous throughput. The grinding stage must be decoupled. For this purpose, a material cache logistically separates the two processing steps with a storage queue, and the Thermomatic inspection system marks the regions surrounding the identified flaws with specific color. This coloring enables the subsequent retrieval of information, i.e. the location, size and quantity of the flaws. A cache, containing four circular billets is shown in Figure 1.1a.

Formerly, the color markings, each indicating a surface flaw, were detected by eye and the cracks were ground out using a hand grinder (see Figure 1.1b). This incurs a number of disadvantages that are associated with the subjective nature of this manual task: First of all, the manual optical inspection is prone to errors; Moreover, the unbalanced handling of the hand-held grinder may lead to a surface of inferior quality; and finally, this process is very labor- and concentration-intensive and thus, it bears a high accident risk for the working personnel. In addition to the avoidance of these disadvantages, the following benefits are obtained by an artificial vision guided grinding system: Full automation of the machining process leads to an increased effectiveness of this production step; Furthermore, the controlled guidance of the grinding unit causes a uniform metal removal of defined depth for the machined surface; and finally, the connection of this production step to a logistics data-base network yields administrative advantages. As a consequence, the motivation for the present work to fully automate this grinding process was to overcome the major drawbacks of the manual grinding process and to reap the contemplated benefits.

(a) Billets at the material cache **(b) Manual grinding**

Figure 1.1: Figure (a) shows four circular billets at the material cache, which separates the continuous operating Thermomatic inspection from the subsequent machining process. The frontmost billet exhibits two color markings that indicate the presence of surface flaws. The image to the right (b) depicts the manual grinding process.

This work presents a new concept for the fully automated repair of surface flaws on billets using an artificial vision guided robotic grinder [3, 4]. Thereby, three image processing systems are used for a reliable detection of the color markings, the measurement of three-dimensional coordinates of a patch that captures the geometry of the billet surface surrounding the color marking, the guidance of an industrial robot with a belt grinding unit on its end-effector and the control of the material conveyor.

1.1 Problem statement

The challenge of this work was to develop an image processing system that fulfills the requirements for this production step. Thereby, the following tasks are of decisive significance:

- The system has to be capable of handling the following product specifications of axially symmetric steel bars:
    - Billets, having a circular cross section with diameters in the range of 50mm to 250mm;
    - Billets, characterized by a rectangular shape with rounded edges and a side-length in the range of 50mm to 250mm.

  The length of the bars is from 4m to 16m, their temperature is similar to the mean ambient air temperature in the production hall, approximately 15°C.
- The necessity for reliable color detection, while the bars are moving on the conveyor;
- The sequence of tasks that comprise color detection and control of the material handling conveyor has to be performed in real-time;
- The optical measurement system must meet the accuracy requirements for the guidance of the grinding robot;

- The system has to be resistant to the harsh environmental conditions, e.g. the vibrations and the produced grinding swarf;
- A proper calibration concept and target are required for the individual calibration and the subsequent spatial registration of the image processing systems and the grinding robot.

Chapter 2

System Concept

This chapter introduces the system concept of the presented fully automated grinding facility. For this purpose, the topic is divided into two sections: The first half gives a configuration overview of all plant components. Essential specifications of individual parts are enumerated and described briefly; The second section details the operating sequence of procedures to fully automate the repair of surface flaws by describing the entire working cycle that takes place to inspection and repair a steel billet.

2.1 Configuration of the components of the facility

The components of the grinding facility may be divided into two groups: The first group comprises the elements of the optical servoing unit, namely a color camera, two plane-of-light scanners, a programmable logic controller (PLC), the lighting and the mounting enclosure hardware; The second group contains the machinery that includes an industrial 6R robot with a belt grinding unit mounted on its end-effector, two billet clamps and the billet conveyor. The configuration of these components is shown in Figure 2.1, a detailed description is appends stated below.

The components of the optical servoing unit are:

- Color camera: The task of the color camera is the observation of the steel billets for color markings that indicate the presence of a surface defect. Appropriate for this purpose is a megapixel resolution, progressive scan, color model with high quality, low distortion lens. A data conversion module, which translates the camera's digital output signal to standard Gigabit-Ethernet signal to enable reliable transmission of image data at a frame rate of 30 Hertz. Finally, a sealed enclosure with a front window shields the optical components from the harsh steel mill environment;

- Plane-of-light scanners: The purpose of the plane-of-light scanners is the measurement of billet cross sections. Each of these scanners consists of a line-projecting laser module and a monochrome camera. Again, the applied camera is a progressive scan model, having megapixel resolution and high quality lens. Also, the data transmission hardware and protocol are similar to the one, used with the color camera. Furthermore, the laser operates at a wavelength of 660nm with a narrow-band pass filter, aligned to this wavelength, that is mounted in front of the camera lens.

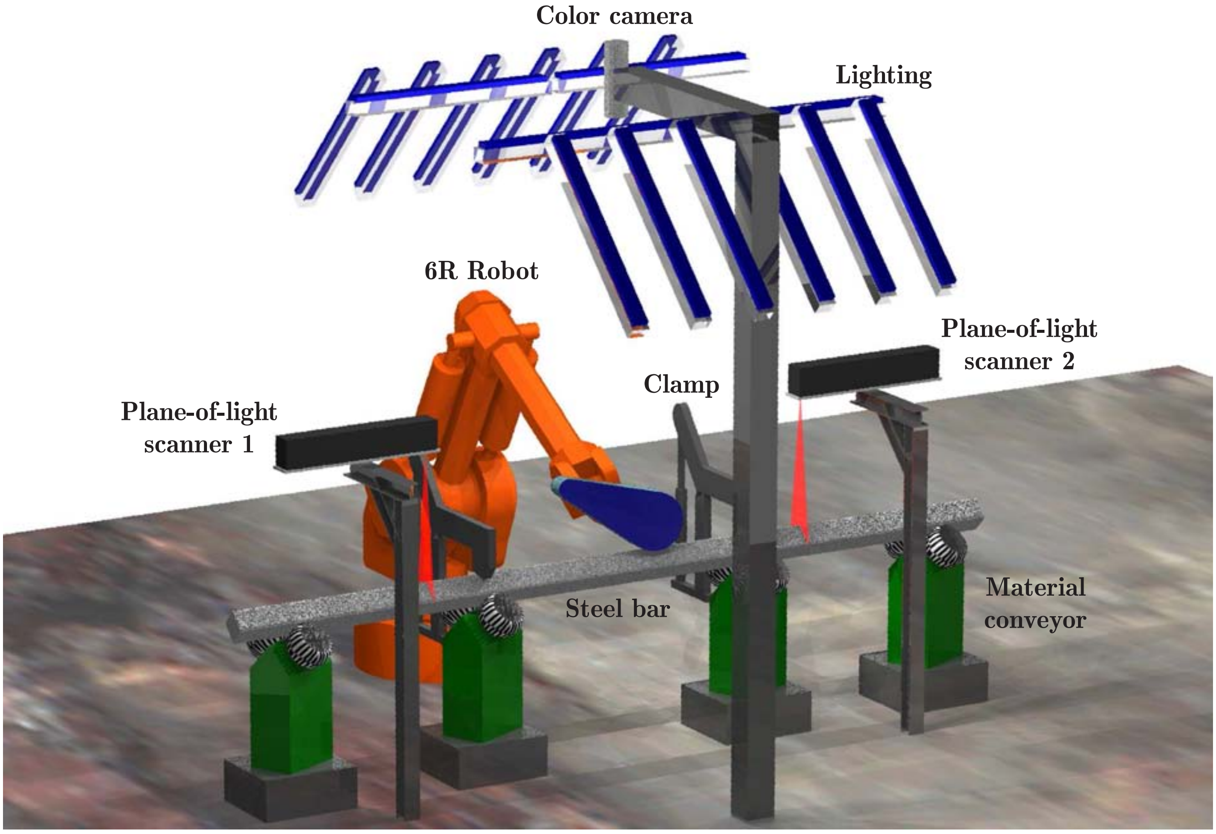


Figure 2.1: Configuration of the components of the fully automated grinding facility. The plant's main components are the three optical units, namely the color camera and two plane-of-light scanners, and finally, the industrial 6R robot with a belt grinding unit mounted on its end-effector, that satisfies the requirements for the plant's machining part.

This makes the scanner insensitive to ambient light. A sturdy sealed enclosure protects the hardware against impact contaminants and holes the camera and the laser module in a firm, fixed spatial interrelationship;

- Programmable logic controller: The PLC (not depicted in Figure 2.1) acts as supervisor of the fully automated grinding facility. In so doing, a high speed central processing unit (CPU) performs the image processing of the acquired images in real-time and controls the facility's actuators and other working parts via analogue and digital I/O-ports. The communication to the grinding robot is achieved through a software interface;

- Lighting and housing: A proper adjustment of several light sources, e.g. as shown in Figure 2.1, provides a constant illumination of the scene. Thereby, fluorescent tubes that emit diffuse light in the range of the daylight spectrum serve as light sources. Furthermore, the housing (not shown in Figure 2.1), enclose the entire facility and protects the camera field of view from possible degeneration caused by ambient light sources. All these arrangements are made to ensure reliable color detection.

The mechanical components of the grinding facility consist of:

- Grinding robot: An industrial robot with six rotational joints performs the surface machining of the steel billets. For this purpose, a belt grinder is mounted on its end-effector. A laser triangulation sensor is mounted on the grinding unit, that compensates possible local variations of the billet geometry as well as verifies metal removal satisfactory;
- Clamps: The clamps serve to fix the position of the billets for the duration of the machining process. This is necessary since, grinding force exerted by the robot may shift the workpiece after a flaw was detected, its patch coordinates determined and subsequently transferred to the robot. The clamps are fixed cycle event-driven by means of binary on-off signals;
- Material conveyor: It consists of stilts that are aligned in a straight line, wherein each stilt contains a criss-cross tire-pair, which is arranged in a way so that their treads establish a rectangular pocket. These supper the billets (see Figure 2.1) and impose its screw-like advance along the conveyor when the conveyor is in motion. This design also accommodates both: square as well as cylindrical billet of various size. The direction, velocity and braking of the conveyor are controlled by the PLC via analogue and digital ports.

The coordination of the four main components of the facility, i.e. the color camera, the two plane-of-light scanners and the grinding robot, requires a fixed installation of these parts relative to each other. From this point of view, it is obvious that any movement of the constituent components must be inhibited after they have been intercalibrated. Consequently, the optical components are mounted on heavy beams that are deeply anchored to building foundation and the robot is set on a specially-built base plate. These structural measures ensure the required resistance of the plant to externally imposed forces and vibrations.

2.2 Operating sequence

The operating sequence of the fully automated grinding facility is illustrated in Figure 2.2. This process is launched by the transfer of a billet from the material cache to the entry of the billet handling conveyor. Subsequently, the conveyor is activated and the material is positioned in the operating range of the grinding robot. Thereby, due to the design of the conveyor, the billets move forward and rotate simultaneously with a screw-like motion. A color camera, placed at a suitable position above the robot's operating range, observes the passing material surface for color marked defects. An adequate field of view of the camera and the described motion of the material, guarantee the complete survey of the billet surface. Consequently, this color detection procedure is repeated until an inductive sensor, embedded at the end of the robot's operating range, detects the passing of the billet-tail. This work cycle is performed in real-time at the color camera's maximum frame rate of 30 Hertz. Finally, after the billet has left the operating range, it leaves the conveyor and a new one, if available, is lifted onto the beginning of the material conveyor and the work-cycle procedure begin anew.

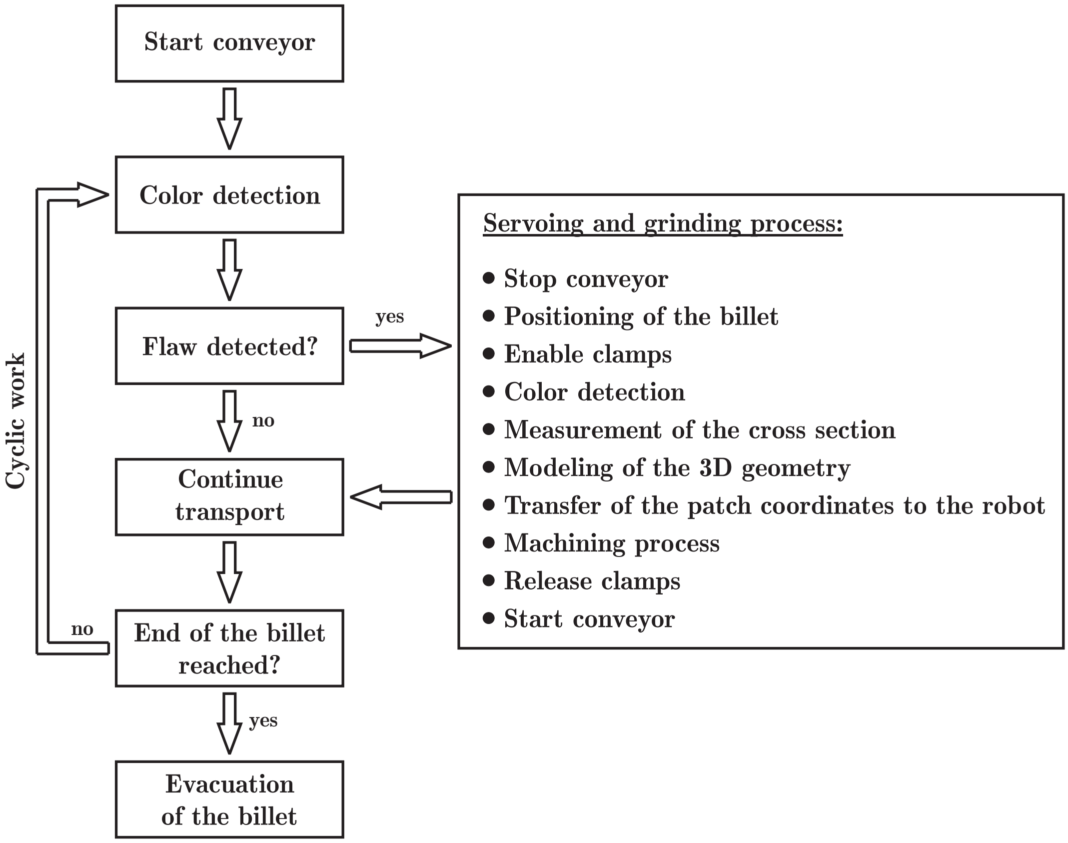


Figure 2.2: Operating sequence of the fully automated grinding facility: The essence of the operating sequence is the 'cyclic work', which comprises the detection of color marked flaws in real-time. Upon detection of a surface defect, the conveyor is stopped and the inner servoing loop is executed, the parameters of a patch, that represents the area surrounding the color marking, are determined and subsequently transferred to the robot, which ultimately performs the grinding process. After the removal of the defect the procedure returns to cyclic work.

When a color patch marked surface defect is detected by the color camera, the material conveyor is stopped immediately. In the case of a square billet, the workpiece may have to be moved somewhat to ensure that it is stably positioned with a corner in pockets between the pairs of rubber tires that advance the billet. When a billet is suitably located, pneumatic clamps accurately hold it in position. The color detection is used to locate patch corner coordinates of the color markings and the plane-of-light scanners determine the cross section of the material on ether side of the robots workspace. These combined results are used to calculate the parameters of a three-dimensional patch that describes the area to be ground, surrounding the color marking. This information is transmitted to the grinding robot, which performs the machining process. After the removal of a surface defect the clamps release, the conveyor restarts and the procedure returns to 'cyclic work', the outer loop in the work-cycle described by the flow chart in Figure 2.2.

Part II

Theoretical and Practical Background

Chapter 3

Color Detection

The automated surface machining process starts with the detection of color markings indicating the presence of surface defects on steel billets. As described in Chapter 2, a color camera is placed at a suitable position above the material conveyor, while a billet is moving along the conveyor. The material simultaneously advances along and rotates about its axis. This motion and an adequate camera field of view guarantees a complete survey of the bar surface to detect color marked defects. An image processing system to detect these defects is introduced in this chapter. Detection data is gleaned from the image, captured by the camera sensor. System output contains the pixel coordinates of the color marking's bounding box. To achieve this, several color image processing techniques were applied in sequence. Assignment of appropriate methods to these processing steps is necessary to satisfy the requirements of this application. Color marking detection must be reliable. This is essential for the uninterrupted productivity of the machining plant. Moreover, color detection is performed while the observed scene is moving and has to be carried out before the scene leaves the camera's range of vision. Consequently, all processing has to be performed in real-time, consistent with the selected camera model and frame rate. Finally, the result obtained by detection is used for metric measurements. That requires precisely determined bounding box coordinates. Sensing element and image processing algorithm performance must meet these strict requirements.

This chapter is divided into two key parts: the first part introduces the theory of how light is reflected by rough surfaces. Phenomenal underlying image formation are presented. This knowledge is essential to choose the individual processing steps that constitute the image processing pipeline sequence to perform the color detection, which is explained in the second half of this chapter. Every step in the pipeline entails the execution of carefully selected color image processing techniques. Starting with image capture, the benefits of the chosen algorithms are explained up to and including the stage where the final rendered image is analyzed by thresholding and morphological operators. This processing sequence is illustrated with an example to give the reader an appreciation of that the color detection process is well suited to its industrial application.

3.1 Reflectance model

Reflectance models are used to describe the scattering of light incident to a surface and describe energy and direction of the reflected light. It is obvious that only light rays which are reflected or scattered in the direction of the sensor cause an image of the observed scene. Also the energy perceived at any point in the captured image is intimately related to surface reflectance properties at the corresponding point in the scene. Especially in color imaging, knowledge of the reflection process is essential, since the perceived color is not only dependent on its chrominance (color) but also on its luminance (intensity). For example, if a surface reflects monochromatic red light and produces a completely black or white image, no useful information can be recovered because the reflected energy is not within the sensor's sensitivity range.

Generally there are two approaches to the study of reflections: A physical model is given by Beckmann and Spizzochino [5]; and a geometrical approach is published by Torrance and Sparrow [6]. Nayer *et al.* [7] compared both models and proposed that both have been developed to describe specular reflection mechanisms and both have been found to fit experimental data well. The physical model is a very precise theoretical formulation of the reflection process. It is based on electromagnetic wave theory and is applicable to very smooth as well as rough surfaces. However, the complexity of this approach makes it difficult to apply it to the color detection problem. The simpler geometrical model is more suited to machine vision. The disadvantage of this model is that it applies only to rough surfaces. It is invalid when the wavelength of incident light is comparable to the dimensions of the irregularities on the reflecting surface. That is the case with mirror-like surfaces and is not limited to color imaging. In the following, the principle of the Torrance and Sparrow model is presented.

This geometrical approach defines three contributing parts to describe the surface reflection of incidence light from rough surfaces. These primary reflection components are the diffuse lobe I_{dl}, the specular lobe I_{sl} and the specular spike I_{ss}. This is shown for a fixed incidence direction and a variable sensor angle in Figure 3.1. The intensity of the light reflected into the sensor is the sum of all three components, given by:

$$I = I_{dl} + I_{sl} + I_{ss} \tag{3.1}$$

The determination of the three contributing components requires the following assumption [7]: the reflectance surface consists of a collection of infinitesimal mirror-like micro-facets. Considering a patch on that surface, a mean normal vector **n** can de determined. Furthermore, each micro-facet in the patch is characterized by its slope α, which is determined as the angle between **n** and the individual normal vector of each micro-facet. Assuming that all the slopes are normally distributed, the roughness of the surface can be described by the standard deviation σ_α of the slope angles α. Thus, for a light source, emanating from the focus of a parabolic reflector to achieve parallel rays, the components can be determined.

- **Diffuse lobe:** The diffuse lobe represents the internal scattering mechanism. Its magnitude is independent of the viewing angle and can be described by the Lambertian model:

$$I_{dl} = \kappa_{dl}\ L_i\ cos\theta_i, \tag{3.2}$$

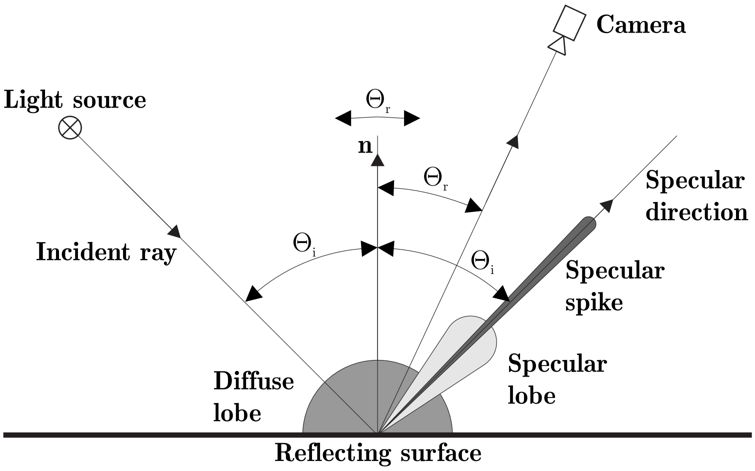


Figure 3.1: The three reflection components as functions of a fixed source direction and varying angle for the sensor.

where κ_{dl} determines the fraction of the incident energy L_i that is diffusely reflected by the surface and θ_i is the angle of the incident light.

- **Specular lobe:** The specular lobe represents a single reflection of the incident light by rough surfaces and is distributed around the specular direction. Torrance and Sparrow [6] give the mathematical model for the specular lobe as:

$$I_{sl} = K_{sl}\ exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right)\ F\left(\theta_i', \eta'\right)\ G\left(\theta_i, \theta_r, \phi_r\right), \tag{3.3}$$

where K_{sl} is the magnitude of the specular lobe. The exponential part of this equation gives the distribution function of the normally distributed slopes of the micro-facets.

$F\left(\theta_i', \eta'\right)$ is the Fresnel refraction coefficient, with θ_i' as the local angle of incidence on the micro-facets and η' gives the refractive index of the medium. The Fresnel coefficient determines the fraction of reflected light per micro-facet and may be obtained theoretically from the Fresnel equation [8]. In practice, for both metals and non-metals, $F\left(\theta_i', \eta'\right)$ varies only slightly for incident angles in the range 0^o-70^o and can be assumed to be constant at a value of approximately 0.8 [6]. Another characteristic is the dependence of the reflected wavelength on the angle of incidence. The color of the reflected light may change with incidence angle. It is evident that this effect only becomes important as the incident angle is greater than 45^o. A model for the calculation of this shift is given by Cook *et al.* [9].

$G\left(\theta_i, \theta_r, \phi_r\right)$ is the geometric attenuation factor which takes the shadowing and masking between adjacent micro-facets into account. θ_r and ϕ_r gives the direction of the camera as illustrated in Figure 3.2. The attenuation factor can be derived from the following equation:

$$G\left(\theta_i, \theta_r, \phi_r\right) = min\left(1, \frac{2\ cos\alpha\ cos\theta_r}{cos\theta_i'}, \frac{2\ cos\alpha\ cos\theta_i}{cos\theta_i'}\right). \tag{3.4}$$

Nayer *et al.* [7] proposed that for angles of incidence smaller than 60^o the surface radiance attains maximum values and thus, $G = 1$ for all values of θ_r and $\theta_i < 60^o$.

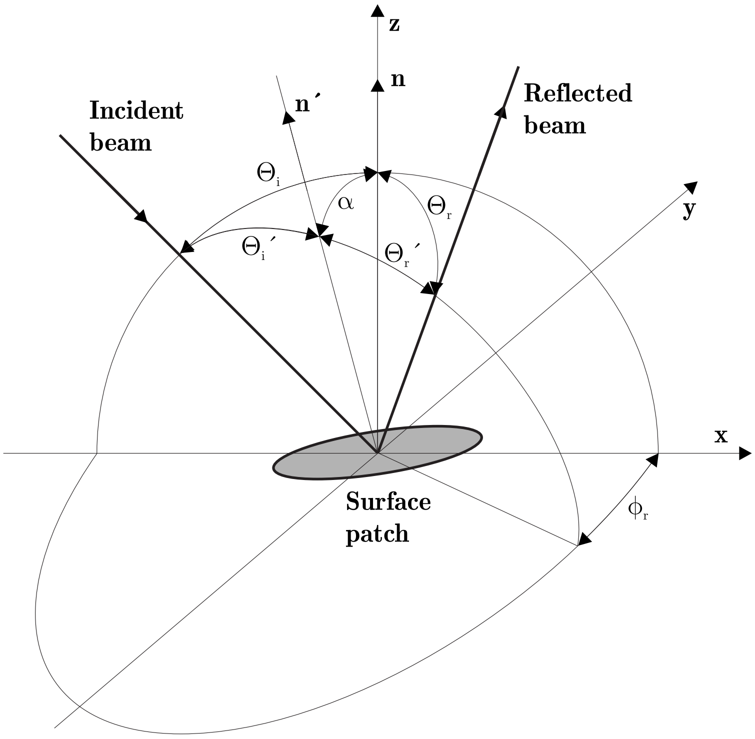


Figure 3.2: Used coordinate system for the derivation of the reflectance model.

This factor should be taken into account, only for greater angles of incidence.

- **Specular spike:** This component is caused by a mirror-like reflection. It can be approximated by the following double-delta Dirac function:

$$I_{ss} = K_{ss}\ \delta(\theta_i - \theta_r)\ \delta(\phi_r), \tag{3.5}$$

  where K_{ss} represents the strength of the specular lobe. The specular spike is concentrated in small regions around the specular direction.

The magnitude of the specular lobe and the specular spike are determined by the roughness of the surface. For very rough surfaces the specular lobe is dominant and the specular spike is negligible. The contrary is true for smooth surfaces, i.e. the specular spike contribution is many orders of magnitude greater than that of the specular lobe. Beckmann and Spizzochino [5] made the assumption that the specular spike becomes negligible when the standard deviation of the surface height to the wavelength exceeds 1.5 ($\sigma_h/\lambda > 1.5$). In other words, both components are only significant within a small range of roughness values.

3.2 The image processing pipeline

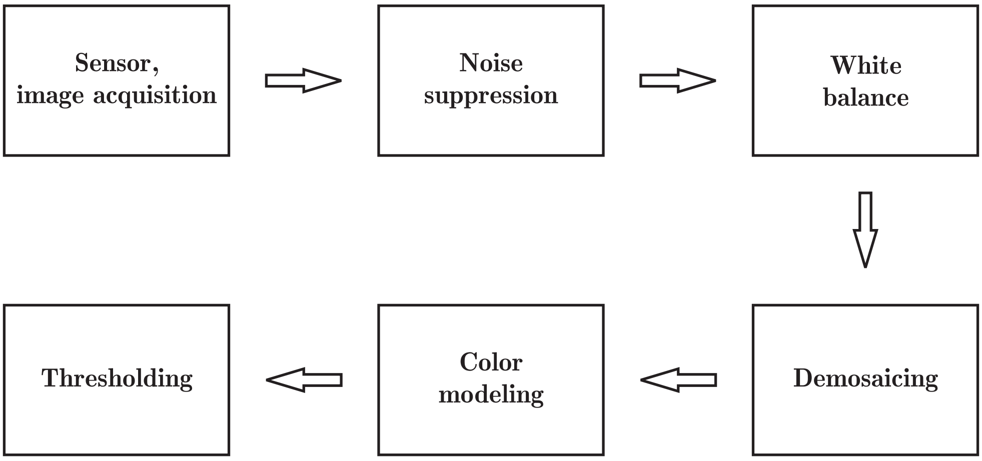


Figure 3.3: Scheme of the color image processing pipeline used for color detection: There are six main stages: 1. Image acquisition, 2. Noise suppression, 3. White balance correction, 4. Color filter array demosaicing, 5. Color modeling and 6. The color detection performed by thresholding with morphological analysis. The choice of the image processing steps is strongly connected to the requirements of the application and the selected camera model.

The signal flowchart shown in Figure 3.3 summarizes the processing used in color detection. Several color image processing steps are carried out in sequence to create a pipeline [10]. The process is start with image acquisition. The following steps are noise reduction, white balance correction, color filter array demosaicing, color modeling and finally the intrinsic color detection, performed by thresholding with morphological analysis. Each of these blocks is fine tuned to improve performance. The organization of the following sections is dedicated these processing steps. It is shown how each stage affects the final rendered image until the bounding box corner coordinates are extracted.

3.2.1 Sensor, image acquisition

A digital camera acquires images with its sensing unit, where photosensitive elements convert light energy to electrical energy for every pixel. This is easily performed for monochrome images, but for color imaging there is a need to measure a minimum of three bands at each pixel location. Consequently, that requires a detector for each channel. It also includes spectrally selective filters to enable each sensor to detect a particular color, e.g. red, green or blue. Some professional cameras use such a design, but fact is, the use of more than one sensor makes cameras susceptible to thermal expansion leading to misalignment and drives up their cost [11]. A cheaper and more robust solution is the use of a camera model using a color filter array (CFA) in front of the sensor. A CFA is a two-dimensional mosaic of color filters, where each pixel on the detector is formed according to the CFA model with intensity values determined by the reflectance and illumination of the observed scene. The advantage of using CFAs is that all required bands for color imaging are detected by a single sensor. On the other hand, such a design leads to a mosaic of color components where the missing colors for every pixel have to be estimated from the

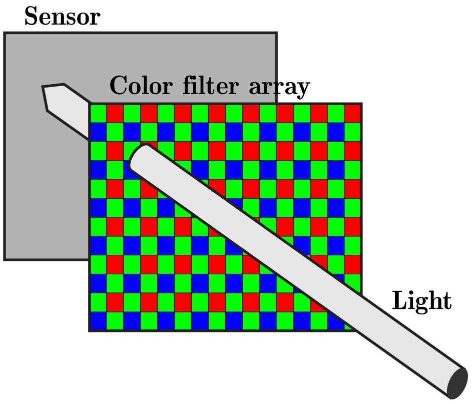


Figure 3.4: In Bayer based color detectors, pixels on the image sensor are covered with a mosaic of red, green and blue transmissive filters. The resulting image is a mosaic of color components, where 50% of the pixels are sensitive to the green, 25% to the red and 25% to the blue color.

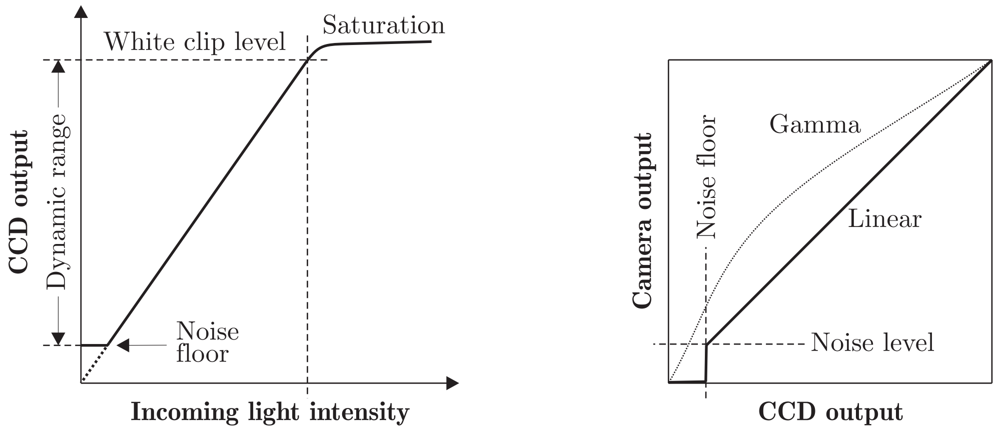
spatially adjacent CFA data. This process is known as CFA interpolation or 'demosaicing'.

The Bayer CFA [12] is commonly used in industrial cameras and was chosen for this application. A Bayer color filter array is arranged such that each pixel is interspaced among pixels of the other two colors. Green pixels are dominant so as to mimic human vision, wherein definition is produced mostly by light at medium wavelengths. Figure 3.4 is a schematic of the Bayer color detector. The resulting image is a mosaic of 50% green and 25% of both, red and blue pixels.

3.2.2 Noise suppression

Sensor intensity signals are corrupted by noise. The first step in the processing chain is to reduce it while ensuring that noise suppression does not introduce artifacts that will appear later. Most of the noise is caused by solid state phenomenon inherent in charged coupled device (CCD) sensors. It is called 'dark current'. Other contributing sources are due to charge and subsequent signal amplication and digital conversion. The sum of all this noise is specified as the camera's 'noise floor'.

For effective noise reduction it is necessary to reduce dark current. Dark current is a thermally driven electron diffusion in the sensor substrate. It combines with the photocurrent and is dependent on the substrate temperature and the image exposure time. In general, this effect limits imaging at low-light-level when the noise energy is comparable to the signal energy and thus determines the lower limit of the sensor's dynamic range. There are ways to handle this noise. Firstly, the amount of produced noise is measured with dark pixels, averaged and subtracted from the active ones. This method is satisfactory for most applications and is standard in CCD cameras. An improved option also handles the 'fixed pattern noise'. This is the remaining variable noise after removing the average value and is derived from dark current variations among the pixels. The fixed pattern noise is handled by subtracting a correction matrix from all entire images. This correction matrix contains the individual dark current values for each pixel and is created using an opaque mask in front of the sensor. This process performs well, but the pixel by pixel subtraction increases computational complexity. The disadvantage of both described methods is that they try to handle the produced noise instead of avoiding its generation and that is why the dynamic range of the sensor is not increased thereby. The only technique that enhances the dynamic range is artificial cooling to keep the temperature of the sensor at a uniformly low level. This is an expansive accessory and probably only worthwhile in delicate scientific applications. A more detailed introduction to CCD noise and dark current is given in [13, 14].



(a) CCD response **(b) Noise suppression**

Figure 3.5: The left image shows a typical response curve obtained by a CCD sensor. Its dynamic range is limited by the noise floor and the white clip level. The LUT concept to suppress the sensor's noise floor is given in the right image. The trend of the LUT curve determines the relation between sensor output and camera output.

Another method to suppress the remaining noise floor is provided by using an adjustable signal conversion look-up table (LUT) [15]. Every pixel signal is replaced by an appropriate value from a one-dimensional LUT. The idea is to clip the noise floor from the sensor response.

A typical response curve obtained by a CCD sensor is given in Figure 3.5a. As can be seen, the response from the CCD sensor is linear over a wide range. The dynamic range is defined between the noise floor and the white clip level, whereby the white clip level is set beneath the saturation knee, where the CCD output flattens out. All intensities above this border are treated as white. Finally, the slope of the curve defines sensitivity. Figure 3.5b shows the LUT concept to suppress the remaining noise. The trend of the LUT curve determines the relation between input and output signal. Normally this conversion is flash processed onboard by a field programmable gate array (FPGA) inside the camera. In this case, the processed data is the camera output. The noise is clipped by defining a knee just above the noise floor. That causes the corrupted intensity values below the noise floor to be set to zero and the real intensities to remain stationary. Signals within the full dynamic range of the sensor make up to the camera output.

For the sake of completeness, it should be noted that there are more benefits to use a LUT conversion for onboard signal processing, not just noise suppression. If the camera is not only for machine vision, but to display or plot the acquired images or to give results pleasing to human perception, then the non-linearities of the displaying device and/or the human eye have to be compensated. The shape of the LUT curve can be easily adapted to comply with such requirements. For instance, the human eye is logarithmically, not linearly sensitive to light intensity. Consequently, the LUT filter must accommodate this characteristic and is called gamma correction [16]. A gamma curve is shown in Figure 3.5b. Another advantage is to provide a slope to digital conversions of different resolutions. Typically there are 10 and 8 bits, respectively, for CCD sensor and camera output. That significantly reduces data mission bandwidth and allows the transmission of higher frame rates. Communication bandwidth is often the time limiting factor in a camera system.

3.2.3 White balance

The human visual system has the ability to adapt to ambient light conditions and makes subjective adjustments in order that objects, known to be white, continue to appear white. That is to say, on observing the color of a white patch under varying illuminations, the color is perceived as white by our eye-brain system, although the different impinging light spectra generate different colors; this phenomenon is called color constancy [10]. Digital cameras do not have this ability and thus, the subject of white balance is to ensure true colors for digital imaging.

White balance algorithms map the obtained color information pixelwise to color descriptors, which are independent of the illuminanion. This mapping is linear [17] and can be performed via a 3 × 3 diagonal matrix for trichromatic color systems [18], e.g. the red-green-blue (RGB) color model. The coefficients of this diagonal matrix are derived from the ambient illuminant conditions when the image is captured and do not change from pixel to pixel. Consequently, white balance can be treated as adjusting the gain for every band of the imaging sensor. The equation for the mapping is given with:

$$\mathbf{d} = \mathsf{A}\mathbf{p} \tag{3.6}$$

where $\mathbf{d}$ is the illuminant independent output vector, $\mathbf{p}$ is the input vector, containing the obtained color information, and A includes the conversion coefficients for the mapping. For trichromatic color systems, Equation 3.6 becomes:

$$\begin{bmatrix} d_{Red} \\ d_{Green} \\ d_{Blue} \end{bmatrix} = \begin{bmatrix} a_{Red} & 0 & 0 \\ 0 & a_{Green} & 0 \\ 0 & 0 & a_{Blue} \end{bmatrix} \begin{bmatrix} p_{Red} \\ p_{Green} \\ p_{Blue} \end{bmatrix}. \tag{3.7}$$

The complexity of white balance algorithms lies in the determination of the diagonal mapping matrix A. Only an accurate estimation of the mapping matrix guarantees color constancy. It is aggravating that the information about the illumination at the instant of image capture is normally not available. Thus, the coefficients of the matrix have to be estimated from the captured image data. In general, there are two strategies to perform this estimation. While the fixed white-point correction is easy to implement, there exist various methods for the automatic estimation of A. The selection of the algorithm to use is strongly dependent on the content of the captured scene, the prevalent illumination and available computational resources of the camera system.

3.2.3.1 Fixed white-point correction

The principle of this approach is that gray color must induce equal camera responses in all three channels. Derived from this idea, the intensity values of the output vector of a gray illuminated pixel have to be:

$$d_{Red} = d_{Green} = d_{Blue}. \tag{3.8}$$

To attain this, a calibration image of a gray colored patch is captured and the average values in the red, green and blue channel are calculated. For n color triplets the mean values are:

$$\begin{aligned} \overline{p}_{Red} &= \tfrac{1}{n} \sum_{i=1}^{n} p_{Red}, \\ \overline{p}_{Green} &= \tfrac{1}{n} \sum_{i=1}^{n} p_{Green}, \\ \overline{p}_{Blue} &= \tfrac{1}{n} \sum_{i=1}^{n} p_{Blue}. \end{aligned} \tag{3.9}$$

The ratios of these mean values and a referenced color channel give the deviations between the bands. The reciprocal values of the deviations lead to the sought camera gains. Using the green channel as the referenced one yields:

$$\begin{aligned} a_{Red} &= \tfrac{\overline{p}_{Green}}{\overline{p}_{Red}}, \\ a_{Green} &= 1, \\ a_{Blue} &= \tfrac{\overline{p}_{Green}}{\overline{p}_{Blue}}. \end{aligned} \tag{3.10}$$

In applications, where constant illumination is guaranteed, this method provides satisfactory results. The advantage is that once the camera system is calibrated, no computational resources are needed, other then to amplify the color channels with appropriate gains. This makes the fixed white-point correction suitable for machine vision. In contrast, manual calibration, required for changing light conditions, limits its usage to common photography.

3.2.3.2 Automatic white-point correction

The demand for accurate color quality in digital imaging even under varying illuminations has led to the development of several techniques for the automatic white-point correction. One of the simplest representative of this group is based on gray world theory. The principle of this assumption is that all colors in an image will integrate to gray. From that point on, the strategy for the calculation of camera gains is similar to the one used for the fixed white-point correction. All intensities of the captured image are averaged for each color channel and their ratios, with respect to a reference color, yield the conversion coefficients needed for the mapping. The disadvantage of this method is the dependence of the resulting color quality on the content of the captured scene. Processing the snapshot of a rainbow may ensure success, but using this algorithm for highly chromatic images will fail. Taking the snap of a green plant for instance, the chances of its image averaging to gray are about nil. A further development of this algorithm is the implementation of constraints. The exclusion of brightly colored pixels from averaging yields better results, but introduces the problem of defining a suitable threshold for the boundary.

More sophisticated and potentially very complex algorithms were published on this topic. Forsyth [17] presented a novel approach using the constraint that the observed colors in the image must be dependent on the color of the illuminant. This arises from the fact that surfaces can reflect no more light than is cast upon them. Laine *et al.* [19] published a method which estimates the prevailing illumination conditions on the basis of additional information, captured at the time when the image was taken. Finally, a good review of white balance algorithms is given in Barnard's PhD thesis [20].

3.2.4 Demosaicing

The preprocessed image still appears as if acquired with a monochrome camera. As described in Section 3.2.1, the chosen camera model contains a Bayer CFA to obtain a two-dimensional mosaic of color components. That is because only one color is measured at each pixel of the resulting image. A real color image, needed for further processing, requires at least three color samples at each pixel location. To obtain this, the missing color components have to be estimated. This challenge is called CFA interpolation or demosaicing. All demosaicing algorithms use information for the interpolation from the spatially adjacent CFA data and thus, the applied method is closely related to the CFA pattern used. In other words, an algorithm for the demosaicing of a cyan-magenta-yellow-green (CMYG) pattern is not suitable for a Bayer CFA. That requires the knowledge of the exact arrangement of the color components in the CFA for the selection of a proper demosaicing algorithm. However, the task of demosaicing is usually the most intensive computational step in the image processing chain.

Many different demosaicing strategies were proposed [21, 22, 23] with various results. Most of these restoration algorithms are performed to faithfully reproduce the human perception or to give good results for printing and other reproduction processes. These methods produce errors associated with edges, making them show up as fringes. Furthermore, classical linear demosaicing techniques, e.g. bilinear interpolation tend to smooth the image and result in noticeable color artifacts along sharp features. This is why these methods perform poorly for color detection and machine vision. Algorithms that perform well [24, 25] are computationally complex and can only be considered for offline processing applications, where processing time is not critical.

In this application, where color detection is combined with metric measurement, the reconstruction of the RGB color triplet must be efficient and fast, reproducing the original color information with high fidelity. The following very simple demosaicing algorithm overcomes the disadvantages of conventional techniques and fulfills all the requirements of this application:

The principle of the introduced demosaicing method is shown in Figure 3.6. The idea is to design the required RGB triplets from spatially adjacent pixel quadruplets that con-

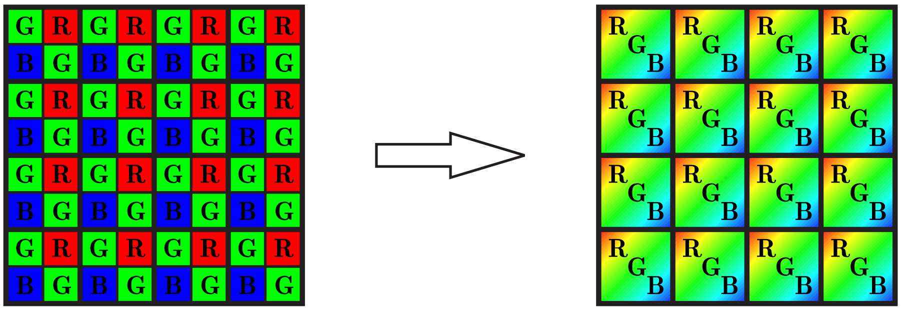


Figure 3.6: Demosaicing scheme used. Left: The sensor's Bayer color filter array arrangement. Right: Resulting RGB data after combining four neighboring pixels of the CFA to a RGB triplet.

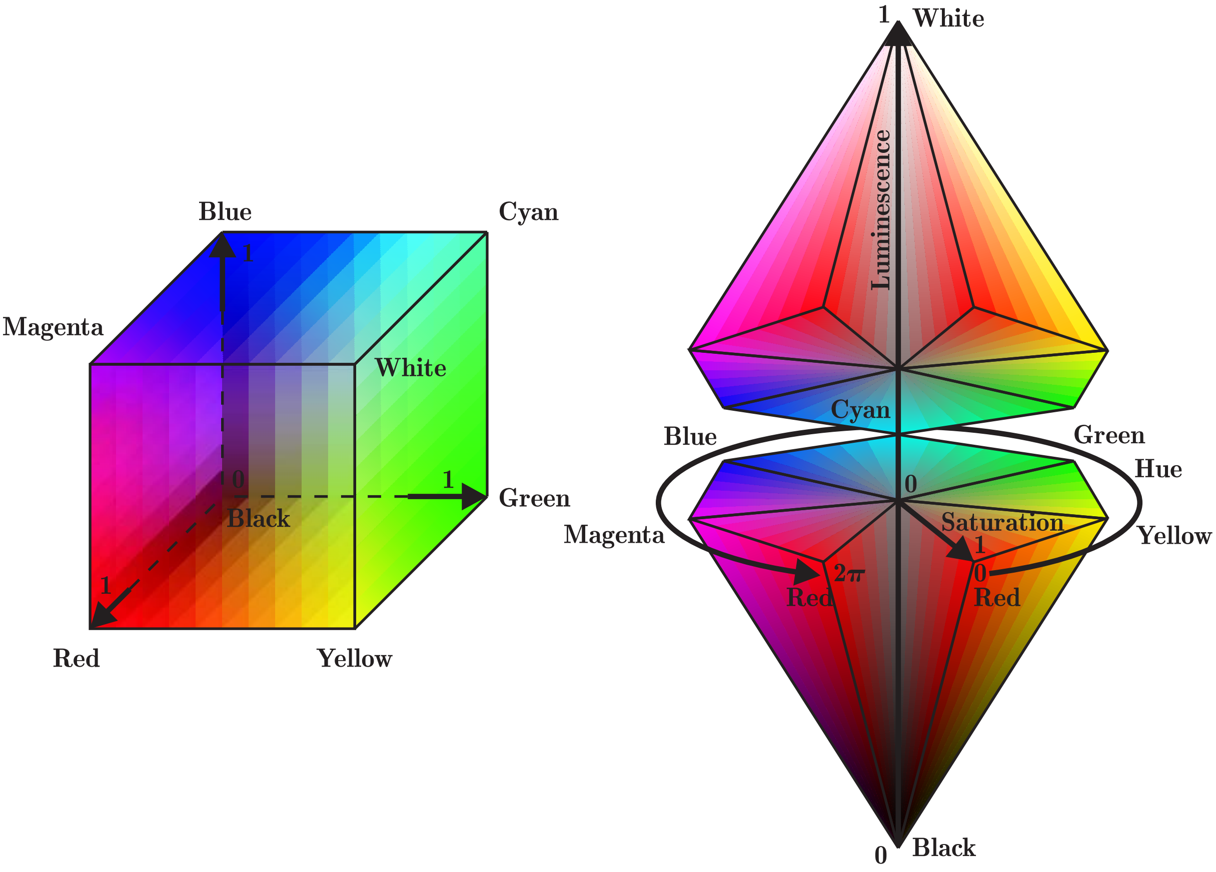
tain the needed color information. Considering any square quadruplet in the CFA pattern, the red and blue values are assigned directly to the output and the green values are averaged. Going through the whole image using this method, the $n \times m$ resolution from the starting Bayer array is decimated to $n/2 \times m/2$ for the resulting RGB image.

The quality of the RGB image obtained in this manner is comparable to that of a three chip camera solution. The three color values for a resulting pixel conform exactly to the measured information for the particular area on the sensor. No geometric errors are generated on interpolating pixel data. Thus, no further processing, except averaging the green values, is required for each resulting color triplet, the calculating time is kept at a minimum and the demosaicing can be performed in real-time.

3.2.5 Color modeling

One of the primary problems in color detection is color constancy. Due to the non Lambertian scattering properties of the observed surface, the appearance of color is dependent on the angle of the incident light (see Section 3.1). Taking the varying texture, position and orientation of the target surface into account, the angle of incident is variable and yields to variations in the appearance of color among markings. All the three color components of the RGB space are subject to variations due to the effect mentioned above and makes it difficult to specify a desired color in the RGB space. This significantly complicates the reliable separation of the required color from the background and makes this color model inapplicable for color detection. Thus, the task of color modeling is to transform the color values from the RGB space into a suitable color model for a reliable color detection to be performed via thresholding.

Many different methods and applications for color detection were proposed [26, 27, 28, 29, 30], consequently, it was shown that the hue-saturation-luminescence (HSL) color model is suitable for color detection. In contrast to the RGB model, the HSL space is an unrendered color model, i.e. it is not convenient for hardware output media such as monitors or printers. The HSL model is patterned on the way the human perceptual system interprets color information [31]. This leads to the definition of the HSL descriptor triplet. The hue value refers to the perceived color, i.e. the dominant wavelength of the color. The property of the saturation value is to distinguish the purity of the color and the luminescence value stands for the brightness. The advantage of this color model is the separation of the chrominance component, represented by hue and saturation, from the luminescence information. This makes the HSL model convenient for applications where the chrominance is of importance rather than the overall color perception, e.g. color recognition in machine vision.



(a) RGB color model

(b) HSL color model

Figure 3.7: Left: The RGB color cube, based on a Cartesian coordinate system. Right: The HSL color space, described by a double-hexcone in a cylindrical coordinate system.

The RGB and the HSL color models are illustrated in Figure 3.7a and 3.7b. In the RGB model, each color is a sum of the primary spectral components, i.e. red, green and blue. Its normalized color space is described by the unit cube in Cartesian coordinate system, where the primary colors are given along the three axis. This leads to the following dye distribution - black is given at the origin, white at the furthermost corner from the origin, the pure colors at the corners which are intersected by the axis and their complementary colors at the three remaining corners of the cube. The gray scales are distributed along the diagonal between black and white.

Referring to Figure 3.7b, the HSL model can be derived from the RGB model via coordinate transformation. The HSL space is nonlinear and is described by a double-hexcone, which corresponds to the straightened up RGB cube. Its three components are given in a cylindrical coordinate system. The luminescence is defined between both cone ends along the vertical axis from black to white. The hue is measured by the angle around the luminescence axis and wraps-around the perimeter at $2\pi = 0$. Finally, the saturation extends radially outwards till the borders of the hexcone. In other words, given a constant luminescence, every conic section is a chromatic plane where color is specified by its hue and saturation.

For normalized RGB values the conversion to the HSL space is defined by the following equations [32, 33]:

$$\begin{aligned} L &= 0.5\,\big(max(R,G,B)+min(R,G,B)\big), \\ S &= \begin{cases} \frac{max(R,G,B)-min(R,G,B)}{max(R,G,B)+min(R,G,B)} & \ldots\ L < 0.5 \\ \frac{max(R,G,B)-min(R,G,B)}{2-(max(R,G,B)+min(R,G,B))} & \ldots\ L \geq 0.5 \end{cases}, \\ H &= \begin{cases} \frac{2\pi}{6}\,\frac{G-B}{R-min(R,G,B)} & \ldots\ R = max(R,G,B) \\ \frac{2\pi}{6}\left(2+\frac{B-R}{G-min(R,G,B)}\right) & \ldots\ G = max(R,G,B) \\ \frac{2\pi}{6}\left(4+\frac{R-G}{B-min(R,G,B)}\right) & \ldots\ B = max(R,G,B) \end{cases}, \end{aligned} \tag{3.11}$$

with the exceptions: $L = 0$ or $L = 1 \rightarrow S = 0$, $H = 0$ and $S = 0 \rightarrow H = 0$. These definitions have to be specified, since the hue becomes singular in achromatic points.

As the color modeling is a part of the real-time environment, this transformation is performed by means of look-up tables for every pixel in the image.

3.2.6 Thresholding with morphological analysis

All the previous processing described was concerned with refining and processing the acquired image to provide suitable input data for reliable color detection: this is the task of this final step in the image processing pipeline. Several methods are combined to extract the sought information from the preprocessed images. Firstly, a thresholding procedure is used to separate the sought color from the background. Moreover, morphological operators [31, 32] are applied to the color patches found so as to smooth their boundaries and to reject artifacts. Subsequently, the remaining color markings are labeled [31, 34], framed and finally, depending on their location, merged to obtain the sought pixel coordinates of the marking's surrounding rectangle. Thus, every detected color patch $\mathbf{x}_i$ is described by a four dimensional vector

$$\mathbf{x}_i = \begin{bmatrix} x_{Left} \\ x_{Top} \\ x_{Right} \\ x_{Bottom} \end{bmatrix}, \tag{3.12}$$

where x_{Left}, x_{Top}, x_{Right} and x_{Bottom} are the pixel coordinates of the vertices of the axially parallel rectangular bounding box.

3.2.6.1 Thresholding

The method of thresholding is used for image segmentation. The purpose is to make a distinction between the color marking and the surrounding background. Therefore, a segmentation criterion is determined, consisting of three pairs of threshold values. Pixels matching this segmentation criterion are to be separated from the background. In this manner, the HSL color image $H(x, y)$, $S(x, y)$ and $L(x, y)$ is converted to a binary image $I_b(x, y)$, according to:

$$I_b(x,y) = \begin{cases} 1 \ \ldots & \begin{array}{ccccc} \big(H_{Min} & \leq & H(x,y) & \leq & H_{Max}\big) \ \wedge \\ \big(S_{Min} & \leq & S(x,y) & \leq & S_{Max}\big) \ \wedge \\ \big(L_{Min} & \leq & L(x,y) & \leq & L_{Max}\big) \end{array} \\ 0 \ \ldots & else \end{cases} . \tag{3.13}$$

The threshold values, representing the sought color, define a sphere in the body of the HSL model; whereby, the determination of colors is performed by the hue (see Figure 3.7b). Considering, that the hue is meaningless when the intensity is close to its extremes, i.e. black or white, and unstable for small saturation values an achromatic region can be determined. The exclusion of this region from thresholding leads to the definition of the saturation and the luminescence thresholds. Plataniotis *et al.* [32] defined the borders of the achromatic region as:

$$\begin{aligned} S_{Min} = 0.1 \quad &, \quad S_{Max} = 1.0, \\ L_{Min} = 0.1 \quad &, \quad L_{Max} = 0.9. \end{aligned} \tag{3.14}$$

The remaining thresholds for the hue are dependent on the sought color and adjusted by the system user. Thus, this method can be used for the detection of all chromatic colors.

3.2.6.2 Filtering

Since thresholding is a pixel-based technique and color detection decision depends solely on color features at individual pixels, the remaining binary output still contains small artifacts. These artifacts have to be removed before analyzing the image. This is performed via morphological operators [31, 32]. Morphological operations are based on the set theory and are specified as interactions between objects within an image and a predefined structural element. The two basic operations in morphology are dilation and erosion. These primary operators form the basis of all other morphological operations and are introduced below. The following definitions are considered for binary images, having their sets in $\mathbb{Z}^2$.

Dilation:
The morphological operator dilation $\oplus$ is the transformation which combines two sets using vector addition of set elements. If I_b and B are sets in $\mathbb{Z}^2$, the dilation of I_b by B is the set of all possible vector sums of pairs of elements, one from each of the sets I_b and B, and is defined with:

$$I_b \oplus B = \left\{ \ x \in \mathbb{Z}^2 \mid x = i_b + b \text{ for some } i_b \in I_b \text{ and } b \in B \ \right\}. \tag{3.15}$$

I_b is the binary image, x is a point and B is the structuring element. The size and shape of B was specified through experiments. A square shape at the size of 3×3 was determined to give best results for this application:

$$B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & \boxed{1} & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,, \tag{3.16}$$

where $\boxed{1}$ denotes the origin of the structuring element. The size of objects within an image are expanded by dilation.

Erosion:
Erosion is the dual translation of dilation. If I_b and B are sets in $\mathbb{Z}^2$, the erosion of I_b by B is the set of all elements of x for which $x + b \in I_b$ for every $b \in B$. In other words, the morphological operator erosion $\ominus$ is the transformation which combines two sets using vector subtraction of all set elements according to:

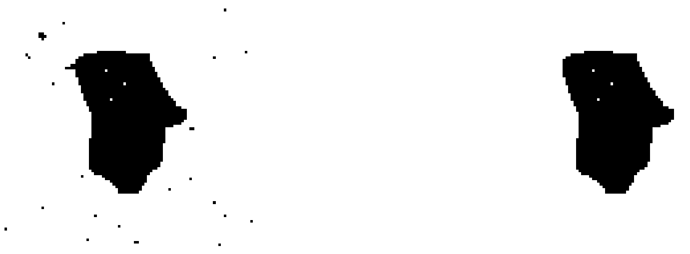
$$I_b \ominus B = \left\{\; x \in \mathbb{Z}^2 \mid x + b \in I_b \text{ for every } b \in B \;\right\}. \tag{3.17}$$

The operation of erosion shrinks the size of image objects.

Opening:
This morphological operator is a composition of both primary operators. The opening operator $\circ$ is defined as an erosion followed by a dilation and given as:

$$I_b \circ B = (I_b \ominus B) \oplus B. \tag{3.18}$$

The objects within an image are first shrunk and then re-expanded, not necessarily to original shape and size. Firstly, the erosion decreases the size of all objects, whereas sets smaller than the structuring element are removed. The subsequent dilation enlarges the objects again. The morphological operation of opening tends to smooth contours of the objects and removes small artifacts. The effect of opening is shown in Figure 3.8.



(a) Original image **(b) Processed image**

Figure 3.8: Illustration of the opening operation: (a) Original image after thresholding, influenced by artifacts. (b) Resulting image after applying the morphological opening operator.

This operation is used for the removal of artifacts, arising from thresholding. The shape and size of the structuring element, shown in Equation 3.16, were derived from the size of these artifacts.

3.2.6.3 Object identification

The desired information, the unique identification of all individual objects, needs to be extracted from the filtered binary image. This is obtained on applying a labeling algorithm [31, 34] to the image, which performs a clustering of spatially neighboring pixels to form disjoint objects. The outcome of this method are vectors $\mathbf{x}_i$, as shown in Equation 3.12, containing the coordinates of the axially parallel bounding box for every disjoint object within the input image.

Labeling algorithm:
The filtered binary image I_b may be partitioned into n subregions $R_1 \ldots R_n$, taking the following considerations into account:

$$\begin{aligned} &\text{(a)} \quad I_b = \bigcup_{i=1}^{n} R_i, \\ &\text{(b)} \quad R_i \text{ is a coherent region, } i = 1, 2, \ldots, n, \\ &\text{(c)} \quad R_i \bigcap R_j = \emptyset, \text{ for all } i \neq j, \end{aligned} \tag{3.19}$$

where $\bigcup$ is indicates the union, $\bigcap$ denotes intersection and $\emptyset$ is defined as the null set in the set theory. The first condition denotes that the segmentation is complete. Term (b) indicates that every region is coherent, i.e. the points within a region are connected. Finally, the last condition takes into account that the different regions do not intersect.

The labeling algorithm conforms to Equation 3.19 by assigning different labels to the disjoint regions. This is performed in two passes through the image data I_b as follows:

1. The binary image I_b is scanned from the upper left to the lower right, pixel by pixel and row by row. The idea of this algorithm is to assign a label v to every non-zero pixel, whereby the value of v is dependent on the label values of the pixel's neighbors as described below. Neighbors are defined by their connectivity status. Having connectivity status 4, the upper and the left adjacent neighbors are considered, have status 8, the two upper diagonal adjacent neighbors are also taken into account.
    - The pixel values of the considered neighbors are zero. In this case the pixel belongs to a new region and an unused label is assigned.
    - The pixel value of at least one considered neighbor is non-zero and in case of more than one non-zero neighbors, their labels are identical. The contemplated pixel belongs to the same region and the label of its neighbors is assigned.
    - The pixel value of at least two considered neighbors are non-zero and their labels are different. The pixel connects two regions and the label of ether of its neighbors is assigned. A note in the equivalence table is made to indicate that the two labels are equivalent.

At the end of the scan all pixels having non-zero values are labeled, but some of these labels may have different values even through equivalence, due to label collision in coherent regions. A second pass is necessary to resolve these collisions.

2. The pairs of equivalent labels are sorted into equivalence classes according to the information stored in the equivalence table and re-labeled with a new label. This yields distinct labels for all disjoint regions.

To fruit a better way to handle disjoint objects, the coordinates of their axially parallel bounding box are extracted during the labeling process. This is easily performed when storing the coordinates of every processed pixel in an individual list, according to its label value. Finally, after finishing the second pass, the minima and maxima of the x- and y-components in each list are extracted. These extreme values are the vertices of the object's bounding box and can be stated as given in Equation 3.12 for all objects within the image.

3.2.6.4 Merging

All detected color markings indicate the presence of a surface defect on steel billets. The aim of this work is the fully automatic guidance of a grinding robot for the machining of these surface defects. To achieve efficient but simple grinding performance, the merging of two or more adjacent color markings is entertained when the footprint of two adjacent particles fall within the active area of the grinding belt. However, machining two or more surface defects at once requires some special algorithmic procedures that are described below:

The color markings are characterized by the pixel coordinates of the marking's bounding box $\mathbf{x}_i$, which were determined in the prior processing step. Thus, for n detected color markings, the input for the merging algorithm is:

$$[\mathbf{x}_1 \ldots \mathbf{x}_n] = \begin{bmatrix} x_{Left,1} & x_{Left,2} & \cdots & x_{Left,n} \\ x_{Top,1} & x_{Top,2} & \cdots & x_{Top,n} \\ x_{Right,1} & x_{Right,2} & \cdots & x_{Right,n} \\ x_{Bottom,1} & x_{Bottom,2} & \cdots & x_{Bottom,n} \end{bmatrix}. \quad (3.20)$$

Referring to Figure 3.9, the criterion for merging is based on the distances Δx and Δy between adjacent rectangles. If both distances are below user selected thresholds Δx_{Max} and Δy_{Max} the two objects are combined. The values of these thresholds are dependent on the architecture and size of the belt grinding unit. This algorithm produces a sorted list, containing the coordinates of all detected color markings, which may be merged in accordance with criterion stated.

In the following, the function of the algorithm is described:

1. The algorithm starts by resetting the counter variable $i = 1$. Furthermore, the coordinates of the objects are sorted by their ascending x_{Left} coordinate, starting with the smallest value. That yields a list, wherein the rectangles are sorted by their left border from left to right:

$$[\mathbf{x}_1 \ldots \mathbf{x}_n] = \begin{bmatrix} x_{Left,1} & \leq & x_{Left,2} & \leq & \cdots & \leq & x_{Left,n} \\ x_{Top,1} & & x_{Top,2} & & \cdots & & x_{Top,n} \\ x_{Right,1} & & x_{Right,2} & & \cdots & & x_{Right,n} \\ x_{Bottom,1} & & x_{Bottom,2} & & \cdots & & x_{Bottom,n} \end{bmatrix}. \quad (3.21)$$

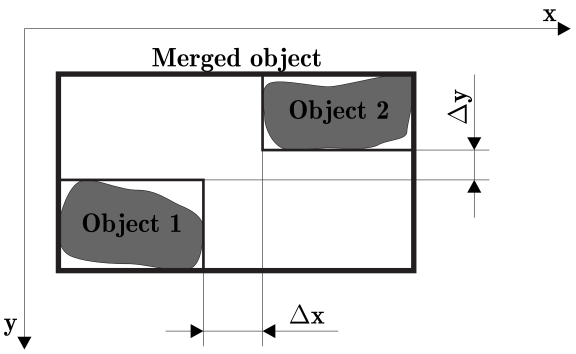


Figure 3.9: If the distances Δx and Δy between two objects are smaller than user defined thresholds then the objects are merged. The thresholds are dependent on the architecture of the belt grinding unit and set by the user of the system.

2. The value of the counter i is compared with the number of objects n in the list. If i reaches n the process terminates and the merging is complete.

3. The distances Δx and Δy between two neighboring objects are calculated:

$$\begin{aligned} \Delta x &= x_{Left,i+1} - x_{Right,i}, \\ \Delta y &= \begin{cases} x_{Top,i} - x_{Bottom,i+1} & \ldots \; x_{Bottom,i+1} < x_{Bottom,i} \\ x_{Top,i+1} - x_{Bottom,i} & \ldots \; x_{Bottom,i+1} \geq x_{Bottom,i} \end{cases} . \end{aligned} \tag{3.22}$$

4. The calculated distances Δx and Δy are compared with thresholds Δx_{Max} and Δy_{Max}. That yields two possible cases:

   - **if** $\big((\Delta x \leq \Delta x_{Max}) \wedge (\Delta y \leq \Delta y_{Max}) \;=\; \text{TRUE}\big)$, **then**
     The criterion for merging is fulfilled. The object at the position i in the list is replaced by a new rectangle, according to the bounding box of the merged objects. The merged coordinates are:

     $$\begin{aligned} x_{Left,i} &= x_{Left,i}, \\ x_{Top,i} &= min(x_{Top,i}, x_{Top,i+1}), \\ x_{Right,i} &= max(x_{Right,i}, x_{Right,i+1}), \\ x_{Bottom,i} &= max(x_{Bottom,i}, x_{Bottom,i+1}), \end{aligned} \tag{3.23}$$

     and the object at the position $i+1$ is erased. The counter i is reset to $i = 1$ and the algorithm restarts at step 2.

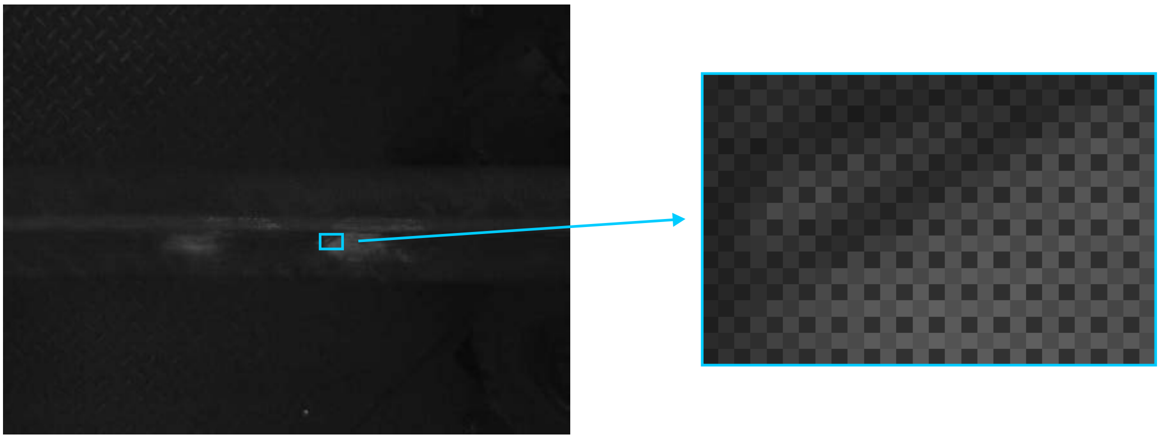
   - **if** $\big((\Delta x \leq \Delta x_{Max}) \wedge (\Delta y \leq \Delta y_{Max}) \;=\; \text{FALSE}\big)$, **then**
     At least one coordinate difference between the objects is too large. No merge takes place, the counter i is increased by 1 and the algorithm returns to step 2.

3.3 Example of the color detection

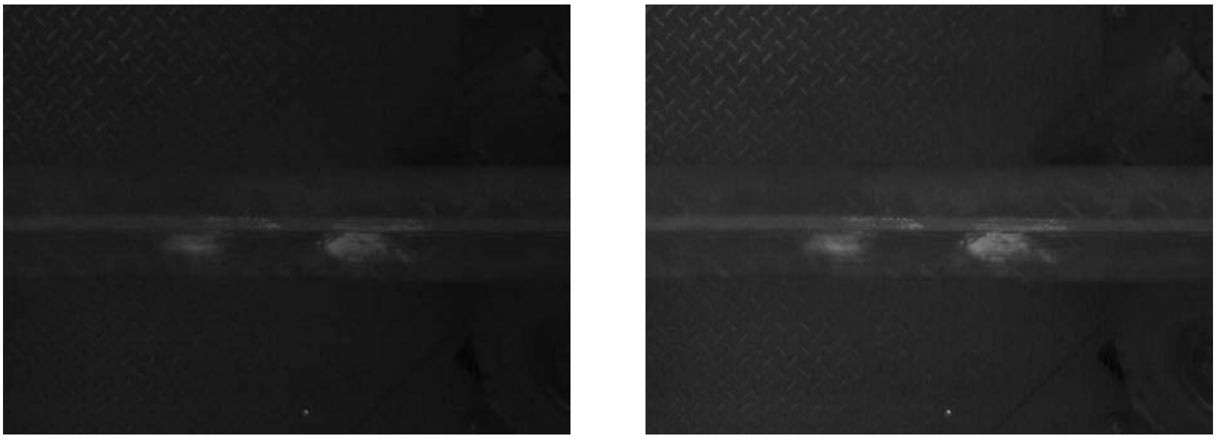
The example presented here is an image chosen at a random sample, from the continuous production process of an optical servoing unit, which is part of a fully automated machining facility. The image contains a square billet with color markings, indicating the presence of surface defects. The camera used is a progressive scan model, having one megapixel resolution and a Bayer CFA to obtain the color information. The snapshot was made while the billet was moving along the billet conveyor. The aim is to obtain the pixel coordinates of the color marking's bounding box, which are required for the guidance of the grinding robot.

The Figures 3.10a - 3.10i, illustrate the billet surface and the color detection process. This is summarized as follows:

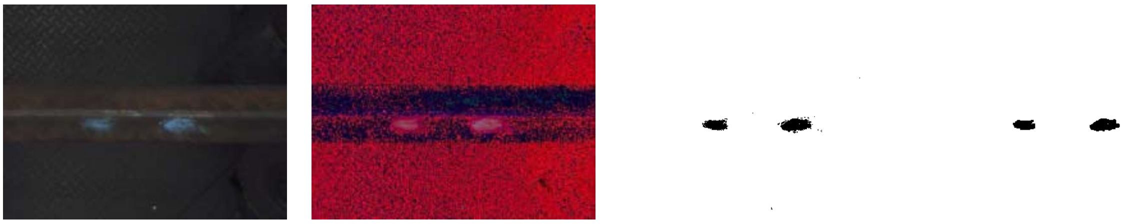
- Figure 3.10a: The snapshot of the event is shown. The image appears to be monochrome. A billet is assumed to be horizontally located in the middle of the scene.
- Figure 3.10b: An enlarged detail from the original image is highlighted. As can be seen, the image consists of a mosaic of color components, where each pixel is formed according to the Bayer CFA array with intensity values determined by the reflectance and illumination of the observed scene.
- Figure 3.10c, shows the image after noise suppression. The intensity values below the noise floor are set to zero, the real intensities remain stationary.
- Figure 3.10d: The image after white balancing. Each color channel was amplified by its individually assigned gain to obtain true colors for the ambient illumination. The conversion coefficients were determined via fixed white-point correction during calibration.
- Figure 3.10e: Demosaicing; spatially adjacent pixel quadruplets are combined to form a RGB pixel, whereby the resolution of the resulting image is halved in both directions. The bluish color markings on the surface of the billet becomes visible.
- Figure 3.10f, illustrates the image after mapping the RGB data into the HSL space. The scene appears improper since the HSL model is an unrendered color space. Thus, the monitor/printer displays the hue channel as red, the saturation as green and the luminescence as blue. The advantage of the HSL space is the separation of the chrominance information from the luminescence, which enables the color detection performed by thresholding.
- Figure 3.10g: The binary image after thresholding is displayed. The bluish color markings are extracted from the background using proper thresholds for the hue channel. As expected, the generated binary image contains small artifacts due to the pixel-based nature of thresholding.
- Figure 3.10h: The artifacts are removed using the morphological opening operator. Subsequently, the sought coordinates are determined on performing the object identification.
- Figure 3.10i: The detected bounding box of the color patches are illustrated in the RGB image. These coordinates are the basis for the guidance of the robot.



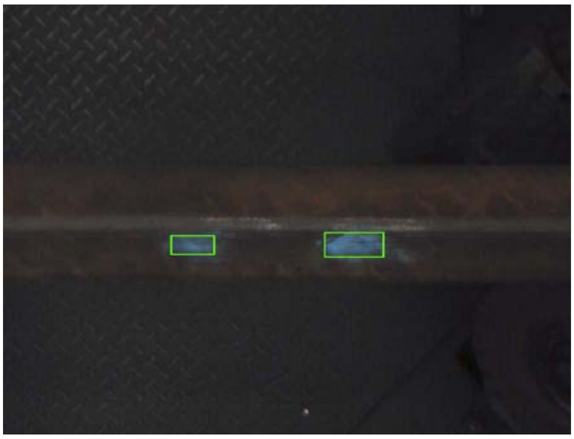
(a) Original image **(b) Enlarged detail**



(c) Signal preprocessing **(d) White balance**



(e) Demosaicing **(f) Color modeling** **(g) Thresholding** **(h) Morphology**



(i) Color detection

Figure 3.10: Example of the color detection: Illustrated are the acquired raw data, the intermediary results as image data flows through and is processed in the pipeline and the final outcome of the color detection process. It shows how the appearance of the image changes after each processing step.

Chapter 4

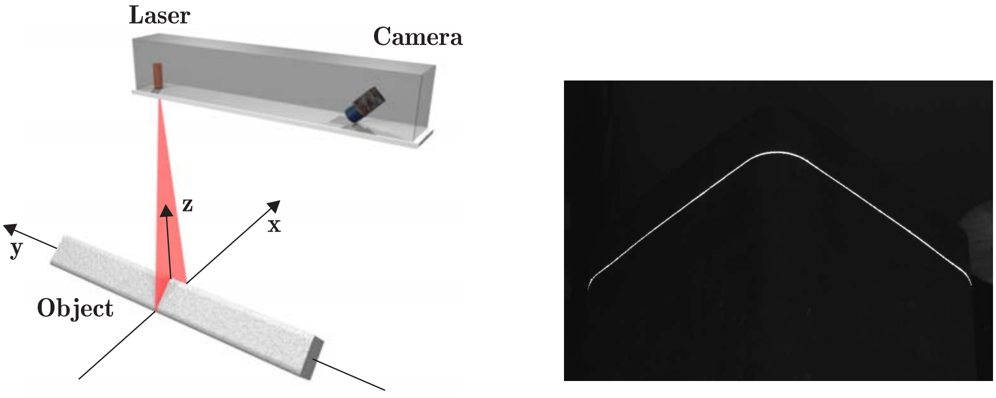
Measurement of the Cross Section

The guidance of the industrial 6R robot, that performs the surface machining, requires an exact determination of the position, orientation and geometry of the steel bars. Furthermore, the knowledge of these basic attributes enables the control of the conveyor, positioning the material, to attain a stable emplacement for the duration of the grinding process. Consequently, an accurate measurement of the cross section is indispensable for this application.

This chapter is concerned with the measurement of the cross section of two types of steel billet relevant to this application: Circular billets, characterized by a circular cross section and square billets, distinguished by a square cross section with rounded corners of circular shape. The chosen measuring principle uses light-sectioning technology, which is a suitable arrangement of stationary optics that will function and survive within steel production environment. For this purpose, an image of the scene is captured with a monochrome camera, positioned inside the plane-of-light scanner enclosure. Once the image is available, the peak intensity position of the light-plane intersection curve is converted into pixel-coordinates. The center of gravity calculation, introduced for this purpose is a powerful tool that exploits the spread of the laser line in the image to calculate the peak position with sub-pixel precision. These pixel-coordinates of the light section are subsequently mapped to real-world coordinates by means of a collineation matrix that is obtained during the calibration (see Chapter 6) of the measurement setup. In this way, the undistorted data points, describing the profile of the billet, are obtained. Additionally, several fitting routines are applied to determine parameters that describe the cross section geometry. These routines fit circles to circular cross sections and, in case of square billets, a combination of a fitting algorithm for orthogonal lines and a novel fitting algorithm for circles, which are defined by two intersecting tangents, is used. In contrast to the real-time performance required for color detection, the time required for this task is less critical, since the measurement is carried out on a stationary billet. This permits the use of more accurate, and accordingly, computationally more expensive routines to fit the geometry, e.g. iterative algorithms. Using statistical confidence level estimation techniques, a sample calculation of square billet cross section parameters is presented to demonstrate the high level of accuracy attained.

4.1 Principle of the light sectioning method

The method of light sectioning is a contact-free and non-destructive range imaging technique for the determination of cross sections [35, 36, 37, 38]. An object, whose cross section is to be surveyed, is intersected by a plane-of-light from one direction. Usually a laser with a line optics is used as light source. It projects a contour line on the surface of the object which is observed by a monochrome camera. Through the fixed relation between the plane-of-light and the camera chip, the collineation between them is uniquely determined. Consequently, the undistorted contour can be calculated and the cross section fitted with suitable algorithms. The collineation is defined once and for all when the measurement system is calibrated. On acquiring a sequence of several cross sections, while the object is moving in one direction, it is possible to reconstruct its complete surface profile with this method. The principle of light sectioning is illustrated in Figure 4.1a, a captured image of the intersection curve is shown in 4.1b.



(a) Light sectioning **(b) Captured image**

Figure 4.1: The principle of the light sectioning method: (a) A directed plane-of-light is intersected by an object, whose cross section is to be determined. The resulting intersection curve (b) is captured by a camera from another position. The undistorted cross section can be calculated via the fixed arrangement of the laser and the camera.

Light sectioning is primarily limited by two factors. Firstly, occlusions may occur due to shadowing effects in the optical path of the measurement system, whereby two different types are distinguished: On the one hand, camera occlusions appear when a part of the intersecting curve is located in a region, outside the camera's field of view. On the other hand, laser occlusions occur when the laser light does not reach areas surveyed by the camera. However, both cases lead to a loss of data points in the occluded area. The occlusion problem can be overcome with a proper selection of the camera and/or laser location or the use of multiple camera and/or laser systems. Referring to the placement of the camera, laser and object in Figure 4.1a, it is obvious that only the top side of the object's surface can be examined with such a setup. The bottom side is occluded by the object itself. To achieve full coverage, the use of a multiple camera/laser system is required. Such multiple view systems increase the complexity of a measurement system considerably. Each camera used leads to an individual coordinate system which must be mapped to a common world coordinate system. This alignment is called camera registration. However, the assignment of a multiple view system is not necessary for this application, because partial light contours are sufficient to fit accurate cross section. Thus, further discussions of this are not required. A detailed overview of methods to manage occlusions and the associated registration problem are given in [39, 40].

The second restriction which has to be considered is the limited sampling rate of such measurement systems. This is due to the need to acquire a visible intensity image of the contour line of the object for every cross section. In doing so, the time limiting aspect arises due to required buffering of the large amount of acquired image data. Typical state-of-the-art cameras have resolutions of approximately 1 megapixel with a 10 to 14 bit pixel word length. The standard interface for data transfer in industrial cameras is Camera Link^{TM} (base configuration), having a maximum transmission rate of 2.38 gigabit/second [41]. According to this, the theoretical sampling rate for the chosen model is 238 frames per second, whereas the practically realizable rate is well below this upper value. In applications, where the surface of the object is to be scanned, the effective sampling rate affects the feed rate and finally the scan time of the object significant. To overcome this problem there are several approaches to increase the sampling rate. The simplest one is the reduction of the sensor resolution, but is inevitably associated with a loss of precision. Another technique to increase the transferred frame rate is the usage of a more advanced Camera Link^{TM} (full configuration) transfer standard. The transmission rate is three times higher than that of the base configuration, but this technology is not readily available yet. However, a camera model delivering 500 frames per second at a resolution of 1.3 megapixel is already available [42]. To exploit it, the most effective and maybe the costliest approach increasing the sampling rate is to implement a field programmable gate array (FPGA) inside the camera. The FPGA converts the image data into range data which is transferred to the programmable logic controller (PLC). Such sensors reach a sampling rate up to 20000 profiles per second at a lateral resolution of 1536 pixels [43].

4.2 Extraction of the intersection curve

Once an image of the intersected object is acquired, the peak position of the intersection curve needs to be extracted and converted into range data to perform the subsequent mapping and fitting. To achieve an accurate and robust calculation of the cross section this extraction has to be done with the highest possible accuracy. Thus, the resulting range data to localize the contour line needs to be specified with sub-pixel precision to meet the requirements.

The following method for the extraction of the intersection curve can be split into two parts: The first part gives a coarse determination of the contour line's position. This yields an array of points which is verified by a line walk algorithm to reject outlier points from the curve found; In a second step, a center of gravity (COG) calculation, applied to the tentative line, provides the needed sub-pixel precision for the localization.

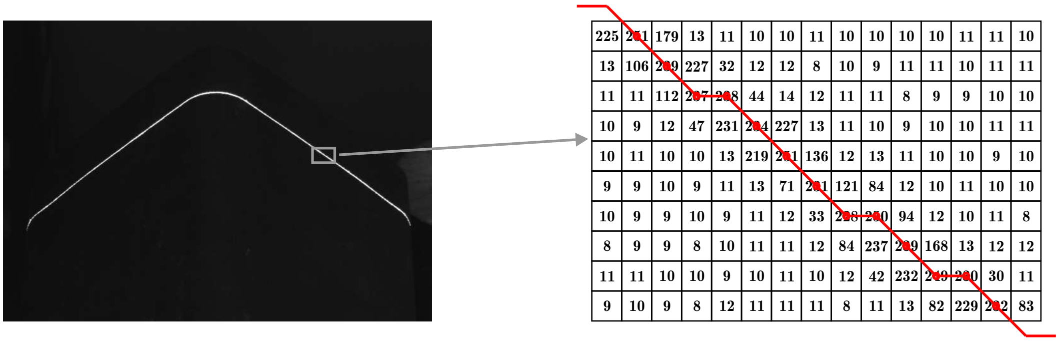
The required accuracy and stability of the described method is attainable in this application only by considering the following arrangements:

- A narrow-band interference filter, centered at the wavelength of the laser, is mounted in front of the camera lens. This rejects most of the ambient light and ensures that the laser is the dominant light source impinging upon the camera sensor.
- The arrangement of the laser and camera relative to each other is as shown in Figure 4.1a: A laser plane intersects the object and is parallel to the cross section while the camera observes the resulting contour with its optical axis oblige to both, the billet axis and the plane-of-light. That yields a flattered, obtuse intersection curve in the acquired image as illustrated in Figure 4.1b.
- Due to the physical nature of the laser, the imaged line is spread over several pixels [35] and conforms to a distribution which is exploited in the sub-pixel interpolation [44]. Consequently, it is very important that the captured image is not overexposed and the intensity values of the pixels, imaging the laser line, stay below the white clip level of the CCD-sensor.

4.2.1 Coarse detection of the intersection curve

The image of the captured intersection curve is scanned, column by column, to find the brightest pixel in every column. If the intensity value of this picture element is higher than a predefined threshold τ_I, its coordinates are stored in an array. In so doing, the resulting trend line is taken as a rough estimation of the position of the intersection curve. The value of the threshold τ_I is set to a medium intensity value to avoid the detection of doubtful pixels at low intensity values. Such artifacts may be caused by some ambient light passing the interference filter.

The result of this image acquisition process is shown in Figure 4.2. The left image shows the captured image; the right one, states the intensity matrix of the enlarged detail and the determined trend of the laser line.



225	251	179	13	11	10	10	11	10	10	10	10	11	11	10
13	106	239	227	32	12	12	8	10	9	11	11	10	11	11
11	11	112	237	238	44	14	12	11	11	8	9	9	10	10
10	9	12	47	231	244	227	13	11	10	9	10	10	11	11
10	11	10	10	13	219	251	136	12	13	11	10	10	9	10
9	9	10	9	11	13	71	241	121	84	12	10	11	10	10
10	9	9	10	9	11	12	33	238	250	94	12	10	11	8
8	9	9	8	10	11	11	12	84	237	239	168	13	12	12
11	11	10	10	9	10	11	10	12	42	232	249	240	30	11
9	10	9	8	12	11	11	11	8	11	13	82	229	242	83

(a) Captured image **(b) Coarse progression of the laser line**

Figure 4.2: Coarse detection of the intersection curve: An image, containing a light section (a) is scanned to identify the brightest pixel in every column matching a threshold τ_I. The coordinates of these pixels yield a coarse estimation of the position of the intersection curve (b).

The demand for high accuracy requires the rejection of outlier points. Even a single contour point that is accepted can cause a significant error, if used to determine the geometry of the cross section. Possible sources of detection failures may be reflections of the laser light or dead pixels on the camera chip. However, such outliers are isolated and discontinuous with respect to the partially continuous intersection curve and constitute very short segment lengths.Therefore, the intersection curve is taken as the longest continuous segment within the array that contains the coordinates of the contour line.

A line walk algorithm [45, 46] is used to segment the found coordinates into continuous portions. Starting with the second entry of the array, the position of this point is compared with the prior entry. The examined point belongs to the actual segment, if the distance between both points is within a maximum distance threshold τ_D or else, a new segment is started. On incrementing the iteration counter this search continues until the last entry of the array is verified. All segments below a minimum length threshold τ_L are rejected and the remaining segments are merged taking the criterion of the maximum distance threshold τ_D into account. The longest remaining segment is taken as a first approximation of the intersection curve. Thresholds τ_D and τ_L are derived from a-priori knowledge of the measured cross sections.

4.2.2 Center of gravity calculation

The accurate and robust localization of the contour line is achieved through the calculation of the weighted center of gravity. The center of gravity is considered to be the equilibrium point at which the entire mass of an object is concentrated. In machine vision, the mass of the object is replaced by its intensity and this enables us an intensity weighted determination of the object's center with sub-pixel precision. Consequently, this method is widely used in different applications in the field of image processing [45, 46, 47, 48].

The definition of the weighted center of gravity $\mathbf{x} = [x, y]^T$ for gray level images is [45]:

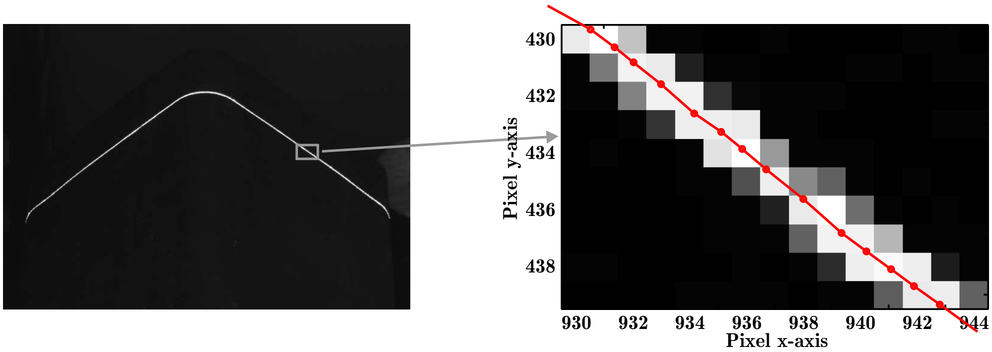
$$\mathbf{x} = \left[\frac{\sum\limits_{x,y\,\epsilon\,\Omega} x\, I^p(x,y)}{\sum\limits_{x,y\,\epsilon\,\Omega} I^p(x,y)}, \frac{\sum\limits_{x,y\,\epsilon\,\Omega} y\, I^p(x,y)}{\sum\limits_{x,y\,\epsilon\,\Omega} I^p(x,y)} \right]^T, \tag{4.1}$$

where Ω defines the size of a tile, containing the pixels that are associated with the object to be localized. $I(x, y)$ are the discrete intensity values in Ω and p is a power factor to emphasize ($p > 1$) bright pixels.

Considering Equation 4.1, it is obvious that the result of the center of gravity calculation is strongly dependent on the choice of the location, size and content of the tile Ω. From there, some a-priori knowledge about the object is necessary to determine Ω and to achieve best results. In our application the imaged object is laser light, which is reflected from the surface of a steel bar. Consequently, the appearance of the laser line in the image is determined by the reflection of the laser light from rough surfaces (see Section 3.1), the constructive and destructive interference of coherent light [35] and the blooming effect of the CCD sensor [13, 49]. All these effects cause the imaged curve to be spread over several pixels with varying width in an direction orthogonal to the laser-plane. Furthermore, the intensity values along this spread are distributed in the neighborhood around the sought center of the curve. This distribution is exploited in the sub-pixel interpolation algorithm and the verification of pixels accepted in the calculation determine the accuracy of the resulting center of gravity location.

Assen *et al.* [50] analyzed the influence on the precision of the calculation in involving pixels that do not belong to the object. They proposed that a threshold τ_B should be used to separate, from the background, the pixels that are to be used in the calculation. A window of fixed size is centered around an approximately estimated object center. The intensity values of all pixels inside this window are investigated and those whose intensities are lower than the threshold τ_B are set to zero. The resulting contiguous area is Ω and is used to perform the calculation of the weighted center of gravity. On varying the location of the window and the value of the threshold τ_B, they concluded that a small threshold leads to a biased object center. On the other hand, a high threshold value, which causes only few pixels entering the calculation, yields in lower precision, i.e. the variance of the calculated center is high. Furthermore, Assen *et al.* [50] figured out that the application of a proper threshold implies that the variance becomes independent of small window misplacement with respect to the window size. Concisely, an optimal threshold value will effectively separate object from noise.

The result of the center of gravity calculation is illustrated in Figure 4.3b, taken from an enlarged detail in Figure 4.3a. It is shown that the calculated trend gives an accurate estimation of the true center of the laser line. The chosen parameters for the calculation are a maximum window size for Ω of 5×5 pixel, a threshold value of $\tau_B = 25$, which corresponds to approximately three times the background noise, a power factor of $p = 1$ and the coordinates of the coarsely determined points as initial points.



(a) Captured image

(b) Calculated center of gravity

Figure 4.3: Center of gravity calculation: The exact location of the intersection curve (a) is determined on performing a center of gravity calculation, whereby a threshold τ_B is used to separate pixels contributing to the result from the background. The calculated curve progression is visualized in (b).

4.3 Calculation of the cross section

In the prior step the pixel-coordinates of the intersection curve were determined. These measurement points are now mapped to the real-world via a collinear transformation (see Chapter 5), which is determined during the calibration of the system (see Chapter 6). That yields the undistorted metric-coordinates of the curve, from which the cross section of the steel bars is to be calculated. As is usual in measurements, the obtained points are perturbed and according to this, it is required to fit the known geometric shape to the scattered data. This is performed by using suitable fitting algorithms for the different bar profiles.

In this application two different types of cross section are to be determined:

- circular billets, characterized by a circular cross section and;
- square billets, having square cross sections with rounded edges of specific radius.

The sought parameters of the calculated geometry are: Position and radius for circular shapes; and position, orientation, edge radius, center point of the edge radius and side-length in case of square billets. In the following sections, several mathematical models are proposed to achieve an accurate computation of these parameters. Circular billets are modeled with an iterative fitting algorithm for circular arcs. Square ones are fitted using a combination of fitting algorithms orthogonal lines and a novel fitting algorithm for circles, defined by two intersecting tangents.

4.3.1 Fitting of circular cross sections

Two different methods for fitting circles to scattered data are presented below. The first one is an algebraic approach in the least squares sense. The advantage of this method is its simplicity and consequently, its low computational cost. However, the disadvantage of this method is that the geometric significance of the objective function is unclear [51]. Especially for the fitting of shallow circular arcs, where the noisy data points are not uniformly distributed around the circumference of the circle, the algebraic approach delivers unsatisfying results. The second algorithm is a geometrical approach, which minimizes the sum of the squares of the distances from the given points to the fitted circle. Such routines, minimizing the geometric distances deliver more exact results, are iterative and computational more complex than algebraic solutions [51].

In this application, the measurement of billet cross section is not part of the cyclic work (see Section 2.2, Figure 2.2) and therefore, accurate measurement is more important than fast computation. Thus, the algebraic method is used to determine the initial parameters of the circle as a starting point for the more exact, iterative geometric algorithm.

4.3.1.1 Fitting of circles, an algebraic approach

The equation of a circle in the plane is given by:

$$(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 = 0, \tag{4.2}$$

where $\mathbf{x}_i = [x_i, y_i]^T$ are the coordinates of a point lying on the circle with the center point $\mathbf{x}_0 = [x_0, y_0]^T$ and the radius r. On using Grassmannian tetra-circular coordinates the above equation becomes [52]:

$$c_1\left(x_i^2 + y_i^2\right) + c_2 x_i + c_3 y_i + c_4 = 0. \tag{4.3}$$

Taking into account that the measurement data is scattered, the algebraic distance e_i of a point $\mathbf{x}_i$ to the circle is:

$$c_1\left(x_i^2 + y_i^2\right) + c_2 x_i + c_3 y_i + c_4 = e_i. \tag{4.4}$$

Having n data points ($n \geq 4$) to describe the circle, Equation 4.4 can be rewritten as:

$$\begin{bmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^2 + y_n^2 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}, \tag{4.5}$$

or in abbreviated form:

$$\mathsf{D}\mathbf{v} = \mathbf{e}, \tag{4.6}$$

where D is the design matrix: The last column of D contains only ones, which are statistically invariant with respect to the data points and would yield a fitted hyper-plane $\mathbf{v}$ not on the centroid of the cloud of scattered data points. O'Leary *et al.* [53] showed that the calculation of the total least squares requires the column of ones to be removed, which ensures the center being part of the fitted hyper-plane. Furthermore, for the special case of fitting circles this removal is essential for the stability of the results. The goal is the partition of the matrix D into:

$$\mathsf{D} = [\mathsf{D}_1\ \mathsf{D}_0] = \left[\begin{array}{ccc|c} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^2 + y_n^2 & x_n & y_n & 1 \end{array}\right], \tag{4.7}$$

from which a new design matrix $\widehat{\mathsf{D}}$ is sought, which is the orthogonal residual of D_1 with respect to D_0, i.e. the portion of D_1 not predicted by D_0. This decomposition is called orthogonal residualization [53] and is given by:

$$\widehat{\mathsf{D}} = \mathsf{D}_1 - \mathsf{D}_0 \mathsf{D}_0^+ \mathsf{D}_1. \tag{4.8}$$

The pseudo-inverse D_0^+ is defined by [54]:

$$\mathsf{D}_0^+ \triangleq \left(\mathsf{D}_0^T \mathsf{D}_0\right)^{-1} \mathsf{D}_0^T. \tag{4.9}$$

For a vector containing n ones the pseudo-inverse becomes:

$$\mathsf{D}_0^+ = \frac{1}{n} \mathsf{D}_0^T, \tag{4.10}$$

and back-substituting into Equation 4.8 yields:

$$\widehat{\mathsf{D}} = \mathsf{D}_1 - \mathsf{D}_0 \frac{1}{n} \mathsf{D}_0^T \mathsf{D}_1, \tag{4.11}$$

which is equivalent to the removal of the mean value from each column of the matrix D_1 and the function becomes:

$$\widehat{\mathsf{D}}\mathbf{v}^* = \begin{bmatrix} x_1^2 + y_1^2 - \overline{(x^2+y^2)} & x_1 - \overline{x} & y_1 - \overline{y} \\ \vdots & \vdots & \vdots \\ x_n^2 + y_n^2 - \overline{(x^2+y^2)} & x_n - \overline{x} & y_n - \overline{y} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}. \tag{4.12}$$

Consequently, the cost function E, which is to be minimized is:

$$E = \sum_{i=1}^{n} e_i^2. \tag{4.13}$$

This minimization is performed by applying singular value decomposition (SVD) [55] to the new design matrix, which yields the required solution: $\widehat{\mathsf{D}} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$, where $\mathbf{v}^*$ is the right singular column vector of V that corresponds to the smallest singular value in S. The SVD ensures the constraint $\|\mathbf{v}^*\| = 1$ to obtain a non-trivial solution for the vector $\mathbf{v}^*$. Moreover, this constraint imposes that the residuals e_i are the normal distances of the data points to the fitted plane.

The remaining coefficient c_4 of the tetra-circular coordinates that describe the circle is determined via the following back-substitution:

$$c_4 = -\mathsf{D}_0^+ \mathsf{D}_1 \mathbf{v}^*; \tag{4.14}$$

$$c_4 = -\frac{1}{n} \mathsf{D}_0^T \mathsf{D}_1 \mathbf{v}^*; \tag{4.15}$$

$$c_4 = -\left[\overline{x^2+y^2}\ \overline{x}\ \overline{y}\right] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}. \tag{4.16}$$

The center point $\mathbf{x}_0 = [x_0, y_0]^T$ and the radius r of the circle are derived from substitution into Equation 4.2 and Equation 4.3:

$$x_0 = -\frac{c_2}{2c_1},\ y_0 = -\frac{c_3}{2c_1} \text{ and} \tag{4.17}$$

$$r = \sqrt{x_0^2 + y_0^2 - \frac{c_4}{c_1}}\ . \tag{4.18}$$

4.3.1.2 Fitting of circles, a geometric approach

Referring to the equation for a circle in the plane, given in Equation 4.2, the geometric or orthogonal distances of scattered data points $\mathbf{x}_i = [x_i, y_i]^T$ to the circle are given by:

$$h_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} - r, \tag{4.19}$$

where $\mathbf{x}_0 = [x_0, y_0]^T$ and r are the center point and the radius of the circle. The object of the geometrical approach is the minimization of the squared sum of these orthogonal distances, which leads to the cost function H:

$$H = \sum_{i=1}^{n} h_i^2. \tag{4.20}$$

This is a nonlinear least squares problem, for which Joseph [56] developed a solution. He proposed an iterative method, which delivers an accurate estimation of the circle parameters even if the input data points describe a circular arc. Assuming that the arc length of the circular arc is long compared with the amplitude of data point scatter, his method can be summarized as follows:

Given an initial circle fit to the data points, the center point and the radius of the circle are iteratively matched to minimize H. The dependence of H on these parameters is approximately linear for small modifications of the circle parameters. Consequently, the cost function is minimized and this procedure is iterated until an acceptable level of accuracy is reached.

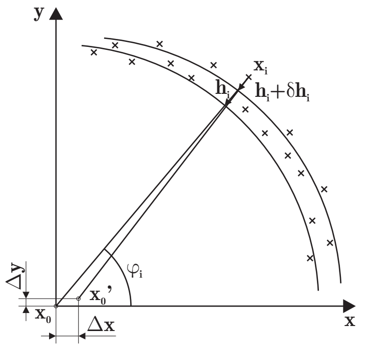


Figure 4.4: Principle of the geometrical approach used for the fitting of circular arcs. It is shown how the orthogonal distance h_i from a data point $\mathbf{x}_i$ to the circle is affected by a marginal alignment of the center point $\mathbf{x}_0$ and the radius r.

On adjusting $\mathbf{x}_0$ and r the orthogonal distance h_i from a data point $\mathbf{x}_i$ to the circle is influenced by δh_i. As shown in Figure 4.4 the two additive parts of δh_i are:

1. Moving the center point from $\mathbf{x}_0$ to $\mathbf{x}'_0$ by a small Δx and Δy yields:

$$\delta h_i|_{r=constant} \simeq -\Delta x\cos(\varphi_i) - \Delta y\sin(\varphi_i); \tag{4.21}$$

2. Changing the radius r by a small amount Δr effects:

$$\delta h_i|_{x_0,\, y_0=constant} \simeq -\Delta r. \tag{4.22}$$

From this it follows that the new cost function after the change becomes:

$$H = \sum_{i=1}^{n} \big(h_i + \delta h_i\big)^2 = \sum_{i=1}^{n} \big(h_i - \Delta x\cos(\varphi_i) - \Delta y\sin(\varphi_i) - \Delta r\big)^2. \tag{4.23}$$

Using the approximation that $\varphi_i = constant$ for small changes of circle parameters, the minimization problem is linear and can be solved by equating the partial derivatives of the cost function with zero. The partial derivatives of H are:

$$\begin{aligned} \frac{\partial H}{\partial \Delta x} &= 2\sum_{i=1}^{n}\big(h_i - \Delta x\cos(\varphi_i) - \Delta y\sin(\varphi_i) - \Delta r\big)\big(-\cos(\varphi_i)\big) = 0, \\ \frac{\partial H}{\partial \Delta y} &= 2\sum_{i=1}^{n}\big(h_i - \Delta x\cos(\varphi_i) - \Delta y\sin(\varphi_i) - \Delta r\big)\big(-\sin(\varphi_i)\big) = 0, \\ \frac{\partial H}{\partial \Delta r} &= 2\sum_{i=1}^{n}\big(h_i - \Delta x\cos(\varphi_i) - \Delta y\sin(\varphi_i) - \Delta r\big)\big(-1\big) = 0. \end{aligned} \tag{4.24}$$

These equations, simplified and rewritten as system of linear equations become:

$$\begin{bmatrix} \sum\cos^2(\varphi_i) & \sum\sin(\varphi_i)\cos(\varphi_i) & \sum\cos(\varphi_i) \\ \sum\sin(\varphi_i)\cos(\varphi_i) & \sum\sin^2(\varphi_i) & \sum\sin(\varphi_i) \\ \sum\cos(\varphi_i) & \sum\sin(\varphi_i) & n \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta r \end{bmatrix} = \begin{bmatrix} \sum h_i\cos(\varphi_i) \\ \sum h_i\sin(\varphi_i) \\ \sum h_i \end{bmatrix}, \tag{4.25}$$

where n denotes the number of data points. The above equation in abbreviated form leads to:

$$\mathsf{A}\mathbf{f} = \mathbf{h}, \tag{4.26}$$

where $\mathbf{h}$ describes the distribution of errors around the arc and A represents the required alignment of the circle parameters to improve the fit.

This system of equations is solved by calculating the inverse of the matrix A and consequently the sought vector $\mathbf{f}$ is:

$$\mathbf{f} = \mathsf{A}^{-1}\mathbf{h}. \tag{4.27}$$

The result is a better fit than the initial one, and the new circle parameters $\mathbf{x}_0' = [x_0', y_0']^T$ and r' are given by:

$$\begin{aligned} x_0' &= x_0 - \Delta x, \\ y_0' &= y_0 - \Delta y, \\ r' &= r - \Delta r. \end{aligned} \tag{4.28}$$

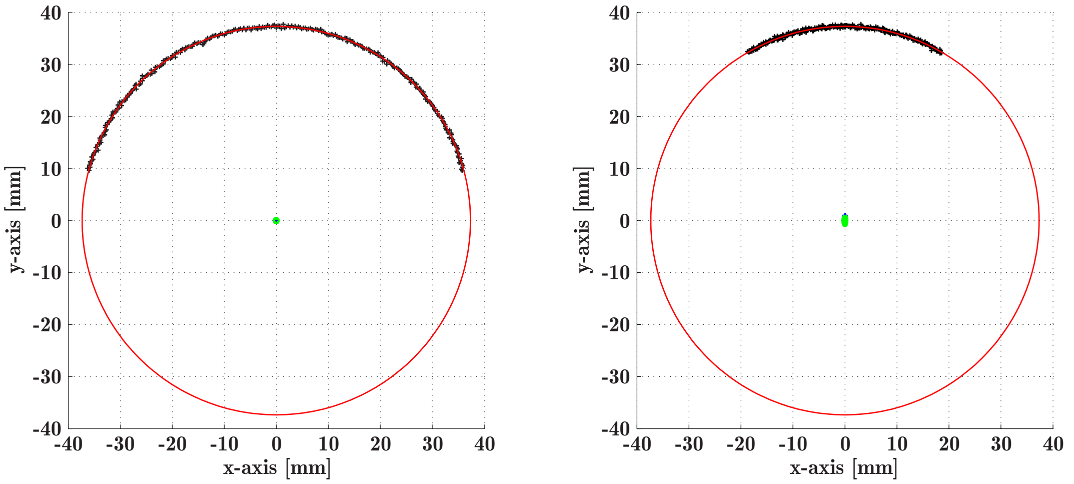
Using these improved circle parameters as initial parameters in a new cycle this process can be iterated until a termination criterion is fulfilled, e.g. an acceptable error for the fit is reached.

4.3.1.3 Comparison of the introduced fitting methods

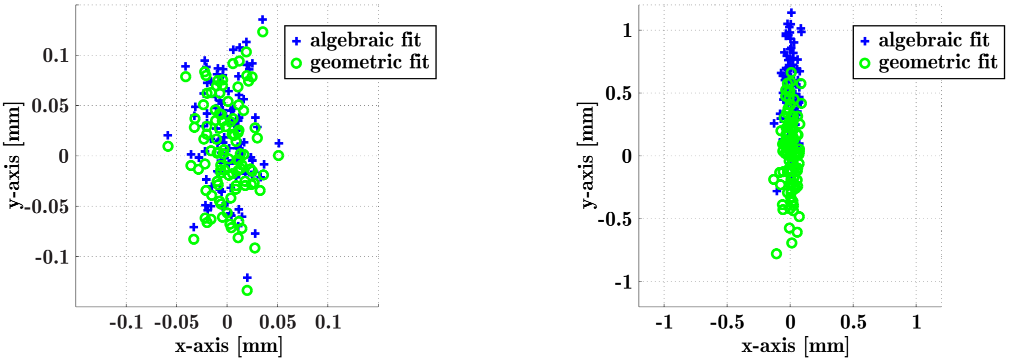
The uncertainties of the introduced algorithms were investigated by performing a Monte Carlo simulation [57]. Consequently, both methods were tested by calculating the circle parameters of 100 synthetic data sets that individually describe a circular arc having a definite arc length. Each data set is derived from a model-circle that consists of 300 scattered points that are spaced equiangularly and with a Gaussian radial distribution. The model is centered on the origin $\mathbf{x}_0 = [0, 0]^T$ and has a radius of $r = 37.5$mm. The standard deviation of the Gaussian noise is $\sigma = 0.2$mm and the comparison was performed with arcs of length $\alpha = 150°$ and $\alpha = 60°$[1]. The results of the simulation are shown in Figure 4.5.

---

[1] A standard deviation of $\sigma = 0.2$mm for the scattering of the data points and an arc length of $\alpha = 60°$ are chosen to be the worst case for real data sets in this application. Experiments in the production process have shown that a standard deviation of $\sigma = 0.2$mm is approximately three times the measured mean standard deviation and an arc length of $\alpha \approx 150°$ is characteristic for the used arrangement of laser and camera. Angles smaller than $\alpha < 100°$ are rare exceptions and occur only when parts of the billet surface are deformed, e.g. already ground.



(a) Circular arc length is 150°
(c) Circular arc length is 60°



(b) Uncertainty of the center points
(d) Uncertainty of the center points

Figure 4.5: Algebraic vs. geometric fitting method: Figure (a) and (c) show the model-circle and a gaussian distributed set of data points having different arc lengths. The images below illustrate the calculated center points of the fitted circles for both methods. Considering the distribution of the center points, one can see that the algebraic result is biased. This bias is increasing on fitting shallow arcs.

Figure 4.5a and 4.5c show the model-circle and one representative scattered set of data points for the chosen arc lengths. The images below, Figure 4.5b and 4.5d, illustrate the corresponding calculated center points of the algebraic as well as the geometric method. As one can see, both methods deliver stable and clear results. The geometric approach reached convergence after no more than two iterations and its time consumption was approximately twice as much that needed by the algebraic method. Considering the distribution of the center points, it is clear that the uncertainty for shallow arcs is larger. Moreover, the uncertainty in the x-direction is smaller than in the y-direction. This is due to the radial noise distribution of the data points, which are centered around the y-axis on the upper half of the circle. The advantage of the geometric approach becomes evident on examining the results of the shallower arc. While the mean y-value of the center points of the geometric method equals the center of the model, the y-values of the algebraic method are biased. This is always the case for algebraic solutions, if the distribution of the scattered data points is not uniform around the whole circumference of the circle, i.e. the data points describe a circular arc.

4.3.2 Fitting of rectangular cross sections

The second cross section to be determined is square with rounded edges, approximating arcs of circular cylinder. Since this geometric shape is as simple as a circular cross section, several fitting algorithms are used to obtain the sought parameters. The first method introduced is an algorithm for the fitting of a line to a set of scattered data points. Derived from this, another approach is presented to fit a pair of orthogonal lines to the measured data. Both methods use the singular value decomposition [55] to calculate the results in a least squares sense. Furthermore, in the special case of fitting lines, these methods minimize the orthogonal distances from the data points to the geometric element. The next fitting algorithm described here is an algebraic approach to determine the parameters of a circle that is defined by two intersecting tangents. The advantage of this method is its simplicity but the results are prone to systematic errors. Thus, an iterative geometric approach for circles with tangents is stated to obtain an unbiased solution. The disadvantage of an increased computing time by using the iterative method is of secondary importance for this application. Finally, an example is presented to demonstrate the practicability of all the introduced routines. It is shown, how the sought parameters are obtained from a real set of measured data points. Moreover, through the estimation of the scattering of the data points, synthetic data sets are created, re-fitted and the uncertainty of the measurement investigated.

4.3.2.1 Fitting of lines

A line in the plane is uniquely defined by two different points $\mathbf{x}_1 = [x_1, y_1, w_1]^T$ and $\mathbf{x}_2 = [x_2, y_2, w_2]^T$. Following that, an arbitrary point $\mathbf{x} = [x, y, w]^T$ in the same plane lies on this line, if it is a linear combination of the points $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e. that implies that their determinant is zero:

$$\begin{vmatrix} x & y & w \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{vmatrix} = 0. \tag{4.29}$$

Expanding this determinant on the minors of the first row yields:

$$x(y_1w_2 - y_2w_1) - y(x_1w_2 - x_2w_1) + w(x_1y_2 - x_2y_1) = 0, \tag{4.30}$$

and rewriting the above equation yields to the definition of the line in the plane:

$$[x \ y \ w] \begin{bmatrix} Y \\ -X \\ N \end{bmatrix} = \mathbf{x}^T\mathbf{l} = 0, \tag{4.31}$$

where the vector $\mathbf{l}$ contains the planar line coordinates [58] representing the line and $\mathbf{x}$ are the homogeneous coordinates of a point lying on this line. In case of fitting a line, a data point $\mathbf{x}_i = [x_i, y_i, 1]^T$ is usually scattered and do not lie exactly on the line and Equation 4.31 becomes:

$$[x_i \ y_i \ 1] \begin{bmatrix} Y \\ -X \\ N \end{bmatrix} = e_i. \tag{4.32}$$

Having n data points ($n \geq 3$) leads to the following system of linear equations:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} Y \\ -X \\ N \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}, \tag{4.33}$$

or in abbreviated form:

$$\mathsf{D}\mathbf{l} = \mathbf{e}, \tag{4.34}$$

with D as design matrix. The last column of D contains only ones and should be removed before calculating the total least squares (see Section 4.3.1.1). This decomposition leads to the new design matrix $\widehat{\mathsf{D}}$ and the system of linear equations is rewritten as:

$$\widehat{\mathsf{D}}\mathbf{l}^* = \begin{bmatrix} x_1 - \overline{x} & y_1 - \overline{y} \\ \vdots & \vdots \\ x_n - \overline{x} & y_n - \overline{y} \end{bmatrix} \begin{bmatrix} Y \\ -X \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}. \tag{4.35}$$

The sought vector $\mathbf{l}^*$ is the right singular vector of V, which is obtained by applying singular value decomposition [55] to the new design matrix $\widehat{\mathsf{D}} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$. The orthogonal residualization implies that the centroid of the data points $\mathbf{x}_0 = [\overline{x}, \overline{y}]^T$ is part of the fitted line and the SVD ensures the constraint $||\mathbf{l}^*|| = 1$ which yields the residuals e_i to be the orthogonal distances from the fitted line to the data points. The remaining coefficient N of the planar line coordinates is obtained via back-substitution:

$$N = -[\overline{x}\ \overline{y}] \begin{bmatrix} Y \\ -X \end{bmatrix}. \tag{4.36}$$

The resulting computed line is represented by the vector $\mathbf{l} = [Y, -X, N]^T$.

4.3.2.2 Fitting of orthogonal lines

A line in the plane is defined by the equation:

$$[x\ y\ w] \begin{bmatrix} Y \\ -X \\ N \end{bmatrix} = \mathbf{x}^T\mathbf{l} = 0, \tag{4.37}$$

where the vector $\mathbf{l}$ contains the planar line coordinates describing the line. The first two entries of this vector Y and $-X$ give the orientation, the third N affords the offset of the line to the origin. Considering the planar line coordinates of two orthogonal lines $\mathbf{l}_1$ and $\mathbf{l}_2$, it is obvious that their orientation is interdependent, i.e. rotated by 90° and one can write:

$$\mathbf{l}_1 : \ [x\ y\ w] \begin{bmatrix} Y \\ -X \\ N_1 \end{bmatrix} = 0 \quad \text{and} \quad \mathbf{l}_2 : \ [x\ y\ w] \begin{bmatrix} X \\ Y \\ N_2 \end{bmatrix} = 0. \tag{4.38}$$

Both equations rewritten in a more convenient way to compose the design matrix:

$$\mathbf{l}_1 : \ [x\ y\ w] \begin{bmatrix} Y \\ -X \\ N_1 \end{bmatrix} = 0 \quad \text{and} \quad \mathbf{l}_2 : \ [-y\ x\ w] \begin{bmatrix} Y \\ -X \\ N_2 \end{bmatrix} = 0. \tag{4.39}$$

Given a cluster of $n+m$ scattered data points describing the orthogonal lines $\mathbf{l}_1$ and $\mathbf{l}_2$, whereby the set n ($n \geq 3$) defines $\mathbf{l}_1$ and the set m ($m \geq 3$) defines $\mathbf{l}_2$, the design matrix for this fitting problem can be formulated as:

$$\mathsf{D}\mathbf{l}_{12} = \left[\begin{array}{cccc} x_{1,1} & y_{1,1} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{1,n} & y_{1,n} & 1 & 0 \\ \hline -y_{2,1} & x_{2,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -y_{2,m} & x_{2,m} & 0 & 1 \end{array}\right] \left[\begin{array}{c} Y \\ -X \\ N_1 \\ N_2 \end{array}\right] = \left[\begin{array}{c} e_{1,1} \\ \vdots \\ e_{1,n} \\ \hline e_{2,1} \\ \vdots \\ e_{2,m} \end{array}\right]. \tag{4.40}$$

Eliminating the last two columns via orthogonal residualization (see Section 4.3.1.1) forces the centroid of each data point set to be a part of the corresponding fitted line and yields the equation:

$$\widehat{\mathsf{D}}\mathbf{l}^*_{12} = \left[\begin{array}{cc} x_{1,1} - \overline{x}_1 & y_{1,1} - \overline{y}_1 \\ \vdots & \vdots \\ x_{1,n} - \overline{x}_1 & y_{1,n} - \overline{y}_1 \\ \hline -y_{2,1} - \overline{y}_2 & x_{2,1} - \overline{x}_2 \\ \vdots & \vdots \\ -y_{2,m} - \overline{y}_2 & x_{2,m} - \overline{x}_2 \end{array}\right] \left[\begin{array}{c} Y \\ -X \end{array}\right] = \left[\begin{array}{c} e_{1,1} \\ \vdots \\ e_{1,n} \\ \hline e_{2,1} \\ \vdots \\ e_{2,m} \end{array}\right]. \tag{4.41}$$

Again, the sought vector $\mathbf{l}^*_{12}$ is the right singular vector of V that is obtained by applying singular value decomposition [55] to the design matrix $\widehat{\mathsf{D}} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$. Furthermore, the SVD ensures the constraint $||\mathbf{l}^*_{12}|| = 1$ which yields the residuals e_i to be the orthogonal distances from the fitted construct to the data points. The remaining coefficients N_1 and N_2 are obtained via back-substitution:

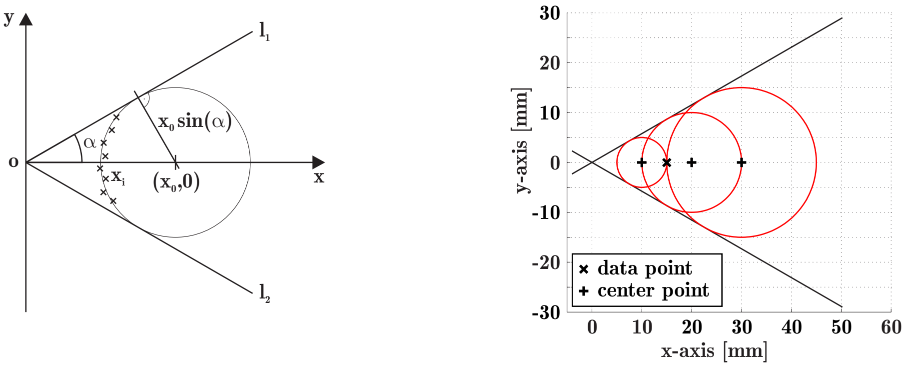
$$N_1 = -\left[\overline{x}_1\ \overline{y}_1\right] \left[\begin{array}{c} Y \\ -X \end{array}\right] \quad \text{and} \quad N_2 = -\left[\overline{x}_2\ \overline{y}_2\right] \left[\begin{array}{c} X \\ Y \end{array}\right]. \tag{4.42}$$

Consequently, the sought orthogonal lines can be represented by their planar line coordinates $\mathbf{l}_1 = [Y, -X, N_1]^T$ and $\mathbf{l}_2 = [X, Y, N_2]^T$.

4.3.2.3 Fitting of circles with tangents, an algebraic approach

Given two intersecting lines $\mathbf{l}_1$ and $\mathbf{l}_2$ and a set of data points $\mathbf{x}_i = [x_i, y_i]^T$ in the plane, the task of this fitting routine is to find a circle k with tangents $\mathbf{l}_1$ and $\mathbf{l}_2$ so that the sum of the squared algebraic distances of the points $\mathbf{x}_i$ to k is minimized.

Referring to Figure 4.6a, it is obvious that the centers of all possible circles with tangents $\mathbf{l}_1$ and $\mathbf{l}_2$ must lie on the bisector of these tangents. Considering now a Cartesian coordinate system where the origin $\mathbf{o} = [0, 0]^T$ is the point of intersection of these tangents and the x-axis is in the direction of the tangent's bisector, the center points of all valid circles $k(x_0)$ that fulfill the fitting requirements are given with $\mathbf{x}_0 = [x_0, 0]^T$. In so doing, it may be necessary to apply a suitable coordinate transformation (see Section 5.2) on the coordinates of the measured data points $\mathbf{x}_i$ and the line coordinates of $\mathbf{l}_1$ and $\mathbf{l}_2$. However,



(a) Used coordinate system **(b) Fitted circles with tangents**

Figure 4.6: Principle of the algebraic method used for the fitting of circles that are defined by two intersecting tangents. Figure (a) shows the coordinate system used: The origin is the point of intersection of the tangents and the x-axis is in the direction of the tangent's bisector. The right image (b) illustrates the three circles satisfying the cubic polynomial that must be found. The 'best' circle has the greatest radius.

with respect to the described coordinate system the equations of all valid circles become:

$$k(x_0): \ (x_i - x_0)^2 + y_i^2 - x_0^2 \sin^2(\alpha) = 0, \tag{4.43}$$

where $\mathbf{x}_i = [x_i, y_i]^T$ are the coordinates of a point lying on the circle and α is the angle between the tangents and their bisector. Considering that there are n scattered data points ($n \geq 1$) the algebraic distance is:

$$(x_i - x_0)^2 + y_i^2 - x_0^2 \sin^2(\alpha) = e_i. \tag{4.44}$$

The above equation leads to the cost function to be minimized:

$$E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \big((x_i - x_0)^2 + y_i^2 - x_0^2 \sin^2(\alpha)\big)^2. \tag{4.45}$$

This minimization problem can be solved by equating the first derivative of the cost function to zero. Calculating the first derivative leads to the cubic polynomial:

$$\frac{\partial E}{\partial x_0} = a_0\, x_0^3 + a_1\, x_0^2 + a_2\, x_0 + a_3 = 0, \tag{4.46}$$

with the coefficients:

$$a_0 = -n \cos^4(\alpha); \tag{4.47}$$

$$a_1 = 3 \cos^2(\alpha) \sum_{i=1}^{n} x_i; \tag{4.48}$$

$$a_2 = -\sum_{i=1}^{n} \big[2x_i^2 + \cos^2(\alpha)(x_i^2 + y_i^2)\big]; \tag{4.49}$$

$$a_3 = \sum_{i=1}^{n} x_i(x_i^2 + y_i^2). \tag{4.50}$$

Calculating the real zeros of this cubic polynomial yields the parameters $\mathbf{x}_0 = [x_0, y_0]^T$ and r of the fitted circles which are given by:

$$\begin{aligned} x_0 &= x_0, \\ y_0 &= 0, \\ r &= x_0 \sin(\alpha). \end{aligned} \tag{4.51}$$

The three solutions satisfying Equation 4.46 are illustrated in Figure 4.6b. The input parameters for the calculation were $\alpha = 30°$ and one data point with the coordinates $\mathbf{x}_1 = [15, 0]^T$. In this case, all three zeros are real and, as expected, one resulting circle is to the left of the given data point, one is to the right of it and the third solution is located between the other two circles.

In this application we are interested in the calculation of a square billet corner radius. Derived from the measurement setup, the obtained data points associated with the edge radius are distributed along the circumference of the circle between both tangent contact points. Consequently, the measured data points must be on the left side of the calculated circle and consequently the real zero with the largest radius is the solution being sought.

4.3.2.4 Fitting of circles with tangents, a geometric approach

Considering a circle in the plane $k(x_0)$ with tangents $\mathbf{l}_1$ and $\mathbf{l}_2$, embedded in a metric coordinate system as described in Section 4.3.2.3, the geometrical distances of scattered data points $\mathbf{x}_i = [x_i, y_i]^T$ to the circle with the center point $\mathbf{x}_0 = [x_0, 0]^T$ are defined by:

$$h_i = \sqrt{(x_0 - x_i)^2 + y_i^2} - x_0 \sin(\alpha), \tag{4.52}$$

where α is the half-angle between the tangents. The goal of this fitting approach is the minimization of the squared sum of these orthogonal distances and with n data points the cost function becomes:

$$H = \sum_{i=1}^{n} h_i^2. \tag{4.53}$$

This fitting problem is nonlinear and again an iterative approximation is used to solve it. Given an initial fit, e.g. the algebraic result, leads to the circle parameters $\mathbf{x}_0$ and r. Joseph [56] suggested that the dependence of H on these parameters is approximately linear for marginal alignments of the circle parameters. Following this, the cost function can be minimized and this procedure is iterated until a termination criterion is fulfilled, e.g the circle parameters reach convergence.

The only adjustable circle parameter in this case is x_0.[2] Because of this, we have to augment x_0 by Δx to minimize H. On moving the center point from $\mathbf{x}_0$ to $\mathbf{x}'_0$, the orthogonal distance h_i from a data point $\mathbf{x}_i$ to the circle is influenced by δh_i. With the approximation that $\varphi_i = \textit{constant}$ for small Δx the change δh_i is given by (see Figure 4.7):

$$\delta h_i \simeq \Delta x \cos(\varphi_i) - \Delta x \sin(\alpha). \tag{4.54}$$

---

[2] The radius r is dependent on x_0 and is given with $r = x_0 \sin(\alpha)$.

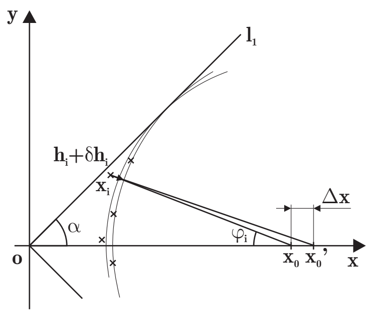


Figure 4.7: Principle of the geometrical approach used for the fitting of circles with two given tangents. On moving the center point from $\mathbf{x}_0$ to $\mathbf{x}'_0$ the orthogonal distance h_i from a point $\mathbf{x}_i$ to the circle is altered by δh_i.

Now the new cost function, after adding δh_i, becomes:

$$H = \sum_{i=1}^{n} \big(h_i + \delta h_i\big)^2 = \sum_{i=1}^{n} \Big(h_i + \Delta x\big(\cos(\varphi_i) - sin(\alpha)\big)\Big)^2. \tag{4.55}$$

The minimization is performed by equating the first derivative to zero:

$$\frac{\partial H}{\partial x_0} = 2\sum_{i=1}^{n} \Big(h_i + \Delta x\big(\cos(\varphi_i) - sin(\alpha)\big)\Big)\Big(\cos(\varphi_i) - sin(\alpha)\Big) = 0. \tag{4.56}$$

Finally, extracting Δx from the above equation leads to the sought solution:

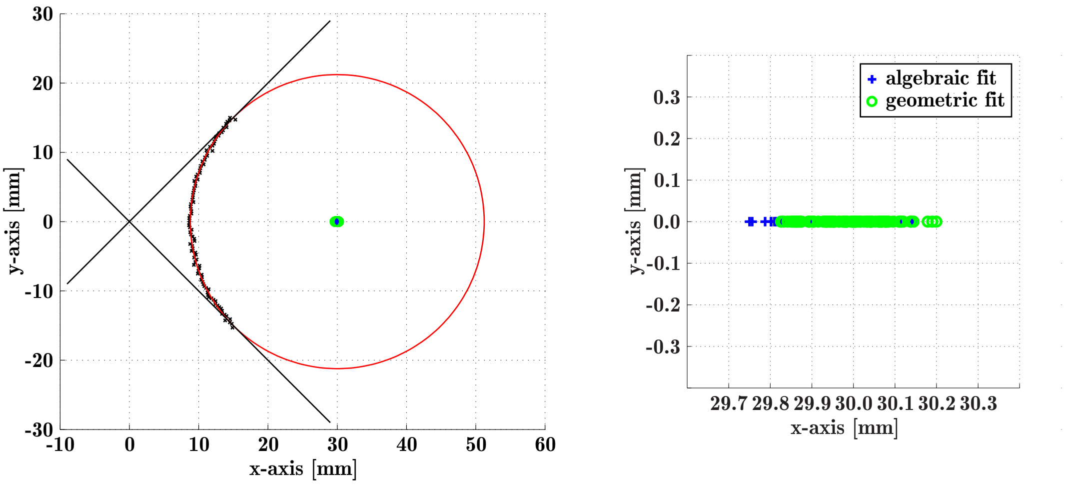
$$\Delta x = \frac{\sum\limits_{i=1}^{n} h_i \,\big(\cos(\varphi_i) - sin(\alpha)\big)}{\sum\limits_{i=1}^{n} \big(\cos(\varphi_i) - sin(\alpha)\big)^2}. \tag{4.57}$$

The parameters of the improved fit are now given by:

$$\begin{aligned} x'_0 &= x_0 - \Delta x, \\ y'_0 &= 0, \\ r' &= x'_0 \sin(\alpha). \end{aligned} \tag{4.58}$$

These parameters can be used as initial parameters for a new cycle and this process is iterated until convergence for the circle parameters is reached.

Again, a Monte Carlo simulation [57] was used to compare the introduced methods. The results of applying the algebraic and the geometric method to 100 synthetic sets of data points are given in Figure 4.8. Each data set is derived from a model-circle and consists of 100 scattered data points that have a radial Gaussian distribution and at equiangular displacement along the circle circumference, between the two points of tangently contact. The half-angle between tangents was chosen as $\alpha = 45°$ and the center point of the model was located at $\mathbf{x}_0 = [30, 0]^T$. The standard deviation of the Gaussian noise was set to $\sigma = 0.2$mm, which is approximately three times the mean standard deviation for actual measurements and considered to be the worst case for data sets in this application.



(a) Model-circle and scattered data points **(b) Uncertainty of the center points**

Figure 4.8: Comparison of the fitting routines for circles with tangents: Figure (a) shows the model-circle, the intersecting tangential lines and a scattered set of data points. The left image (b) pictures the calculated center points of 100 synthetic sets for both methods. It is shown that the algebraic results are biased.

The left image in Figure 4.8 shows the model-circle, the two intersecting tangent lines and one representative scattered data point set. The image to the right, shows an enlarged detail containing the distribution of the calculated center points of the fitted circles. Once more, stable and clear results were achieved with both methods and dependent on the desired accuracy of the convergence, the computational time consumption using the geometric algorithm is about two to three times higher than with the algebraic approach. On the other hand, the results derived from the algebraic method are somewhat biased. This suspicion arises upon comparing the distribution of the calculated center points, pictured in Figure 4.8b, and the calculation of the mean values of x_0 confirmed this systematic error.

4.3.2.5 Example for calculating the cross section of a square billet

The following example is derived from an arbitrarily sampled image, obtained from the billet grinding production line, that shows an actual, valid line on a square billet. To be valid, contour line must show an intersection curve that is constituted by two sides of the bar. The analysis of an image, containing a curve, where due to the location of the billet, only one flank is visible, cannot yield a cross section measurement. In that case, the material must travel along the conveyor until it rotates enough to present a valid image. However, Figure 4.9 shows a valid set of data points and the cross section of the billet can be determined. The sought parameters therefore are the billet's position, characterized by the point of intersection of the tangents to the lateral surfaces, its orientation, given through the tangent's bisector, and the parameters of the edge radius. The aim of this example is to show that the fitting algorithms achieve their objectives. Furthermore, the uncertainty of the achieved parameters is investigated to this end.

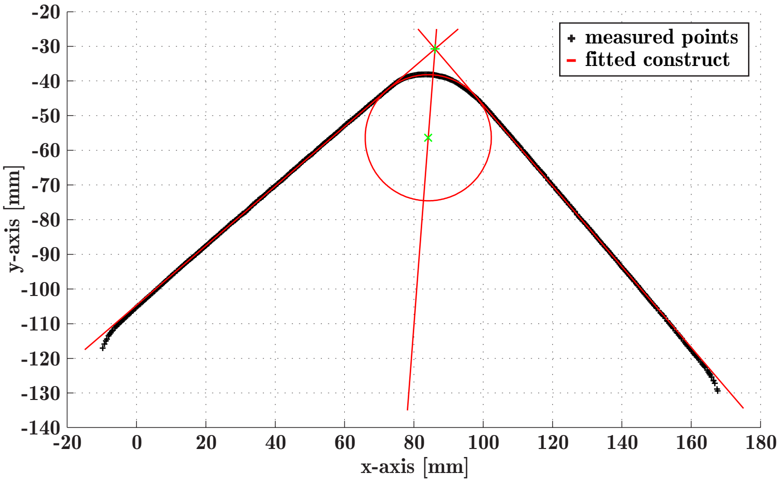
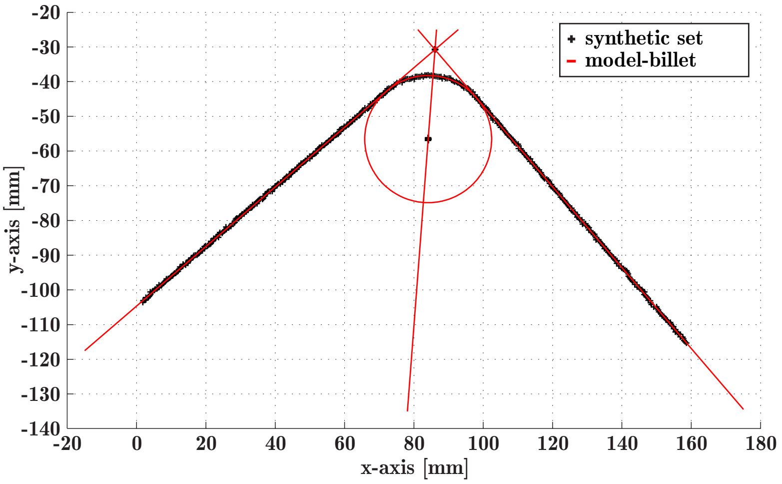


Figure 4.9: Fit rectangular cross sections: The algorithms for the fitting of orthogonal lines and circles that are defined through a pair of intersecting tangents are used to determine the cross section of square billets.

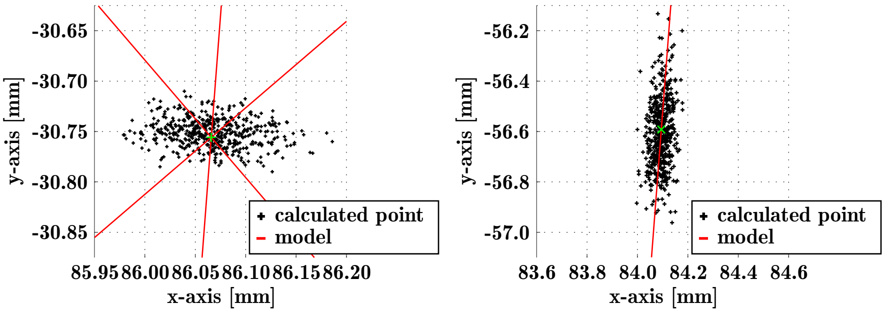
On considering a valid contour line on a square billet, it is possible to distinguish three different elementary geometric shapes. The measurement points describing the light section can be separated into groups corresponding to a tangent line to the left, a corner arc between and another tangent line to the right. The transition from one shape to the next is continuous and the tangents intersect is a right angle. This segmentation of the billet's cross section into three shapes is used for the following precis of cross section shape parameter calculation process:

Fitting starts with the assignment of the data points to the corresponding shape. For this purpose, the set of data points is segmented into two clusters by the point having the maximum y-coordinate. Furthermore, each cluster is quartered, the first and last quarter are rejected and a pair of orthogonal lines is fitted to the remaining points. The resulting lines are a good approximation of the billet's tangents and used as model to assign the data points. This is performed on verifying the orthogonal distance of each data point to the model. Points matching a threshold τ_T belong to the corresponding tangent, points located between the tangents are assigned to the edge rounded corner and all the rest are rejected. Applying the algorithm for the fitting of orthogonal lines to the points representing the tangents leads to the planar line coordinates that constitute the tangents of the billet. The lines' intersection point specifies the position of the bar and the bisector gives the orientation. The measurement is completed on using the remaining set of data points, that corresponds to the rounded corner arc, and the line coordinates to perform the fitting of a circle with tangents. That yields the center point and the radius of the corner. The result of this fitting procedure is shown in Figure 4.9.

It should be noted that there is no way to perform an accurate measurement of the distance between parallel sides of the billet with the existing measurement setup. An exact determination of the side-length would require the location of a second tangent contact point to ether of the partial corner arcs that appear in Figure 4.9 at lower left and lower right. Data points there are ether been rejected or are occluded, hence inavailable. To overcome this, the assignment of a multiple laser/camera system to cover more than the upper half of the billet is needed. Consequently, the only opportunity to estimate actual side of a square billet is to count the data points on one side. This leads to a very rough estimate with an accuracy of no better than ±5% of the true size. Nevertheless, since this dimension is not used to position the robot end effector but rather to define limits to its required workspace, this estimate is good enough for its flaw grinding activities.



(a) Model-billet with one scattered data set



(b) Point of intersection **(c) Center point**

Figure 4.10: Analysis of the uncertainty of the measurement. Figure (a) shows one of 500 synthetic data sets generated by a Monte Carlo simulation and the model-billet. The images below (b) and (c) illustrate uncertainties in the calculated point of intersection of the tangents and the center point of the edge radius respectively.

On performing a Monte Carlo simulation [57] the uncertainty of the measurement was analyzed. Due to this, the fitted construct from the above example was used as model to create a sequence of 500 scattered data sets. Derived from the measurement data, each synthetic set consists of 484 equidistant data points that have Gaussian distribution in the direction orthogonal to the reference line, used to establish equidistant spacing. The noise level was set to a standard deviation of $\sigma = 0.2$mm, which is approximately three times the mean value of the noise amplitude of the production process. Again, this is considered to be the worst case for actual measurements. The model and one scattered data set are shown in Figure 4.10a.

It should be noted that for the evaluation of the reference standard deviation for scattering, the two sides of the square billets were modeled with a quadratic curve instead of straight line. This is necessary since the sides of some billets are not exactly planar and thus, a second order function gives a better fit to the profile of a billet flank than an exactly straight line. In so doing, the obtained standard deviation of the geometric distances is less influenced by systematic deviations of the actual bar geometry from the target shape, and gives a better representation of actual measurement point scatter.

The results of the simulation are presented in Figure 4.10b and Figure 4.10c: The left figure shows the enlarged details of the uncertainties in the calculated point of intersection of the tangents; the right figure illustrates the distribution of the center points of the edge radius. As expected, good results were achieved for all data sets. Moreover, the results are not prone to systematic errors since the centers of the uncertainty-clouds conform to the original parameters of the model. The uncertainty of the point of intersection is smaller in comparison to the one, derived from the center point of the edge radius. This is due to the influence of the fitting of circles with tangents in the y-direction. The results of the measurement of the billet's cross section are summarized in Table 4.1.

Position	Measured value	Standard deviation
Point of intersection	$x_{PoI} = 86.07$mm	$\sigma = 0.036$mm
	$y_{PoI} = -30.75$mm	$\sigma = 0.014$mm
Orientation[3]	$\phi = -4.36°$	$\sigma = 0.021°$
Edge radius	$x_0 = 84.10$mm	$\sigma = 0.029$mm
	$y_0 = -56.60$mm	$\sigma = 0.127$mm
	$r = 18.33$mm	$\sigma = 0.093$mm
Side-length	$l = 130$mm	

Table 4.1: Results of the measurement of the billet's cross section.

---

[3] The orientation of a billet is given by the inclination ϕ of the tangents' bisector to a vertical line.

Chapter 5

Modeling of the 3D Geometry

In prior chapters of this work the detection of color markings, indicating a surface defect and the measurement of billet cross section were presented in Chapters 3 and 4, respectively. Color detection provides the coordinates of the vertices of a box surrounding the color marking; Cross section measurement delivers the parameters of the fitted geometry of the bar. The aim of this chapter is the combination of both results to establish a box that depicts the billet surface surrounding the color marking. Then the parameters characterizing the location and limits of the box are determined and supplied to the industrial 6R robot, so that it may perform the grinding process.

The color detection process obtains two-dimensional pixel-coordinates with respect to the camera chip origin. The subsequent interpretation of this result requires the mapping of these coordinates to the metric real-world. In Chapter 4, where cross section measurement is presented, the transformation of the pixel-coordinates of the intersection curve to the metric real-world is omitted. Therefore, this chapter begins with an introduction to collineations, a method to map coordinates from one plane to another. It is shown how planar collineations are determined and subsequently applied to camera pixel-coordinates to obtain real-world coordinates. This yields metric results from data, obtained in the color detection and of the cross section measurement process. Then, three-dimensional coordinate transformations are applied to remove the projective distortion from the measured cross sections and to register the results, obtained in individual frames, with the world coordinate system. Later, the distortion free cross sections are used to create a three-dimensional model of the billet surface, upon which the color marking's bounding box is projected to create a virtual image of the region to be ground and the 3D coordinates of the box corners in the world frame that unambiguously locate the region so as to be accessible to and compatible with the robotic grinder. These parameters are flaw descriptors, suitable for both: circular as well as square billets.

5.1 Collineations

A collinear transformation κ is the mapping of homogeneous coordinates from $\mathbb{P}^n \rightarrow \mathbb{P}'^n$ by a (n+1)×(n+1)-matrix A, that cannot be a zero matrix $\mathsf{0}$ [59]:

$$\mathbf{x}' = \mathsf{A}\mathbf{x}. \tag{5.1}$$

In Equation 5.1, $\mathbf{x}$ and $\mathbf{x}'$ are the homogeneous coordinates of corresponding points with respect to their coordinate systems S and S' respectively. If the mapping is uniquely invertible or bijective then every point $\mathbf{x}$ in $\mathbb{P}^n$ has an image $\mathbf{x}'$ in $\mathbb{P}'^n$ and vice versa, this transformation is called a homography, projective transformation or collineation [59]. This is the case if the determinant of the matrix A exists ($\det \mathsf{A} \neq 0$). The characteristic property of a collineation is that mapped sets retain their projective properties, e.g. a line maps to a line.

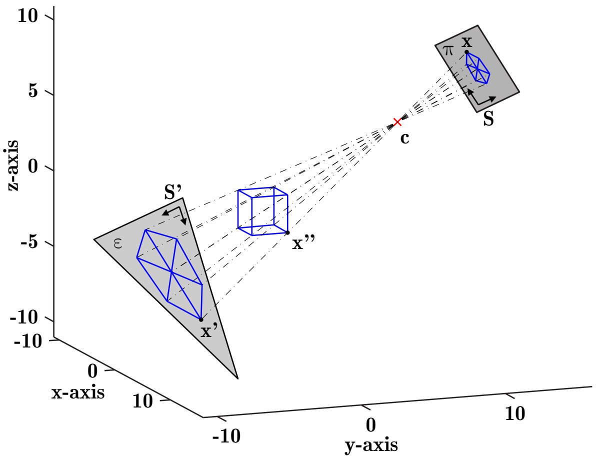


Figure 5.1: Camera projection: A camera projection can be used to map three-dimensional points $\mathbf{x}''$ through an eye or central point $\mathbf{c}$ to a plane π. If that all three-dimensional points $\mathbf{x}'$ are element of the plane ϵ and $\mathbf{c}$ is not on ϵ, the collineation from ϵ to π via $\mathbf{c}$ is uniquely defined by a 3×3 matrix A of full rank.

Following from the above definition, matrix A is 4×4 of full rank for three-dimensional collineations. A special case of such a three-dimensional collineation is a central-projection or camera-projection [59]. This is a projection in $\mathbb{P}^3$ from an eye $\mathbf{c}$ (center, focal-point) upon a plane π (camera-chip), i.e. every three-dimensional point $\mathbf{x}''$, except the eye $\mathbf{c}$, generates an unique but not necessarily distinct image point $\mathbf{x}$ on π through the eye $\mathbf{c}$ (see Figure 5.1). The size of the matrix A for a central-projection is also 4×4, but due to the fact that the coordinates of the eye $\mathbf{c}$ yield no image on π, it is of rank 3. Using such a central-projection subject to the condition that all three-dimensional points $\mathbf{x}'$ are elements of the plane ϵ ($\mathbf{x}' \in \epsilon$) and the eye $\mathbf{c}$ is not element of π nor ϵ ($\mathbf{c} \notin \epsilon, \pi$), produces an uniquely invertible planar collineation between π and ϵ via $\mathbf{c}$. This collineation can be described by a 3×3-matrix A of full rank and Equation 5.1 becomes:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}. \tag{5.2}$$

5.1.1 Isotropic scaling

The calculation of the homography yields the collineation matrix A that contains pixel-coordinates, metric-coordinates, their products and a constant. This implies that the matrix A is poorly conditioned in a numerically sense and may cause a dependence of the result on some specific property of the input data, e.g. the choice of the origin in the image. To avoid this, input data should be normalized to a canonical coordinate frame before calculating the homography. In addition, the input data becomes invariant to arbitrary scaling. Normalization, called isotropic scaling [60, 61], applied independently to all data points in both frames (metric- and pixel-coordinates) can be separated into the following two steps:

1. The data $\mathbf{x} = [x_i, y_i]^T$ is made mean-value free, i.e. the origin is taken as the centroid of all data points in a frame:

$$\tilde{x}_i = x_i - \overline{x} \quad \text{and} \quad \tilde{y}_i = y_i - \overline{y}; \tag{5.3}$$

2. The data is scaled in such a way that the average distance of points from the origin is equal to $\sqrt{2}$ and its coordinates are of the form $\mathbf{x} = [1, 1, 1]^T$. An isotropic scaling factor α is chosen so that the x- and y-axis are scaled equally. Therefore, the scaling factor α becomes:

$$\sqrt{2} = \frac{1}{n}\sum_{i=1}^{n} \sqrt{\left((\alpha\,\tilde{x}_i)^2 + (\alpha\,\tilde{y}_i)^2\right)}; \tag{5.4}$$

$$n\sqrt{2} = \alpha \sum_{i=1}^{n} \sqrt{\left(\tilde{x}_i^2 + \tilde{y}_i^2\right)}; \tag{5.5}$$

$$\alpha = \frac{n\sqrt{2}}{\sum\limits_{i=1}^{n} \sqrt{\left(\tilde{x}_i^2 + \tilde{y}_i^2\right)}}, \tag{5.6}$$

where n is the number of data points. The transformation matrix that performs the isotropic scaling therefore assums:

$$\mathsf{T}_s = \begin{bmatrix} \alpha & 0 & -\alpha\,\overline{x} \\ 0 & \alpha & -\alpha\,\overline{y} \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.7}$$

Suppose that the pixel-coordinates $\mathbf{x}$ are transferred as $\widehat{\mathbf{x}} = \mathsf{T}_s\,\mathbf{x}$ and the metric-coordinates $\mathbf{x}'$ are mapped by $\widehat{\mathbf{x}}' = \mathsf{T}'_s\,\mathbf{x}'$, the following homography is implied:

1. Isotropic scaling is according to the transformations:

$$\widehat{\mathbf{x}} = \mathsf{T}_s\,\mathbf{x} \quad \text{and} \quad \widehat{\mathbf{x}}' = \mathsf{T}'_s\,\mathbf{x}'; \tag{5.8}$$

2. The projection matrix $\widehat{\mathsf{A}}$ fulfills the equation:

$$\widehat{\mathbf{x}}' = \widehat{\mathsf{A}}\widehat{\mathbf{x}}; \tag{5.9}$$

3. If $\widehat{\mathsf{A}}$ is calculated via scaled coordinates, the back-transformation is:

$$\mathsf{A} = \mathsf{T}'^{-1}_s\,\widehat{\mathsf{A}}\,\mathsf{T}_s. \tag{5.10}$$

5.1.2 Determination of the projection matrix for planar collineations

Based on the definition of collineations, the mapping κ of homogeneous coordinates from $\mathbb{P}^2 \rightarrow \mathbb{P}'^2$ is uniquely determined, if the 3×3 projection matrix A is non-singular:

$$\mathsf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \det \mathsf{A} \neq 0. \tag{5.11}$$

The projection is then given by $\mathbf{x}' = \mathsf{A}\mathbf{x}$, where $\mathbf{x}$ and $\mathbf{x}'$ are the homogeneous coordinates of corresponding points in their respective planes. The matrix A can be normalized on any of its non-zero entries without invalidating the transformation. This leads to eight independent ratios among the entries in A that are also referred to as the eight degrees of freedom, inherent in planar collineations [60]. However, on knowing the individual coordinates of a rigid point in both planes, each point places two constraints on A and consequently, a minimum of four spatially, linearly independent points[1] are necessary to fully determine the system of linear equations. If exactly four points are used to calculate A, one obtains an exact or minimal solution for the homography. In general no homography will perfectly satisfy a set of more than four corresponding points, whose measurement is inevitably subject to some measurement error, however small. The task is to find the solution that best fits such an over-determined system of equations that are meant to determine A. The direct linear transformation (DLT) algorithm [59, 60] suggests a way to determine the eight independent ratios among the elements of A. It is accomplished using the singular value decomposition [55].

A planar collineation κ is described by the equation $\mathbf{x}'_i = \mathsf{A}\mathbf{x}_i$, where the vectors $\mathbf{x}_i = [x_i, y_i, 1]^T$ in $\mathbb{P}^2$ are mapped to $\mathbf{x}'_i = [x'_i, y'_i, w'_i]^T$ in $\mathbb{P}'^2$. This implies that the homogeneous vectors $\mathbf{x}'_i$ and $\mathsf{A}\mathbf{x}_i$ are linearly dependent, i.e. $\mathbf{x}'_i - \lambda\mathsf{A}\mathbf{x}_i = \mathbf{0}$, $\lambda \neq 0$, thus, if A is a 3×3-matrix the following cross product yields a zero vector. In addition, a linear system to obtain the entries of A can be formulated:

$$\mathbf{x}'_i \times \mathsf{A}\mathbf{x}_i = \mathbf{0}. \tag{5.12}$$

Let $\mathbf{a}_j^T$ be the $j = 1, 2, 3^{th}$-row of A then the cross product becomes:

$$\begin{bmatrix} y'_i \mathbf{a}_3^T \mathbf{x}_i & - & w'_i \mathbf{a}_2^T \mathbf{x}_i \\ w'_i \mathbf{a}_1^T \mathbf{x}_i & - & x'_i \mathbf{a}_3^T \mathbf{x}_i \\ x'_i \mathbf{a}_2^T \mathbf{x}_i & - & y'_i \mathbf{a}_1^T \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.13}$$

The above equation is rewritten as a system of equations that are linear in matrix elements $\mathbf{a}_{ij}$, contained in $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]^T$ that is separated in Equation 5.14 by rearranging Equation 5.13:

$$\begin{bmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \\ -y'_i \mathbf{x}_i^T & x'_i \mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix} = \mathbf{0}, \tag{5.14}$$

or in abbreviated form:

$$\mathsf{D}\mathbf{a} = \mathbf{0}. \tag{5.15}$$

---

[1] Non of the points is a linear combination of the others, i.e. no three points are collinear.

Only two of the three rows of D, called the design matrix, are linearly independent. If $w'_i \neq 0$, then the third row is a combination of the first and the second row after scaling with $-\frac{x'_i}{w'_i}$ and $-\frac{y'_i}{w'_i}$, respectively. Consequently, the last row may be removed. Expanding Equation 5.15 to accommodate a number of points n, $n \geq 4$, produces Equation 5.16:

$$\begin{bmatrix} 0 & 0 & 0 & -w'_1 x_1 & -w'_1 y_1 & -w'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 \\ w'_1 x_1 & w'_1 y_1 & w'_1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -w'_n x_n & -w'_n y_n & -w'_n & y'_n x_n & y'_n y_n & y'_n \\ w'_n x_n & w'_n y_n & w'_n & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \mathbf{0}. \tag{5.16}$$

The non-trivial solution of this equation is obtained by applying singular value decomposition [55] to the design matrix $\mathsf{D} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$. The desired vector $\mathbf{a}$, containing the nine elements of A, is the right singular vector of V that corresponds to the smallest singular value in S. The SVD implies the constraint $||\mathbf{a}|| = 1$ so as to obtain a non-trivial solution for the vector $\mathbf{a}$. Since the collineation is unique invertible, the inverse of the matrix A exists and the inversion of the transformation is given by:

$$\mathbf{x} = \mathsf{A}^{-1}\mathbf{x}'. \tag{5.17}$$

5.2 Coordinate transformation in 3D

An arbitrary point $\mathbf{x}$ is completely described in $\mathbb{R}^3$ by the relation [62]:

$$\mathbf{x} = x\mathbf{o}_x + y\mathbf{o}_y + z\mathbf{o}_z, \tag{5.18}$$

where x, y and z specify the point's position in a Cartesian frame and the vectors $\mathbf{o}_x$, $\mathbf{o}_y$ and $\mathbf{o}_z$ denote the orientation of that coordinate frame. Furthermore, let $\mathbf{o}_x$, $\mathbf{o}_y$ and $\mathbf{o}_z$ be the principal unit vectors of this system with an origin $\mathbf{o}$. Therefore, the position vector of point $\mathbf{x}$ is written as:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{5.19}$$

Consider a second Cartesian coordinate system on the origin $\mathbf{o}'$ and with frame orientation $\mathbf{o}'''_x$, $\mathbf{o}'''_y$ and $\mathbf{o}'''_z$ with respect to the first frame, and let $\mathbf{x}$ and $\mathbf{x}'$ be the coordinates of a fixed point in the respective frames. Then the relationship between the coordinates in the different frames is given by:

$$\mathbf{x} = \mathbf{o}' + x'\mathbf{o}'''_x + y'\mathbf{o}'''_y + z'\mathbf{o}'''_z, \tag{5.20}$$

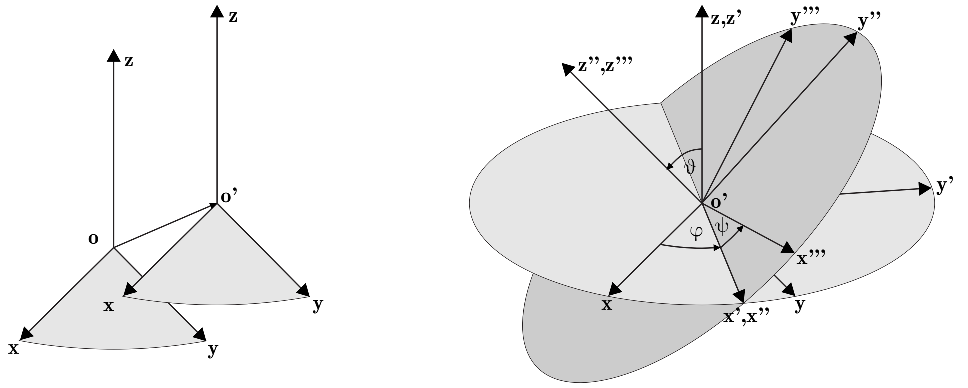
where $\mathbf{o}'$ is the position vector of the 2^{nd} frame's new origin with respect to $\mathbf{o}$. To achieve a compact notation the three orientation vectors can be inserted columnwise from left to right, to formulate the proper orthogonal 3 × 3 rotation matrix R and Equation 5.20 becomes:

$$\mathbf{x} = \mathbf{o}' + \mathsf{R}\mathbf{x}', \tag{5.21}$$

where

$$\mathsf{R} = \begin{bmatrix} \mathbf{o}_x''' & \mathbf{o}_y''' & \mathbf{o}_z''' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{5.22}$$

Consequently, Equation 5.21 represents the three-dimensional coordinate transformation between two coordinate systems, where the first part describes the translation of the origin and the second part the rotation of the coordinate frame. The translation and the rotation are illustrated in Figure 5.2a and 5.2b, respectively.



(a) Translation **(b) Rotation**

Figure 5.2: Coordinate transformation in 3D: The left figure (a) shows the shift of the origin from $\mathbf{o}$ to $\mathbf{o}'$. Following this, three successive rotations about the axes z, x' and z'', possibly using Euler angles φ, ϑ and ψ, complete the 3D coordinate transformation (b).

A possibly more convenient form of Equation 5.21 is obtained by using homogeneous coordinates to represent vectors $\mathbf{x}$ and $\mathbf{x}'$. As a result, the coordinate transformation in 3D can be written as a simple matrix-vector product:

$$\mathbf{x} = \mathsf{T}\mathbf{x}', \tag{5.23}$$

where T is a 4 × 4 homogeneous transformation matrix:

$$\mathsf{T} = \begin{bmatrix} \mathsf{R} & \mathbf{o}' \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \Delta x \\ r_{21} & r_{22} & r_{23} & \Delta y \\ r_{31} & r_{32} & r_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.24}$$

The inverse of this transformation is obtained by multiplying Equation 5.23 by T^{-1}, thus:

$$\mathbf{x}' = \mathsf{T}^{-1}\mathbf{x}. \tag{5.25}$$

Having specified the origins and orientations of both coordinate frames, the entries of the matrix T may be determined in the following manner:

Referring to Figure 5.2a, one can see that the translation vector is the position of the new origin $\mathbf{o}'$ with respect to the first coordinate system with the origin $\mathbf{o}$. Consequently, the values of Δx, Δy and Δz are the entries in $\mathbf{o}'$.

The nine elements in R constitute a highly redundant description of frame orientation [62, 63]: In fact, the rotation matrix R is orthonormal and due to this, the nine entries of R are related by six constraints to reduce its essential description to a triplet of three independent parameters that are sufficient to describe any rotation in space. Each parameter is represented by an angle that defines a rotation angle imposed about a principal axis in the coordinate frame. A sequence of three such rotations fully determines a general rotation matrix R, but one has to ensure the independence of the individual rotations imposed, i.e. the specification excludes successive rotations about parallel axes. Thus, only 12 out of 27 possible permutations of successive rotations may be enumerated and their independent angle triplets are called Euler angles.

The convention used here is given by the angles φ, ϑ and ψ and the corresponding axes about which the rotations are carried out are the z-, x'- and the z''-axis. The resulting frame orientation is the combination of these successive rotations and is shown in Figure 5.2b. If the individual rotations are given in terms of the rotation matrices R_φ, R_ϑ and R_ψ then the sought generic rotation matrix R is obtained by their post-multiplication[2]:

$$\mathsf{R} = \mathsf{R}_\varphi \mathsf{R}_\vartheta \mathsf{R}_\psi. \tag{5.26}$$

The individual rotation matrices are derived from the orientation of the coordinate frame after the rotation about the specified axis. Referring to Figure 5.2b, the orientation vectors of the frame, upon performing the first rotation by an angle φ about the z-axis with respect to the reference frame, are given by:

$$\mathbf{o}'_x = \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{bmatrix}, \mathbf{o}'_y = \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ 0 \end{bmatrix}, \mathbf{o}'_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{5.27}$$

where positive rotation angle is produced by counter-clockwise rotation of the frame. The orientations of the frame after executing the rotations about the x'- and the z''-axis can be determined in a similar manner. Finally, collecting the obtained orientation vectors in the individual rotation matrices yields:

$$\mathsf{R}_\varphi = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathsf{R}_\vartheta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta) & \sin(\vartheta) \\ 0 & -\sin(\vartheta) & \cos(\vartheta) \end{bmatrix}, \mathsf{R}_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.28}$$

---

[2] It should be noted that matrix multiplications are in general not commutative and thus, the composition depends on the order of the single rotations.

5.3 Calculation of the 3D geometry

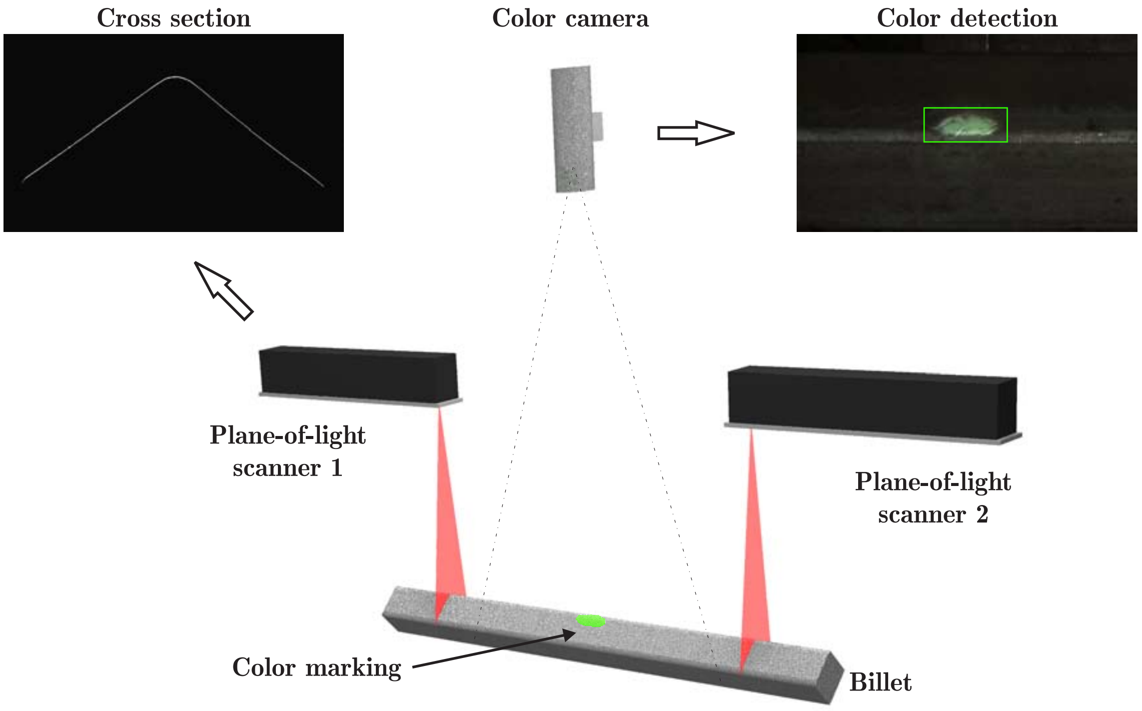


Figure 5.3: Calculation of the 3D geometry: Two plane-of-light scanners are used to measure a cross section of the bar to the left and the right of the robot's working range. From this, a three-dimensional model of the material surface is calculated and the two-dimensional coordinates of the color marking's bounding box are projected upon the model to define the region to be ground out.

The principle of and set-up for calculating 3D geometry are shown in Figure 5.3: Two plane-of-light scanners are used to acquire a cross section of the material to the left and the right of the robot's working range. The two profiles obtained in this manner are subsequently combined in a common coordinate system in which the three-dimensional model of a billet surface is calculated. Color detection is then used to obtain the coordinates of a rectangle, surrounding the color marking. Taking into account, that the metric-coordinates of this bounding box are registered on a true view of the surface defect, the projection of the color marking upon the three-dimensional model of the bar generates the required patch.

The following sections show the application of collineations to produce metric-coordinates, the use of 3D coordinate transformations to correct the distortion of the measured cross section and the registration of the individual results with a world coordinate system. Finally, the parameters that characterize the surface defects are summarized for both, circular and square billets.

5.3.1 Mapping of the pixel-coordinates to the real-world

5.3.1.1 Color detection

The color camera is placed at a suitable position above the material conveyor, observing the surface of the bars to detect color marked defects. By using this camera arrangement, the coordinates obtained from the color detection are the true view of the color marking's bounding box. Consequently, a planar collineation A_{Cam}, between the camera chip and a virtual plane, which is specified by the color camera's coordinate system, can be defined. Referring to the definition of planar collineations, given by Equation 5.2, the homogeneous mapping between pixel-coordinates and metric real-world coordinates of the color marking's bounding box is uniquely determined by:

$$\mathbf{x}_{Cam} = \mathsf{A}_{Cam}\mathbf{x}_p. \tag{5.29}$$

In this equation, $\mathbf{x}_{Cam} = [x_{Cam}, y_{Cam}, w]^T$ are the homogeneous metric-coordinates of a vertex of the bounding box, $\mathbf{x}_p = [x_p, y_p, 1]^T$ are the pixel-coordinates of the corresponding point in the acquired image and A_{Cam} is the 3×3-matrix that specifies the collineation. This collineation matrix and the color camera's coordinate system are determined during the calibration of the system (see Chapter 6).

Since the resulting vector $\mathbf{x}_{Cam}$ is homogeneous, the inhomogeneous or affine coordinates of the rectangle vertices are obtained by scaling the homogeneous point constitute vector with its last element:

$$x_{Cam,a} = \frac{x_{Cam}}{w}, \quad y_{Cam,a} = \frac{y_{Cam}}{w}. \tag{5.30}$$

5.3.1.2 Measurement of the cross section

The definition of a planar collineation is also the basis for the calculation of metric-coordinates from pixel-coordinates of a light plane intersection curve image. With reference to the principle of light sectioning, where an object is intersected by a plane-of-light and the arising intersection curve is acquired by a fixed but arbitrarily positioned camera, it is ensured that all measured contour points fall on the plane-of-light and workpiece intersection curve. On defining two coordinate frames, one on the laser plane and one in the acquired image, the relationship between the camera chip and the laser plane can be described via a planar collineation. This yields metric-coordinates, which may be distorted by relative positioning of the plane-of-light with respect to the workpiece. The undistorted profile of the object is obtained by performing an appropriate coordinate transformation to map the points from the laser-plane to a predefined three-dimensional coordinate system. The orientation of this three-dimensional coordinate system is such that its xz_{Scan}-plane is parallel to the cross sectional area of the object. Assuming that the cross section is uniform over the length of the bar, the measurement of the undistorted profile is guaranteed and enables one to extract a faithfully accurate cross section.

The sequence of a planar collineation A_{Scan} and a three-dimensional coordinate transformation T_{Scan} leads to a new collinear transformation, which can be described by a

4×3-matrix $\tilde{\mathsf{A}}_{Scan}$:

$$\tilde{\mathsf{A}}_{Scan} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \Delta x \\ r_{21} & r_{22} & r_{23} & \Delta y \\ r_{31} & r_{32} & r_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} \\ \tilde{a}_{21} & \tilde{a}_{22} & \tilde{a}_{23} \\ \tilde{a}_{31} & \tilde{a}_{32} & \tilde{a}_{33} \\ \tilde{a}_{41} & \tilde{a}_{42} & \tilde{a}_{43} \end{bmatrix}. \qquad (5.31)$$

The left matrix contains the entries $\Delta x, \Delta y, \Delta z$ and r_{ij}, which are the translation and rotation elements describing the coordinate transformation. The elements a_{ij} of the matrix to the right, characterize the planar collineation. The intermediate matrix adds a third coordinate ($z_{Laser} = 0$) to the points that are part of the laser-plane. Finally, substituting the matrix A for $\tilde{\mathsf{A}}_{Scan}$ in Equation 5.1 yields the transformation to map pixel-coordinates to the undistorted metric real-world:

$$\mathbf{x}_{Scan} = \tilde{\mathsf{A}}_{Scan}\mathbf{x}_p. \qquad (5.32)$$

The vector $\mathbf{x}_{Scan} = [x_{Scan}, y_{Scan}, z_{Scan}, w]^T$ contains the homogeneous metric-coordinates of a point of the intersection curve, $\mathbf{x}_p = [x_p, y_p, 1]^T$ are the pixel-coordinates of the corresponding point in the acquired image and $\tilde{\mathsf{A}}_{Scan}$ is the 4×3-matrix that specifies the collinear transformation. The matrix $\tilde{\mathsf{A}}_{Scan}$ and the scanner's three-dimensional coordinate system are determined during calibration of the system (see Chapter 6).

The inhomogeneous representation of the required metric-coordinates is achieved by dividing a homogeneous point coordinate vector by its last element:

$$x_{Scan,a} = \frac{x_{Scan}}{w}, \quad y_{Scan,a} = \frac{y_{Scan}}{w}, \quad z_{Scan,a} = \frac{z_{Scan}}{w}. \qquad (5.33)$$

Through the arrangement of the scanner's three dimensional coordinate-system, the undistorted coordinates, used for the fitting of the geometry, are represented by the $x_{Scan,a}$- and $z_{Scan,a}$-coordinate of the inhomogeneous vector $\mathbf{x}_{Scan,a}$. The measurement results from the second plane-of-light scanner are obtained in a similar manner.

5.3.2 Coordinate registration and calculation of the patch coordinates

The optical part of the servoing unit consists of three image processing systems: two plane-of-light scanners and one color camera. Each system possesses its individual coordinate system. To calculate patch 3D geometry, it is necessary to register these individual coordinate systems with a world coordinate system. The origin and the orientation of the axes of the individual coordinate frames and the world coordinate system are established by calibrating the system (see Chapter 6). In addition, the registration is uniquely determined and is accomplished via homogeneous coordinate transformations in 3D. Referring to Equation 5.23, the three individual transformations are given by:

$$\begin{aligned} \mathbf{x}_{Cam,w} &= \mathsf{T}_{Cam,w}\mathbf{x}_{Cam,m}, \\ \mathbf{x}_{Scan1,w} &= \mathsf{T}_{Scan1,w}\mathbf{x}_{Scan1,m}, \\ \mathbf{x}_{Scan2,w} &= \mathsf{T}_{Scan2,w}\mathbf{x}_{Scan2,m}. \end{aligned} \qquad (5.34)$$

The subscripts w and m denote that the associated homogeneous vectors are referred to the world coordinate system and the individual metric coordinate systems, respectively.

Moreover, the 4×4 matrices $\mathsf{T}_{Cam,w}$, $\mathsf{T}_{Scan1,w}$ and $\mathsf{T}_{Scan2,w}$ describe the particular transformations.

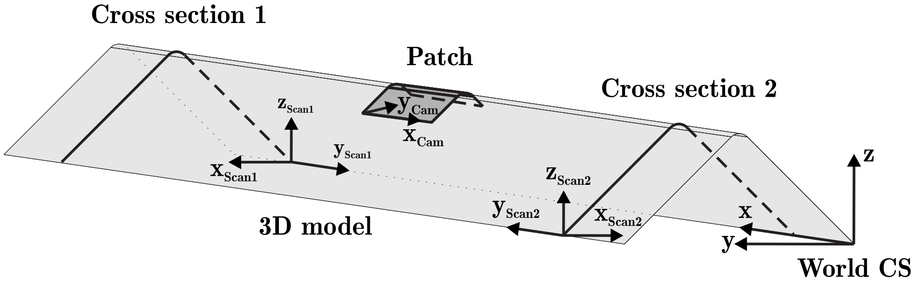


Figure 5.4: Coordinate registration and calculation of the patch coordinates: The measurement results from the plane-of-light scanners and the color camera are registered with a world coordinate system. From this, on extruding the measured profile along the world x-axis, a 3D model of the bar is generated upon which the color marking's bounding box is projected to calculate the coordinates of the required patch.

Once the measurement results are registered and assuming that billet cross sectual profile is uniform along its length, the acquired cross sections, obtained with the plane-of-light scanners, are used to create a three-dimensional model of the bar surface. As shown in Figure 5.4, the 3D model is generated on extruding the profile along the x-axis of the world coordinate system between both light planes. From this, the color marking's bounding box is projected upon the virtual surface of the model to obtain the required defect patch bounding box. The position of this patch is described by the three-dimensional coordinates of its vertices: The x- and y-coordinates are obtained from the color detection, the z-coordinate results from calculating the height of the cross section at a given y-value.

Finally, surface machining by the grinding robot can begin when it is supplied with cross section parameters and the coordinates of the patch thus, determining the position and limits of the defect. Dependent on shape of bar cross section necessary parameters are summarized in Table 5.1.

Circular billet	Square billet
Coordinates of the patch	Coordinates of the patch
Center point	Point of intersection
Radius	Orientation
	Center point of the edge radius
	Edge radius
	Side length

Table 5.1: Parameters required for the guidance of the grinding robot.

Chapter 6

Calibration of the System

The role of the calibration system is to survey in, with sufficient accuracy, the local reference frames of the color camera and the two plane-of-light scanners and to register these with the frame of the grinding robot. This survey is necessary to establish the fixed parameters that relate these four frames. Thus, variable measurements that define the bounding box surrounding color marked flaws, may be readily and reliably transformed to robot joints coordinate sequences. This grinding robot end effector path planning operation must be carried out in real-time and achieve an accuracy of a few tenths of a percent of some characteristic length associated with the robot workspace, i.e. of the same order of magnitude as the robot's specified repeatability.

Numerous accounts, of visual system calibration to extract metric information from acquired images, are to be found in the literature. Citations [64, 65, 66, 67, 68, 69] refer to a few of these pertinent to modern robot vision systems. However, details are omitted here because the automated billet flaw grinding artificial vision system calibration is much more complicated than any and requires considerable innovation to implement. The calibration concept presented resembles the one that is described in [70, 71, 72], but contrary to this, a further development in the determination of homographies by using direct linear transformation (DLT) is presented herein. This algorithm requires the coordinates of at least four rigid calibration points, in tetrahedral vertex configuration, with respect to their projective planes to determine the homography. In metric vision, the real-world coordinates are commonly known and the pixel-coordinates are refined to desired shape parameters by means of appropriate fitting algorithms. Because of perspective distortions, the acquired pixel coordinates normally do not exactly match the true position of the calibration points. The resulting homography is prone to systematic errors. An original way to overcome these errors is introduced. This involves the insertion of a non-linear homography approximation subsequent to the initial DLT algorithmic step. Nevertheless, it should be noted that the presented concept does not correct lens distortions and thus, these were avoided by using lenses of high quality.

This chapter is structured as follows: Essential system elements and procedures are outlined. Then the underlying mathematical methodology is introduced. These methods are Gauss-Newton approximation and fitting routines for ellipses and planes. Furthermore, the application of these techniques is subsequently explained in the context of calibrating the individual parts of the grinding facility. This yields collineation or transformation matrices and distinct coordinate frames of the separate components. Finally, this chapter concludes with an explanation of how measurement information in these individual frames is registered with a common coordinate system.

6.1 Calibration concept

Calibration was carried out on and with the equipment, shown in Figure 6.1 and determines the parameters that relate individual frames. These parameters are necessary to combine measurement data, obtained via the individual frames, into the common world frame to allow grinding path planning calculations to be carried out in real-time. Once the individual frames have been adjusted in registration with the robot frame and aligned in the direction of work piece advance along the conveyor, they remain fixed.

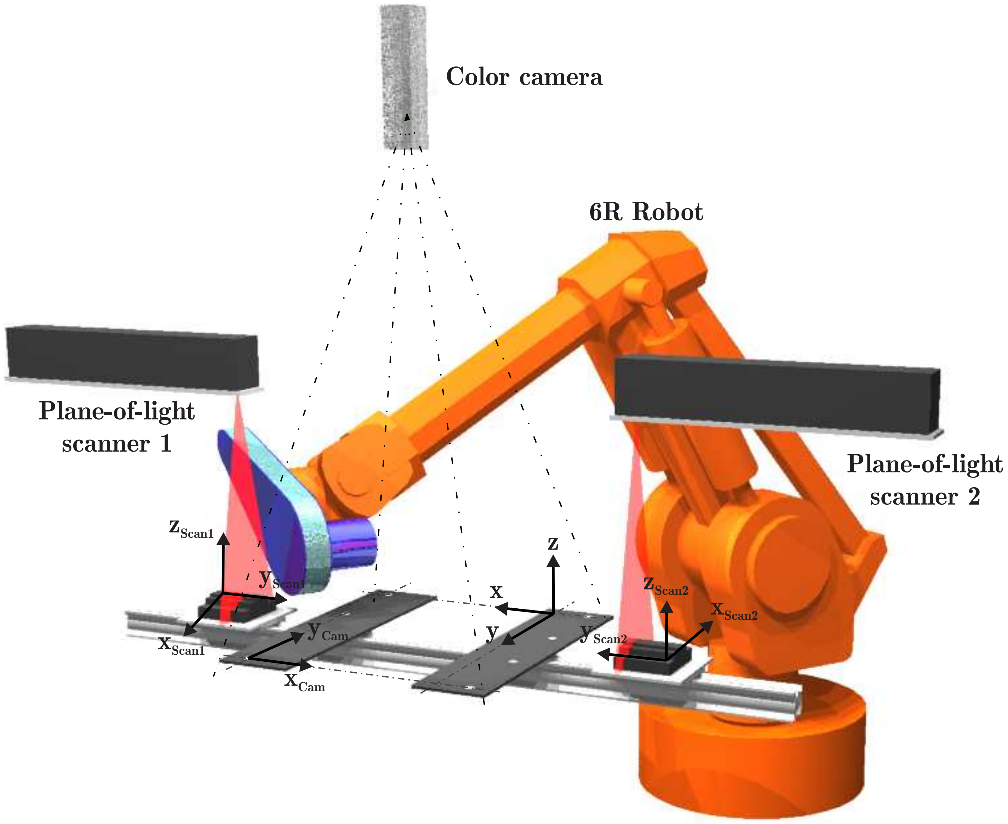


Figure 6.1: Calibration of the system: Overall view of calibrating the automatic grinding facility. The individual coordinate frames of the four main components are determined individually and subsequently registered so as to have one common coordinate system. This Cartesian frame is called the world coordinate system of the facility.

This calibration concept enjoys the advantages of a very simple hardware setup without moving parts. Since the algorithms used to perform the DLT are based on projective geometry, the calculation of the homographies between the image planes and the real-world are accomplished without reference to camera geometry, e.g. focal length. Instead, the collineation matrices are derived via the accessible and accurately laid out geometry of the calibration jig (see Appendix A).

The calibration equipment consists of:

- two planar calibration panels, with simple registration spots, used to calibrate the color camera and to determine the origin and orientation of the world coordinate system;
- two step targets, also with registration spots to calibrate the plane-of-light scanners;
- a precision rail on which the calibration panels and targets are mounted;
- two robust, height adjustable brackets that rigidly support the calibration equipment (not shown in Figure 6.1).

6.2 Essential mathematical tools for the calibration

6.2.1 Gauss-Newton approximation

Least squares fitting of functions, in general, leads to nonlinear minimization problems, e.g. the computation of geometric distances is a nonlinear task. Gauss-Newton approximation, a numerical procedure to iteratively solve nonlinear least squares problems is presented below:

A nonlinear function $f_i\left(\mathbf{x}_i, \mathbf{u}\right) = 0$, where $\mathbf{x}_i$ are the coordinates of data points, approximately constrains unknown coefficients $\mathbf{u} = [u_1, \ldots, u_m]^T$. A minimum number of n, $n \geq m$, data points is required to solve for m parameters that optimally fit the function to the data points. Taking into account that, measurement data is somewhat scattered, the function $f_i\left(\mathbf{x}_i, \mathbf{u}\right) = \mathbf{e} \neq \mathbf{0}$. There exists a residual e_i for every point $\mathbf{x}_i$ and the system of n nonlinear equations becomes:

$$\mathbf{f}\left(\mathbf{x}_i, \mathbf{u}\right) = \left[f_1(\mathbf{x}_1, \mathbf{u}), \ldots, f_n(\mathbf{x}_n, \mathbf{u})\right]^T = \mathbf{e}. \tag{6.1}$$

Therefore, the aim is to minimize the sum of the squares of the residuals, under the condition that the number of data points n is greater than the quantity of unknowns m: $n > m$. Accordingly, the cost or objective function for n data points is:

$$E = \sum_{i=1}^{n} \mathbf{e}^2, \text{ a minimum.} \tag{6.2}$$

Explicitly:

$$E = \mathbf{f}^T\left(\mathbf{x}_i, \mathbf{u}\right) \mathbf{f}\left(\mathbf{x}_i, \mathbf{u}\right). \tag{6.3}$$

This problem can be solved by equating the first derivative of the cost function to zero, which yields:

$$\frac{\partial E}{\partial \mathbf{u}} = \frac{2\,\partial \mathbf{f}^T\left(\mathbf{x}_i,\mathbf{u}\right)}{\partial \mathbf{u}}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}\right) = \mathbf{0}. \tag{6.4}$$

From this point of view, the goal is the evaluation of multiple roots that satisfy Equation 6.4. Making the following substitution:

$$\mathsf{G}\left(\mathbf{x}_i,\mathbf{u}\right) = \frac{\partial \mathbf{f}^T\left(\mathbf{x}_i,\mathbf{u}\right)}{\partial \mathbf{u}}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}\right), \tag{6.5}$$

and expanding the matrix $\mathsf{G}\left(\mathbf{x}_i,\mathbf{u}\right)$ in a Taylor series [73, 74] about the sought vector $\mathbf{u} = \mathbf{u}_0 + \Delta\mathbf{u}$ leads to:

$$\mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0+\Delta\mathbf{u}\right) = \mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \frac{\partial \mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\Delta\mathbf{u} + \frac{\partial^2 \mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{2\,\partial \mathbf{u}_0^2}\,\Delta\mathbf{u}^2 + \ldots, \tag{6.6}$$

where $\mathbf{u}_0$ is the vector containing the parameters of a starting point solution and $\Delta\mathbf{u}$ is an increment to more closely approach an optimum. Then, according to Newton-Raphson approximation [75], only terms of first order are retained:

$$\mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0+\Delta\mathbf{u}\right) \approx \mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \frac{\partial \mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\Delta\mathbf{u}. \tag{6.7}$$

Setting this equation to zero and evaluating $\mathsf{G}\left(\mathbf{x}_i,\mathbf{u}_0\right)$ according to Equation 6.5 yields:

$$\tfrac{\partial \mathbf{f}^T(\mathbf{x}_i,\mathbf{u}_0)}{\partial \mathbf{u}_0}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \left(\tfrac{\partial^2 \mathbf{f}^T(\mathbf{x}_i,\mathbf{u}_0)}{\partial \mathbf{u}_0^2}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \tfrac{\partial \mathbf{f}^T(\mathbf{x}_i,\mathbf{u}_0)}{\partial \mathbf{u}_0}\,\tfrac{\partial \mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}{\partial \mathbf{u}_0}\right)\Delta\mathbf{u} = \mathbf{0}. \tag{6.8}$$

Due to the Gauss-Newton approximation [76], the second derivative of $\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right)$ is neglected and Equation 6.8 becomes:

$$\frac{\partial \mathbf{f}^T\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \frac{\partial \mathbf{f}^T\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\frac{\partial \mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\Delta\mathbf{u} = \mathbf{0}. \tag{6.9}$$

After simplification, this equation is approximated by Equation 6.10,

$$\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) + \frac{\partial \mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0}\,\Delta\mathbf{u} = \mathbf{0}. \tag{6.10}$$

Taking into account that matrix, Equation 6.11:

$$\mathsf{J}_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)} = \frac{\partial \mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right)}{\partial \mathbf{u}_0} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_1,\mathbf{u}_0)}{\partial u_{0,1}} & \cdots & \frac{\partial f_1(\mathbf{x}_1,\mathbf{u}_0)}{\partial u_{0,m}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n(\mathbf{x}_n,\mathbf{u}_0)}{\partial u_{0,1}} & \cdots & \frac{\partial f_n(\mathbf{x}_n,\mathbf{u}_0)}{\partial u_{0,m}} \end{bmatrix} \tag{6.11}$$

is the Jacobian [73, 74] of the function $\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)$, Equation 6.10 can be rewritten as:

$$\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) = -\mathsf{J}_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\Delta\mathbf{u}. \tag{6.12}$$

Finally, increment $\Delta\mathbf{u}$ is given by:

$$\mathsf{J}^T_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right) = -\left(\mathsf{J}^T_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\mathsf{J}_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\right)\,\Delta\mathbf{u}, \tag{6.13}$$

$$\Delta\mathbf{u} = -\left(\mathsf{J}^T_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\mathsf{J}_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\right)^{-1}\mathsf{J}^T_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right), \tag{6.14}$$

$$\Delta\mathbf{u} = -\mathsf{J}^+_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}\,\mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_0\right), \tag{6.15}$$

where $\mathsf{J}^{+}_{\mathbf{f}(\mathbf{x}_i,\mathbf{u}_0)}$ is the pseudo-inverse [54] of the Jacobian. An improved solution is attained by adding the increment to the initial parameters:

$$\mathbf{u} = \mathbf{u}_0 + \Delta\mathbf{u}. \tag{6.16}$$

The revised parameters are then used as initial parameters for a new cycle: beginning again with Equation 6.6. These iterations continue until convergence, specified by some acceptable small value of $\Delta\mathbf{u}$, is reached.

6.2.2 Fitting of ellipses

A general conic[1] is represented by Equation [59]:

$$[x\ y\ w]\begin{bmatrix} c_1 & \frac{c_2}{2} & \frac{c_4}{2} \\ \frac{c_2}{2} & c_3 & \frac{c_5}{2} \\ \frac{c_4}{2} & \frac{c_5}{2} & c_6 \end{bmatrix}\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{x}^T\mathsf{C}\mathbf{x} = 0, \tag{6.17}$$

Homogeneous coordinates of a point on the conic are given by vector $\mathbf{x}$ and C is the symmetric conic coefficient matrix. Euclidian invariants, that determine the conic type, are given by:

$$D = \det\mathsf{C}, \tag{6.18}$$

$$\Delta = \begin{vmatrix} c_1 & \frac{c_2}{2} \\ \frac{c_2}{2} & c_3 \end{vmatrix} = c_1c_3 - \frac{c_2^2}{4}, \tag{6.19}$$

$$H = c_1 + c_3, \tag{6.20}$$

and finally, if $D = \Delta = 0$, the fourth invariant becomes:

$$K = c_6H - \Big(\frac{c_4^2}{4} + \frac{c_5^2}{4}\Big). \tag{6.21}$$

If $D \neq 0$ and $\Delta > 0$, the coefficients of the matrix C describe an ellipse, if $D \neq 0$ and $\Delta < 0$ the shape of the conic is hyperbolic and finally, if $D \neq 0$ and $\Delta = 0$, it is a parabola. A more detailed introduction to the Euclidian classification of conics is given in [59, 73, 74].

This work focuses on elliptical solutions and thus, the elliptical parameters: the homogeneous coordinates of the center point $\mathbf{x}_0$, the direction $\mathbf{e}_1$, $\mathbf{e}_2$ and the length a_1, a_2 of the major and the minor principal axes are of interest and given by:

$$\mathbf{x}_0 = \mathbf{c}_1 \times \mathbf{c}_2, \tag{6.22}$$

$\mathbf{c}_1$ and $\mathbf{c}_2$ are the first and the second column vector in C. In addition, the eigenvectors [74] $\mathbf{e}_1$ and $\mathbf{e}_2$ of the sub-matrix:

$$\mathsf{C}_1 = \begin{bmatrix} c_1 & \frac{c_2}{2} \\ \frac{c_2}{2} & c_3 \end{bmatrix}, \tag{6.23}$$

---

[1] Actually a generalized conic is to be fit. However, since the data always presents a closed curve the fitting of an ellipse is guaranteed. It is not necessary to impose the quadratic constraint $c_2^2 - 4c_1c_3 = 0$.

determine the direction of the major and the minor axes of the ellipse, and finally half the length of these axes is given by:

$$a_1 = \sqrt{\left|\frac{D}{\Delta\lambda_1}\right|}, \quad a_2 = \sqrt{\left|\frac{D}{\Delta\lambda_2}\right|}, \tag{6.24}$$

where, λ_1 and λ_2 are the eigenvalues [74] of C_1.

Expanding Equation 6.17 yields the scalar point equation of the conic:

$$c_1x^2 + c_2xy + c_3y^2 + c_4xw + c_5yw + c_6w^2 = 0. \tag{6.25}$$

Every point $\mathbf{x}_i = [x_i, y_i, 1]$ imposes a constraint on the coefficients of the conic. Because multiplication of Equation 6.25 by a non-zero constant is of no consequence, five given points fully define a conic. Consequently, with $n \geq 6$ data points one obtains the following system of linear equations:

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}, \tag{6.26}$$

this is abbreviated as:

$$\mathsf{D}\mathbf{c} = \mathbf{e}, \tag{6.27}$$

where D is the design matrix. For purposes of numerical stability the last column of D, that contains only ones, should be removed via orthogonal residualization (see Section 4.3.1.1):

$$\widehat{\mathsf{D}}\mathbf{c}^* = \begin{bmatrix} x_1^2 - \overline{x^2} & x_1y_1 - \overline{xy} & y_1^2 - \overline{y^2} & x_1 - \overline{x} & y_1 - \overline{y} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 - \overline{x^2} & x_ny_n - \overline{xy} & y_n^2 - \overline{y^2} & x_n - \overline{x} & y_n - \overline{y} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}, \tag{6.28}$$

This minimization problem is solved by applying singular value decomposition [55] to the design matrix $\widehat{\mathsf{D}} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$. As a result, the sought vector $\mathbf{c}^*$ is the rightmost column vector in matrix V. Moreover, the SVD imposes the constraint $\|\mathbf{c}^*\| = 1$, which yields the residuals e_i as normal distances from data points to the hyper-plane described by the conic coefficients. Finally, the remaining coefficient c_6 is obtained via back-substitution:

$$c_6 = -\left[\overline{x^2}\ \overline{xy}\ \overline{y^2}\ \overline{x}\ \overline{y}\right] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}. \tag{6.29}$$

6.2.3 Fitting of planes

A plane is fully determined by the equation:

$$[x\ y\ z\ w]\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{bmatrix} = \mathbf{x}^T\boldsymbol{\pi} = 0, \tag{6.30}$$

where the vector $\boldsymbol{\pi}$ contains the coordinates describing the plane and elements of $\mathbf{x}$ are the homogeneous coordinates of any point on this plane. The first three entries of $\boldsymbol{\pi}$: $\mathbf{n} = [\pi_1, \pi_2, \pi_3]^T$ are the direction numbers of a vector normal to the plane, and if $\|\mathbf{n}\| = 1$, $\mathbf{n}$ is called the unit normal-vector of the plane. Moreover, the zero vector $\mathbf{n} = [0, 0, 0]^T$ is omitted, according to the definition of planes [73, 74], because it cannot belong to any plane in the Euclidian space. The fourth value of $\boldsymbol{\pi}$ defines the offset of the plane from the origin.

Data points $\mathbf{x}_i = [x_i, y_i, z_i, 1]^T$, obtained by measurement, are somewhat scattered hence, almost never exactly coplanar. Equation 6.30 becomes:

$$[x_i\ y_i\ z_i\ 1]\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{bmatrix} = e_i \neq 0. \tag{6.31}$$

Furthermore, given a cluster of n scattered data points ($n \geq 4$), describing the plane to be fitted, leads to the following system of linear equations:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1\\ \vdots & \vdots & \vdots & \vdots\\ x_n & y_n & z_n & 1\end{bmatrix}\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{bmatrix} = \begin{bmatrix}e_1\\ \vdots\\ e_n\end{bmatrix}, \tag{6.32}$$

or briefly:

$$\mathsf{D}\boldsymbol{\pi} = \mathbf{e}, \tag{6.33}$$

where D is the design matrix. Eliminating the last, unit column from the design matrix via orthogonal residualization (see Section 4.3.1.1) places the centroid of the data points on the fitted plane and yields the equation:

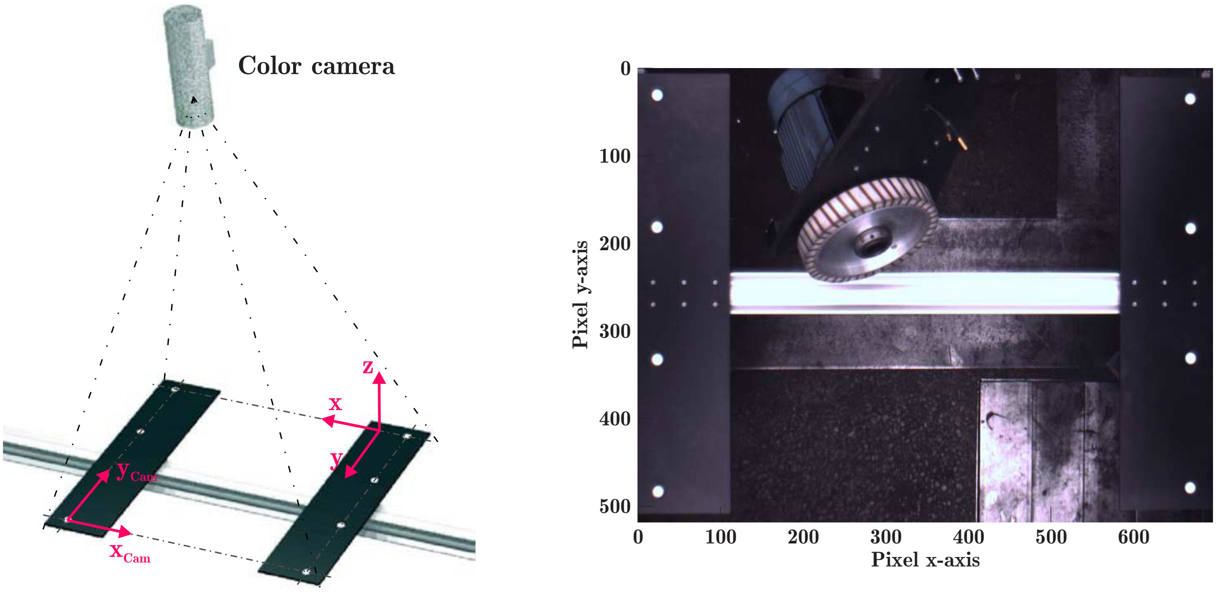
$$\widehat{\mathsf{D}}\mathbf{n} = \begin{bmatrix} x_1-\overline{x} & y_1-\overline{y} & z_1-\overline{z}\\ \vdots & \vdots & \vdots\\ x_n-\overline{x} & y_n-\overline{y} & z_n-\overline{z}\end{bmatrix}\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\end{bmatrix} = \begin{bmatrix}e_1\\ \vdots\\ e_n\end{bmatrix}. \tag{6.34}$$

Consequently, the minimization of $\mathbf{e}$ in the least squares sense is performed by applying singular value decomposition [55] to the design matrix $\widehat{\mathsf{D}} \xrightarrow{\text{SVD}} [\mathsf{U}, \mathsf{S}, \mathsf{V}]$. As a result, the sought vector $\mathbf{n}$ is the rightmost singular vector of matrix V. Moreover, the SVD imposes the constraint $\pi_1^2 + \pi_2^2 + \pi_3^3 = 1$, which yields residuals e_i as *normalized* normal distances of the data points to the fitted plane. Finally, the remaining coefficient π_4 is obtained with the following back-substitution:

$$\pi_4 = -[\overline{x}\ \overline{y}\ \overline{z}]\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\end{bmatrix}. \tag{6.35}$$

6.3 Calibration of the color camera

Two calibration panels, each equipped with four collinear reference spots, are used to calibrate the color camera. Each reference spot is a circular light source that is embedded flush with the panel surface. It is at a precisely measured location and has a known diameter. As shown in Figure 6.2a, together the panels establish a virtual plane that is defined via the color camera's coordinate system $[x_{Cam}, y_{Cam}]$. The origin of this frame is chosen to be the center point of the spot in the south-west corner. Calibration begins with the acquisition of an image that shows both panels, Figure 6.2b. Having obtained the coordinates of the center points of the spots with respect to the camera's coordinate frame and pixel coordinate system, respectively, the homography between both planes is uniquely determined as A_{Cam}.



(a) Calibration of the color camera **(b) Captured image**

Figure 6.2: Calibration of the color camera: The principle of calibration is shown in the left figure (a). Two calibration panels, arranged to be coplanar, each equipped with four collinear light spots, define the color camera's coordinate system. Via the known location of the spots with respect to the real-world and the pixel coordinate system, respectively, the homography between both planes is uniquely determined. The corresponding calibration image is shown in the figure to the right (b).

6.3.1 Determination of the homography

Having a calibration image, as shown in Figure 6.2b, the determination of the homography that fulfills the mapping $\mathbf{x}_{Cam} = \mathsf{A}_{Cam}\mathbf{x}_p$, where $\mathbf{x}_{Cam} = [x_{Cam}, y_{Cam}, w]^T$ and $\mathbf{x}_p = [x_p, y_p, 1]^T$ are homogeneous coordinates of corresponding points with respect to the metric real-world and the pixel coordinate system, is done in the following three steps:

6.3.1.1 Acquisition of corresponding coordinates of the reference points

The metric-coordinates of the center points of the eight reference spots $[\mathbf{x}_{Cam,1}, \dots, \mathbf{x}_{Cam,8}]^T$, with respect to the individual coordinate system of the color camera, are derived from the geometry of the calibration panels and the distance between both plates.

The pixel-coordinates have to be extracted from the captured calibration image. Therefore, the acquired color image is mapped from the acquired RGB color space to the HSL color space (see Section 3.2.5). The advantage of using the HSL model is that the monochrome image of the scene, required in subsequent image processing, is then available directly from the luminescence channel. In addition, the boundary points of the calibration spots are extracted via thresholding and the subsequent application of a contour algorithm [34] to the regions of interest. This yields edge contour coordinates of the eight reference spots $[\mathsf{X}_{c,1}, \dots, \mathsf{X}_{c,8}]^T$, where $\mathsf{X}_{c,i} = [\mathbf{x}_{pc,1}, \dots, \mathbf{x}_{pc,n}]$ is a matrix containing the pixel-coordinates of n contour points on the i^{th}-calibration spot.

6.3.1.2 Making an initial guess for the homography

An initial guess of the homography parameters is made using the DLT algorithm, presented in Section 5.1.2. This homogeneous mapping requires at least four reference points in both planes. Real-world coordinates $[\mathbf{x}_{Cam,1}, \dots, \mathbf{x}_{Cam,8}]^T$ of these points were acquired in the prior step. Corresponding pixel-coordinates $[\mathbf{x}_{p,1}, \dots, \mathbf{x}_{p,8}]^T$ are obtained by fitting ellipses (see Section 6.2.2) to the contour coordinates of the eight reference spots. Consequently, inserting these points in Equation 5.16 yields the homography A_{Cam}.

The matrix A_{Cam} can be normalized by setting one of its non-zero entries to unity, e.g. $a_{ij} = 1$, without altering the projectivity. Below, $a_{33} = 1$ was selected:

$$\mathsf{A}_{Cam} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}. \tag{6.36}$$

With this choice the origin $\mathbf{x} = [0, 0]^T$ can never be mapped to the line at infinity, i.e. to the vanishing line: $l_\infty = a_{31}x + a_{32}y + 1$. However, this is not a serious restriction in metric vision applications.

6.3.1.3 The nonlinear approximation of the homography

This approximation optimizes the entries of the collineation matrix A_{Cam} so that geometric distances among corresponding points in both planes are minimized in a least squares sense. A function f that defines the relationship between these corresponding points, using the parameters of the homography, is formulated; This function is derived from the shape of the calibration spots.

Given x_0, y_0 and r, which are the center coordinates and the radius of a circular light spot, the geometric distance of a datum or contour point to the fit circle is defined as:

$$f(\mathbf{x}_{pc}, \mathbf{a}) = \sqrt{\left(x_0 - x_{a,i}\right)^2 + \left(y_0 - y_{a,i}\right)^2} - r, \tag{6.37}$$

where $x_{a,i}$ and $y_{a,i}$ are the inhomogeneous metric-coordinates of a data point. Having the pixel-coordinates of the contour points $\mathbf{x}_{pc} = [x_{pc}, y_{pc}, 1]^T$, the corresponding metric-representation of a point is given by:

$$x_{a,i} = \frac{x_i}{w_i} = \frac{a_{11}x_{pc} + a_{12}y_{pc} + a_{13}}{a_{31}x_{pc} + a_{32}y_{pc} + 1} \quad \text{and} \quad y_{a,i} = \frac{y_i}{w_i} = \frac{a_{21}x_{pc} + a_{22}y_{pc} + a_{23}}{a_{31}x_{pc} + a_{32}y_{pc} + 1}. \tag{6.38}$$

Substituting these two equations into Equation 6.37, yields the cost function:

$$f(\mathbf{x}_{pc}, \mathbf{a}) = \sqrt{\left(x_0 - \frac{a_{11}x_{pc} + a_{12}y_{pc} + a_{13}}{a_{31}x_{pc} + a_{32}y_{pc} + 1}\right)^2 + \left(y_0 - \frac{a_{21}x_{pc} + a_{22}y_{pc} + a_{23}}{a_{31}x_{pc} + a_{32}y_{pc} + 1}\right)^2} - r. \tag{6.39}$$

This function is in general not zero since the data points do not usually define any perfect circle:

$$f(\mathbf{x}_{pc}, \mathbf{a}) = e_{pc} \neq 0. \tag{6.40}$$

The Gauss-Newton method (see Section 6.2.1) is applied to minimize the sum of the squares of these residuals $\sum e_{pc}$. The normalization of the matrix A_{Cam} by setting $a_{33} = 1$ implies that there are only eight non-zero partial derivatives. The coefficient vector increment is:

$$\Delta\mathbf{a} = -\mathsf{J}^+_{\mathbf{f}(\mathbf{x}_{pc}, \mathbf{a}_0)} \mathbf{f}(\mathbf{x}_{pc}, \mathbf{a}_0), \tag{6.41}$$

where $\mathsf{J}^+_{\mathbf{f}(\mathbf{x}_{pc}, \mathbf{a}_0)}$ is the pseudo-inverse [54] of the Jacobian. Every contour point, from each of the eight calibration spots, yields a row in the Jacobian and thus, its definition for m points is given by:

$$\mathsf{J}_{\mathbf{f}(\mathbf{x}_{pc}, \mathbf{a}_0)} = \frac{\partial \mathbf{f}\left(\mathbf{x}_{pc}, \mathbf{a}_0\right)}{\partial \mathbf{a}_0} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_{pc,1}, \mathbf{a}_0)}{\partial a_{11}} & \cdots & \frac{\partial f_1(\mathbf{x}_{pc,1}, \mathbf{a}_0)}{\partial a_{32}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n(\mathbf{x}_{pc,m}, \mathbf{a}_0)}{\partial a_{11}} & \cdots & \frac{\partial f_n(\mathbf{x}_{pc,m}, \mathbf{a}_0)}{\partial a_{32}} \end{bmatrix}. \tag{6.42}$$

Using the substitutions:

$$dx = \left(x_0 - \frac{a_{11}x_{pc} + a_{12}y_{pc} + a_{13}}{a_{31}x_{pc} + a_{32}y_{pc} + 1}\right) \quad , \quad dy = \left(y_0 - \frac{a_{21}x_{pc} + a_{22}y_{pc} + a_{23}}{a_{31}x_{pc} + a_{32}y_{pc} + 1}\right), \tag{6.43}$$

$$\Psi = \sqrt{dx^2 + dy^2}\left(a_{31}x_{pc} + a_{32}y_{pc} + 1\right); \tag{6.44}$$

the eight partial derivatives of $f(\mathbf{x}_{pc}, \mathbf{a})$ become:

$$\frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{11}} = \frac{-dx\, x_{pc}}{\Psi} \quad , \quad \frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{12}} = \frac{-dx\, y_{pc}}{\Psi} \quad , \quad \frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{13}} = \frac{-dx}{\Psi}; \tag{6.45}$$

$$\frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{21}} = \frac{-dy\, x_{pc}}{\Psi} \quad , \quad \frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{22}} = \frac{-dy\, y_{pc}}{\Psi} \quad , \quad \frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{23}} = \frac{-dy}{\Psi}; \tag{6.46}$$

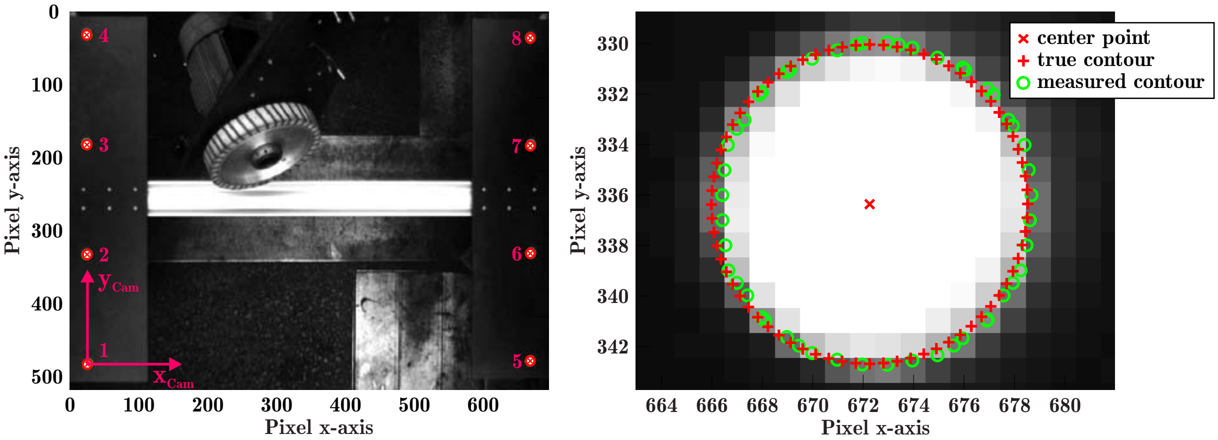
$$\frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{31}} = \frac{dx\left(a_{11}x_{pc} + a_{12}y_{pc} + a_{13}\right)x_{pc} + dy\left(a_{21}x_{pc} + a_{22}y_{pc} + a_{23}\right)x_{pc}}{\Psi\left(a_{31}x_{pc} + a_{32}y_{pc} + 1\right)}; \tag{6.47}$$

$$\frac{\partial f(\mathbf{x}_{pc}, \mathbf{a}_0)}{\partial a_{32}} = \frac{dx\left(a_{11}x_{pc} + a_{12}y_{pc} + a_{13}\right)y_{pc} + dy\left(a_{21}x_{pc} + a_{22}y_{pc} + a_{23}\right)y_{pc}}{\Psi\left(a_{31}x_{pc} + a_{32}y_{pc} + 1\right)}. \tag{6.48}$$

According to Equation 6.45-6.48, the numerical parameters of the function $f(\mathbf{x}_{pc}, \mathbf{a}_0)$ and the Jacobian $\mathsf{J}_{\mathbf{f}(\mathbf{x}_{pc},\mathbf{a}_0)}$ can be calculated. The offset $\Delta\mathbf{a}$, is determined using the initial guess $\mathbf{a}_0$. The improved matrix of the homography is obtained by adding the offset to the initial guess:

$$\mathbf{a} = \mathbf{a}_0 + \begin{bmatrix} \Delta\mathbf{a} \\ 0 \end{bmatrix}. \tag{6.49}$$

Note, the initial guess $\mathbf{a}_0 = [a_{11}, \ldots, a_{32}, 1]^T$ contains nine entries of the matrix A and is improved by adding eight parameters $\Delta\mathbf{a} = [\Delta a_{11}, \ldots, \Delta a_{32}]^T$ corresponding to the eight non-zero partial derivatives. Finally, the improved parameter vector $\mathbf{a}$ can be used as initial parameters of a new cycle where $\mathbf{a}_0 = \mathbf{a}$. This procedure is repeated until convergence is reached when $\Delta\mathbf{a}$ is deemed to be sufficiently small.



(a) Processed image

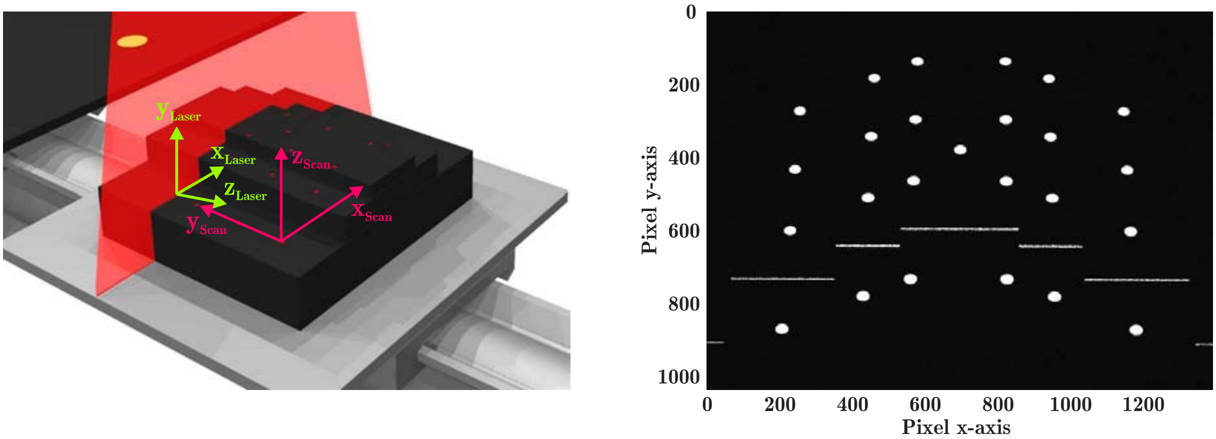
(b) Enlarged calibration mark no. 6

Figure 6.3: Approximation of the homography: The left image (a) shows the processed calibration image, containing the eight calibration spots and the color camera's coordinates system. In addition to this, the homography between the real-world and the image plane was calculated. The image to the right (b) gives an enlarged detail of a calibration mark. As one can see, the measured contour points of the light source and the calculated border points are almost identical.

The result of calibrating the color camera is shown in Figure 6.3a. The contour points of the eight reference spots were determined. Furthermore, the origin of the color camera's coordinate system was chosen to be the center point of the lower reference spot on the left calibration panel. From this, the x- and y-axes are aligned orthonormal, such that the y-axis is along the collinear marks on the left panel and the x-axis intersects the lower spot on the right panel. An enlarged detail of such a calibration mark is shown in Figure 6.3b. As one can see, the measured contour points of the light source and the calculated border points are almost identical. The cause of a small but perceptible bias is due to lens distortions. The magnitude of the distortion of the high quality lens used is so small that its effect is negligible. A detailed treatment of lens distortions and how to deal with them is given in [77, 78].

6.4 Calibration of a plane-of-light scanner

The aim of calibrating a plane-of-light scanner is the determination of a collinear relationship between the camera plane, i.e. the coordinates of the captured intersection curve, and a predefined three-dimensional coordinate system. This calibration is carried out with a step-target of known, specific geometry, built for the purpose. It is shown in Figure 6.4a. The step target consists of three parallel planes, where each stage is equipped with a pattern of coplanar, embedded circular light sources that serve as reference spots. The target is located in the plane-of-light so that the laser-plane intersects all three planes. A typical calibration image that captures the scene is shown in Figure 6.4b.



(a) Calibration of a plane-of-light scanner

(b) Captured image

Figure 6.4: Calibration of a plane-of-light scanner: Referring to the left figure (a), a target is used to fully calibrate a plane-of-light scanner. Having acquired a calibration image, capturing the scene as shown the figure to the right (b), the known geometry of the step target is sufficient in order to calculate the homography between the laser plane and the image plane. A proper choice of the scanner's three-dimensional coordinate system $[x_{Scan}, y_{Scan}, z_{Scan}]$ enables the scanner to determine the unsmeared profile of the intersected workpiece, without reference to the orientation of the laser plane.

This method of calibrating a plane-of-light scanner requires the definition of two coordinate frames (see Figure 6.4a.):

1. A three-dimensional coordinate system $[x_{Scan}, y_{Scan}, z_{Scan}]$ is chosen to be located on the centroid of the south-west light source of the lowermost plane. Its orientation is

that the xy_{Scan}-plane is parallel to the stages of the target and the y_{Scan}-axis is along the collinear reference marks. This frame is referred to as the scanner's individual coordinate system;

2. A three-dimensional coordinate system $[x_{Laser}, y_{Laser}, z_{Laser}]$ is defined such that its xy_{Laser}-plane is coplanar to the plane-of-light and the origin is positioned at the point of intersection between the laser-plane and the y_{Scan}-axis. This frame is called laser coordinate system.

Having acquired a calibration image (see Figure 6.4b), the calibration is performed in two steps: Firstly, position and orientation of the laser coordinate system with respect to the scanner's individual coordinate frame are specified. In addition, the homogeneous coordinate transformation matrix T_{Scan}, that defines the mapping between these two frames, is determined; Secondly, the homography A_{Scan} between the image plane and the laser coordinate system is determined. As a result of calibration, the pixel-coordinates of the captured intersection curve can be projected to the metric laser coordinate system and from there, to the scanner's individual coordinate frame. This combination of a planar collineation and a 3D coordinate transformation yields a new collinear transformation $\tilde{\mathsf{A}}_{Scan}$ and specifies the required mapping $\mathbf{x}_{Scan} = \tilde{\mathsf{A}}_{Scan}\mathbf{x}_p$ (see Section 5.3.1.2).

6.4.1 Determination of the laser coordinate system

The aim of determining the laser coordinate system is the specification of an equation that defines the laser plane with respect to the scanner's individual coordinate system. Once orientation and location of the laser plane are known, the translation and rotation entries of the homogeneous transformation matrix T_{Scan} are derived from the coefficients of the plane-equation.

A plane is uniquely determined by a minimum of two distinct, intersecting or parallel lines or three non-collinear points. The laser plane is intersects the calibration target producing a line image on every stage (see Figure 6.4b). According to this, one has to determine the coordinates of at least two lines that are located on two different stages, or correspondingly a minimum of three points lying on these two lines, to determine the plane-of-light. This is explained in detail below:

Using the nonlinear approximation of homographies (see Section 6.3.1), the collineation matrices A_i between the three stages $i = a, b, c$ of the calibration target and the camera chip are determined (Thereby, the embedded light sources within every stage are used as calibration spots.). In addition, the position of the laser line is extracted from the calibration image by appropriate image processing. Depending on the stage i where the laser lines are located, their pixel-coordinates are mapped to the real-world using the associated homography A_i. This yields metric x_{Scan}- and y_{Scan}-coordinates of points that are elements of the laser-plane. The corresponding z_{Scan}-coordinates are obtained from the geometrical distances between the respective stages and the origin of the scanner's individual coordinate system. Finally, the coefficients $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \pi_4]^T$ of the plane-equation are determined with the fitting approach, presented in Section 6.2.3, and the laser-plane is fully determined by:

$$\mathbf{x}_{Scan}^T \boldsymbol{\pi} = 0, \tag{6.50}$$

The final step in specifying the laser coordinate system is the determination of a matrix T_{Scan} that describes the three-dimensional coordinate transformation from the laser coordinate system to the scanner's individual coordinate frame:

$$\mathbf{x}_{Scan} = \mathsf{T}_{Scan}\mathbf{x}_{Laser}, \tag{6.51}$$

where, $\mathbf{x}_{Scan}$ and $\mathbf{x}_{Laser}$ are homogeneous coordinates of corresponding points with respect to their individual coordinate frames and T_{Scan} is the 4×4 homogeneous transformation matrix, of the form:

$$\mathsf{T}_{Scan} = \begin{bmatrix} \mathsf{R} & \mathbf{o} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \Delta x \\ r_{21} & r_{22} & r_{23} & \Delta y \\ r_{31} & r_{32} & r_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.52}$$

The translation vector $\mathbf{o}$ and the rotation matrix R within the matrix T_{Scan} are derived from the plane coefficients $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \pi_4]^T$. The translational part is determined via the relative location of the origin of the laser coordinate system with respect to the scanner's individual coordinate frame. Thus, this displaced origin position vector is determined by substituting $x_{Scan} = 0$ and $z_{Scan} = 0$ in the plane-equation:

$$y_{Scan}\,\pi_2 + \pi_4 = 0; \tag{6.53}$$

$$y_{Scan} = -\frac{\pi_4}{\pi_2}. \tag{6.54}$$

Consequently, the translation vector $\mathbf{o}$ becomes:

$$\mathbf{o} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{\pi_4}{\pi_2} \\ 0 \end{bmatrix}. \tag{6.55}$$

The rotation matrix R is specified by the three Euler angles φ, ϑ and ψ (see Section 5.2). These angles are determined by the orientation of the laser coordinate system with respect to the scanner's individual coordinate frame. The orientation or direction of a frame is given by its z-axis. Figure 6.5a shows the two coordinate frames having the directions: $\mathbf{o}_{z,Scan} = [0, 0, 1]^T$ and $\mathbf{o}_{z,Laser} = \mathbf{n} = [\pi_1, \pi_2, \pi_3]^T$. Thus, the three Euler angles are determined as follows:

1. The normal vector $\mathbf{n}$ is orientated parallel to the y_{Scan}-axis but in the negative direction so:

   if $(\pi_2 > 0)$, **then** $(\mathbf{n} = -\mathbf{n})$;

2. The angle φ is the angle between the x_{Scan}- and the x_{Laser}-axis. The orientations of these axes are:

$$\mathbf{o}_{x,Scan} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \mathbf{o}_{x,Laser} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} -\pi_2 \\ \pi_1 \\ 0 \end{bmatrix}; \tag{6.56}$$

3. The angle φ between the vectors $\mathbf{o}_{x,Scan}$ and $\mathbf{o}_{x,Laser}$ is computed as follows:

$$\varphi = \arccos\left(\frac{\mathbf{o}_{x,Scan}\,\mathbf{o}_{x,Laser}}{\|\mathbf{o}_{x,Scan}\|\,\|\mathbf{o}_{x,Laser}\|}\right) = \arccos\left(\frac{-\pi_2}{\sqrt{\pi_1^2+\pi_2^2}}\right); \tag{6.57}$$

4. The relative orientation of the x_{Laser}-axis to the x_{Scan}-axis determines the sign of the angle φ:

   if $\big(\mathbf{o}_{x,Laser}(2) \geq 0\big)$, **then** $(\varphi = \varphi)$, the rotational direction is counter-clockwise and;

   if $\big(\mathbf{o}_{x,Laser}(2) < 0\big)$, **then** $(\varphi = -\varphi)$, the rotational direction is clockwise;

5. The angle ϑ is the angle between the z_{Scan}- and the z_{Laser}-axis. The orientations of these axes are:

$$\mathbf{o}_{z,Scan} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad , \quad \mathbf{o}_{z,Laser} = \mathbf{n} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix}; \tag{6.58}$$

6. In addition, the angle ϑ between the vectors $\mathbf{o}_{z,Scan}$ and $\mathbf{o}_{z,Laser}$ is calculated by Equation 6.59:

$$\vartheta = \arccos\left(\frac{\mathbf{o}_{z,Scan}\,\mathbf{n}}{\|\mathbf{o}_{z,Scan}\|\,\|\mathbf{n}\|}\right) = \arccos\left(\frac{\pi_3}{\sqrt{\pi_1^2+\pi_2^2+\pi_3^2}}\right); \tag{6.59}$$

7. The test and negate step 1 ensures $\vartheta \geq 0$;

8. The final rotation through angle ψ causes a rotation of the laser coordinate frame about an axis normal to the laser plane and does not affect its orientation. Consequently, ψ can be set to zero:

$$\psi = 0. \tag{6.60}$$

As a result, the rotation matrix R is obtained by the multiplication of the successive rotations about the corresponding axes by the three Euler angles φ, ϑ and ψ. The individual rotation matrices are given by:

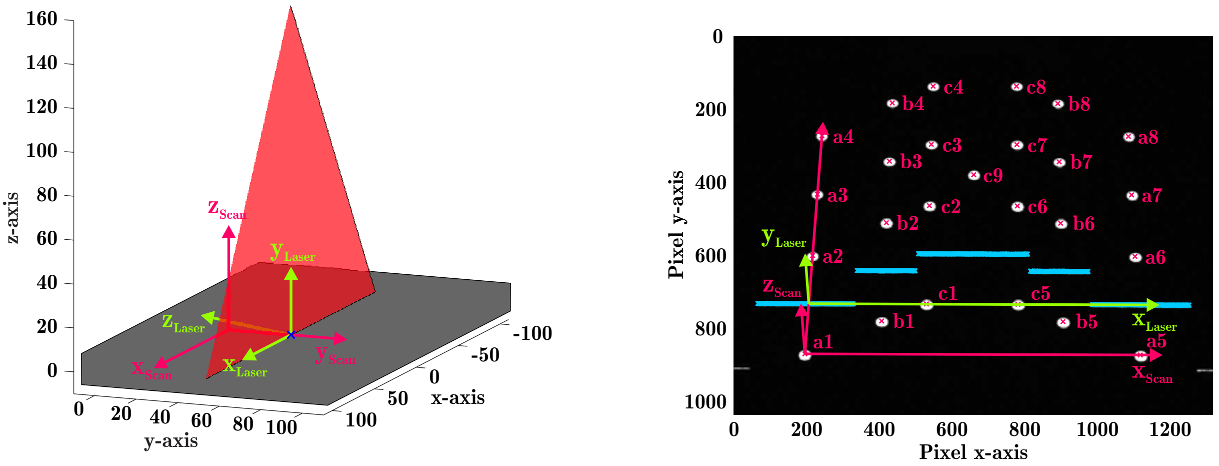
$$\mathsf{R}_\varphi = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathsf{R}_\vartheta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta) & \sin(\vartheta) \\ 0 & -\sin(\vartheta) & \cos(\vartheta) \end{bmatrix}, \mathsf{R}_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.61}$$

which yields the generic rotation matrix R:

$$\mathsf{R} = \mathsf{R}_\varphi \mathsf{R}_\vartheta \mathsf{R}_\psi = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{6.62}$$

It should be noted that the inverse transformation T_{Scan}^{-1}, which defines the three-dimensional coordinate transformation from the scanner's individual coordinate system back to the laser coordinate system is given by:

$$\mathsf{T}_{Scan}^{-1} = \begin{bmatrix} \mathsf{R}^T & -\mathsf{R}^T\mathbf{o} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{6.63}$$



(a) Coordinate systems **(b) Processed image**

Figure 6.5: Determination of the laser coordinate system: The figure to the left (a) shows the alignment of the laser coordinate system with respect to the scanner's individual coordinate frame. The processed calibration image to the right (b), depicts the analyzed reference marks, the localized segments of the laser line and finally, the position and orientation of both coordinate systems.

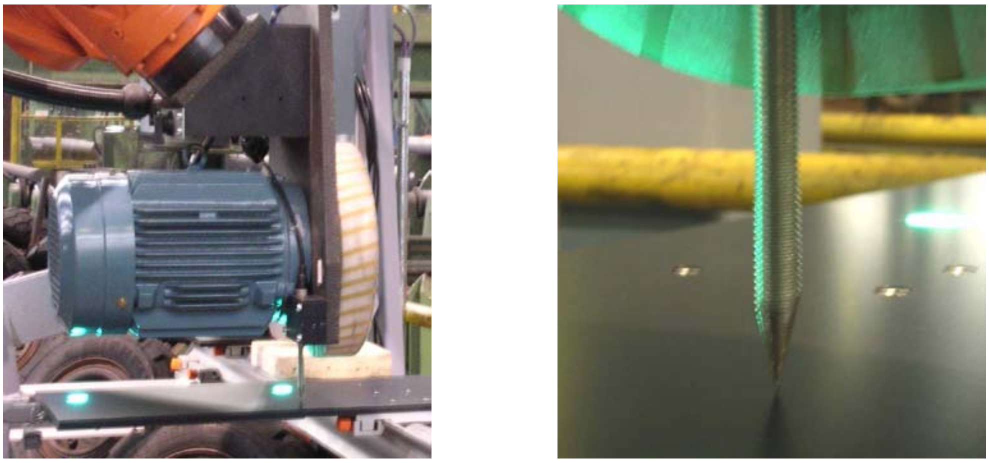
The results of determining the laser coordinate system are shown in Figure 6.5a and Figure 6.5b. The left figure shows the alignment of the laser coordinate system with respect to the scanner's individual coordinate frame. The calibration image to the right, shows the designated calibration spots, labelled $a1 - a8$, $b1 - b8$ and $c1 - c9$, and the determined positions of the laser lines.

6.4.2 Determination of the homography between the image plane and the laser plane

The next step in calibrating a plane-of-light scanner is to determine the homography A_{Scan} between the image plane and the laser coordinate system. To calculate this matrix, the coordinates of at least four points of the intersection curve from the plane-of-light and the step target, in general position with respect to both frames, are required: The image-coordinates of these points are known through the location of the laser line in the image, the corresponding metric-coordinates are obtained by mapping the intersection points from the scanner's individual coordinate system to the laser coordinate system using T_{Scan}^{-1}. Thus, the homography A_{Scan} is calculated using the DLT algorithm (see Section 5.1.2).

6.5 Calibration of the grinding robot

The grinding robot was calibrated by its manufacturer so as to identify all of its intrinsic parameters. Consequently, calibration of the robot in the billet flaw grinding system is confirmed to register it with the system world coordinate frame. This is performed by



(a) Robot's end effector **(b) Calibration spike**

Figure 6.6: Registration of the grinding robot: A triplet of small-drill holes specify the origin and the orientation of the world coordinate system. The registration of the robot with this coordinate frame is achieved by navigating a calibration spike, mounted on the end-effector, to the drill-holes manually.

navigating a calibration spike, mounted on the end-effector, into holes, drilled in the calibration panels for this purpose. These marks determine the origin and the orientation of the world coordinate system. End effector, tapered spike and registration holes used in this manual teaching process are shown in Figure 6.6.

6.6 Registration of the individual coordinate systems

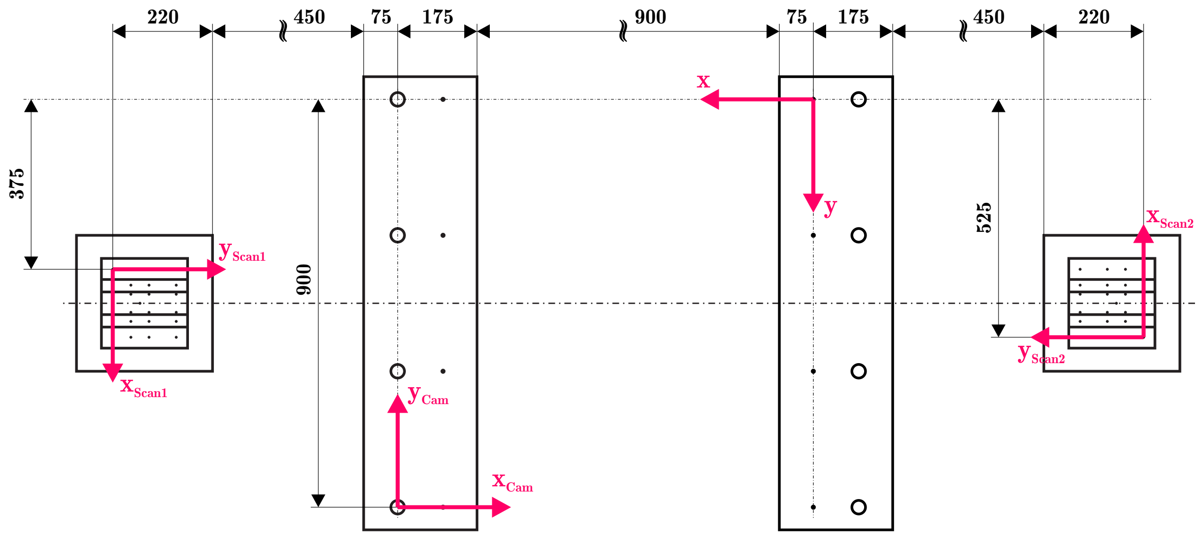


Figure 6.7: Registration of the individual coordinate frames: The position and the orientation of the individual frames with respect to the world coordinate system are determined with the calibration assembly.

The individual coordinate frames of the color camera and the two plane-of-light scanners have to be registered with the world coordinate system. The position and the orientation of all coordinate frames are determined with the calibration equipment assembly. This arrangement is shown in Figure 6.7.

The homogeneous transformation matrices $\mathsf{T}_{Cam,w}$, $\mathsf{T}_{Scan1,w}$ and $\mathsf{T}_{Scan2,w}$ that describe the registrations (see Section 5.3.2) are determined by their individual translation vector $\mathbf{o}$ and rotation matrix R, as summarized in Table 6.1.

Device	$\mathbf{o} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$	$\mathsf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$
Color camera	$\mathbf{o} = \begin{bmatrix} 1150 \\ 900 \\ 0 \end{bmatrix}$	$\mathsf{R} = \begin{bmatrix} \cos(\pi) & -\sin(\pi) & 0 \\ \sin(\pi) & \cos(\pi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Plane-of-light scanner 1	$\mathbf{o} = \begin{bmatrix} 1895 \\ 375 \\ 40 \end{bmatrix}$	$\mathsf{R} = \begin{bmatrix} \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) & 0 \\ \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Plane-of-light scanner 2	$\mathbf{o} = \begin{bmatrix} -845 \\ 525 \\ 40 \end{bmatrix}$	$\mathsf{R} = \begin{bmatrix} \cos(-\frac{\pi}{2}) & -\sin(-\frac{\pi}{2}) & 0 \\ \sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 6.1: Position and orientation of the individual coordinates frames with respect to the world coordinate system.

Part III

Operational Reliability

Chapter 7

Operational Results

The final part of this work starts with the presentation of operational results to verify the full functionality of the introduced concept. The proposed examples are derived from five installed and successfully commissioned grinding facilities of similar design at two different steel production plants. Results are given for the industrial implementation, the color detection and the measurement of the cross section. The quality of the machining process is provided through a comparison of the material surface before and after the grinding process. In addition, it is shown that the accuracy of the system is sufficient, because the robot reshapes the bar precisely. All these operational results are verified for both types of billet.

7.1 Industrial implementation

The first fully automated grinding facility was developed early in 2005. Prior to this, a laboratory setup was put together to determine general system configuration parameters that are needed to implement the industrial prototype. Experience in the lab formed the basis for the following industrial implementation:

The color camera, shown in Figure 7.1a, is placed approximately 3.0m above the material conveyor. This distance ensures an appropriate field of view to accommodate robot workspace while ensuring adequate clearance to avoid interference with moving parts. Inspection and repair must encompass the entire surface of the billets. Because of this, the robot workspace is established as:

- Since the image processing system guides the robot, effective workspace is limited to the color camera's field of view;
- The inspection and repair of the whole surface of a billet is ensured, if workspace accommodates a complete axial revolution of the billet on the conveyor;
- Due to the design of the material conveyor, the billets advance and rotate simultaneously. So billet rate of advance is directly proportional to the billet's circumference. Furthermore, the calculation of the exact axial feed rate is complicated by unknown, uncontrollable slip between material and conveyor.

The required minimum working region was determined by carrying out a test sequence, wherein a billet of maximum circumference is moved through its entire length along the

(a) Illumination and color camera **(b) Plane-of-light scanner**

Figure 7.1: Industrial implementation of the color camera, the lighting and a plane-of-light scanner. The assembly of the individual components is derived from a feasibility study in the plant, which was carried out before the industrial implementation.

conveyor. As a result of this, the motion of a square billet with a side-length of 250mm imposed the choice of a minimum working range of 1.2m in the feed direction.

Figure 7.1a also shows the illumination of the grinding facility. It consists of a 1KW bank of 20 fluorescent tubes and emits diffuse daylight spectrum light to ensure uniform illumination of the working region. This arrangement of light sources is a compromise among ideal illumination, the required uniformity for reliable color detection and an acceptable placement of lamps outside the robot workspace.

The measurement of the billet's cross section is performed by two plane-of-light scanners that are arranged on ether side of the robot's working range. One of these scanners is shown in Figure 7.1b. The scanners are centered on the conveyor at approximately 90cm above the material. Additionally, this height was shown to provide the optimal coverage of the cross section of all billet dimensions, without necessity to change the position of the scanner and/or camera and laser position to cover the measurement range. Scanner mounts have no moving parts and this too is of great advantage to ensure reproducible measurement accuracy.

Moreover, all electronic and optical component enclosures conform to protection class IP65[1] to protect image processing components from air burn contamination, e.g. grinding swarf and aerosols that are prevalent in the production hall. The optical paths are protected by means of heat resistant glass that can withstand contact with hot grinding swarf.

The industrial robot with a belt grinding unit on its end-effector is shown in Figure 7.2a. The robot was especially designed for industrial applications and its design with parallel

---

[1] International protection classes according to DIN EN 60529

(a) Grinding robot (b) Safety fence

Figure 7.2: The industrial robot with a belt grinding unit on its end effector (a) and a security fence (b) that is coupled with an emergency stop mechanism are shown.

linkages provides a sturdy structure connected with a high-speed and high contact force grinding process without compromising accurate path control. Additionally, automatically guided robot motion is potentially hazardous to production hall workforce. This requires the construction of a safety fence surrounding the robot's workspace. An emergency stop mechanism, immediately freezes the robot's six rotational joints and cuts-off power to the grinding unit, if a security door (see Figure 7.2b) is inadvertently opened. Because the billet conveyor entry port and the security interlock equipped door are the only access to the grinding facility it conforms to a desirable high level of industrial safety. Note that the development of the belt grinding unit as well as the programming of the grinding robot was not a part of this work and was carried out by *Carat robotic innovation GmbH*[2].

The final part of the industrial implementation was the enclosing of the whole facility (see Figure 7.3a and Figure 7.3b). For this purpose, metal panels were used to isolate the facility to avoid safety hazard and the entry of stray light that might interfere with color detection.

The system concept proved to be fully functional. The prototype was accepted and commissioned after several weeks of supervised testing. In addition to this successful industrial implementation, three more plants of similar design were installed as 'turn key' operations in 2006 in the same production hall. Moreover, all four automated grinding facilities were connected to a common supply of billets to maximize their productivity and to minimize the effect of maintenance downtime. From their operational onset, these grinding facilities ensure the high surface quality of the semi-finished products in this steel production plant.

---

[2] See http://www.carat-robotic.de/

(a) Housing under construction **(b) Finished housing**

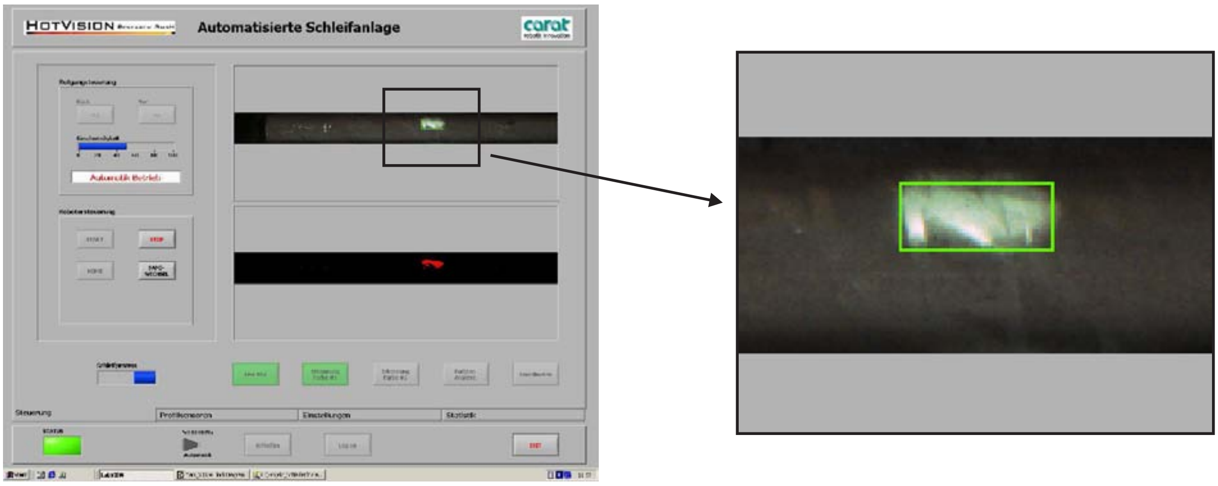
Figure 7.3: The housing that encloses the whole facility protects the color detection from possible influences of stray light and minimizes personnel accident risk.

7.2 Graphical user interface and color detection

Several months of testing allowed for necessary minor image processing algorithm adjustments and improved the real-time performance of the plant. During the same period the graphical user interface (GUI) of the grinding facility was adapted to satisfy customer requirements on one hand, and to meet the need for a user friendly GUI on the other. The graphical user interface is described below along with some typical results concerning color detection effectiveness.

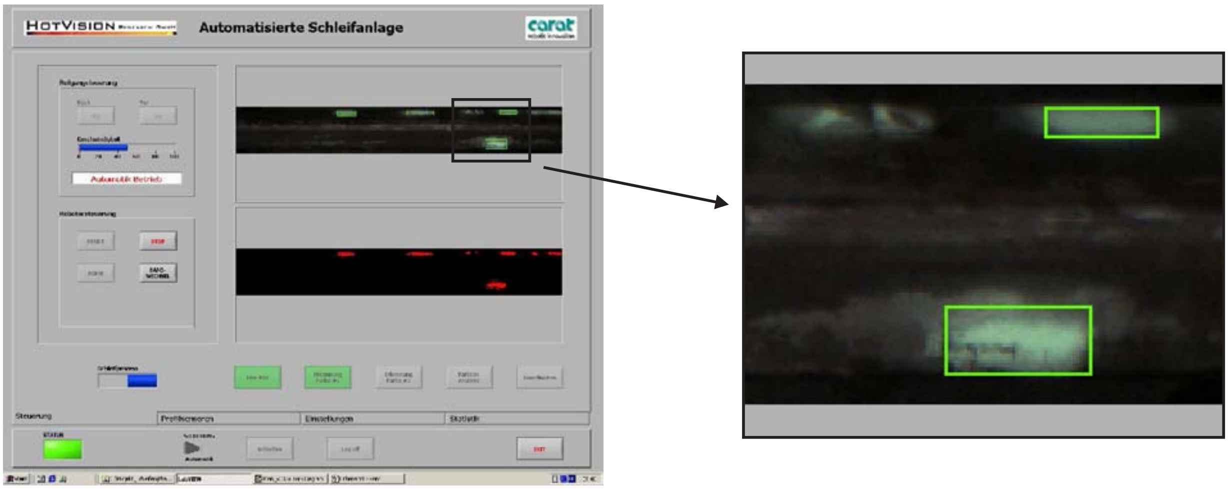
The main window of the graphical user interface and some color detection postprocessed patch images for circular and square billets are shown in Figure 7.4 and Figure 7.5 respectively. The GUI provides visual recordings of events, e.g. flaw appearance or measurement of billet cross sections. It shows the images produced by all three cameras in real-time, contains virtual 'push-buttons' for manual control of the actuators and instruments and offers controls to adjust the system parameters, e.g. thresholds for color detection or camera gains for white balance correction. Moreover, it gives a status overview of the connected hardware and contains a log of relevant events. All changes of the system parameters that are made with the GUI are stored and recovered after a programmed shut-down of the programmable logic controller.

Color detection is the main task in the proposed work cycle, which is performed in real-time at the color camera's maximum frame rate of 30 Hertz. As a consequence, all the required high speed processing, e.g. color image acquisition and processing, is programmed in *C++* and integrated into the main operating program by means of library functions. The processing is reliably performed on color markings on all relevant shape and size of billet and for any patch color chosen by the client. The outcome of the color detection is shown in Figure 7.4b and Figure 7.5b: The captured image is analyzed according to specific color, which is identified by a color threshold triplet. Once a color marking is detected, the coordinates of the surrounding box are calculated and the servoing and



(a) **Color marking on a circular billet** (b) **Enlarged detail**

Figure 7.4: The left image (a) shows the main window of the graphical user interface. The GUI enables the visual exposition of the scene, the flaw appearance, the measurement of the cross section, a log file and the hardware status. It provides an operator interface for the manual control of the material conveyor and the grinding robot and finally, controls to alter the system parameters. The enlarged inset depicts details of a green marked flaw that was detected (b).



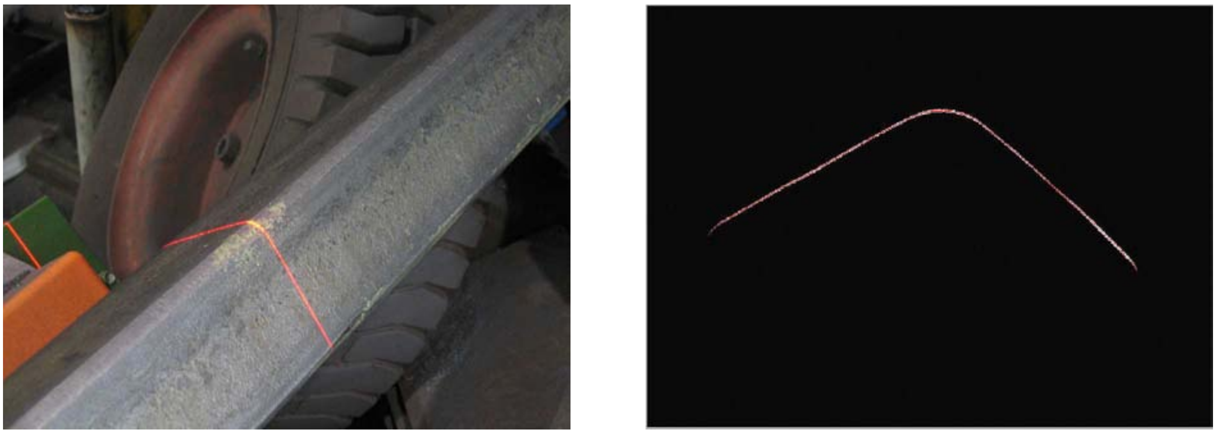
(a) **Color markings on a square billet** (b) **Enlarged detail**

Figure 7.5: Appearance of simultaneous billet surface flaw patches: Color detection is the main task in the real-time work cycle at the color camera's maximum frame rate of 30 Hertz. If several color markings are detected simultaneously, the patch coordinates of all markings are determined, queued and transmitted to the grinding robot.

grinding sequence begins. As this happens, the bounding boxes that appear on the GUI contain the calculated coordinates of the three-dimensional patch. Finally, if several color markings are detected simultaneously, the coordinates are queued and transferred to the grinding robot.

7.3 Measurement of the cross section

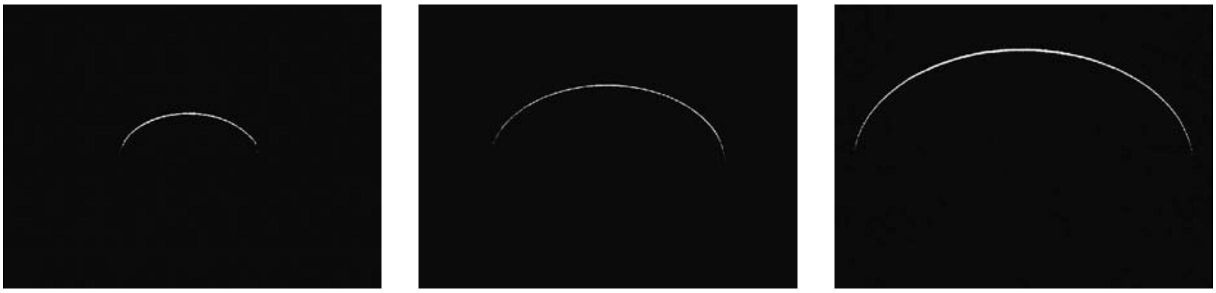
This section presents typical results of cross section measurement for both types of billet. The following six examples were carefully sampled from a production line to verify the accuracy and robustness of the fitting algorithms that were developed. This shows that these fitting routines are satisfactory.



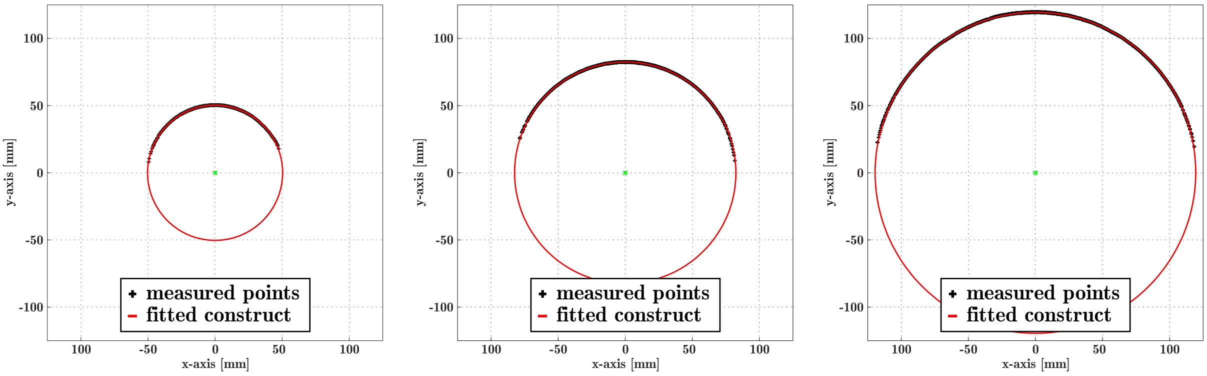
(a) Intersection curve **(b) Visual exposition of the intersection curve**

Figure 7.6: Intersection curve (a), which is obtained from a square cross section and its visual exposition (b): As depicted in the right image, only the laser line remains in the acquired image. This is achieved by using a narrow optical band filter in front of the camera's lens. The filter's band is adjusted to the appropriate laser wavelength.

Figure 7.6 shows a snapshot of an intersection curve between the laser light plane and the billet, and the corresponding representation of the scene on the GUI. The coordinates of the intersection curve are subsequently determined with sub-pixel precision by means of a center of gravity calculation. In the next step, a collinear transformation is used to map the resulting pixel-coordinates to the metric real-world. The undistorted coordinates obtained are used in geometric fitting.



(a) Captured image **(c) Captured image** **(e) Captured image**



(b) Measurement result **(d) Measurement result** **(f) Measurement result**

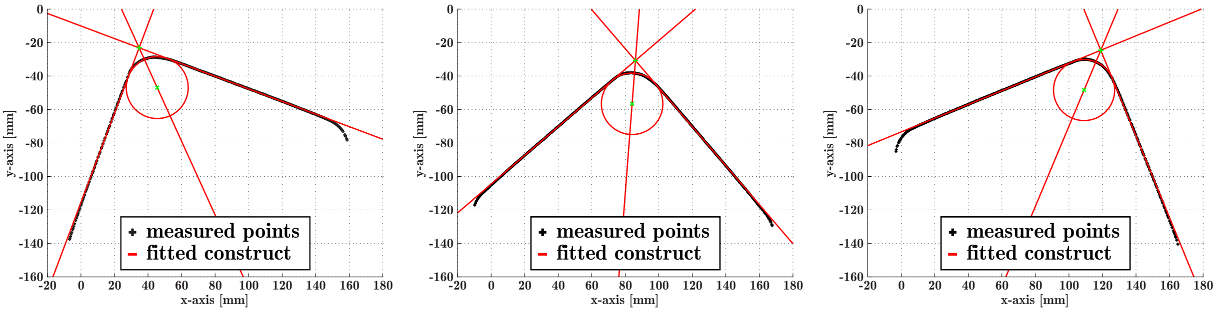
Figure 7.7: Measurement of circular cross sections: The images in the upper row show the acquired measurement images, the corresponding diagrams below depict the fitted cross sections. For the purpose of comparison, the data points were centered at the origin.

Position	Figure 7.7b	Figure 7.7d	Figure 7.7f
Center Point	$x_0 = 65.7\text{mm}$	$x_0 = 62.8\text{mm}$	$x_0 = 83.0\text{mm}$
	$y_0 = -155.9\text{mm}$	$y_0 = -118.2\text{mm}$	$y_0 = -102.3\text{mm}$
Radius	$r = 50.3\text{mm}$	$r = 82.5\text{mm}$	$r = 119.5\text{mm}$
Standard deviation	$\sigma = 0.13\text{mm}$	$\sigma = 0.09\text{mm}$	$\sigma = 0.14\text{mm}$

Table 7.1: Results of the measurement of circular cross sections: Because measurement images were taken from different calibration setups, the absolute coordinates of the center points are subjective. According to product specifications of circular billets, the target radii are $r = 50\text{mm}$, $r = 82.5\text{mm}$ and $r = 120\text{mm}$. These were product specifications not the result of any measurements. Therefore the true dimensions are deemed to be unknown.

The measurement results for circular billets of different dimension are presented in Figure 7.7 and Table 7.1. A comparison of the results with actual data is difficult, since the coordinates of the center points are subjective on the one hand, and the stated dimensions of the bars are just good estimates rather than exact values on the other. For this reason, results are evaluated by calculating the standard deviation σ of the geometrical distances of all data points. Additionally, the radial standard deviation of the pixel generated data point set, from the fitted circle, is sufficiently small so as to conclude that the adopted geometrical approach to establish billet cross section dimensions, is quite satisfactory.

(a) Captured image **(c) Captured image** **(e) Captured image**



(b) Measurement result **(d) Measurement result** **(f) Measurement result**

Figure 7.8: Measurement of a square cross section at different orientations: The images of the upper row show the acquired measurement images, the corresponding diagrams below depict the fitted cross sections.

Position	Figure 7.8b	Figure 7.8d	Figure 7.8f
Point of intersection	$x_{PoI} = 34.7\text{mm}$	$x_{PoI} = 86.1\text{mm}$	$x_{PoI} = 100.7\text{mm}$
	$y_{PoI} = -23.2\text{mm}$	$y_{PoI} = -30.8\text{mm}$	$y_{PoI} = -24.6\text{mm}$
Orientation	$\phi = 24.4°$	$\phi = -4.4°$	$\phi = -22.7°$
Edge radius	$x_0 = 45.5\text{mm}$	$x_0 = 84.1\text{mm}$	$x_0 = 109.0\text{mm}$
	$y_0 = -46.9\text{mm}$	$y_0 = -56.6\text{mm}$	$y_0 = -48.4\text{mm}$
	$r = 18.4\text{mm}$	$r = 18.3\text{mm}$	$r = 18.3\text{mm}$
Side-length	$l = 134\text{mm}$	$l = 130\text{mm}$	$l = 134\text{mm}$
Standard deviation	$\sigma = 0.10\text{mm}$	$\sigma = 0.09\text{mm}$	$\sigma = 0.08\text{mm}$

Table 7.2: Results of the measurement of a square cross section at different orientations: Stable and consistent results are achieved with the introduced fitting approach. On comparing the values of the edge radius r, it is clear that the results are not affected by square billet cross section orientation.

The geometry of two sides of a square billet was investigated at different orientations. Additionally, the measurement results for the captured intersection curves in Figure 7.8 are summarized in Table 7.2. On comparing the results, one can conclude that the algorithms deliver consistently correct results, independent of the orientation of the billet. Again, the geometric distances of all data points to the section contour were determined and presented by their standard deviation σ, since the true values of the results are unknown.

7.4 Grinding process

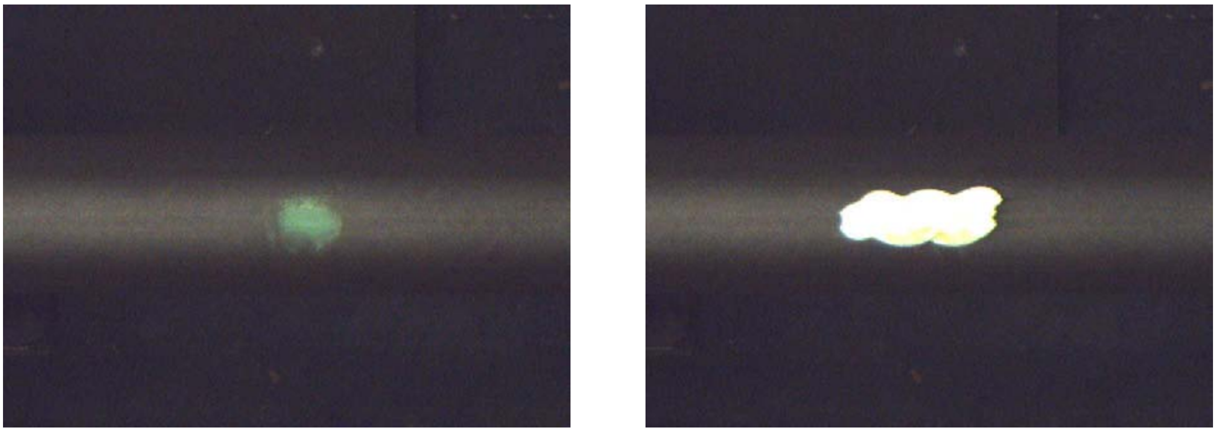
The results from the color detection and the measurement of the cross section are used to calculate three-dimensional coordinates of a patch that contains the billet surface surrounding the color marking. These parameters are subsequently used by the grinding robot controller that executes the machining process to remove flaws. This section presents the operational results of this machining process.

(a) Grinding process **(b) Ground surface**

Figure 7.9: The surface machining of a square billet is shown in the left image (a). For the duration of the grinding process, the position of the billet is fixed to the conveyor with two clamps. The figure to the right (b) shows the surface after the defect has been ground out. Thereby, adequate metal removal is verified by a triangulation sensor that is mounted on the robot's grinding unit.

The grinding process is shown in Figure 7.9. After receiving the patch coordinates from the image processing system, the robot verifies the local geometry of the billet near the color marking with a triangulation sensor that is mounted on the grinding unit. This step is necessary since the cross section of the bar is determined while it is outside the robot's workspace and the robot may have to handle possible local changes in patch location. In addition, the robot calculates the required grinding tool passes, aligns the belt grinding unit to the material surface and starts the machining procedure. After the execution of this process, the grinding depth is verified by the triangulation sensor and, if sufficient metal has not been removed another grinding pass is made. Minimum grinding depth is a setpoint, adjustable via the graphical user interface. Normally, the grinding depth is set to the depth of cracks, determined by of the Thermomatic inspection facility that precedes the grinding facility. In so doing, the complete removal of all flaws is ensured. A typical value for this threshold is $\tau = 1\text{mm}$.

An example of the machining of the surface of a circular billet is given in Figure 7.10. The left image shows the billet before, and the right figure gives the scene after the grinding procedure. On comparing both images, it becomes obvious that the robot reshapes the surface of the bar accurately. Grinding paths are exactly parallel to the curvature of the



(a) Color marking, before grinding **(b) Color marking, after grinding**

Figure 7.10: The surface of a circular billet before (a) and after (b) the machining process. On comparing both images, it becomes obvious that the ground area is equivalent to the entire surface that is specified by the color marking. Moreover, the robot reshapes the surface of the bar accurately and thus, the achieved surface quality is better than that achieved by conventional hand grinding.

surface. In case of grinding non-planar surfaces, i.e. circular shapes or edge radii, the motion of the abrasive belt is cycled to attain a constant attrition. This increases the life of the belt significantly and does not depreciate the quality of the ground surface. Contrary to this, machining a planar surface with a non-uniformly worn belt would yield an uneven face. However, the achieved surface quality is better than that produced by conventional hand grinding. The above example shows that the ground area completely removes the patch. Consequently, the machined surface shows that the introduced vision system for the guidance of the grinding robot fulfills the planned objectives.

Chapter 8

Conclusion and Outlook

Conclusion

An approach for a fully automated repair of surface flaws on steel billets, using a robotic grinder, was presented in this thesis. The adopted technique is based on a machine vision system for the inspection of the billet surface to identify color marked defects and to establish flaw dimensions so as to guide an industrial 6R robot that grinds out the defects. The conclusion of this work may be summarized as follows:

- The continual improvement of the image processing system during the feasibility study and the subsequent test run have led to stable and robust processing algorithms;
- The real-time performance of the work-cycle was achieved. The determining factors were: the use of efficient driver software for image acquisition; a careful selection of proper algorithms for use in the color image processing pipeline sequence; the programming of these drivers and algorithms in *C++* and their implementation by means of library functions. Furthermore, the reliability of the color detection shows that the camera's maximum frame rate of 30 Hertz is sufficient for seamless examination for flaws on the entire billet surface;
- The quality of the ground surfaces was improved considerably above that achieved by manual grinding;
- The use of three independent image processing systems for the guidance of a grinding robot has led to the development of a new calibration concept which allows independent calibration of individual components as a first step and finally, their spatial registration with a world coordinate frame;
- The operation of the grinding plant has shown that the system is capable of handling required product specifications of axially symmetric steel billets. Thereby, the challenge was to find a proper design for the plane-of-light scanners, which incorporates the desired accuracy as well as the needed measurement range;
- The system can withstand the prevailing harsh environmental conditions in the plant. Typical enclosures of high protection class and a robust construction of the mounting of the optical components and the grinding robot ensure the necessary resistance to vibration and grinding swarf;

- The implementation of several automated grinding workcells with a product database that controls the flow of material to be processed, can maximize the productivity and reduce maintenance downtime cost in this production step;
- Finally, the full functionality of the system concept is demonstrated, since all five grinding facilities, of the same fundamental design, are operating at two different steel production plants. Both operators, *Deutsche Edelstahlwerke GmbH* in *Witten/Germany* and *Mittal Steel Ruhrort GmbH* in *Duisburg/Germany*, are more than satisfied with the quality and productivity of the fully automated grinding facilities.

Outlook

The presented nonlinear concept for the determination of a homography can be used to self-calibrate an apparatus for the automatic localization of an object to enable automatic processing with a robot [79]. A possible application is the deburring of railway tracks, where a plane-of-light scanner is used to determine and orient the position of the rail to guide a robot precisely along the butt-end edge contour to enable its deburring. Therefore, one or more plane-of-light scanners are placed in a position so that the required geometric shape of the workpiece can be measured. The relative position of the robot with respect to the plane-of-light scanner(s) is irrelevant. The calibration proceeds as follows:

1. The robot places a cylinder with known radius so that a minimum of four points, in tetrahedral vertex configuration, are within the view of the camera. The metric-coordinates of the center point of the cylinder are obtained from the robot joint coordinates;
2. The contour points of the intersection curve between the cylinder and the plane-of-light are extracted from the image;
3. The homography between the camera chip and the laser plane is determined using the nonlinear calibration procedure developed herein.

The advantages of this processing in comparison to prior calibration techniques are:

a. No calibration equipment is required for the calibration and the registration of the plane-of-light scanner(s) and the robot;

b. The calibration procedure can be full automated. This enables an automatic re-calibration to avoid inaccuracies of the positioning due to changes in temperature that may cause mechanical deformation of the apparatus;

c. No fitting technique is required to determine the geometric shape, position and/or the orientation of the workpiece in order to guide a deburring robot.

Appendix A

Geometry of the Calibration Equipment

This section contains drawings of equipment for the calibration of the color camera and the plane-of-light scanners. The metric-coordinates of the reference marks with respect to their individual coordinate frames can be derived from the geometry of these components. The following drawings are included:

- A planar calibration panel, equipped with four collinear reference marks;
- A calibration step target, consisting of three parallel planes, each containing eight coplanar calibration marks. The calibration mark located in the center of the upper plane may be used for orientation in the algorithm that assigns correct coordinates to the light sources.

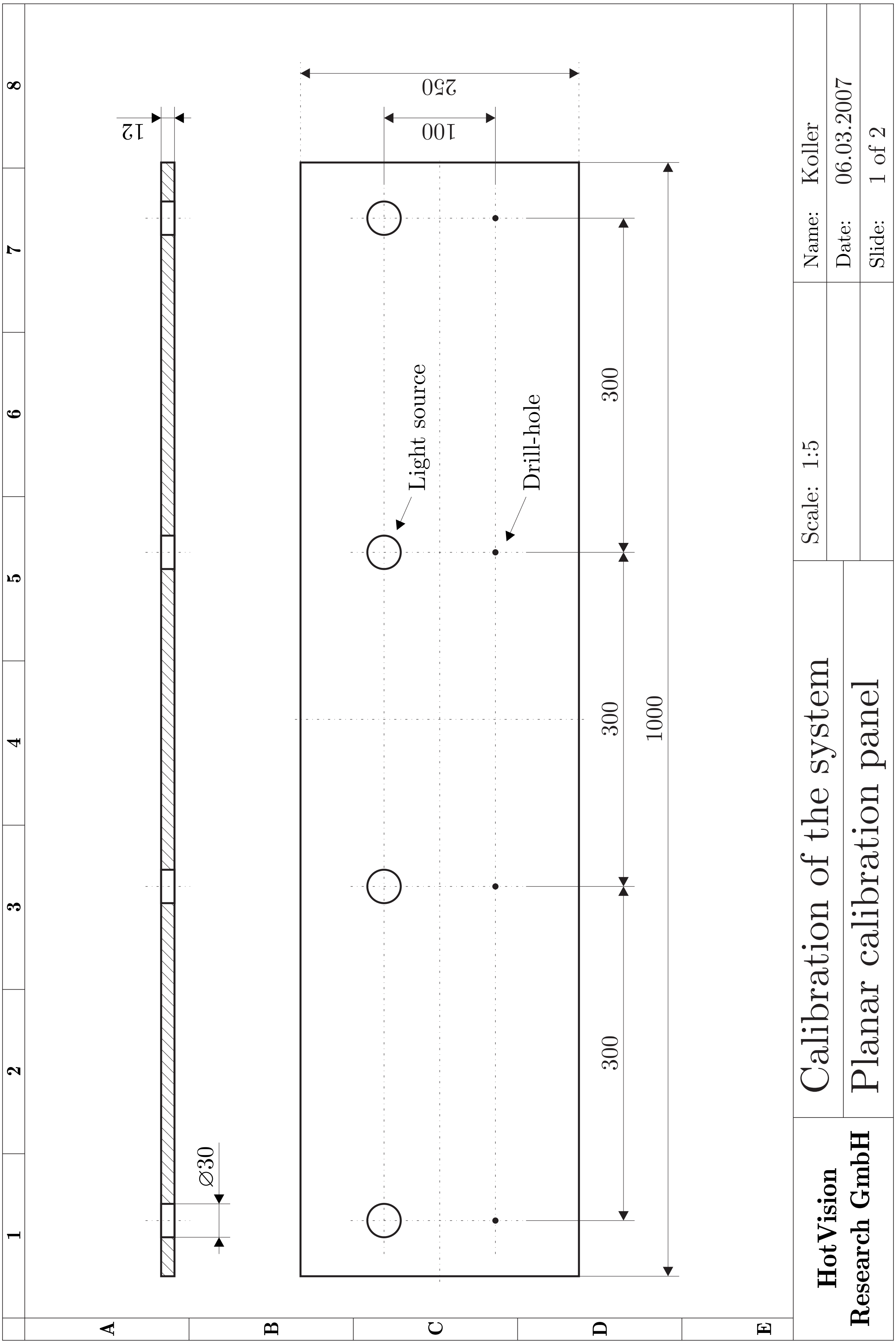


Figure A.1: Geometry of a planar panel for the calibration of the color camera.

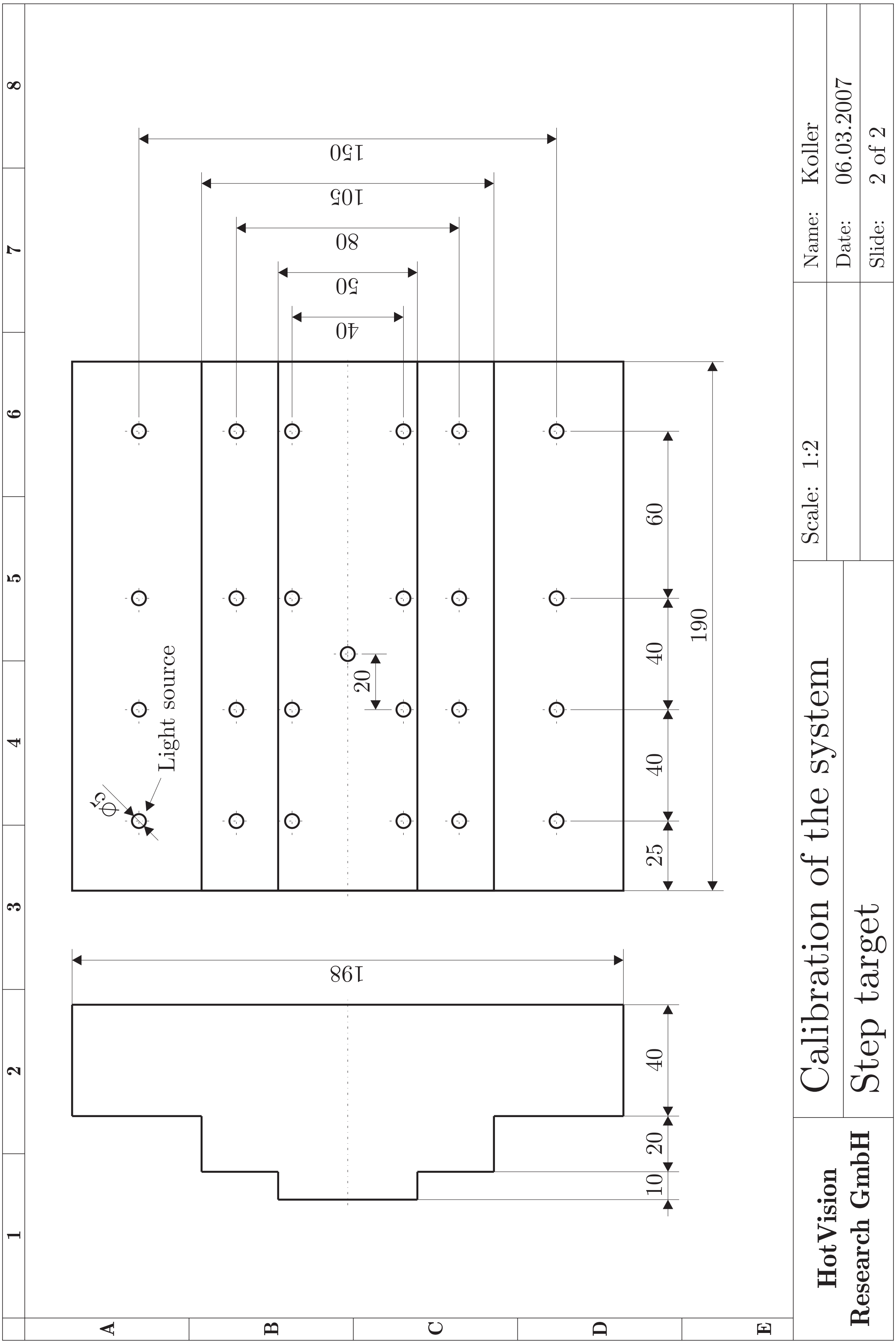


Figure A.2: Geometry of a step target for the calibration of the plane-of-light scanners.

Author's Publications

- Ewald Fauster, Norbert Koller, Ronald Ofner and Paul O'Leary, "Optical Inspection of Steel Bars for Automatic Grinding of Surface Defects", in Proceedings of the XXV. Verformungskundliches Kolloquium, Planner Alm, pp. 5-17, March 2006.
- Norbert Koller, Ronald Ofner, Paul O'Leary and Ewald Fauster, "Optical Servoing for Industrial Surface Machining", in Proceedings of SPIE, Machine Vision Applications in Industrial Inspection XIV, 6070, pp. 1-11, January 2006.
- Norbert Koller, Paul O'Leary and Peter Lee, "Comparison of CMOS and CCD Cameras for Laser Profiling", in Proceedings of SPIE, Machine Vision Applications in Industrial Inspection XII, 5303, pp. 108-115, January 2004.
- Norbert Koller and Ewald Fauster, "Vollautomatische Roboterschleifanlage, Bedienungsanleitung zur Bildverarbeitungsanlage", technical report for EWK, Witten, Germany, March 2007.
- Norbert Koller and Peter Schalk, "Ethercam Control Panel V1.0, User manual", technical report, Institute for Automation, University of Leoben, Leoben, Austria, November 2003.
- Peter Schalk and Norbert Koller, "How to Set Up a Development Environment for the Ethercam Modbus Server Application", technical report, Institute for Automation, University of Leoben, Leoben, Austria, April 2003.
- Norbert Koller, "Vergleich der Dynamik einer CMOS-Kamera und einer CCD-Kamera", technical report, Institute for Automation, University of Leoben, Leoben, Austria, January 2003.

List of Figures

List of Tables

Bibliography

[1] Elkem, "Detection of flaws in metal members", tech. rep., Patent No. Re32166, 1979.

[2] Elkem, "Method of flaw detection in billets", tech. rep., Patent No. US4480928, 1984.

[3] N. Koller, R. Ofner, P. O'Leary and E. Fauster, "Optical Servoing for Industrial Surface Machining", in *Proceedings of SPIE, Machine Vision Applications in Industrial Inspection XIV*, **6017**, pp. 1–11, January 2006.

[4] E. Fauster, N. Koller, R. Ofner and P. O'Leary, "Optical Inspection of Steel Bars for Automatic Grinding of Surface Defects", in *Proceedings of the XXV. Verformungskundliches Kolloquium, Planner Alm*, pp. 5–17, March 2006.

[5] P. Beckmann and A. Spizzochino, *The Scattering of Electromagnetic Waves from Rough Surfaces*, Artech House Publishers, New York, USA, 1963.

[6] K. E. Torrance and E. M. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces", *Journal of the Optical Society of America* **57**(9), pp. 1105–1114, December 1967.

[7] S. K. Nayar, K. Ikeuchi and T. Kanade, "Surface Reflection: Physical and Geometrical Perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(7), pp. 611 – 634, July 1991.

[8] E. M. Sparrow and R. D. Cess, *Radiation* H*eat* T*ransfer*, Series in Thermal and Fluids Engineering, Augmented ed., New York, USA, 1978.

[9] R. L. Cook and K. E. Torrance, "A Reflectance Model for Computer Graphics", *ACM Transactions on Graphics (TOG)* **1**(1), pp. 7–24, January 1982.

[10] R. Ramanath, W. E. Snyder, Y. Yoo and M. S. Drew, "Color Image Processing Pipeline", *IEEE Signal Processing Magazine* **22**(1), pp. 34–43, January 2005.

[11] J. Adams, K. Parulski and K. Spaulding, "Color Processing in Digital Cameras", *IEEE Micro* **18**(6), pp. 20–30, November-December 1998.

[12] B. E. Bayer, "Color Imaging Array", tech. rep., Patent No. US3971065, July 1976.

[13] G. C. Holst, *CCD Arrays, Cameras and Displays*, JCD Publishing, SPIE Optical Engeneering Press, Winter Park, Florida, USA, 1996.

[14] T. W. McCurnin, L. C. Schooley and G. R. Sims, "Charge-Coupled Device Signal Processing Models and Comparisons", *Journal of Electronic Imaging* **2**(2), pp. 100–107, April 1993.

[15] Pulnix, "Technical Note, TM-1020 LUT (Look-Up Table) Function", tech. rep., Jai Pulnix, Inc., San Jose, California, USA, July 15, 2000.

[16] J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics - Principles and Practice*, Addison-Wesley Publishing Company, Inc., Boston, Massachusetts, USA, July 1995.

[17] D. A. Forsyth, "A Novel Algorithm for Color Constancy", *International Journal of Computer Vision* **5**(1), pp. 5–36, August 1990.

[18] G. D. Finlayson, M. S. Drew and B. V. Funt, "Color constancy: generalized diagonal transforms suffice", *Journal of the Optical Society of America A* **11**(11), pp. 3011–3019, November 1994.

[19] J. Laine and H. Saarelma, "Illumination-Based Color Balance Adjustments", *The Proceedings of the IS&T, Eighth Color Imaging Conference* , pp. 202–206, 2000.

[20] K. Barnard, *Practical Color Constancy.* doctoral thesis, Burnaby, Canada, 1999.

[21] R. Ramanath, W. E. Snyder and G. L. Bilbro, "Demosaicking Methods for Bayer Color Arrays", *Journal of Electronic Imaging* **11**, pp. 306–315, July 2002.

[22] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer and R. M. Mersereau, "Demosaicking: Color Filter Array Interpolation", *IEEE Signal Processing Magazine* **22**, pp. 44–54, January 2005.

[23] M. R. Gupta and T. Chen, "Vector Color Filter Array Demosaicing", in *Proceedings of SPIE, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II*, **4306**, pp. 374–382, January 2001.

[24] R. Kimmel, "Demosaicing: Image Reconstruction from Color CCD Samples", *IEEE Transactions on Image Processing* **8**(9), pp. 1221–1228, September 1999.

[25] R. Lukac and K. N. Plataniotis, "Normalization of the Color-Ratio Model for CFA Interpolation", *IEEE Transactions on Consumer Electronics* **50**, pp. 737–745, May 2004.

[26] B. D. Zarit, B. J. Super and F. K. H. Quek, "Comparison of Five Color Models in Skin Pixel Classification", *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Sytems, Corfu, Greece* , pp. 58–63, 1999.

[27] D. Grest, J.-M. Frahm and R. Koch, "A Color Similarity Measure for Robust Shadow Removal in Real-Time", tech. rep., Vision, Modeling and Visualization 2003, Munich, Germany, November 19 - 21, 2003.

[28] J. Lu, Q. Gu and K. N. Plataniotis, "A Comparative Study of Skin-Color Models", tech. rep., International Conference on Image Analysis and Recognition (ICIAR), Toronto, Canada, September 28 - 30, 2005.

[29] J. B. Martinkauppi, M. N. Soriano and M. H. Laaksonen, “Behavior of Skin Color under Varying Illumination seen by different Cameras at different Color Spaces”, in *Proceedings of SPIE, Machine Vision in Industrial Inspection IX*, **4301**, pp. 102–113, January 2001.

[30] F. Naccari, S. Battiato, A. Bruna, A. Capra and A. Castorina, “Natural Scenes Classification for Color Enhancement”, *IEEE Transactions on Consumer Electronics* **51**, pp. 234–239, February 2005.

[31] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, USA, 1993.

[32] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer-Verlag, Berlin, Germany, 2000.

[33] R. Altmann, *Digitale Fotografie und Bildbearbeitung*, MIDAS-Verlag, Zürich, Switzerland, 2003.

[34] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, Brooks/Cole Publishing Company, Pacific Grove, California, USA, 1998.

[35] R. Ofner, *Three-Dimensional Measurement via the Light-Sectioning Method.* doctoral thesis, Leoben, Austria, 2000.

[36] F. Pernkopf and P. O’Leary, “Image acquisition techniques for automatic visual inspection of metallic surfaces”, *NDT&E International* **36**, pp. 609–617, 2003.

[37] P. J. Besl, “Active Optical Range Imaging Sensors”, in *Sanz, Jorge L. C. (Ed.), Advances in Machine Vision*, pp. 1–53, Springer-Verlag, (Heidelberg, Germany), 1989.

[38] M. Johannesson, *SIMD architectures for Range and Radar Imaging.* doctoral thesis, Linköping, Sweden, 1995.

[39] D. Page, A. Koschan, Y. Sun and M. Abidi, “Laser-based Imaging for Reverse Engineering”, *Sensor Review, Special Issue on Machine Vision and Laser Scanners* **23**(3), pp. 223–229, July 2003.

[40] M. Tratnig, *Calibration and Registration Approaches for Light-Sectioning Setups Featuring Small Fields of View.* doctoral thesis, Leoben, Austria, 2005.

[41] AIA, “Camera Link, Specifications of the Camera Link Interface Standard for Digital Cameras and Frame Grabbers, Version 1.1”, tech. rep., Automated Imaging Association, Ann Arbor, Michigan, USA, January 2004.

[42] Basler, “Basler A 500 Series, Product Specifications”, tech. rep., Basler Vision Technologies, Ahrensburg, Germany, October 2003.

[43] IVP, “Ranger C, Technical Description”, tech. rep., Integrated Vision Products, Linköping, Sweden, October 2005.

[44] R. B. Fisher and D. K. Naidu, “A comparison of algorithms for subpixel peak detection”, in *Sanz, Jorge L. C. (Ed.), Image Technology*, Springer-Verlag, (Heidelberg, Germany), 1996.

[45] R. Ofner, P. O'Leary and M. Leitner, "A Collection of Algorithms for the Determination of Construction Points in the Measurement of 3D Geometries via Light-Sectioning", *WESIC, 2nd Workshop on European Scientific and Industrial Collaboration promoting: Advanced Technologies in Manufacturing* , pp. 505–512, 1999.

[46] M. Leitner, R. Ofner and F. Pernkopf, "Comparison of algorithms for line detection in light-sectioning images", tech. rep., Institute for Automation, University of Leoben, Leoben, Austria, 2000.

[47] A. LaCruz, "Accuracy Evaluation of Different Centerline Approximations of Blood Vessels", in *Data Visualization 2004, Visualization Symposium Proceedings*, pp. 115–120, EUROGRAPHICS/IEEE TCVG, May 2004.

[48] A. Patwardhan, "Subpixel position measurement using 1D, 2D and 3D centroid algorithms with emphasis on applications in confocal microscopy", *Journal of Microscopy* **186**(3), pp. 246–257, June 1997.

[49] N. Koller, P. O'Leary and P. Lee, "Comparison of CMOS and CCD Cameras for Laser Profiling", in *Proceedings of SPIE, Machine Vision Applications in Industrial Inspection XII*, **5303**, pp. 108–115, January 2004.

[50] H. C. van Assen, M. Egmont-Petersen and J. H. C. Reiber, "Accurate object localization in gray level images using the center of gravity measure: accuracy versus precision", *IEEE Transactions on Image Processing* **11**(12), pp. 1379–1384, 2002.

[51] W. Gander, G. H. Golub and R. Strebel, "Least-Squares Fitting of Circles and Ellipses", in *Numerical analysis (in honour of Jean Meinguet)*, editorial board Bulletin Belgian Mathematical Society, ed., pp. 63–84, 1996.

[52] P. O'Leary, P. Schalk, R. Ofner and A. Gfrerrer, "Instrumentation and Analysis-Methods for the Measurement of Profiles using Light Sectioning", in *Proceeding of the 23rd IEEE Instrumentation and Measurement Technology Conference*, pp. 1108–1113, April 2006.

[53] P. O'Leary, M. Harker and P. Zsombor-Murray, "Direct and least square fitting of coupled geometric objects for metric vision", in *Vision, Image and Signal Processing, IEEE Proceedings-Publication*, **152**(6), pp. 687–694, December 2005.

[54] E. W. Weisstein, "Moore-Penrose Matrix Inverse." From *MathWorld*–A Wolfram Web Resource, 1999-2007. Available: http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html.

[55] G. H. Golub and C. F. V. Loan, *Matrix Computations*, THE JOHNS HOPKINS UNIVERSITY PRESS, Baltimore, Maryland, USA, 1996.

[56] S. H. Joseph, "Unbiased Least Squares Fitting of Circular Arcs", *CVGIP: Graphical Model and Image Processing* **56**(5), pp. 424–432, 1994.

[57] W. H. Press, S. A. Teukolsky, W. T. Vettering and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, THE CAMBRIDGE UNIVERSITY PRESS, New York, USA, 1988-1992.

[58] P. O'Leary, "Fitting Geometric Models in Image Processing using Grassmann Manifolds", in *Proceedings of SPIE, Machine Vision Applications in Industrial Inspection X*, **4664**, pp. 22–33, January 2002.

[59] A. Gfrerrer, *Einführung in die analytische projektive Geometrie*, Institute for Geometry, Graz University of Technology, Graz, Austria, 2003.

[60] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, Second Edition*, THE CAMBRIDGE UNIVERSITY PRESS, Cambridge, UK, 2003.

[61] R. I. Hartley, "In Defence of the 8-point Algorithm", in *Proceedings of the Fifth International Conference on Computer Vision (ICCV'95)*, pp. 1064–1070, June 1995.

[62] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators, Second Edition*, Springer-Verlag, London, UK, 2000.

[63] E. W. Weisstein, "Euler Angles." From *MathWorld*–A Wolfram Web Resource, 1999-2007. Available: http://mathworld.wolfram.com/EulerAngles.html.

[64] Y. F. Li and S. Y. Chen, "Automatic Recalibration of an Active Structured Light Vision System", *IEEE Transactions on Robotics and Automation* **19**(2), pp. 259–268, April 2003.

[65] Z. Zhang, "A flexible new Technique for Camera Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), pp. 1330–1334, 2000.

[66] A. M. McIvor, "Calibration of a Laser Stripe Profiler", in *Proceedings of IEEE, 2nd International Conference on 3D Imaging and Modeling*, pp. 92–98, 1999.

[67] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation* **3**(4), pp. 323–344, August 1987.

[68] J. Heikkilä and O. Silvén, "A Four-step Camera Calibration Procedure with Implicit Image Correction", in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 1106–1112, 1997.

[69] J. Xu, Z. Fang, A. A. Malcolm and H. Wang, "Camera Calibration for 3-D Measurement with Micron Level Accuracy", in *Proceedings of 5th International Conference on Mechatronics Technology*, pp. 56–61, June 2001.

[70] E. Fauster, P. Schalk and M. Tratnig, "Calibration Method for Light Sectioning Measurement Systems", in *Proceedings of WSEAS, Advances in Multimedia, Video and Signal Processing Systems, WSEAS Press*, pp. 6–11, 2002.

[71] N. Koller, "Profilmessung von glühenden Stäben und Knüppeln", diploma thesis, Institute for Automation, University of Leoben, Leoben, Austria, November 2002.

[72] M. Tratnig, P. O'Leary, H. Hlobil and J. Reisinger, "Camera Calibration, Data Segmentation, and Fitting Approaches for a Visual Edge Inspection System", in *Proceedings of SPIE, Machine Vision Applications in Industrial Inspections XII*, **5303**, pp. 42–51, January 2004.

[73] H.-J. Bartsch, *Mathematische Formeln, 22. Auflage*, VEB Fachbuchverlag Leipzig, Leipzig, Germany, 1989.

[74] I. Bronstein and K. Semendjajew, *Taschenbuch der Mathematik, 24. Auflage*, Teubner Verlagsgesellschaft Leipzig, Leipzig, Germany, 1989.

[75] E. W. Weisstein, "Newton's Method." From *MathWorld*–A Wolfram Web Resource, 1999-2007. Available: http://mathworld.wolfram.com/NewtonsMethod.html.

[76] W. Alt, *Nichtlineare Optimierung - Eine Einführung in Theorie, Verfahren und Anwendungen*, Friedrich Vieweg & Sohn Verlagsgesellschaft, Braunschweig/Wiesbaden, Germany, 2002.

[77] R. R. Shannon and J. C. Wyant, *Applied Optics and Optical Engineering, Vol. XI*, Academic Press, New York, USA, June 1992.

[78] J. Weng, P. Cohen and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(10), pp. 965–980, October 1992.

[79] P. O'Leary, M. Harker and I. Reindl, "Concept for a Patent: Apparatus and Procedure for the Automatic Localization of an Object to Enalble Automatic Processing with a Robot", tech. rep., Institute for Automation, University of Leoben, Leoben, Austria, November 2006.