Development of a system to identify unmarked objects based on 3D-Object recognition and multi-sensor information

by

Marc Ungen

Thesis presented in fulfilment of the requirements for the degree of

Master of Engineering (Engineering Management)

in the Faculty of Engineering at Stellenbosch University

This thesis has also been presented at Reutlingen University in terms of a double-degree agreement

Supervisor: Prof. L. Louw

Co-Supervisor: Prof. D. Palm

March 2021

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third-party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2021

Abstract

Automatic identification (Auto-ID) technology serves as an interface between the real physical world and the virtual world of information by synchronising the physical flow of material with the virtual flow of information. Through this characteristic, Auto-ID technologies represent the core component for the implementation of the Internet of Things (IoT) in the context of Digitalisation and Industry 4.0. The Auto-ID technologies established over the years are almost exclusively based on the use of artificial identifiers for the purpose of identification, with barcodes and radio-frequency identification transponders being the most commonly used ones. In fact, the use of artificial identifiers causes additional effort, additional costs and is not even always applicable. By using the natural features of objects as identifiers, such drawbacks are avoided.

Combining methods of 3D-Object recognition from the field of machine vision (MV) with further multi-sensor information and the data master from product data management, this thesis develops a novel multi-sensor Auto-ID system for the direct identification of unpackaged piece goods. Based on in-depth literature research, requirements for such a system are defined and transformed into a concept using engineering design methods. Subsequently, a first mechatronic prototype of the multi-sensor Auto-ID system is constructed and implemented, which includes both hardware and software development. The verification and functional test of the implemented system is then carried out by checking the prototype against the established requirements and by conducting a practical experiment.

The results, based on the practically implemented prototype, show that reliable automatic identification of unmarked and unpackaged piece goods can be accomplished using multi-sensor information. This Auto-ID system prototype requires no artificial identifiers for the purpose of identification thus avoiding additional efforts, additional costs and issues with application. Furthermore, the use of multi-sensor information improves identification in terms of distinctiveness and accuracy, compared to purely optical instance-level 3D-Object recognition.

This research contributes mainly to the existing scholarship in the field of Auto-ID technology. In particular, theoretical and practical contributions are made to the hitherto little-studied field of direct identification based on natural object features.

Keywords: 3D-Object recognition, Automatic identification (Auto-ID), Digitalisation, Direct identification, Industry 4.0, Internet of Things (IoT), Machine vision (MV), Multi-sensor information, Natural identifiers, Product data management (PDM)

Opsomming

Outomatiese identifikasie (Outo-ID) tegnologie dien as ‘n koppelvlak tussen die werklike fisiese wêreld en die virtuele wêreld van informasie deur die fisiese vloei van materiaal met die virtuele vloei van informasie te sinkroniseer. Deur hierdie eienskap verteenwoordig Outo-ID-tegnologie die kernkomponent vir die implementering van die “Internet van Dinge” (IoT) in die konteks van Digitalisering en Industrie 4.0. Die Outo-ID-tegnologieë wat deur die jare heen gevestig is, is byna uitsluitlik gebaseer op die gebruik van kunsmatige identifiseerders vir identifikasie, waar strepieskodes en radiofrekwensie-identifikasie transponders die algemeenste gebruik word. Trouens, die gebruik van kunsmatige identifiseerders het verskeie nadele wat insluit addisionele moeite sowel as addisionele onkostes. Die tipe indentifiseerders is, in realiteit, ook nie eens altyd van toepassing nie. Deur die natuurlike kenmerke van voorwerpe as identifiseerders te gebruik, is sulke nadele vermy.

Deur 3D-objek herkenning vanuit die veld van masjienvisie (MV) met verdere veelvoudige-sensor-inligting en die data-meester uit produkdata-bestuur te kombineer, kan daar ‘n nuwe veelvoudige-sensor Outo-ID-stelsel vir die direkte identifikasie van onverpakte goedere ontwikkel word. Op grond van in-diepte teoretiese navorsing word die vereistes vir so ‘n stelsel gedefinieer en omgeskep in ‘n konsep met behulp van ingenieursontwerpmetodes. Vervolgens word ‘n eerste megatroniese prototipe van die veelvoudige-sensor Outo-ID-stelsel gebou en geïmplementeer, wat die ontwikkeling van beide harde- en sagteware insluit. Die verifikasie en funksionele toets van die geïmplementeerde stelsel word dan uitgevoer deur die prototipe te toets teen die vasgestelde vereistes. Daarna word dit verder getoets deur ‘n praktiese eksperiment uit te voer.

Die resultate, gebaseer op die prakties geïmplementeerde prototipe, toon dat betroubare outomatiese identifikasie van ongemerkte en onverpakte goedere moontlik is met behulp van veelvoudigei-sensor inligting. Hierdie prototipe van die Outo-ID-stelsel vereis geen kunsmatige identifiseerders vir die identifikasie proses nie, dus vermy dit addisionele arbeid, addisionele koste en probleme met toepassing. In vergelyking met die suiwer optiese 3D-voorwerpherkenning metode bied die gebruik van veelvoudige-sensor-inligting verbeterde identifikasie in terme van onderskeidbaarheid sowel as akkuraatheid.

Hierdie navorsing dra hoofsaaklik by tot bestaande studie in die veld van Outo-ID-tegnologie. Teoretiese en praktiese bydraes word spesifiek gelewer tot die relatiewe onbekende areas van direkte identifikasie op grond van natuurlike objek kenmerke.

Sleutelwoorde: 3D-Objectherkenning, Outomatiese identifikasie (Outo-ID), Digitalisering, Direkte identifikasie, Industrie 4.0, Internet of Things (IoT), Masjienvisie (MV), Veelvoudige-sensor-inligting, Natuurlike identifiseerders, Produkdata-bestuur (PDB)

Acknowledgements

"If we knew what it was we were doing, it would not be called research, would it?"

- Albert Einstein (1879-1955)

According to this maxim, I would like to thank everyone who accompanied, supported and motivated me during the process of writing my master's thesis.

Firstly, I would like to express my sincere gratitude to Prof. Daniel Palm, Prof. Louis Louw and Mr Julian Herrmann for their guidance, helpful suggestions and constructive criticism during the entire research period.

I would also like to thank all my family and friends who have accompanied me throughout this time and have always provided me with spiritual balance and encouragement.

Finally, I would like to thank my fellow students from the DIME programme, who have so generously shared their impressions and thoughts with me, inspiring me to do this research and supporting me in challenging times.

Thank you very much!

The Author

September 2020

Table of contents

List of figures

List of tables

List of acronyms

1D-Code .... One-dimensional code
2D-Code .... Two-dimensional code
3D-Code .... Three-dimensional code
6-DoF .... Six degrees of freedom
ADC .... Analog-to-digital converter
AIS .... Automatic identification systems
Auto-ID .... Automatic identification
BLE .... Bluetooth low energy
Brep .... Boundary representation
CAx .... Computer-aided x
CBSE .... Component-based software engineering
CIM .... Computer-integrated manufacturing
CPS .... Cyber-physical system
CSG .... Constructive solid geometry
CV .... Computer vision
CVFH .... Clustered viewpoint feature histogram
DSR .... Design science research
EBOM .... Engineering bill of materials
EPROM .... Erasable programmable read-only memory
ERP .... Enterprise resource planning
FDM .... Fused deposition modelling
FPFH .... Fast point feature histogram
HDMI .... High definition multimedia interface
HMI .... Human-machine interface
HR .... Hardware requirement
HSF .... Hardware subfunction
HSP .... Hardware solution principle
HSVA .... Hardware solution variant
ICP .... Iterative closest point
ID .... Identification number
IDE .... Integrated development environment

IoT .... Internet of things

JSON .... Java Script Object Notation

k-d trees .... K-dimensional trees

LED .... Light-emitting diode

LIDAR .... Light detection and ranging

LSA .... Laser surface authentication

MES .... Manufacturing execution system

MV .... Machine vision

NFC .... Near-field communication

NNS .... Nearest neighbour search

OCR .... Optical character recognition

PC .... Personal computer

PCL .... Point cloud library

PDM .... Product data management

PLC .... Programmable logic controller

PLM .... Product lifecycle management

PLY .... Polygon file format

PRO .... Primary research objective

PROM .... Programmable read-only memory

PRQ .... Primary research question

RAM .... Random access memory

RFID .... Radio-frequency identification

RGB .... Red green blue

SCADA .... Supervisory control and data acquisition

SDC .... Shape distribution component

SHOT .... Signature of histogram of orientation

SR .... Software requirement

SRQ .... Subordinate research questions

SSF .... Software subfunctions

STEP .... Standard for the exchange of product model data

STL .... Standard tessellation language

ToF .... Time-of-flight

UID .... Unique identifier

UML .... Unified modelling language

USB .... Universal serial bus

UWB .... Ultra-wideband

VFH .... Viewpoint feature histogram

Nomenclature

$\boldsymbol{a, b, c, d, l}$	*Distance or length*
$\boldsymbol{c_0}$	*Propagation velocity of light*
$\boldsymbol{C}$	*Centre of mass (Point)*
$\boldsymbol{d_{L2}}$	*Euclidean distance*
$\boldsymbol{d_{\chi^2}}$	*Chi − squared distance*
$\boldsymbol{D}$	*Distance (const.)*
$\boldsymbol{f}$	*Focal length*
$\boldsymbol{f_n}$	*Identification feature (Object)*
$\boldsymbol{F}$	*Force*
$\boldsymbol{F_k}$	*Identification feature (Knowledge base)*
$\boldsymbol{g}$	*Gravitational acceleration*
$\boldsymbol{i, j, k, m, n}$	*Variables*
$\boldsymbol{I}$	*Electric current*
$\boldsymbol{I_l}$	*Identity information*
$\boldsymbol{K}$	*Set of pairs (Knowledge base)*
$\boldsymbol{O}$	*Origin or observing point*
$\boldsymbol{p}$	*Image point (Pixel)*
$\boldsymbol{P, S}$	*Space point*
$\boldsymbol{P_j}$	*Pair (Object description in knowledge base)*
$\boldsymbol{R}$	*Electrical resistance*
$\boldsymbol{S_f}$	*Set (Identification features of object)*
$\boldsymbol{S_F}$	*Set (Identification features in knowledge base)*
$\boldsymbol{S_I}$	*Set (Identity information in knowledge base)*
$\boldsymbol{t}$	*Time period*
$\boldsymbol{T}$	*Distance (const.)*
$\boldsymbol{U}$	*Electrical tension*
$\boldsymbol{x, y, z}$	*Space coordinates*
$\boldsymbol{\alpha, \beta, \gamma}$	*Angles*
$\boldsymbol{Z}$	*Spatial depth*

1 Introduction

The purpose of this chapter is to provide insight into the research undertaken. The chapter commences with a brief background of the trends and demands placed on the industry in the course of digitalisation, followed by an explanation of the research problem, the formulation of the research questions and the research objectives. Then the scope of the work is delineated together with the research design and methodology followed to solve the identified problem. The chapter concludes with the design or roadmap of the study.

1.1 Background and rationale of the research

Digitalisation and Industry 4.0 are far-reaching fields of action with high relevance for society, education and the economy in all areas of life worldwide. The core ideas of Digitalisation and Industry 4.0 are provided by research and development activities in the natural and engineering sciences, which use and link anew the latest technologies in the information, communication and automation sectors. (Deckert 2019, pp. 1–6)

One field of action to which particular importance is attributed within Digitalisation and Industry 4.0 is the interconnection of the real and the virtual world. The concept of linking physical real-world things with the virtual world of information has been widely referred to as the Internet of Things (IoT) (Uckelmann et al. 2011, p. 2). The Internet of Things utilises interfaces to the real world, which on the one hand provide data from the environment, and on the other hand also allow the environment to be influenced (Milenkovic 2020, pp. 1–5). For some time now, real-time positioning technology, sensor technology, actuator technology and automatic identification technology (Auto-ID) have been regarded as the key technologies for this merger (Koch and Deiters 2007, pp. 193–195; Lampe and Flörkemeier 2005, pp. 69–70; Uckelmann et al. 2011, pp. 2–3; Milenkovic 2020, pp. 280–282).

Probably the most significant contribution to the fusion of the physical and virtual world according to the concept of IoT is provided by the aforementioned Auto-ID technology. This is also evident from the fact that the term Internet of Things was introduced by the founders of the original MIT Auto-ID Center (Santucci 2010). Explicitly, Auto-ID technology acts as an interface between the real physical world and the virtual world of information, synchronising the physical flow of material and the virtual flow of information (Arnold et al. 2008, p. 816). The field of Auto-ID consists of a variety of individual technologies, of which the best-known representatives are barcode identification and radio-frequency identification (RFID).

Sharing over 70 % of the market, the barcode is still the most widely used technology for identification today (Ten Hompel et al. 2008, p. 9; VFC Research 2018). With its invention and subsequent patent in 1952 (U.S. patent number: US2612994A), the barcode looks back on a long tradition and is therefore considered a very well-established technology. However, it must be mentioned here that there have been numerous further developments of the classical one-dimensionally arranged bar pattern to this day (see Figure 1.1). For this reason, this technology is also often referred to as optical or visual code identification technology (Hippenmeyer and Moosmann 2016, pp. 22–23). For identification purposes, optical codes must be attached to the objects to be identified, which is referred to as marking. Two basic marking methods are distinguished (Ten Hompel and Schmidt 2007, p. 197). Direct marking, also known as direct labelling, describes the application or embossing of information on the surface of an identification object. Indirect marking, also referred to as indirect labelling, describes the application of artificial information carriers to an identification object. In the industrial environment, both direct and indirect markings are applied to identify objects using various techniques (see Table 1.1) through additional production steps. This results in costs for the necessary machines, working time, label materials and supplies (Hippenmeyer and Moosmann 2016, pp. 27–31; Ten Hompel et al. 2008, pp. 95–103). Likewise, optical codes need to be a minimum size and a minimum distance from each other in order to ensure readability and they can only be applied to suitable (flat) surfaces, which must be considered in advance relating to the object to be identified (Ten Hompel et al. 2008, pp. 99–100). The geometry of an object may therefore have to be adapted purely for the purpose of identification. There are applications in the identification of components where optical codes cannot be used for this reason (Dragon et al. 2011, p. 276). Furthermore, visual codes are rendered useless or are even destroyed by certain production steps, such as painting processes or heat treatment processes (Kropik 2009, p. 109). In order to keep objects in the material flow identifiable, visual codes must therefore be removed and reapplied by means of further additional work steps.

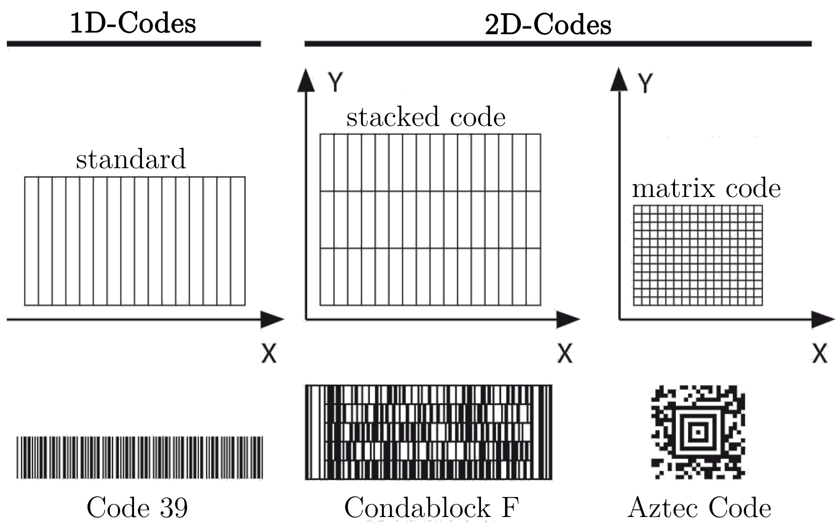


Figure 1.1: Further developments of the standard barcode, adapted from (Finkenzeller 2015, p. 3)

Table 1.1: Direct and indirect labelling techniques for optical codes, summarised from (Ten Hompel and Schmidt 2007, p. 197; Hippenmeyer and Moosmann 2016, pp. 27–31)

Direct labelling techniques	Indirect labelling techniques
– Direct ink-jet printing	– Matrix printing
– Direct laser marking	– Ink-jet printing / Laser printing
– Engraving	– Laser marking
– Embossing / Dot peen marking	– Thermo transfer/ Thermo direct printing
	– Photosetting / Offset printing
	– Screen printing

The RFID technology mentioned above, which is currently gaining acceptance, is attributed a significant role in the implementation of IoT (Uckelmann et al. 2011, p. 2; Japs 2007, pp. 1–2). With RFID technology, the information required for identification is stored electronically on data carriers known as transponders. These transponders have an antenna and a semiconductor memory chip for reading and writing data. Two main types of transponders exist in this context, distinguished by their mode of energy supply (Finkenzeller 2015, pp. 25–27). Passive transponders (see Figure 1.2) have no energy supply of their own but receive the energy necessary for operation only through the magnetic or electromagnetic field of their reading devices. Active transponders, on the other hand, have their own energy supply (e.g. battery), making them independent of any external energy supply. There are a variety of housing designs of RFID transponders such as: glass housing, plastic housing, housings for surface mounting, chip cards and adhesive labels (Finkenzeller 2015, pp. 16–24). In order to attach the transponders to identification objects, they have to be attached in a friction-locked, form-locking or material-locking manner, which also requires additional work steps and thus incurs additional costs. With RFID transponders, an attachment area must also be provided on the object to be identified, although this no longer requires a significant size compared to the optical codes. However, the functional range depends largely on the size of the antenna and thus the size of the transponder (Finkenzeller 2015, p. 18). The use of RFID transponders on or in metal surfaces is also problematic and can lead to a complete failure of the connection between reader and transponder (Hippenmeyer and Moosmann 2016, pp. 24–25). This is another shortcoming of RFID technology, as in case of a malfunction this technology does not offer any emergency strategy, which is why RFID is mostly used in combination with optical codes (Kropik 2009, p. 111). Furthermore, this implies a costly necessity for both an optical code and an RFID infrastructure. The required infrastructure in the form of hardware for reading and writing the less standardised RFID transponders represents a further cost factor that should not be ignored. Such systems require precise planning due to frequency and radio link overlay (Kropik 2009, pp. 110–112). Despite the good protection offered by RFID transponders through their housings, the semiconductor components are susceptible to high temperatures. Therefore, issues arise in production processes that require this property.

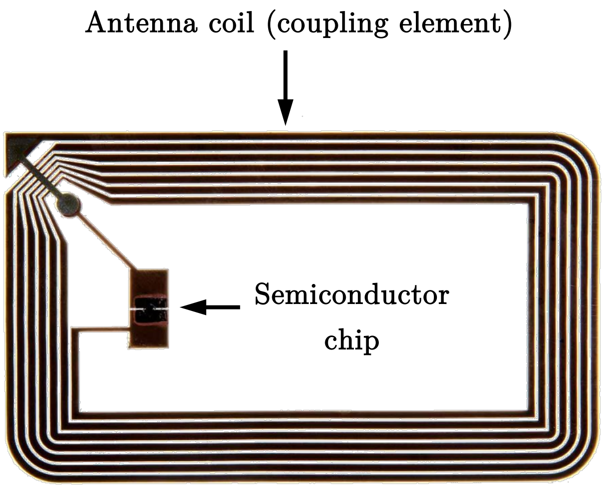


Figure 1.2: Technical design of a passive RFID transponder (Size: 30mm x 50 mm)

Two further subject areas that deal with the recognition and identification of objects are computer vision (CV) and machine vision (MV) respectively (Ahola et al. 2016, p. 1). While CV is more dedicated to the informatics perspective of artificial vision, MV is a branch of systems engineering and applies the methods of CV to technical systems in an industrial context. Vision-based identification systems offer great potential in the environment of Digitalisation and Industry 4.0, with the identification of individual components being of particular importance (Hippenmeyer and Moosmann 2016, p. 34). Vision-based approaches are already widely used in industry to capture optical codes as representatives of artificial identifiers (Ten Hompel et al. 2008, p. 77). However, a recent research approach is to use natural object features in the form of surface structures as a basis for identification. Grinding imprints (Frauenhofer IPM 2017; Breidenstein et al. 2016; Dragon et al. 2011) and the surface structure of paper fibres (Buchanan et al. 2005; Cowburn 2008) are explicitly used for this direct identification based on natural object features so far. Natural object features will gain in importance in the future of identification, since they are available for capturing and analysis without any additional effort and costs (Hippenmeyer and Moosmann 2016, p. 12; Frauenhofer IPM 2017). In particular, the camera-based identification of individual components is regarded as a key technology for the implementation of the Internet of Things, since up to now only load carriers have been identified and the actual object identification has been rather neglected for technical reasons (Hippenmeyer and Moosmann 2016, p. 34). One of these technical reasons is that object recognition from 2D-Image data cannot meet the requirements of practical applications (Dong et al. 2019, p. 243). At the same time, depth perception sensors are becoming increasingly popular and affordable, which in combination with newly developed object recognition algorithms offers great potential for component identification (Mateo et al. 2014, p. 428).

1.2 Research problem statement and questions

Against the background described in the previous section, the Research Problem Statement (RPS) of this thesis is formulated as shown in Table 1.2.

Table 1.2: Research problem statement

RPS:	Identification of objects by artificial identifiers within industrial material flow generates costly additional efforts and is not always applicable.

Additional efforts are:

- Additional process steps to attach artificial identifiers to objects
- Additional process steps to remove artificial identifiers from objects for further processes (e.g. before painting or coating)
- Maintenance of artificial identifiers (e.g. replacement of batteries on active RFID-Transponder)
- Efforts caused due to missing or destroyed artificial identifier (e.g. unidentifiable object due to defective RFID-tag)
- Efforts for integration of artificial identifiers on objects (e.g. mounting hole for RFID-Transponder

Issues with the application are:

- Application is not possible due to object's geometry (e.g. Functional surfaces)
- Lack of an emergency strategy, especially for RFID
- Application is not possible because of object's production steps (e.g. Painting, Hardening processes)

It is reasonable to assume that the rejection of artificial identifiers for the identification of industrial objects would counteract the problems that have been raised. This approach is followed by state-of-the-art CV and MV systems for object recognition, which use different geometric features of objects in the identification process. Known systems are processing either 2D-image data or 3D-depth image data, which are optically acquired by means of suitable sensor technology. However, the exclusive use of optical sensors limits the perception of such an identification system to optically recognisable and distinguishable object properties and thus the potential identification tasks.

This thesis therefore pursues the approach of expanding the perceptive capacity of optical recognition systems by using further sensors recording the natural features of components. It will investigate how this multi-sensor information can lead to an identification system which no longer requires artificial identification features. The primary research question (PRQ) with which this research is concerned can therefore be derived as shown in Table 1.3. In order to underline the primary research question and to give the research more structure, additional subordinate research questions (SRQ) are also asked. Table 1.4 provides an overview of these SRQs.

Table 1.3: Primary research question

PRQ:	How can an automated system for direct identification of industrial objects be accomplished using multi-sensor information instead of artificial identifiers?

Table 1.4: Subordinate research questions

SRQ 1:	What is the state of the art in automatic identification technology, machine vision and product data management (PDM) in industry and can a system using multi-sensor information for direct identification be built on it?
SRQ 2:	How must an automated identification system be designed to allow direct identification based on multi-sensor information?
SRQ 3:	What does a practical implementation of an automated multi-sensor system for direct identification look like?
SRQ 4:	What are the effects of using multi-sensor information in addition to optical 3D-Object recognition for identification purposes?

1.3 Research objectives

In order to answer the research questions raised and thus ultimately solve the research problem, this paper pursues the primary research objective (PRO) presented in Table 1.5. Furthermore, a sequence of subordinate research objectives is intended to guide the implementation of the research, thus enabling a systematic approach and the achievement of the PRO. This is shown in Table 1.6.

Table 1.5: Primary research objective

PRO:	Development of an automated system that is capable of directly identifying objects by matching information from CAD-Data with multi-sensor information.

Table 1.6: Subordinate research objectives

SRO I:	Identify the current state of the art in the related fields of research
SRO II:	Develop a conceptual design of multi-sensor AIS for direct identification
SRO III:	Develop a prototype of the designed multi-sensor AIS
SRO IV:	Implement the developed prototype of the multi-sensor AIS
SRO V:	Examine the effects of using multi-sensor information for the purpose of identification

1.4 Research design and methodology

In order to conduct research systematically, a decision has to be made on the steps to be taken from broad assumptions to detailed methods of data collection, analysis and interpretation. The plan or proposal to conduct research is generally referred to as a research approach. The selection of such an approach depends largely on the nature of the research problem, the personal experience of the researcher and the audiences for the study. (Creswell and Creswell 2018, p. 40)

Essentially, three fundamental approaches can be distinguished when conducting a research project: qualitative, quantitative and mixed methods. Qualitative research is generally characterised by inductive approaches to knowledge building and aims to explore, robustly investigate and learn. Quantitative research is characterised by deductive approaches in the research process, which aim to prove, disprove or give credibility to existing theories. This type of research measures variables and tests relationships between them to reveal patterns and correlations. In research using mixed methods, both quantitative and qualitative data are collected, analysed and integrated. The phases of a research project are integrated or synergistic, where the quantitative phase influences the qualitative phase or vice versa. The integration of quantitative and qualitative data leads to a comprehensive understanding of the phenomenon under investigation (Leavy 2017, pp. 8–10).

According to (Creswell and Creswell 2018) there are three components that interact for a broad research approach (see Figure 1.3). The intersection of philosophical worldviews, research designs and research methods form the overall approach.

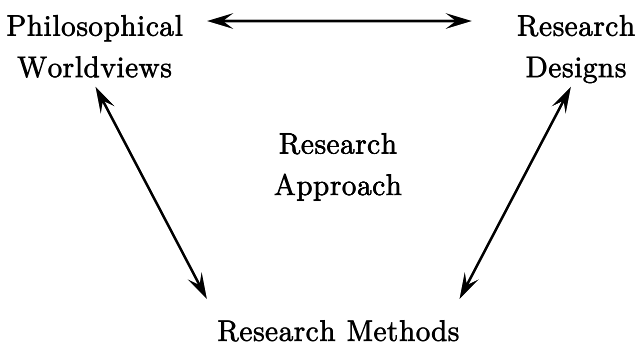


Figure 1.3: Framework for research (Creswell and Creswell 2018, p. 43)

The philosophical worldview describes the general philosophical orientation about the world and the nature of research that a researcher follows. Philosophical ideas remain largely hidden in research, but they still influence the practice of research (Creswell and Creswell 2018, p. 44).

Research designs are the strategies of inquiry within qualitative, quantitative, and mixed-methods approaches that provide specific direction for procedures in a research study. There are also qualitative, quantitative and mixed-methods types of designs (Creswell and Creswell 2018, p. 49)

Research methods are the specific forms of data collection, analysis and interpretation pursued within the approach. The methods can also be qualitative, quantitative or mixed, depending on the data collected, analysed or interpreted. (Creswell and Creswell 2018, p. 53)

This thesis makes use of a pragmatic world view. Pragmatism is not committed to any one system of philosophy or reality, thus allowing the researcher freedom of choice. Multiple methods, different world views, different assumptions as well as different ways of data collection and analysis are enabled using this concept. Pragmatism thus focuses on applications and problem solutions and does not simply follow a black and white approach. This possibility of using all available methods to solve a problem explains why this world view fits very well with a mixed-method approach.

Concerning the research design for this thesis a type of study and a strategy of inquiry have to be selected. Since the research questions are of an exploratory and descriptive nature, a mixed-methods design is used. On the one hand, exploratory research questions require qualitative research in order to develop a deep understanding of the factors under investigation. On the other hand, descriptive research questions can be better answered by quantitative research. Thus, an explorative sequential strategy of inquiry is applied, consisting of a qualitative phase followed by a quantitative phase. The overall research design can therefore be described as a mixed-methods exploratory sequential design.

The selection of explicit research methods for this thesis is based on the methodology of design science research (DSR). The DSR concept aims to develop and design solutions to improve existing systems, solve problems or even create new artefacts (Dresch et al. 2015, p. 56). The problem solution does not have to be an optimal solution right from the start; rather an iterative procedure is used to further optimise the solution. The general methodology of DSR consists of five fundamental phases, which are applied in a so-called design cycle. Figure 1.4 illustrates these phases (bold) and the corresponding knowledge flows (italics) combined in the design cycle.

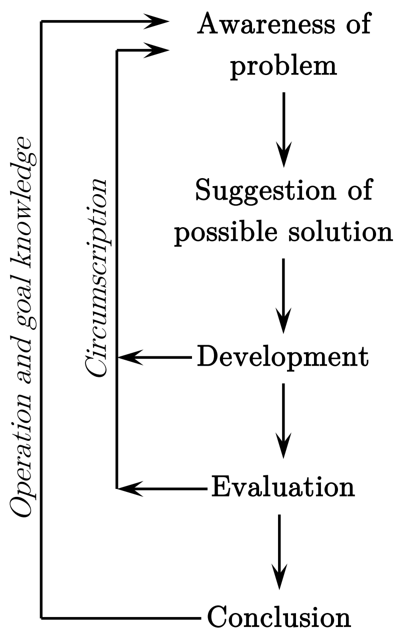


Figure 1.4: General methodology of DSR, adapted from (Vaishnavi and Kuechler 2008, p. 20)

Within the first phase an awareness of a problem is developed. This involves identifying and defining a research problem, which is expressed in the form of a proposal for a new research effort. In the second phase, a preliminary suggestion for a problem solution is drawn abductively from existing knowledge, the theory based on the problem area or a suitable research methodology. The result of this essential creative phase is reflected in a tentative design that envisages a new functionality based on a novel configuration of either existing or new and existing elements. The third phase includes the further development and implementation of the tentative design, in the form of an artefact. Explicit methods are selected and applied according to the type of artefact and the necessary development step. After successful construction, the fourth phase involves the evaluation of the artefact. The qualitative and quantitative criteria for evaluation are established during the phase by hypotheses about the behaviour of the artefact and may also emerge from the first phase. Through the development and evaluation and the associated gain in insight, the knowledge and thus the awareness of the problem is strengthened. This circular flow of knowledge presents itself as a circumscription. The conclusion as the last phase completes the research. The result and the knowledge gained is documented, which may have arisen from a single or multiple recursions of the cycle. (Vaishnavi and Kuechler 2008, pp. 19–22)

Within this work, the DSR methodology is applied in the form of a procedure to successively answer the research questions and research objectives. Figure 1.5 illustrates the steps of the procedure, as well as the questions and objectives addressed. The first two steps serve to create awareness of the problem and thus describe the first phase of the DSR methodology. Step three suggests a possible solution to the problem and can therefore be assigned to the second phase of the design cycle. Steps four and five of the research procedure correspond to the third phase of the DSR methodology and create a prototype of the conceptually designed artefact. The sixth of the research procedure relates to the fourth phase of the DSR methodology and aims to evaluate the prototype. The last procedure step refers to the last phase of the DSR methodology and concludes the research with a conclusion.

Procedure Steps		Questions addressed	Objectives addressed
1	Problem, Questions, Objectives		
2	Literature review	SRQ 1	SRO I
3	Conceptual design of multi-sensor AIS	SRQ 2	SRO II
4	Development of prototype		SRO III
5	Implementation of prototype	SRQ 3	SRO IV
6	Verification and experiment	SRQ 4	SRO V
7	Conclusion	PRQ	

Figure 1.5: Explicit research procedure for thesis

1.5 Delimitations and limitations

Since this thesis is located in an area of research that has been little explored or not explored at all so far, it is of particular importance to point out the delimitations and to disclose the limitations. Delimitations in this context are the explicit boundaries that are set for this research.

Limitations, on the other hand, are the conditions that affect the study from outside and thus cannot be influenced.

This thesis is delimited to the development of a system for direct identification of individual objects. Optical recognition methods from the research field of MV are used as a basis for this. The complex aspects within the field of MV, such as imperfect illumination, noise, clutter and partial image details are considered during development, but cannot be fully addressed due to the scope of this work.

Aspects that cannot be influenced are the available methods of machine vision as well as the sensor technology for capturing the natural identification features. The qualitative data collection is carried out by means of a literature review and is therefore confronted with the limitations associated with this approach. Due to the novel field of research there are no prior research studies on this topic. Thus, this thesis is also limited with regard to the available and the reliability of the existing data.

1.6 Thesis outline

- **Chapter 1** introduces the research. On the basis of the theoretical background the problem definition and the associated research questions are presented. Subsequently, the research objectives are specified and the research design and methodology followed are discussed. After stating the delimitations and limitations, the chapter concludes with the presentation of the thesis outline.
- **Chapter 2** provides a comprehensive literature review of all the scholarship relevant to the research. Three predominant fields are studied: Automatic identification technology, Machine vision and Product data management. At the end of the chapter an interim conclusion is drawn and the findings are discussed.
- **Chapter 3** describes the development process of the multi-sensor automatic identification system for direct identification of piece goods. On the basis of the requirements analysis for such a system, an overall concept consisting of hardware and software components is developed and proposed.
- **Chapter 4** comprises the aspects of implementing the multi-sensor AIS prototype. On the basis of the concept developed in the previous chapter, both the hardware and the software components are implemented in order to obtain a functioning prototype of the proposed system.
- **Chapter 5** encompasses the verification of the developed and implemented prototype. The verification is carried out on the one hand against the criteria laid down in Chapter 3 and on the other hand through a practical experiment. The experiment examines the functionality of the prototype and verifies the advantages of using multi-sensor information compared to purely optical recognition for identification purposes.
- **Chapter 6** commences with a brief and concise summary of the research, followed by a statement of the theoretical and practical research contributions of this work. The chapter as well as this thesis ends with the conclusion and the recommendation for subsequent research.

2 Literature review

The aim of this chapter is to review the existing knowledge in relation to the problem statement and the objectives of the research. As the methodology described in Section 1.4 indicates, the purpose of this chapter is to answer SRQ 1 and achieve SRO I. As the background outlined in Section 1.1 and Figure 2.1 indicates, the research problem lies within the synergy of Auto-ID Technology, MV and PDM. Below is a brief outline of the methods used for this literature review and the reasons for their selection. Sections 2.1, 2.2 and 2.3 describe the current leading literature and research for the respective fields of investigation and provide the basis for the further proceedings within this thesis. A systematic literature review is carried out in Section 2.2.2 to examine all relevant evidence regarding 3D-Object recognition based on CAD-Data and to select a suitable approach for this thesis. Section 2.4 concludes this chapter and critically discusses the findings in literature.

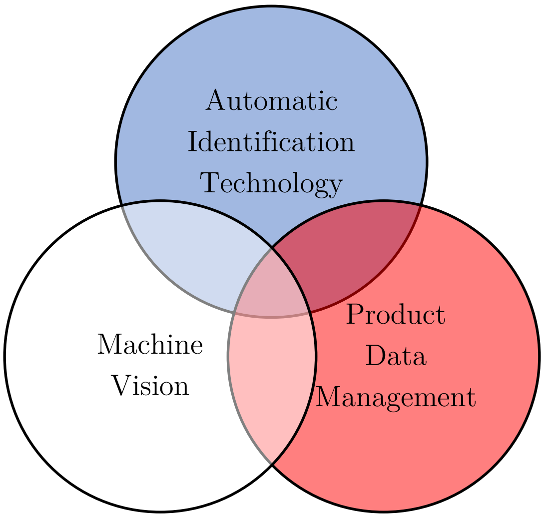


Figure 2.1: Relevant fields of research in the literature concerning the problem

Literature reviews can address research questions with a power that no single study has by integrating findings and perspectives from many empirical findings. A well-conducted review as a research method creates a firm foundation for advancing knowledge and facilitating theory development. Literature reviews can be differentiated into three broad types, including the systematic review, the semi-systematic review and the integrative review. Within the different types a decision must be made based on the nature of the research question and the purpose of the research. (Snyder 2019, pp. 333–334)

Within this thesis a semi-systematic approach of literature review is applied to each of the mentioned fields of investigation. Semi-systematic literature reviews are a traditional method of obtaining an overview of complex research areas. This kind of approach is often used to describe and discuss the leading viewpoints in a comprehensive, critical and objective manner. (Snyder 2019, p. 335)

Additionally, a systematic literature review is conducted within the field of machine vision in order to identify and summarise the latest conditions and explicit methods of CAD-based object recognition. Systematic literature reviews serve to identify, evaluate and interpret all available research relevant to a research question, topic area, or phenomena of interest (Kitchenham 2007, p. 11). Due to the rapidly evolving nature and the huge diversity of applications within the field of MV, this kind of approach enables the specific information of interest for this research to be extracted.

The databases or search engines used for the literature review are shown in Table 2.1

Table 2.1: Databases used for literature review

Database / Search Engine	Hyperlink
Elsevier Scopus	www.scopus.com
IEEE Xplore	www.ieeexplore.ieee.org
Google Scholar	www.scholar.google.com

2.1 Automatic identification technology in industry

Within the automation of piece good processes, automatic identification has a long tradition (Müller 2018, pp. 27–30). The invention and patenting of the barcode in 1950's and the subsequent breakthrough in its mass application in the 1970's, achieved a milestone in the automation of identification, which still dominates logistics today. Automatic identification of logistical objects is an elementary prerequisite for automated logistical processes and over the years many different technologies have been developed to cope with this. The term "Automatic Identification Technology", therefore describes a collection of technologies that are used to automatically identify objects within industrial material flow systems (Wannenwetsch 2014, p. 214). These technologies are used to fully or partially automate processes (Hippenmeyer and Moosmann 2016, p. 33).

Technical systems that are embedded in material flow systems and use Auto-ID technologies are called Automatic identification systems (AIS) and serve to make information available in an up-to-date and object-related manner. In order to establish the reference to the object, it is necessary on the one hand to identify the object directly or indirectly and on the other hand to correctly allocate the data required for informational purposes (Arnold and Furmans 2019, p. 355). Auto-ID technology and also Auto-ID systems hence serve as a linking element between information flow and material flow in logistical material flow systems (Arnold et al. 2008, p. 816). This linking of material flow and information flow makes it possible to manage, monitor and control object-related processes in logistics, which are a basic component of automatically operating systems and enables efficient process design (Ten Hompel et al. 2008, p. 9).

Material flow is the phenomenon of discrete objects moving at regular or irregular intervals along transport routes or conveyor lines. Within material flow science, the totality of all material flows is referred to as the material flow system. All essential technical processes that occur in the material flow can be categorised under the following generic terms: processing, assembling, testing, handling, conveying, transporting, storing, buffering, collecting, distributing, sorting, packing. (Arnold and Furmans 2019, p. 1)

An information flow describes the chain of all processes involved in obtaining, editing, processing as well as in distributing information in systems. The term originates from the field of information logistics, which refers to the scientific teaching of planning, control and monitoring of the information flow in control and information systems (Krämer 2002, p. 51).

In fact, Auto-ID systems fulfil four classes of applications within the processes of industrial material flow systems; which are authentication, item tracking, process effectiveness, and information management applications. Authentication applications are applications in which an accurate identification of an object is required and therefore a verification or an authenticity check of a claimed property of an entity is performed. Item tracking applications gather knowledge about the location and route as well as the state of logistical objects. Applications where AIS are used to reduce data entry times and data entry errors in processes aiming to create more efficient processes are referred to as process effectiveness applications. Information management applications are applications where the main purpose is to access information about logistical objects using the identifiers attached to these objects (Kärkkäinen and Ala-Risku 2003, pp. 1–3).

Providing real and current data from the material flow in a company or production networks, automatic identification systems are data acquisition systems that operate within information networks. Such systems are the basis of information technology, or, more precisely, of information logistics and industrial automation technology (Krämer 2002, p. 77).

Automatic identification technology and its embodiments in the form of automatic identification systems thus aim to (Jünemann and Beyer 1998, p. 91):

- Synchronise the material flow and information flow
- Flexibilise planning and operative processes
- Improve process performance, quality, reliability and ergonomics
- Avoid error-prone manual entries
- Increase transparency of processes

2.1.1 Automation and data acquisition architecture in industry

Today's manufacturing systems in industry are largely characterised by the hierarchical model of the automation pyramid (Forstner and Dümmler 2014, p. 199). In the course of increasing automation, the automation pyramid aims to reduce the complexity of data acquisition and processing in these systems by dividing the processes into individual levels. The resulting visual representation of industrial automation is easy to understand and shows the use of technologies and their limits. Current forms of representation of the pyramid are based on the approaches of Computer-Integrated Manufacturing (CIM) from the 1970s and the so-called "CIM-Pyramid" (Siepmann and Graef 2016, p. 49). Since then, levels have been added, removed or merged and different formulations for this automation model have been developed (Meudt et al. 2017, p. 2).

The six-level automation pyramid according to Siepmann and Graef (see Figure 2.2) represents a very detailed formulation and follows a classical approach (Meudt et al. 2017, p. 5). The different levels are explained below (Siepmann and Graef 2016, pp. 49–50):

- Level 0 (Process Level): The base of the pyramid is formed by the various industrial production processes. This level provides information on product properties and production steps.
- Level 1 (Field Level): The field level, or shop floor level, describes the manufacturing area or site and thus the location of value creation. Sensors and actuators, so-called field devices, are an integral part of this level and provide process data from the information of the process level in the form of output and input signals.
- Level 2 (Control Level): At the control level, the input signals (e.g. sensor data) from the field level are evaluated and converted algorithmically into output signals (e.g. actuator movement data). The output signals are then sent back to the field level, which then physically affects the process level. Technical systems used at this level are called programmable logic controllers (PLC).
- Level 3 (Supervisory Level): Production-relevant processes are visualized and monitored within the supervisory level. Production-relevant processes can be controlled and monitored via process control, human-machine interface (HMI) and supervisory control and data acquisition (SCADA) systems. As an operating and monitoring system, this level influences all subordinate levels.
- Level 4 (Operational Level): Production is managed, controlled and monitored by the operative level. Manufacturing execution systems (MES) are used for detailed production planning and data acquisition. This level serves as a link between the management level and the subordinate, more production-related levels. Production data is sent to the management level for planning future production.
- Level 5 (Management Level): The top of the pyramid represents the management level. At this level, the rough production planning and order processing takes place using enterprise resource planning systems (ERP).

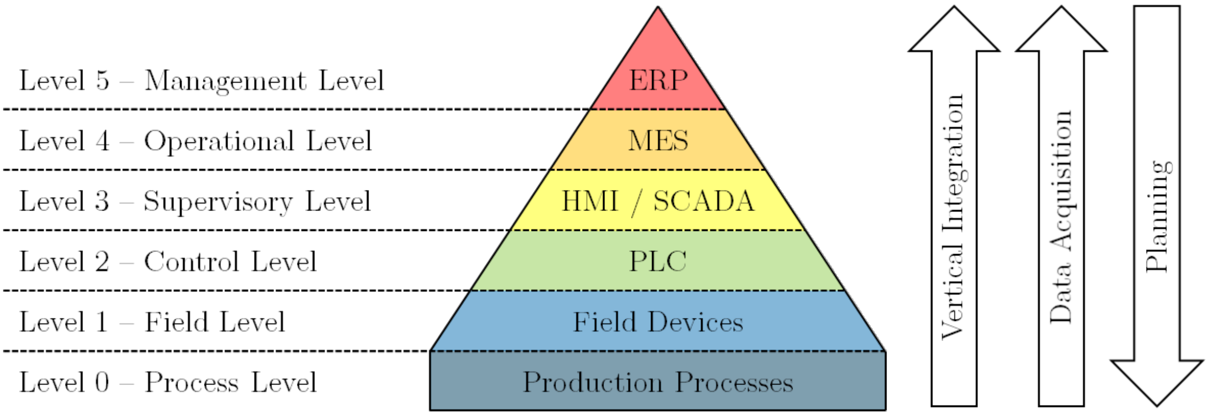


Figure 2.2: Automation pyramid adapted from (Siepmann and Graef 2016, p. 49)

As can be assumed from Figure 2.2, the automation pyramid has interfaces only between its individual levels. For this reason, the levels and the associated control of the levels are considered separately (Schöning and Dorchain 2014, p. 544). Resulting from static vertical integration and a bottom-up orientation of data collection this leads to a strictly centralised concept of automation, with planning from top to bottom and specific encapsulated systems. The limits of this less networked model are thus mainly in the performance of data transmission and processing (Siepmann and Graef 2016, p. 51). This fact casts doubt on the future of this model in view of the requirements that manufacturing systems will face in the future, such as rising data streams, flexibility and decentralised structures. In addition, systems following this approach cannot lead to continuously standardised vertical and horizontal integration (Forstner and Dümmler 2014, p. 199). Literature therefore discusses the dissolution or more precisely the softening of the automation pyramid towards cyber-physical systems (CPS). These are characterised by a linkage of real (physical) objects and processes with information processing (virtual) objects and processes via open, partly global information networks that are connected to each other at any time (Bauernhansl 2014, pp. 15–16). The resulting decentralised and networked structure has no levels. Services, data and hardware components can be distributed to any node of the resulting network (see Figure 2.3), thus forming abstract functional modules from which the automation system is built (VDI/VDE-Gesellschaft 2013, pp. 2–4).

The actual architecture of CPS is currently being researched. A practically applicable and comprehensible version is the 5C architecture. According to this architecture, there are five main components in CPS that build upon each other (see Figure 2.4). Smart Connection attempts to automatically provide data from physical objects through the use of sensors. The processing of data (e.g. from the intelligent connection level) by intelligent algorithms is called data-to-information conversion. Cyber computing acts as a central information hub and processes all information obtained. The cognition component deals with decision-making from information. A feedback from cyberspace into physical space occurs through the configuration level, through the application of decisions made to the system (Lee et al. 2015, pp. 18–20).

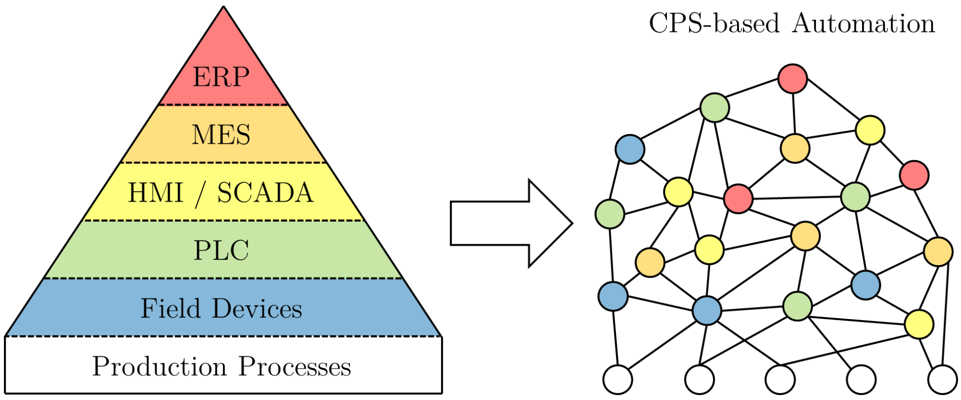


Figure 2.3: Transformation of automation pyramid to CPS-based automation adapted from (VDI/VDE-Gesellschaft 2013, p. 4)

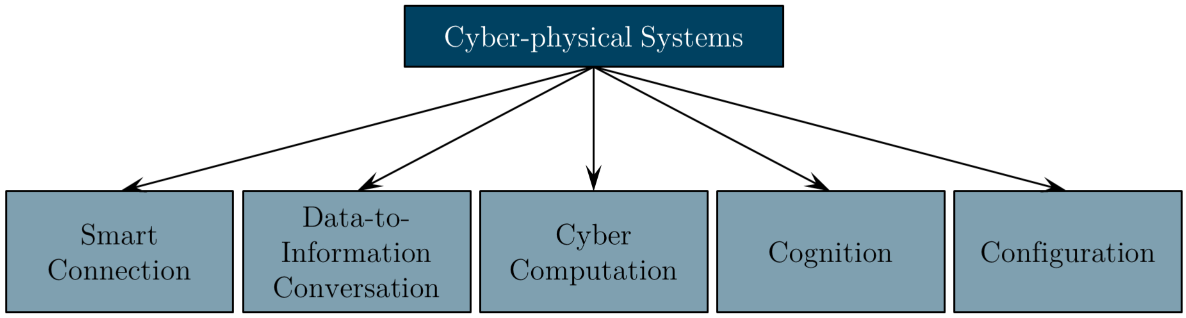


Figure 2.4: 5C-Architecture of CPS according to (Lee et al. 2015)

As mentioned in the previous section AISs are data acquisition systems. They are the basis of information technology and automation technology according to the classical viewpoint, as they collect sensor data from the material flow (Krämer 2002, p. 77). These systems have no control function and only provide information regarding the identity of objects. Within the automation pyramid model, automatic identification systems are consequently field devices and are assigned to the field level due to their proximity to the production processes.

With the transformation of the automation pyramid towards CPS-oriented automation, the data acquisition characteristics of AISs remain the same. Automatic identification systems are particularly the node in the networked CPS that establishes the connection between the virtual and physical world. In the framework of CPS automatic identification technologies are therefore seen as smart connection nodes which are operated in the previously presented smart connection level (Trappey et al. 2016, p. 7363).

2.1.2 Concept of identity

Identification, as mentioned before, describes the process of determining the identity of an object. In literature the term identity or rather the process of identification is not defined uniformly. The different findings from literature are presented in the following:

- Müller defines identification as the determination of the identity of an object of investigation. This can be done on the basis of features that characterise the object naturally or are artificially attached to it. Items may be marked for the purpose of distinctness if they do not have a perceptible or detectable characteristic for a chosen identification process. Marking is thus the making of objects distinguishable, whereas identification means the recognition of objects. The aim of identification is to enable the retrieval and sorting of objects (Müller 2018, p. 27).
- Hippenmeyer and Moosmann emphasise the fact that there are different meanings of the term "Identity". The identity of an object or process in the technical-organisational area should be clearly recognisable and thus detectable everywhere, so that a certain object or process can be clearly distinguished from other, similar objects or processes. The identity of an object is therefore a set of characteristics that is unique to the object, which allows that object to be uniquely recognised and distinguished from other similar objects or processes. The set of characteristics can contain both the natural characteristics of an object (e.g. surface structures) and symbolic artificial characteristics (e.g. barcode) attached to the object (Hippenmeyer and Moosmann 2016, pp. 11–13).
- Jünemann and Beyer understand identification generally as the recognition of known objects. During the identification process, certain features of the object are recorded from the physical scene using sensors. These characteristics are extracted from the sensor data in machine-readable form and compared with characteristics of known objects. If the comparison of the recorded object characteristics with the previously known, stored object characteristics results in a match or sufficient similarity, the object is considered identified. A distinction must be made between direct and indirect identification. Direct identification describes identification using natural reference characteristics. Indirect identification describes identification using artificial reference characteristics (Jünemann and Beyer 1998, pp. 87–88).
- DIN 6763 defines the act of identifying as the unambiguous and unmistakable recognition of an object on the basis of characteristics, so-called identification characteristics, with the accuracy specified for the respective purpose. A characteristic is a particular attribute that serves to describe and distinguish objects from a group of objects or groups of objects from each other (DIN 6763:1985-12, pp. 2–4).
- Fischer and Hofer describe identification as finding an identity in a database. The database contains system-wide valid identifiers of objects, which serve the purpose of identification by comparison. An identity is therefore a system-related unique description of the identifiers of an object (Fischer and Hofer 2011, pp. 414–415).

2.1.3 Numbering in industrial order systems

The term numbering describes the creation, issuance, administration and application of numbers for numbering objects within an ordered system (Dangelmaier 2001, p. 448). Assigning a number consisting of a sequence of numbers and/or characters to a numbering object is, therefore, generally referred to as numbering (DIN 6763:1985-12, p. 6). A number within the concept of numbering is a sequence of characters formed according to certain rules and used to designate objects (DIN 6763:1985-12, p. 4). Numbers can be found in identification numbers, classification numbers and check numbers according to their function (see Figure 2.5). A number that serves for the purpose of identification and thus has identifying properties is called an identification number (DIN 6763:1985-12, p. 2; Kurbel 2016, p. 65). In principle, identification numbers can be assigned both to individual objects and to a group of objects (Wiendahl 2019, p. 168). Object groups, also called object classes, combine objects whose characteristics are identical to a certain defined extent (DIN 6763:1985-12, p. 3). Identification numbers serve as linking elements between objects and information associated with their identity (Hippenmeyer and Moosmann 2016, p. 14). However, the information associated with the object's identity via the identification number must be available from a knowledge base (e.g. database) to make identification possible at all (Ten Hompel et al. 2008, p. 12). Identification numbers are therefore always limited by the availability of the knowledge base applicable to the respective identification process. There are different types of identification numbers that are used in the industrial sector. Two basic types of identification numbers are differentiated (Eigner 2014b, p. 228; Hippenmeyer and Moosmann 2016, pp. 13–16). The first type enables the assignment of a unique identification number to one particular identification object which thus enables a unique identification of this object within a group of similar objects (see Figure 2.6). The second type, on the other hand, only enables a partially unique assignment of an identification number to several similar identification objects and thus only enables the identification of an object as a member of an object group or a subgroup of an object group respectively (see Figure 2.7). Different representations of these number types are used in industry (see Table 2.2). Numbers of the first type are often referred to as serial numbers and are unique identifiers (UID) that are assigned to objects to uniquely identify them within a group of objects (Wiendahl 2019, p. 168). Serial numbers are used to assign specific information (e.g. safety-relevant documentation) to an explicit object and are therefore used for tracking and tracing of one particular object over its entire life cycle as well as protection against counterfeit objects (Ten Hompel and Schmidt 2007, pp. 23–24).

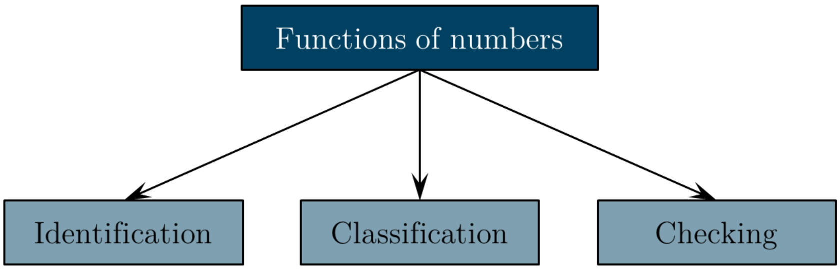


Figure 2.5: Functions of numbers within the concept of numbering (Dangelmaier 2001, p. 449)

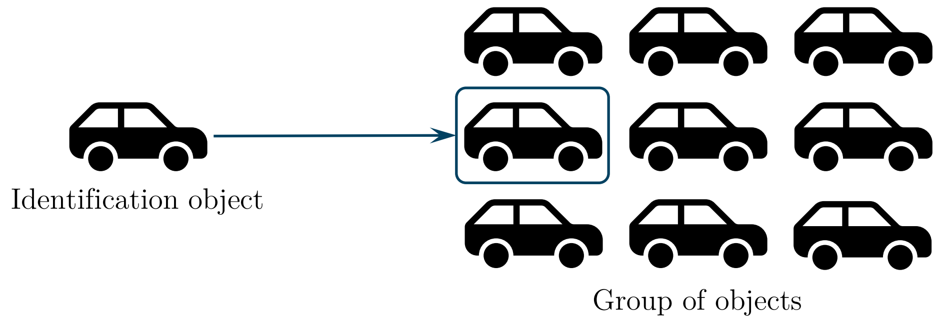


Figure 2.6: Assignment by means of first type identification numbers

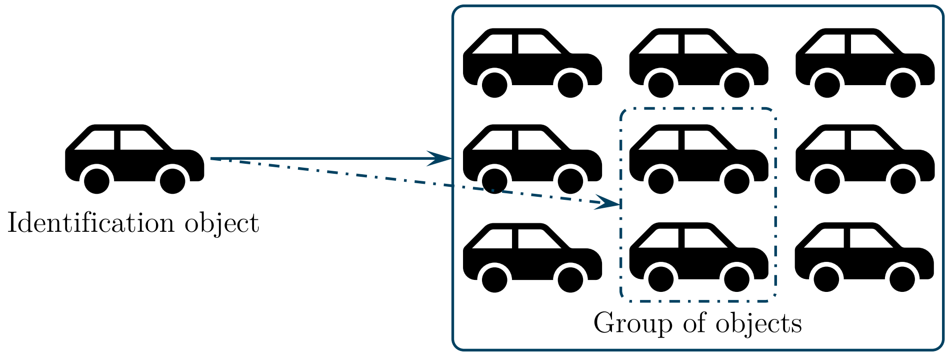


Figure 2.7: Assignment by means of second type identification numbers

Table 2.2: Identification numbers in industry (Hippenmeyer and Moosmann 2016, p. 14)

Numbers for <u>unique</u> identification	Numbers for <u>partially</u> unique identification
– Serial number	– Batch number
– Factory number	– Article number
	– Order number

2.1.4 Process of identification

The process for the identification of objects (see Figure 2.8), on which all automatic identification technologies are based, consists of three subprocesses (Jünemann and Beyer 1998, pp. 87–88; Lolling 2003, pp. 20–24):

1. <u>Data acquisition:</u> From a physical environment, which contains the object of identification, certain features are recorded by means of sensor technology. Only the features describing the object are extracted from the recorded sensor data, while other 'noise' from environmental influences are excluded as far as possible.
2. <u>Data-to-information transformation:</u> The features extracted in the previous step are compared with the stored features of known objects. If the extracted features of the object to be identified sufficiently match these stored features, the object to be identified is considered as identified. Consequently, the data collected were transformed into information

3. **Passing of information:** The information obtained in the form of the identity of the object of identification is passed on to subsequent information systems for further processing.

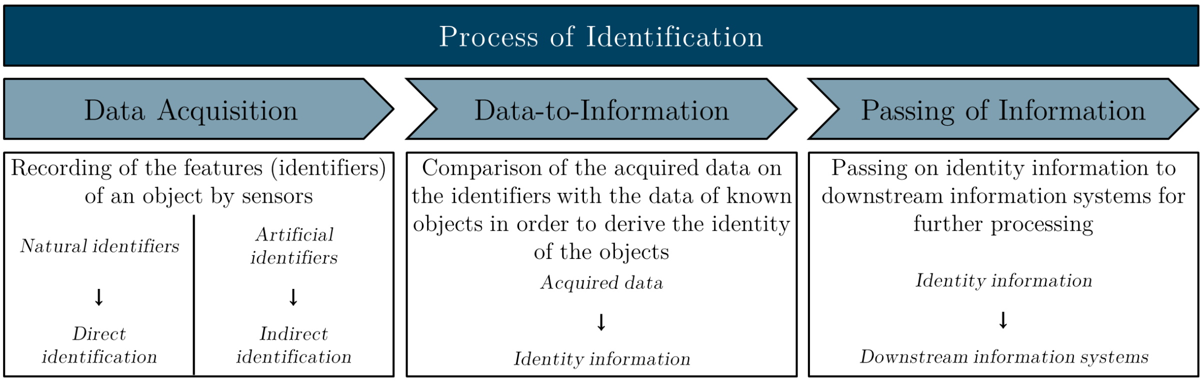


Figure 2.8: Process of identification, according to (Jünemann and Beyer 1998, pp. 87–88; Lolling 2003, pp. 20–24)

The first subprocess of identification involves the recording of features that characterise the object. There are two basic types of features that can be utilised for identification (Arnold et al. 2008, p. 816; Hippenmeyer and Moosmann 2016, pp. 11–12; Ten Hompel et al. 2008, pp. 10–13; Jünemann and Beyer 1998, pp. 87–88):

– **Natural features** are characteristics that an object shows by its natural condition and are therefore called natural identifiers. Natural features used for identification are the so-called **natural identifiers.**
– **Artificial features** are artificially added to an identification object to enable its identification and are therefore called **artificial identifiers.**

All identifiers found in the relevant literature are presented in these two categories in Table 2.3.

Depending on the object features used for identification, a distinction is made between the two generic methods of identification below (Jünemann and Beyer 1998, p. 87):

– **Direct identification** refers to identification based on natural identifiers.
– **Indirect identification** refers to identification by means of artificial identifiers.

When acquiring data, it is essential to ensure that the data reflects the identifiers required for identification as accurately as possible. This is achieved by using adequate sensor technology and post-processing the sensors' raw data (Hippenmeyer and Moosmann 2016, p. 21).

The second subprocess deals with the transformation of data concerning the identifiers of an identification object into identity information. If the data is present in encrypted form, it must first be decrypted to ensure the assignment of the identifiers to their known representation in a database. One example of this is the conversion of a visual code into an alphanumeric code, as is often the case with barcodes. In general, coding is understood to mean a clear, not necessarily reversible assignment of characters from one set of characters (original set) to those of another set of characters (image set) (Ten Hompel et al. 2008, pp. 12–14).

After decoding, the data can be compared with stored data representations of known objects. The prerequisite for automatic object identification is the presence of information about the object in machine-readable form (Wannenwetsch 2014, p. 214). If the data match completely or to a sufficient degree, the identification object is considered identified (Jünemann and Beyer 1998, p. 87). Additional identity information that accompanies the stored data representation, such as the identification number (see Section 2.1.3), can now be retrieved.

In the third and last subprocess, the identity information is forwarded to downstream information systems for further processing via various communication technology interfaces. These include electronic interfaces such as cable or radio connections, as well as visual and auditory interfaces (Lolling 2003, pp. 22–23).

Table 2.3: Natural and artificial identifiers used in industry (Hippenmeyer and Moosmann 2016, pp. 20–27; Jünemann and Beyer 1998, pp. 87–88; Krämer 2002, pp. 84–232; Kropik 2009, pp. 107–114; Ten Hompel et al. 2008, pp. 10–13; Frauenhofer IPM 2017)

Natural identifiers	**Artificial identifiers**
- Weight	- Handwriting
- Geometry (2D/3D)	- Electromagnetic signature
- Length	- Visual code
- Width	- Magnetic signature
- Height	- Artificial acoustic signature
- Volume	- Electronic
- Surface structure	
- Colour	
- Material	
- Temperature	
- Acoustic signature	

2.1.5 Identification objects in the industrial material flow

Material in the sense of material flow theory is a generic term for raw materials, auxiliary and operating materials for processed and finished parts, for assemblies and products of all kinds that still allow a change of location (Arnold and Furmans 2019, p. 1). A more general term for the objects that are handled in the material flow is the term 'goods' (Ten Hompel et al. 2018, p. 4). Goods are defined as things that can be transported and are classified as bulk materials, liquids, gases or piece goods (Pfohl 2018, p. 139; DIN 30781:1989-05, p. 2). Bulk materials are goods that are handled and stored in bulk (e.g. granulate, cement) (Ten Hompel and Heidenblut 2011, p. 275). Individual goods that are handled individually and are included in the transport information separately are referred to as piece goods (DIN 30781:1989-05, p. 2). Each of the four classes of goods can be transformed into packaged goods using packaging material. Packaging material is used to protect packaged goods and to partially or completely enclose or bundle them (DIN 55405:2014-12). Some examples of packaging materials are bags, cans, boxes or barrels (Ten Hompel et al. 2018, p. 12). Piece goods can be considered as packaged goods with or without packaging material (Ten Hompel et al. 2018, p. 9). The combination of packaging items using load carriers is called a loading unit (DIN 30781:1989-05, p. 2). Load carriers are, for example pallets, workpiece carriers or box pallets (Ten Hompel et al. 2018, p. 13). Figure 2.9 summarises this description. In the industrial material flow, the identification objects are therefore piece goods, packages or loading units.

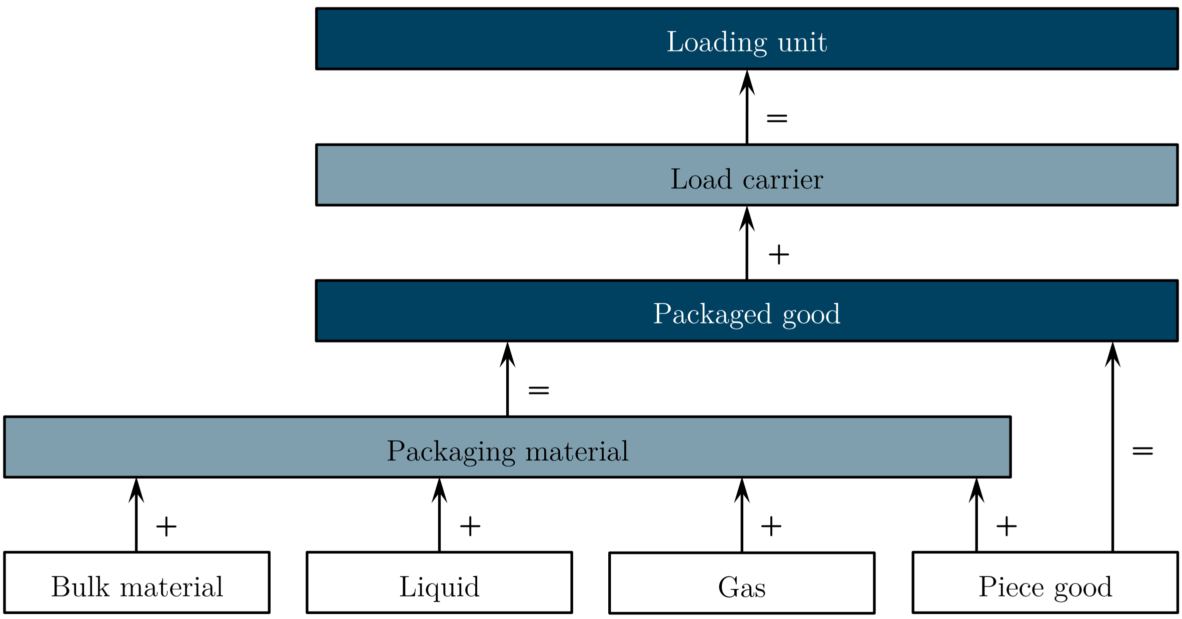


Figure 2.9: Objects in industrial material flow, adapted from (Ten Hompel et al. 2018, p. 10)

2.1.6 Classification of identification technology

The classification of identification systems in the literature is very diverse. The following formulations of classifications were found:

– **Martin** first distinguishes between identification technologies that carry information directly or indirectly. He then divides these two groups again according to their physical principles of the used identifiers, which he defines as follows: mechanical, magnetic, optical and electronic. Explicit technologies are then assigned to these physical principles of action (Martin 2014, p. 505).

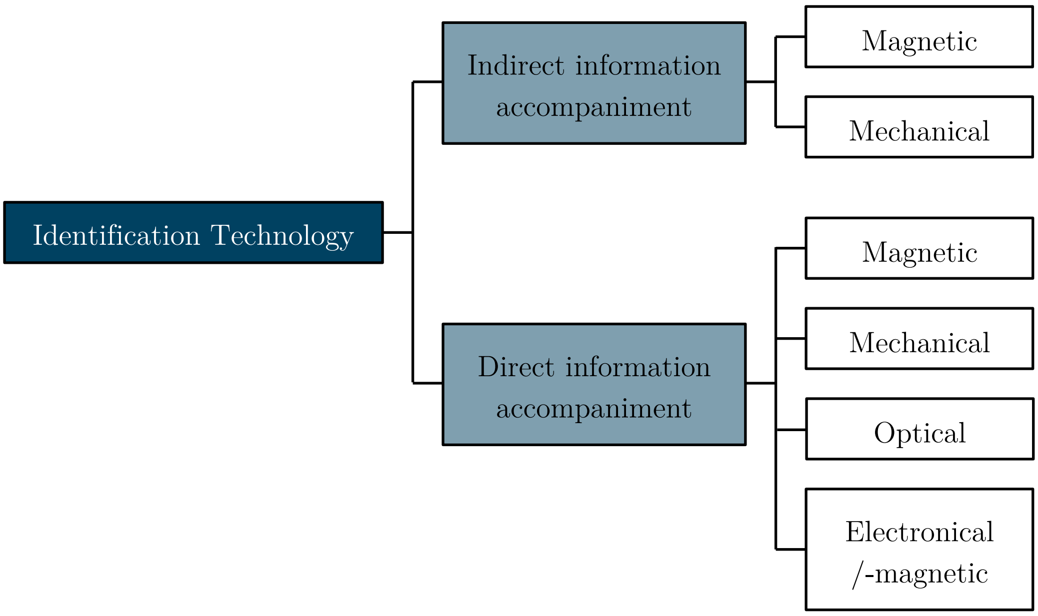


Figure 2.10: Classification of identification technologies (1), according to (Martin 2014, p. 505)

– **Helmus** differentiates between biometric procedures, electronic procedures and character- or symbol-based procedures within identification technologies. Only biometric systems are further subdivided into acoustical procedures and optical procedures. Technologies are further assigned to these procedures (Helmus et al. 2009, p. 199).

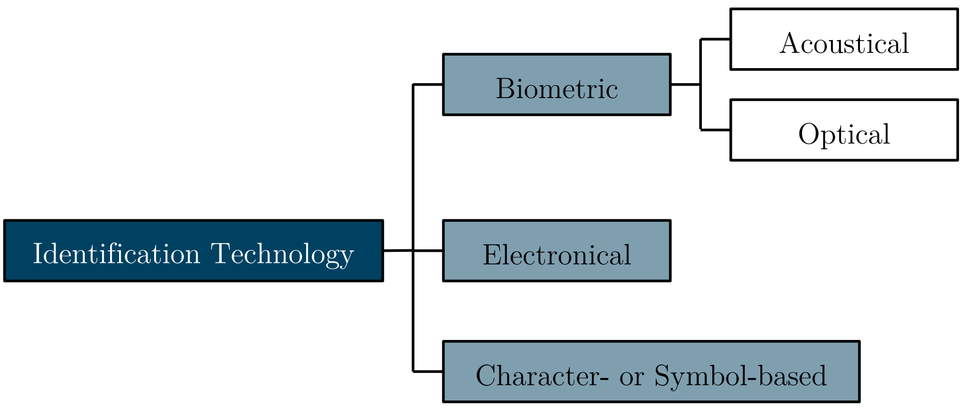


Figure 2.11: Classification of identification technologies (2), according to (Helmus et al. 2009, p. 199)

– **Arnold and Furmans**, **Jünemann and Beyer**, **Krämer**, as well as **Lolling** distinguish identification systems on the basis of their physical principle of data transmission. In particular, they distinguish between mechanical, magnetic, optical and electronic data transmission (Lolling 2003, pp. 41–43; Jünemann and Beyer 1998, pp. 88–90; Arnold and Furmans 2019, pp. 357–358; Krämer 2002, pp. 77–91).

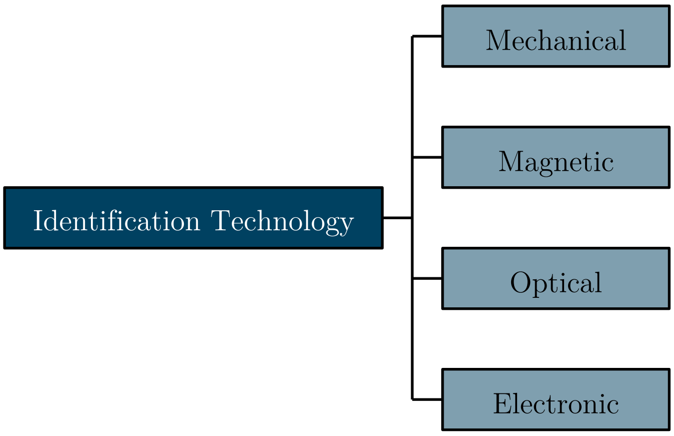


Figure 2.12: Classification of identification technologies (3), according to (Lolling 2003, pp. 41–43; Jünemann and Beyer 1998, pp. 88–90; Arnold and Furmans 2019, pp. 357–358; Krämer 2002, pp. 77–91)

2.1.7 Existing technologies for direct and indirect identification

As shown in the previous section, the classifications of identification technologies in the literature differ considerably. Of particular interest for this work are the technologies that allow direct identification based on the natural features of the object. Those technologies are described in more detail below. For the sake of completeness, all technologies for indirect identification found in literature are presented in Table 2.4.

While there are many approaches for direct identification using biometric features for the purpose of identifying living beings, there are few approaches for the direct identification of industrial components or logistical objects in general (Helmus et al. 2009, pp. 199–205; Hippenmeyer and Moosmann 2016, pp. 20–21).

Table 2.4: Technology for indirect identification [1](Martin 2014, pp. 505–516) [2](Hippenmeyer and Moosmann 2016, pp. 21–28) [3](Arnold and Furmans 2019, pp. 359–399) [4](Ten Hompel et al. 2008, pp. 9–20) [5](Weißflog et al. 2019) [6](Helmus et al. 2009, pp. 199–212) [7](Jünemann and Beyer 1998, pp. 90–95)

Principle of data transmission	Technology for indirect identification
Optical	Optical character recognition (OCR) [1][2][3][4][5][6]
	One-dimensional code (1D-Code) [1][2][3][4][5][6][8]
	Two-dimensional code (2D-Code) [1][2][3][4][5][6]
	Three-dimensional code (3D-Code) [1][2][3][4][5][6]
	Hole patterns [4][7]
Electronic/Electromagnetic	Radio-frequency identification (RFID) [1][2][4][5][6]
	Bluetooth low energy (BLE) [2][5]
	Ultra-wideband (UWB) [5]
	Near-field communication (NFC) [5]
	Programmable read-only memory (PROM) [1][3][7]
	Erasable programmable read-only memory (EPROM) [1][3][7]
	Random access memory (RAM) [3]
Magnetic	Magnetic memory [1][3][7]
Mechanical	Cams [1][3]

In 2005 an identification procedure for the direct identification of documents, plastic cards and packaging was presented by (Buchanan et al. 2005). The method uses the naturally and randomly occurring imperfections on the surfaces of the aforementioned objects as unique identifiers. The procedure exposes rough surfaces with a focused laser, which leads to diffuse scattering, also called laser granulation or speckle (see Figure 2.13). The speckle is subsequently captured by photodetectors arranged at different angles to measure the reflected intensity. By means of statistical analysis a digital fingerprint can be generated from the fluctuation of the intensity around the mean intensity and then translated in a binary descriptor containing only ones and zeros. Comparing such descriptors by means of cross-correlation results in a clear peak in the signal in case of agreement, which does not occur in case of disagreement (Buchanan et al. 2005, p. 475).

One of the co-authors of the aforementioned publication published another paper three years later and named the direct identification procedure laser surface authentication (LSA) (Cowburn 2008, pp. 332–342).

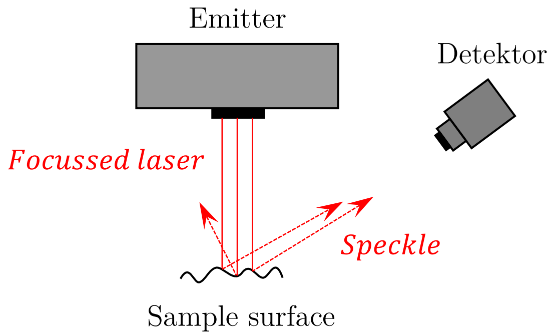


Figure 2.13: Laser surface authentication, own representation according to (Buchanan et al. 2005, p. 475; Cowburn 2008, p. 337)

Under the heading “Fingerprints for Machines – Characterization and Optical Identification of Grinding Imprints”, an approach for direct identification of mechanical components was proposed by (Dragon et al. 2011). The basic distinguishing features for this direct identification are the grinding imprints, which result from the machining of component surfaces with grinding wheels. Grinding imprints can be regarded in the same way as the lines and ridges of a human fingerprint and are also as characteristic as these. The proposed approach uses mainly the surface profile orthogonal to the grinding direction, which can be acquired by optical sensors. After noise removal, this surface profile is transformed into the space-frequency domain with the help of a continuous wavelet transform process. Figure 2.14 shows the generation of the space-frequency domain from a grinding profile. Features are detected from the space-frequency domain and described by individual descriptors. The actual identification is done by comparing feature descriptors extracted from a known object and a sample object (Dragon et al. 2011, pp. 276–280).

Two more recent publications describe the applicability of direct identification using grinding imprints according to the research of (Dragon et al. 2011) in an industrial environment (Frauenhofer IPM 2017, pp. 1–2; Breidenstein et al. 2016, pp. 412–415).

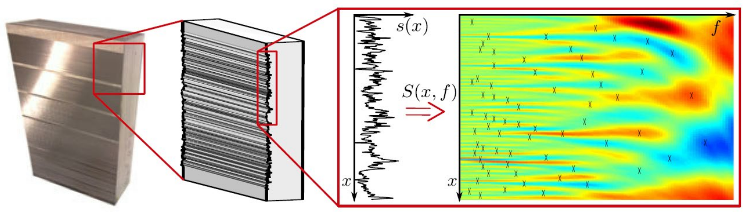


Figure 2.14: Generation of space-frequency domain from grinding imprints (Dragon et al. 2011, p. 276)

2.2 Machine vision

Human vision and cognition is of paramount importance for many industrial applications from the initial design process to manufacturing, assembly and quality control, and final packaging (Batchelor 2012b, p. 4; Beyerer et al. 2016, p. 3). In the progression towards increasing automation and the transfer of human activities to machines, artificial vision is one of the essential skills that machines must learn. Artificial vision does not necessarily attempt to imitate human vision as it is confronted with different requirements such as speed, cost efficiency and reliability, but tries to solve vision-related tasks (Šonka et al. 2015, pp. 1–9; Cognex 2016, p. 5). In the context of artificial vision, there are two terms frequently mentioned in literature: computer vision and machine vision. These terms are often used synonymously, as both involve artificial vision, but there are fundamental differences in attitudes and priorities that make a distinction between them necessary (Batchelor 2012b, pp. 10–15).

Computer vision, as a branch of computer science, deals with the processing of image information from a camera into a decision or new representation by means of computers. (Bradski and Kaehler 2011, pp. 2–6).

Machine vision (MV) is concerned with the engineering of artificial vision systems. These consist of mechanical, optical, electronic as well as software components. MV aims to examine natural objects or materials, human artefacts and manufacturing processes to detect defects and improve the quality, operational efficiency and safety of products and processes. It is also used to control the machines used in manufacturing. Thus, MV is a holistic approach based on the principles of systems engineering that requires the harmonious integration of various areas of study to create useful and practical systems for artificial vision in industrial applications. (Batchelor 2012b, pp. 5–10)

Some of the areas of study MV aims to integrate are (Batchelor 2012b, p. 18):

- Lighting/illumination technology
- Optics and image acquisition technology
- Algorithms/heuristics for image processing
- Communication/networking technology
- Software and systems engineering
- Mechanical engineering
- Production engineering
- Industrial engineering

As can be concluded from the above list, human vision and cognition are important for many different industrial applications. The applications of MV are very wide-ranging. Typically, MV applications fall into one of the following categories (Cognex 2016, pp. 6–10):

- <u>Guidance</u> of robots or machine controllers, by determining position and orientation of an object in a 2D- or 3D-Space.
- <u>Recognition and identification</u> of objects using artificial or natural identifiers for process monitoring and control, traceability and quality assurance.

- Gauging of objects by calculating dimensions between points or geometrical locations in order to check whether these meet specifications.
- Inspection of objects in order to detect defects, contaminants, functional flaws, incompleteness or other irregularities.

MV systems use a processing chain to fulfil all application-specific tasks. The specific steps of this chain are as follows (Beyerer et al. 2016, pp. 10–15):

1. Image acquisition as the initial step, is the most crucial of the whole chain. This step defines the amount and quality of information that is contained in the image data, which is needed for all further processing. Prerequisites for a successful result are suitable illumination, optics and image sensors with regard to the object in the scene.
2. Digitisation of the originally analogue voltage signal from the image sensor chip by discretisation and limitation in terms of space and amplitude. The resulting raw digital image data usually still contains disturbing and irrelevant components, such as noise and inhomogeneities.
3. Preprocessing aims to compensate for the disturbing and irrelevant influences of the raw digital image data in order to obtain improved data for optimal information retrieval.
4. Compression and extraction of information from the improved data. Here the data is segmented into meaningful sections and/or relevant parameters are extracted in the form of features.
5. Decisions can ultimately be made on the basis of the condensed information from the previous steps. These are based on predictions or assumptions that result from the detection, classification or interpretation of the extracted information.

The necessary components of an MV system can already be guessed from the previous explanations. Major hardware components of every machine vision system are: Illumination, optics, image sensors, vision processing and communications hardware, as well as a mechanical handling unit. These components are available commercially off the shelf, either as separate modules or as integrated devices. Software components are all algorithms that are used to review and process the image sensor data as well as extract and communicate the information (Cognex 2016, pp. 11–16; Batchelor 2012b, pp. 6–8).

2.2.1 3D-Object recognition

Object recognition is one of the numerous applications within machine vision. It describes the recognition of one or more physical objects by comparing image input data gathered from a scene with a knowledge base of known objects. More specifically, 3D-Object recognition describes the classification or identification of known objects within input data, which is often associated with determining its position and orientation in three-dimensional space, also known as six degrees of freedom (6-DoF) pose estimation (Arman and Aggarwal 1993, p. 6; Zhao 2012, p. 633; Liu et al. 2019, p. 135; Jain et al. 1995, p. 459; Poggio and Ullman 2014, p. 469; Treiber 2013, pp. 95–96).

In the field of 3D-Object recognition there are two fundamental branches into which the research community has been split (Alhamzi et al. 2014, p. 651; Byne and Anderson 1998, p. 533; Bradski and Kaehler 2011, pp. 520–521):

- 3D-Object recognition using 2D-Input data describes the recognition of objects on the basis of 2D-Image data acquired by conventional image sensors.
- 3D-Object recognition using 3D-Input data describes the recognition of objects on the basis of depth images, 3D-Point clouds or 3D-Meshes acquired by means of 3D-Sensing technology.

However, the last-mentioned branch in particular is considered to be very important for future applications, as the additional integration of depth information creates more possibilities for object recognition and therefore enables more precise and reliable procedures (Bradski and Kaehler 2011, p. 521; Dong et al. 2019, p. 243; Hashimoto et al. 2017, pp. 31–32).

Another important distinction within 3D-Object recognition is made between the level of recognition aimed for (Alhamzi et al. 2014, p. 651; Andreopoulos and Tsotsos 2013, p. 828):

- Instance-level recognition describes the recognition or identification of an explicit object based on its specific features, such as appearance and geometry.
- Category-level recognition is the recognition of the category of an object and is also often referred to as classification.

Due to the scope of this thesis, only methods enabling instance-level recognition are relevant for this work. Methods for the classification of objects are not explained any further, despite the fact that they share some similarities. Depending on the type of description or form of representation of the objects, the following approaches can be pursued to perform 3D-Object recognition on an instance-level (Dong et al. 2019, pp. 243–247; Hashimoto et al. 2017, pp. 31–33; Alhamzi et al. 2014, pp. 652–653):

- Model- or geometry-based approaches use mathematical descriptions representing geometries of objects themselves for 3D-Object recognition. For this purpose, the geometric properties of objects are reconstructed from the sensor input data and mathematically described. These reconstructed descriptions can then be matched against those of known objects.
- View- or appearance-based approaches use descriptions representing one or several views from different perspectives of objects for 3D-Object recognition. The appearance of objects in the sensor input data acquired from a scene can therefore be directly compared with the view descriptions in the knowledge base.
- Feature matching-based approaches use descriptions of geometrical and or appearance features for 3D-Object recognition. The features extracted from the sensor input data of the scene are matched with features in a knowledge base in order to perform the recognition.

There are a number of advantages and disadvantages for each type of approach. Model-based approaches are generally applicable for objects with non-complex shapes, since geometric models need to be established from the sensor input data, which is computationally intensive as well as susceptible to occlusion and noise (Dong et al. 2019, p. 244). View- or appearance-based approaches share these disadvantages, plus they are susceptible to the viewing angles and illumination (Dong et al. 2019, p. 251; Hashimoto et al. 2017, p. 32). In order to compensate for these disadvantages, feature matching-based approaches were developed, whereby the features take on an intermediate level. These approaches achieve good robustness to occlusion, noise and changes in perspective and can furthermore be applied in an model- or view- oriented manner (Dong et al. 2019, p. 251; Hashimoto et al. 2017, pp. 32–33). Modern approaches from research in MV almost exclusively use feature matching-based approaches for 3D-Object recognition.

The generic process for feature-based recognition of 3D-Objects at instance-level consist of three fundamental steps (Bay et al. 2008, p. 346; Jain et al. 1995, pp. 460–462; Liu et al. 2019, p. 136; Poggio and Ullman 2014, p. 469; Dong et al. 2019, p. 243):

1. **Extraction of features** for the purpose of recognition. For 2D-Input data these features can be prominent points (pixels), edges or contours (Lowe 2004, pp. 92–94). In the case of 3D-Input data, properties and configurations of points or surfaces are possible features (Mateo et al. 2014, p. 428).
2. **Description of features** with so-called descriptors. In fact, descriptors are high-dimensional vectors, which encode the properties of the detected features. This step is of exceptional importance as it is important to describe the features as clearly as possible and thus ensure their reliable and time-effective comparability.
3. **Matching** of descriptors acquired from a scene to descriptors of known objects stored in a knowledge base in order to find correspondences. The quality of a correspondence is determined with the help of comparative metrics and thus determines the threshold at which a recognition is considered to be successful.

From the evidence presented so far, it is obvious that there is a wide range of possibilities for the design of machine vision system for instance-level 3D-Object recognition. To design such a system, four essential elements must be determined (Arman and Aggarwal 1993, p. 6):

1. **The type of sensor** for data collection
2. **The methods of constructing a knowledge base** with descriptions of known objects
3. **The means of describing the input data** from the sensor and for the knowledge base
4. **The methods of matching the descriptions** in order to recognize known objects

The generation of the knowledge base can either be done by storing descriptors acquired from real sensor data or on the other hand be derived from suitable object descriptions like CAD-Models (Byne and Anderson 1998, pp. 533–534; Aldoma et al. 2011, pp. 585–586; Ahola et al. 2016, p. 1). Due to the scope of this work, the knowledge base for object recognition will be generated from CAD-Data. The selection of a particular description from one of the approaches described above has a decisive influence on the quality of the recognition process (Dong et al. 2019, pp. 250–251). In order to explore and summarise the best in the field of 3D-Object recognition based on CAD-Data, a systematic literature review is conducted in the following section.

2.2.2 Systematic literature review on 3D-Object recognition using CAD-Data

Within the field of machine vision there is a wide range of methods for 3D-Object recognition using CAD-Data. In particular, the choice of optical sensors, features and descriptors influence the quality of object recognition (Beyerer et al. 2016, pp. 10–15). In order to determine the latest findings in the field of CAD-based object recognition and to gain insight into the subject area, literature will be reviewed. Here, the main focus is on the recognition process itself, including the sensor technology and representations used. Finally, one approach is to be identified how to design a 3D-Object recognition system.

The method of choice for this undertaking is a systematic literature review (SLR), since this kind of approach aims to identify, evaluate and interpret all available research relevant to a particular research question, topic area or phenomenon of interest (Kitchenham 2007, p. 3). In detail, the systematic literature review follows a procedure based on (Snyder 2019, pp. 336–337), which consists of four phases: Designing the literature review, conducting the search, analysing the found documents and writing the review. The following sections describe the individual phases.

2.2.2.1 Designing the systematic literature review

The initial phase is the design or planning of the SLR. This includes the formulation of the questions, the search strategy, and the inclusion and exclusion criteria for found literature (Snyder 2019, p. 336). The questions that the SLR puts to each reviewed document are shown in Table 2.5.

Table 2.5: Questions for SLR

Q1:	Which optical sensor technologies are used regarding input data?
Q2:	How does the recognition process work?
Q3:	Which CAD-Representations and recognition representations are used?
Q4:	Which software is used?

For the research in the above-mentioned databases (see Table 2.1), the definition of a search string is necessary. The search engines, Scopus and IEEE Xplore, search only the related scientific databases and thus provide more structured results. Google Scholar is not directly tied to specific databases and therefore also covers grey literature. With respect to the subject area and the questions asked, the search strings are formulated as follows:

("3D" OR "three-dimensional") AND "object recognition" AND ("cad" OR "computer-aided design")

This search string is first applied to the title, keywords and abstract of the literature in the databases. The criteria displayed in Table 2.6 and Table 2.7 are then used to include or exclude and therefore select documents. The selection is made in two steps:

1. **Application of the inclusion criteria (IC) and exclusion criteria (EC) to the title and abstract** of the found documents in order to pre-select documents for further analysis.
2. **Application of the inclusion criteria (IC) and exclusion criteria (EC) to the full content** of the remaining documents for final selection of appropriate documents.

Subsequently, the SLR is written and the information needed to answer the questions are extracted. Finally, a conclusion is drawn from the findings and one approach is selected as a result of this SLR, applying the quality criteria (QC) shown in Table 2.8 to the documents examined. Each criterion is evaluated using a 4-level Likert scale and is formulated in such a way that if the criterion is met, the document is particularly suitable for this aspect. The scale is formulated as follows: (1) Does not apply, (2) Rather does not apply, (3) Rather applies, (4) Applies. The document with the highest score thus represents the most suitable approach for the purpose of this thesis.

Table 2.6: Inclusion criteria for documents

IC1:	Study representing 3D-Object recognition on basis of CAD-Data

Table 2.7: Exclusion criteria for documents

EC1:	Document does not describe the process of object recognition for multiple different objects from CAD-Data
EC2:	Document does not describe recognition in the sense of identification, but classification
EC3:	Document is not from the field of engineering or computer science or not applicable to industrial component recognition
EC4:	Document is not formulated in German or English language
EC5:	Document is not accessible with existing resources
EC6:	Document is published before 2010

Table 2.8: Quality criteria for final selection

QC1:	Document is clear and understandable
QC2:	Document describes the full process of object recognition in detail
QC3:	Document makes use of open or commercially available software
QC4:	Document states the sensor technology used

2.2.2.2 Conducting the systematic literature review

The search is conducted in the second phase of this systematic literature review. For this purpose, the previously defined search strings are applied to the selected search engines. The number of hits resulting from each search engine using the raw search string can be seen in Table 2.9. IEEE Xplore and Scopus delivered a number of hits with a similar magnitude. Both search engines also offer the possibility to export the hit list into a table format and thus enable the further refinement of search and analysis of documents. Google Scholar only offers the possibility to search whole articles or just their titles. As can be seen from Table 2.9, a very large and a very small number of hits was created by applying the search string to the two possible fields. Even the narrowing down by means of exclusion criteria 6 could not significantly improve the result. Since Google Scholar does not provide any further filter options to limit the hits and the most relevant hits referred to either IEEE or Science Direct, it was decided to use only those two search engines.

Table 2.9: Number of hits after application of the search strings to the search engines

Search engine	Search field	Hits	Date (last updated)
IEEE Xplore	Metadata	304	20.08.2020
Scopus	Title-Abstract-Keywords	348	20.08.2020
Google Scholar	Whole Article	17400	20.08.2020
	Title	15	20.08.2020

2.2.2.3 Analysing the found documents

For the purpose of analysis, exported CSV-Tables from the search engines IEEE Xplore and Scopus were used. In order to select most appropriate documents, the tables containing the search results were examined in two steps. The title and abstract of each document found were first examined to determine whether the study should be included or excluded for further analysis using the criteria shown in Table 2.6 and Table 2.7. After this pre-selection a total of 21 documents were still remaining for more detailed analysis. Table 2.10 shows the exact number of excluded and included documents as well as the number of duplicates discovered. In a second step, the remaining 21 documents were read in their entirety and checked whether they were again to be included or excluded on the basis of the criteria. Table 2.11 shows the final state of the selection with all details. The number of excluded documents by exclusion criteria is shown in Table 2.12.

Table 2.10: Number of documents after applying selection criteria to title and abstract

Search Engine	Hits	Excluded	Included	Duplicates	Selected
IEEE Xplore	304	289	15	5	21
Scopus	348	337	11	5	

Table 2.11: Number of documents after applying selection criteria by reading paper

Search Engine	Hits	Excluded	Included	Duplicates	Selected
IEEE Xplore	304	296	8	4	11
Scopus	348	340	8	5	

Table 2.12: Assignment of the number of exclusions to the criteria

Criterion	Number of excluded documents
EC1	181
EC2	33
EC3	50
EC4	1
EC5	3
EC6	368

2.2.2.4 Reviewing the final selection

In the following summary the analyses of all documents are presented in alphabetical order of the authors' names:

1. Ahola et al. propose a configurable CAD-based object recognition and pose estimation system. In particular, their proposal includes a method for adjusting the resolution of the models in the database with respect to the resolution of the 3D-Sensing technology used. For this purpose, the finer details of the CAD-Models are filtered out, which could not be captured by the used 3D-Sensing technology with their respective resolution anyway. The proposed approach will improve the similarity values between the measured data from the scene and the model in the database, as non-detectable features are not considered. The proposed recognition process (see Figure 2.15) begins with the raw 3D-Point cloud acquired by a selected 3D-Sensing technology. This point cloud is downsampled in a first filtering step, in order to reduce noise and the overall computation effort. Afterwards the point cloud is segmented into individual surface patches groups. On the basis of these segments a so-called

attributed graph model is created that describes the geometrical information and topology of the surface patches. The recognition itself is done by comparing these attributed graphic models of the CAD-Model and the sensor data. In particular for the creation of the graph model from the CAD-Data, the proposed filtering method comes into play. Lower limits for surface patch dimensions are set and surface types can be excluded for this purpose. The measured and modelled graphs are afterwards compared to each other in two phases. First candidate graphs are selected based on the number of detectable surfaces types, then these selected graphs are compared in detail in order to gather their similarity. Resulting in a final score table of similarities, the pairing of measured and modelled surface patches enables pose estimation as well. Table 2.13 summarises the evidence needed to answer the questions of this systematic literature review (Ahola et al. 2016, pp. 1–6).

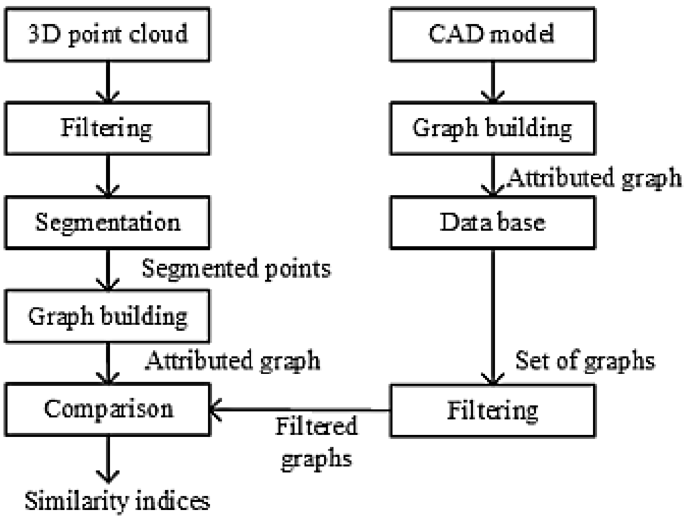


Figure 2.15: Recognition process (Ahola et al. 2016, p. 3)

Table 2.13: Summary (Ahola et al. 2016)

Sensor technology:	3D-Sensing technology, not specified
Recognition process:	See Figure 2.15
CAD-Representation:	STEP-Format
Recognition-Representation:	Aspect graph
Software:	Point cloud library (PCL)

2. Aldoma et al. present a new global descriptor type for 3D-Object recognition and 6 degrees of freedom (6-DoF) pose estimation as an extension of the viewpoint feature histogram (VFH), which they call a clustered viewpoint histogram (CVFH). Global descriptors describe geometry, appearance or both of a whole and partial view of an object given as a point cloud. Aldoma et al. claim, that none of the descriptors presented in literature has tackled the problem of using CAD-Data for creating a database of known objects. The proposed recognition framework, which is nothing other than the underlying recognition process, consists of two parts (see Figure 2.16). The first part is an offline training stage, where the CVFH descriptors for the CAD-Models in the database are calculated. For this purpose, several distinguishable views of each CAD-Model are created by rotating a virtual 3D-Sensor spherically around it. From the resulting point clouds of each individual view, CVFH descriptors describing the geometry and so-called camera roll histograms describing the respective viewing perspective are calculated and stored in a training dataset. The second part of the recognition framework processes the sensor data of the real scene online. For this purpose, the point cloud captured by a sensor is first separated into individual segments that belong together and which describe the object(s) to be detected. CVFH descriptors and roll histograms are then also calculated for each of these segments. For each scene segment, the data set that fits best is now selected from the training set. This is achieved by means of nearest neighbour search using the scene and model CVFH descriptors. Finally, the roll histograms are used to determine the object's pose in space. Table 2.14 summarises the evidence needed to answer the questions of this systematic literature review (Aldoma et al. 2011, pp. 585–592).

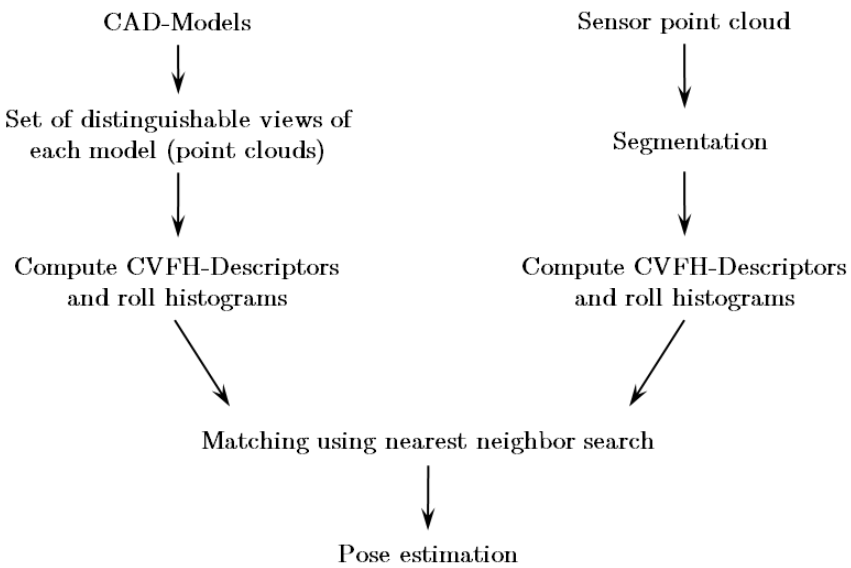


Figure 2.16: Recognition process according to (Aldoma et al. 2011, p. 589)

Table 2.14: Summary (Aldoma et al. 2011)

Sensor technology:	3D-Sensing technology, Microsoft Kinect Sensor
Recognition process:	See Figure 2.16
CAD-Representation:	Not specified
Recognition-Representation:	Clustered Viewpoint Feature Histogram (CVFH) from point clouds
Software:	Not specified

3. **Han and Zhao** studied the recognition of 3D objects from monocular (two-dimensional) images in mobile augmented reality using CAD-Data. The proposed method is based on the assumption that the position of the centre of mass of a material-uniform 3D object in relation to the camera coordinate system can be estimated by using its projection point in the form of the centroid of area on the image plane. This makes it possible to reduce the search space from a 6-DoF to a 3-DoF one using simple image processing algorithms such as contour detection. These remaining 3 rotational degrees of freedom in the form of azimuth parameters can then be derived from rendered views of an associated CAD-Model by matching contours. Furthermore, Han and Zhao provide a modification of their method, which also enables the recognition of objects whose mass centre is not located in the centroid of the object's projective area. Figure 2.17 shows the underlying recognition process. Table 2.15 summarises the evidence needed to answer the questions of this systematic literature review (Han and Zhao 2015, pp. 36–46).

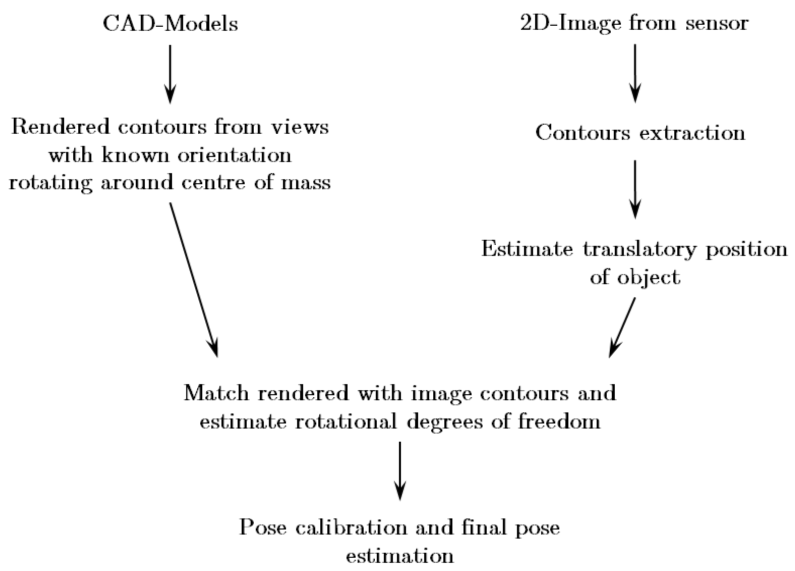


Figure 2.17: Recognition process according to (Han and Zhao 2015)

Table 2.15: Summary (Han and Zhao 2015)

Sensor technology:	2D-Image sensor
Recognition process:	See Figure 2.17
CAD-Representation:	Surface and wire-frame representation, not specified
Recognition-Representation:	Contours from 2D-Image data
Software:	OpenGL

4. Lee et al. propose a visual 3D-Perception system for bin picking applications in automotive sub-assembly automation. The system is specially designed for the recognition of components consisting of cylindrical volumes, such as car alternators. Using a structured light sensor, 3D-Data is first collected from a scene in the form of a point cloud. Afterwards the point cloud is denoised using Gaussian smoothing and the normal of each point is calculated using neighbouring points. A specially developed surface patch segmentation algorithm generates a set of plane and cylinder surface patches using the normal information of the point clouds. The resulting surface patches are then transformed into basic geometric shapes, so-called 'geometric primitives'. For object recognition, geometric primitives are now also derived from CAD-Data and compared with those obtained from the point cloud. If there is sufficient agreement between model and sensor data, the pose is determined from the orientation of the geometric primitives to each other. Figure 2.18 summarises the described recognition process. Table 2.16 summarises the evidence needed to answer the questions of this systematic literature review (Lee et al., pp. 706–713).

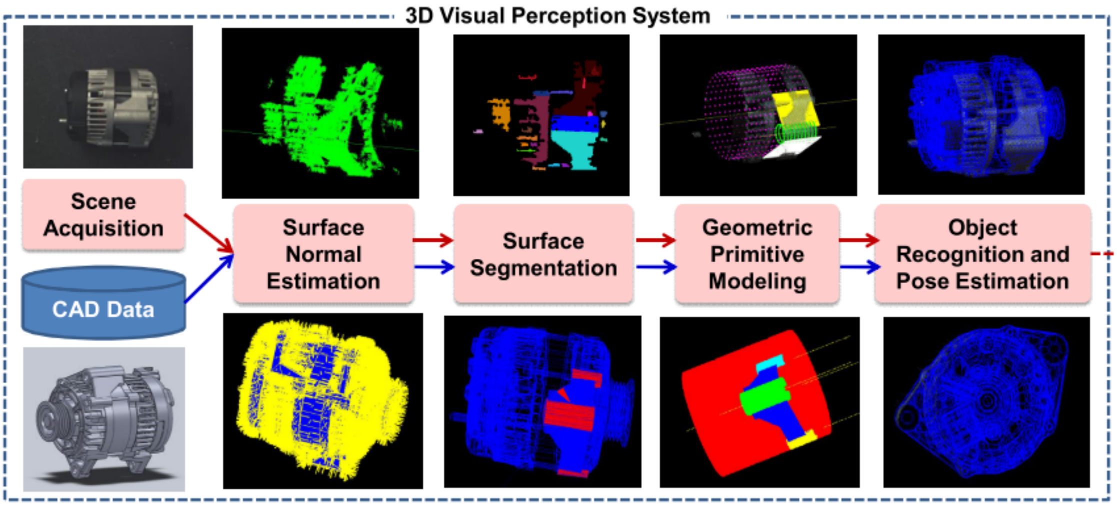


Figure 2.18: Recognition process (Lee et al., p. 707)

Table 2.16: Summary (Lee et al.)

Sensor technology:	3D-Sensing technology, structured light camera
Recognition process:	See Figure 2.18
CAD-Representation:	DXF-Format
Recognition-Representation:	Geometric primitives from point cloud
Software:	Not specified

5. Luo and Kuo propose a scalable modular architecture for a robot arm fetching system including 3D-Object recognition. For this purpose, 3D-Sensing technology is used to capture a point cloud from the scene as an input for the recognition algorithm. The recognition algorithm itself is based on the pipeline described in (Aldoma et al. 2011) and is therefore not explained again (see Figure 2.16). Table 2.17 summarises the evidence needed to answer the questions of this systematic literature review (Luo and Kuo 2015, pp. 269–274).

Table 2.17: Summary (Luo and Kuo 2015)

Sensor technology:	3D-Sensing technology, Microsoft Kinect
Recognition process:	See Figure 2.16
CAD-Representation:	Not specified
Recognition-Representation:	Clustered Viewpoint Feature Histogram (CVFH) from point clouds
Software:	Point cloud library (PCL)

6. Pan et al. enhance the CAD view-based algorithm of (Ulrich et al. 2012) applying image segmentation and efficient shape matching. The recognition process (see Figure 2.19) starts with the 2D-Image data of a scene. This input image is downsampled using a Gaussian image pyramid to avoid the calculation of too many unnecessary pixels in the background clutter. A conversion of the image data from the RGB colour space to the HSV colour space subsequently enables a segmentation according to colour information. By limiting the permissible values of hue (H) and saturation (S), only those pixel segments that are desired are selected for further processing. Morphological image processing methods are then applied to remove unwanted noise in the segmented image data. Since up to now segmentation has only been carried out by means of colour information, the next step will also consider the size and aspect ratio features for further segmentation. Ideally, each created segment now contains only one object of interest and is therefore a region of interest (ROI) for further matching. This ROI is then passed to the algorithm proposed by (Ulrich et al. 2012), which is explained later in this systematic literature review (see Figure 2.25). Table 2.18 summarises the evidence needed to answer the questions of this systematic literature review (Pan et al. 2017, pp. 406–411).

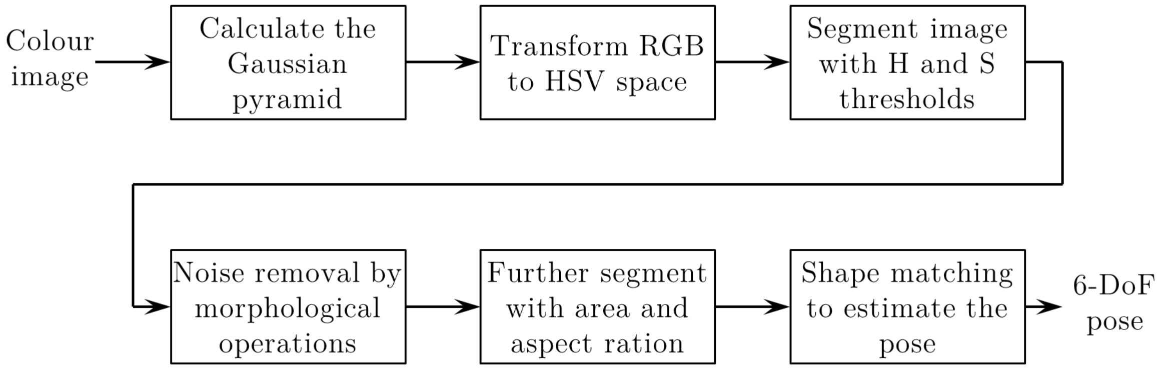


Figure 2.19: Recognition process according to (Pan et al. 2017, p. 407)

Table 2.18: Summary (Pan et al. 2017)

Sensor technology:	2D-Image sensor, not specified
Recognition process:	See Figure 2.19
CAD-Representation:	Not specified
Recognition-Representation:	Contours from 2D-Image data
Software:	Not specified

7. Pretto et al. present a system for recognition and localisation of planar objects for industrial bin-picking using only a standard camera. Due to the application, the planar parts can be placed randomly in a container or on a conveyor belt. From a given CAD-Model for the component to be recognised, a template of its contour is first created. 2D-Image data is collected from the scene depicting the components and the contours contained therein are extracted by means of a so-called LSD detector. A voting scheme based on the conventional Hough-like approach is used to search for potential contour candidates representing a particular object. Subsequently, a rigid body transformation is calculated for each hypothetical contour, which transfers the template contour previously extracted from the CAD-Model into the candidate contour. Finally, by applying special optimisations, the candidate that is best suited for the next grasp in the box is selected. Figure 2.20 represents the described process. Table 2.19 summarises the evidence needed to answer the questions of this systematic literature review (Pretto et al. 2013, pp. 168–175).

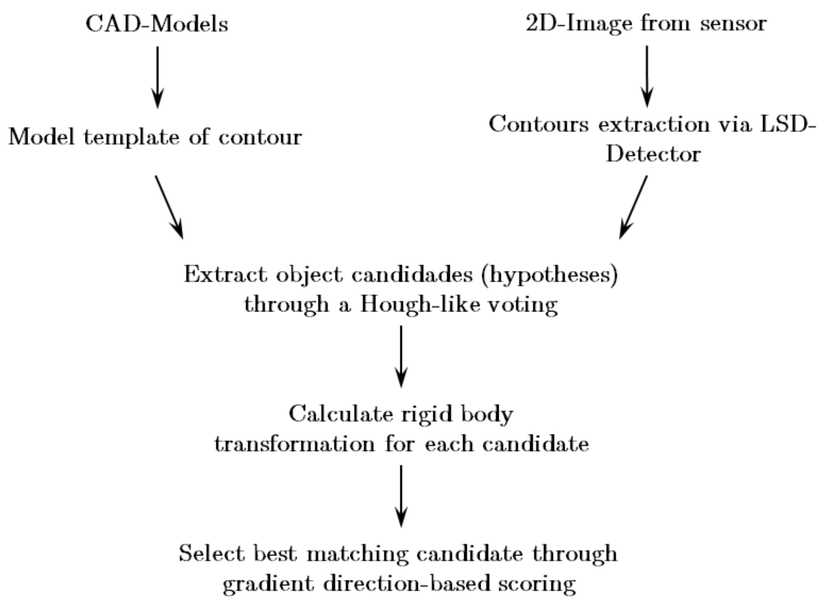


Figure 2.20: Recognition process according to (Pretto et al. 2013)

Table 2.19: Summary (Pretto et al. 2013)

Sensor technology:	2D-Image sensor, not specified
Recognition process:	See Figure 2.20
CAD-Representation:	Not specified
Recognition-Representation:	Contours from 2D-Image data
Software:	Not specified

8. Ravari and Taghirad propose a method for 3D-Object recognition from depth data using a complexity-based representation. Non-uniform rational basis splines (NURBS) are a representation form used by many CAD applications to efficiently represent curves and surfaces of three-dimensional objects. Starting from a point cloud captured from the scene, a first processing and outlier removal takes place, in order to optimise the data (see Figure 2.21). A least-squares fit of NURBS surfaces to the point cloud is then performed, generating a NURBS model of the point cloud. This NURBS model is then transferred into a complexity-based representation, which is invariant for transformations in the sense of Kolmogorov complexity. This complexity-based representation can now be compared with a complexity-based representation created from the NURBS representation of a CAD-Model. For this purpose, the common information between the two object descriptions is compared, using the normalised compression distance metric (NCD) as a comparative measure. If the agreement of common information exceeds a threshold value, an object is considered to be recognised. Table 2.20 summarises the evidence needed to answer the questions of this systematic literature review (Ravari and Taghirad 2014, pp. 902–907).

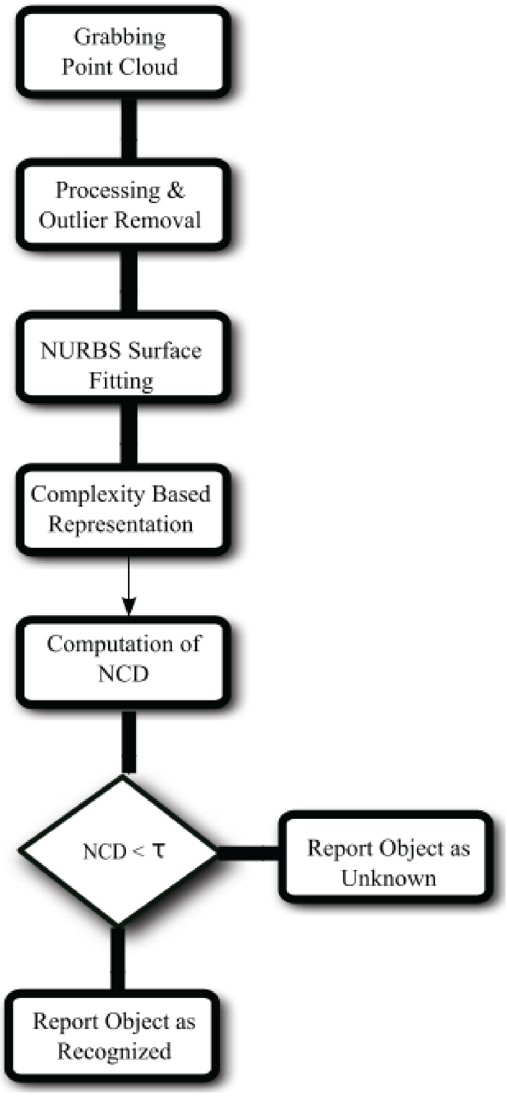


Figure 2.21: Recognition process (Ravari and Taghirad 2014, p. 904)

Table 2.20: Summary (Ravari and Taghirad 2014)

Sensor technology:	3D-Sensing technology, Microsoft Kinect
Recognition process:	See Figure 2.21
CAD-Representation:	Surface representation, NURBS
Recognition-Representation:	Complexity based representation derived from NURBS fitted to point cloud
Software:	Not specified

9. **Song et al.** present a 6-DoF pose estimation system for random bin-picking of multiple objects, containing an object recognition process (see Figure 2.22). As a basis for recognition, a 3D-Model database is created from CAD-Models (see Figure 2.23). A virtual depth camera is rotated spherically around a CAD-Model and the resulting depth image is converted into a point cloud. Subsequently, a voxel grid filter is applied in order to reduce the number of points and therefore the computational effort for the recognition. This filtered point cloud is then used to compute point-pair feature descriptors that are finally stored as a hash table within the 3D-Model database. Each known CAD-Model has one hash table in the database, that serves as an easy to look up description of the object. While this database generation process takes place offline, the actual recognition process takes place online. At the beginning of the online process the depth images of a 3D-Sensor are segmented into elements of interest. These depth image segments are then converted into point clouds. Using a voxel grid filter, the number of points within the point cloud is reduced and the surface normal is then determined for each filtered point. Subsequently, point-pair feature descriptors are calculated for further matching to the database. By means of voting scheme matching, the similarity of the descriptors in respect of hashes of the database entries with the point cloud from the scene is checked. Since transformations can be calculated for similar descriptors which transfer the scene point cloud into the model point cloud, the corresponding pose can be estimated. In order to improve the accuracy of the 6-DoF pose estimation, a pose clustering is then carried out. This refined estimation is then further improved using an iterative closest point (ICP) algorithm, which reduces the distance between the scene and the model point cloud. Table 2.21 summarises the evidence needed to answer the questions of this systematic literature review (Song et al. 2017, pp. 455–470).

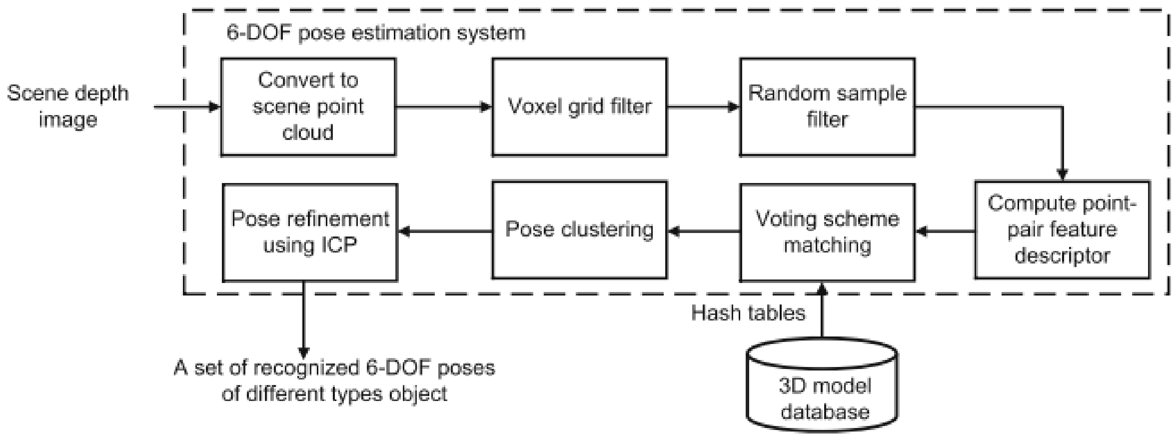


Figure 2.22: Recognition process (Song et al. 2017, p. 460)

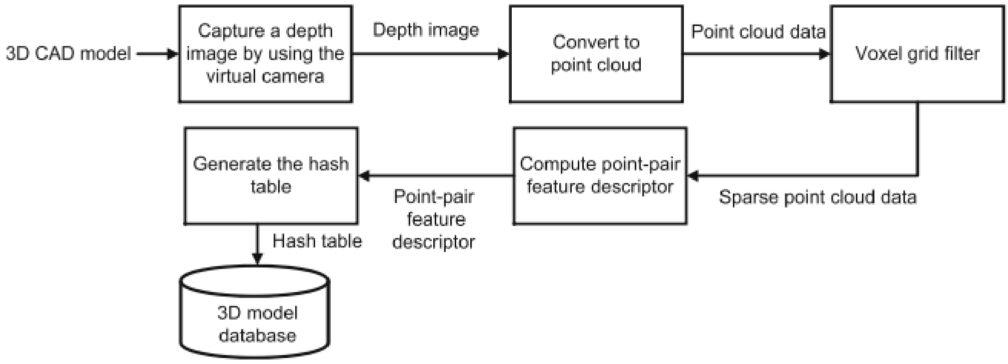


Figure 2.23: Generation of database for recognition (Song et al. 2017, p. 457)

Table 2.21: Summary (Song et al. 2017)

Sensor technology:	3D-Sensing technology, Microsoft Kinect
Recognition process:	See Figure 2.22
CAD-Representation:	Not specified
Recognition-Representation:	Point-pair feature descriptors from point cloud
Software:	Not specified

10. **Tsarouchi et al.** describe a method for detection of randomly placed objects for robotic handling. The proposed method includes an online and an offline system. The offline system uses CAD-Data to identify different poses for every object. Points of interest (POI) are determined for each pose, which are not described in detail. For each pose determined in this way, the coordinates of the POIs are determined from the CAD-Data and made available for the further recognition process. The online system starts with the processing of 2D-Image data from the scene using basic image processing techniques. After the pose of an object has been detected from the scene image, POIs are also extracted and compared with those generated from the CAD-Model. Finally, the pose information stored with the POIs of the CAD-Models is retrieved and can be transferred to a robot system. Figure 2.24 gives an overview of the recognition process and the mentioned systems. Table 2.22 summarises the evidence needed to answer the questions of this systematic literature review (Tsarouchi et al. 2016, pp. 20–27).

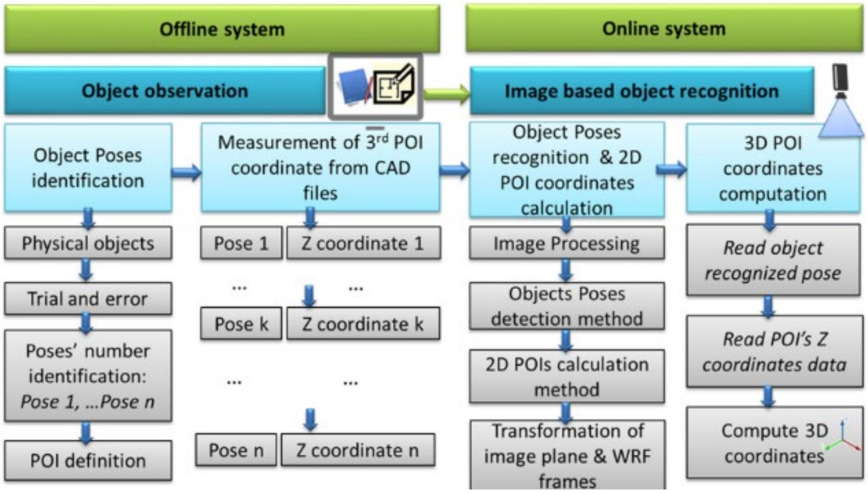


Figure 2.24: Recognition process (Tsarouchi et al. 2016, p. 21)

Table 2.22: Summary (Tsarouchi et al. 2016)

Sensor technology:	2D-Image sensor, not specified
Recognition process:	See Figure 2.24
CAD-Representation:	Not specified
Recognition-Representation:	Not specified
Software:	Matlab

11. Ulrich et al. present an approach for the recognition of 3D-Objects from single camera images, combined with the determination of their pose. In an offline process, different views of a CAD-Model are created by the spherical rotation of a virtual camera around it. The different view data are stored in the form of a special hierarchical tree structure to enable fast and efficient retrieval. For every view, edges and contours are extracted and stored together with the pose information. For recognition itself, 2D-Image data from the scene are analysed and existing edges and contours are also extracted. The tree-like search structure is then searched for matches using edge-based matching techniques. The best view determined from the search structure provides the pose information of the object, and refinement methods can be used to further improve these. Figure 2.25 shows the described process. Table 2.23 summarises the evidence needed to answer the questions of this systematic literature review (Ulrich et al. 2012, pp. 1902–1914).

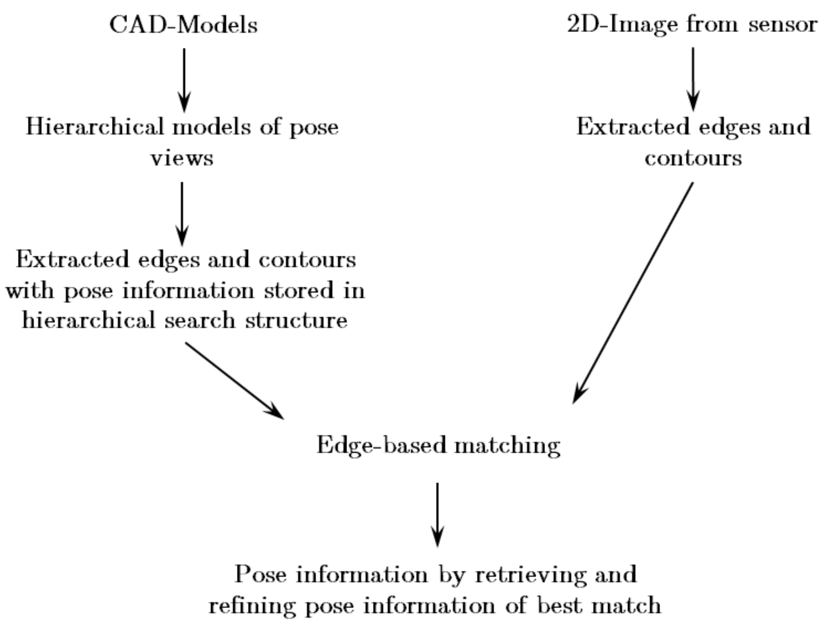


Figure 2.25: Recognition process according to (Ulrich et al. 2012)

Table 2.23: Summary (Ulrich et al. 2012)

Sensor technology:	2D-Image sensor, not specified
Recognition process:	See Figure 2.25
CAD-Representation:	Surface representation, not specified
Recognition-Representation:	Not specified
Software:	Not specified

2.2.2.5 Result of systematic literature review

The aim of this systematic literature review is to explore the latest research and practice in the field of CAD-based object recognition. Out of more than 650 documents, only 11 documents were considered representative. The main reasons for this extreme reduction are (see Table 2.12):

- More than half of the found documents are older than 10 years (EC6)
- Many documents don't describe the complete process for recognition of multiple different objects using CAD-Data as a basis (EC1)

Especially for the latter reason it should be mentioned that many publications only deal with single aspects of object recognition instead of describing the complete processes. This could be due to the fact that recognition systems are usually very specifically designed for a certain application and the definition of a generic process is therefore of limited applicability. Furthermore, most recognition systems are only designed to identify one or more entities of a specific object from a scene. Found systems that can handle several different objects are usually not used for identification but for classification and were excluded using criteria EC2. In particular, the approaches for the classification of objects use methods of artificial intelligence, whereas not a single approach was found which used artificial intelligence for identification.

The first question (Q1) of this systematic literature review deals with the sensor type and the input data used for the recognition algorithm. The evidence presented in Section 2.2.2.4 shows that both 2D-Image data from conventional image sensors (5 out of 11) and 3D-Image data from 3D-Sensing technologies (6 out of 11) are used. However, it is apparent that there is also a strong connection to the application. In particular, low-cost applications or applications on mobile devices like smartphones use 2D-Technology. If the applications require more precise recognition and many different objects are to be distinguished reliably, 3D-Technology is mainly used. The higher quality of 3D-Input data compared with 2D-Input data resulting from a projection onto an image sensor is crucial here. While little information was provided on the 2D-Sensing technology actually used, 3D-Sensors based on the principle of structured light seem to be the preferred choice.

In response to the second question (Q2), the processes extracted from the literature can be found in Section 2.2.2.4. For all processes found, regardless of their input data, recognition takes place by comparing features or descriptors. The basis for this is that both the features or descriptors from the scene and those of the CAD-Models from the database are available. All methods found initially extract the features from the CAD-Models in an offline process and save the features relating to their descriptions in a recognition knowledge base. Using 2D-Input data from the scene, projections of the CAD-Models onto a virtual camera serve as the basis for the extraction of contours and edges as features. Using 3D-Input data from the scene, the surface descriptions of the CAD-Models are either used directly or converted into point clouds. None of the approaches, whether 2D or 3D, used colour information for the recognition process. However, more approaches generate point clouds from the CAD-Models and describe them with special descriptors using the geometrical aspects between points as features. The offline process for generating the recognition database is followed by an online process for processing the sensor data from the scene. The counterpart to the offline process for generating the recognition database is an online process for processing the sensor data coming from the scene. All methods found preprocess the raw sensor data to reduce noise and other imperfections. Subsequently, features were extracted and described in the same way as in the offline process. The comparison of features is done in a matching step, using mathematical operations to calculate comparative measures that describe how well the features of the scene match with those in the recognition database. Since the features extracted from the sensor data are not complete and perfect, unlike the features extracted from the CAD-Models, practically no exact match can occur. A recognition is therefore considered successful if the comparative measure shows sufficient agreement. The extent of the required level of agreement cannot be uniformly defined and must be decided on an application-specific basis. After the successful recognition there is still the possibility of determining the 6-DoF pose of the object in relation to the image sensor.

Question three (Q3) is about the representations used for the CAD-Models as well as for the recognition. The analysed documents show many different forms of representation, which are listed in Section 2.2.2.4. Only two formats were mentioned for the CAD-Representation: STEP and DXF. All other documents refer to surface or wireframe representations of the CAD-Models. Accordingly, only CAD-Formats that describe the geometric characteristics of objects are used. No approach uses texture nor colour features nor CAD-Formats with this information. The recognition representations are the features or descriptors chosen with respect to either 2D- or 3D-Input data. Three documents give specific information on PPF- and CVFH-Descriptors used for recognition from point clouds. No specific representations are given for recognition on the basis of 2D-Image data.

The fourth question (Q4) served to identify the software used for 3D-Object recognition. OpenGL, Matlab and Point cloud library (PCL) were specified in the documents analysed. Only the last-mentioned, PCL, is a library specifically designed for 3D-Object recognition.

Table 2.24 shows the result of the quality assessment and selection of the documents reviewed. The best rated study (Luo and Kuo 2015) describes an architecture for 3D-Object recognition applying the well described algorithms in (Aldoma et al. 2011) and (Aldoma et al. 2012) that additionally make use of the openly accessible software library PCL. On the basis of these studies and the knowledge gained from this systematic literature review, a 3D-Object recognition system can be designed serving the purpose of this thesis. In the following sections the basics are explained, which are necessary to understand the 3D-Object recognition processes within these studies.

Table 2.24: Result of selection according to quality criteria

Document	QC1	QC2	QC3	QC4	Score
(Ahola et al. 2016)	4	4	3	4	15
(Aldoma et al. 2011)	4	3	1	2	10
(Han and Zhao 2015)	3	3	4	3	13
(Lee et al.)	4	4	1	4	13
(Luo and Kuo 2015)	4	4	4	4	16
(Pan et al. 2017)	4	2	1	4	11
(Pretto et al. 2013)	4	3	1	3	11
(Ravari and Taghirad 2014)	4	3	1	4	12
(Song et al. 2017)	4	4	1	4	13
(Tsarouchi et al. 2016)	4	2	3	1	10
(Ulrich et al. 2012)	4	4	2	3	13

2.2.3 Pipelines for 3D-Object recognition from point clouds based on CAD-Models

The sequence of steps required in order to recognise objects through computational algorithms is often referred to as a pipeline. This pipeline in fact consists of two processes, an offline training process and an online recognition process (Aldoma et al. 2012, p. 84)

In the offline training process, the CAD-Models, which are surface descriptive representations of physical objects, are transformed to point clouds. Two methods can be used for this transformation (Corsini et al. 2012, pp. 1–2; Aldoma et al. 2012, pp. 84–85):

1. Virtual rendering methods rotate a virtual camera around the CAD-Model and use ray-tracing algorithms in order to calculate the depth buffer. The rotation takes place either on the surface of a sphere or the vertices of an icosahedron, with the CAD-Model in the centre. By putting multiple snapshots together, the final point cloud is created.
2. Sampling methods sample individual points on the surfaces defined by the CAD-Model. For this purpose, the surface is first converted to a fine resolution polygonal mesh and then a number of points are sampled for each polygon according to its area.

Virtual rendering is especially suitable if only partial point clouds from certain views on the object are of interest (Aldoma et al. 2011, p. 585). The sampling methods are preferably used, if the whole CAD-Model is to be converted to a point cloud. After acquiring the full or a partial point cloud, descriptors are generated that later serve the matching.

With the online process for 3D-Object recognition, a distinction is made between two types of pipelines according to the type of descriptor used:

1. Local recognition pipelines make use of local descriptors. Local descriptors encode the properties of neighbouring points around individual so-called ‘key points’. Figure 2.26 shows the process flow for a local recognition pipeline. The right branch illustrates the mentioned offline process for knowledge base generation. Starting from the CAD-Model a point cloud of it is generated using either virtual rendering or the sampling methods as described above. From this point cloud certain key points are extracted by means of a key point detector or a so-called ‘voxel down-sampling’. Key point detectors should have two main characteristics: repeatability and distinctiveness. Repeatability on this context means that the detector chooses the same key points out of the point cloud, even if it is captured from a different view. Distinctiveness means, that only those key points are selected that are highly characteristic and descriptive and therefore support the matching process. Voxel down-sampling divides the three-dimensional space of the point cloud into voxels, which are cubes with a predefined edge length. Voxels are the equivalents of two-dimensional pixels in the three-dimensional space. Then every voxel is checked to determine if it contains any points. If so, the specific edge point of the voxel is passed as a key point, that is closest to the point or accumulation of points. After extracting the key points, a local description is computed for each one and stored in the overall descriptor. As already mentioned, the description concerns the neighbourhood of one key point. The type of features encoded varies for different

descriptor types and will be explained later. It is worth mentioning that if the chosen radius for the description around a key point is big enough to cover all points of the point cloud, there is no longer any difference from a global description. In this case, local descriptors are basically similar to global descriptors. The offline process ends with this step. Descriptors can be generated for several CAD-Models repeating these steps. The left branch in Figure 2.26 shows the online process, starting from the sensor point cloud. In a similar way to the offline process, key points are extracted using either a key point detector or voxel down-sampling. Then local descriptors are also computed for each key point and passed to the matching step. In the matching step, these scene descriptors are matched against all the descriptors of known objects generated in the offline process. For every possible pair of scene and model descriptors a distance metric is calculated during this step and determined whether it is good enough for applying a threshold. If there are corresponding descriptors between scene and model, these correspondences are passed to the next step. The list of correspondences contains pairs of scene and model key points whose neighbourhoods are similar, which doesn't necessarily mean the model is present in the scene. This is why correspondence grouping is applied in order to check, if the correspondences are geometrically consistent between scene and model. Only rotations and translations are allowed in order to achieve geometric consistency, where at least three correspondences are needed in order to determine a 6-DoF transformation. In this case, a rough estimation for the transformational matrix can be calculated in order to fit the model to the scene, which is called registration or pose estimation. After this step, the recognition has essentially already been carried out, but further steps can be taken to make the recognition even more precise. The first rough registration is refined using iterative closest point (ICP) algorithms, which minimise the error between the scene and the model point cloud. Finally, hypothesis verification serves to further reject false-positive recognition. In this step the overall similarity between scene and model point cloud are tested by leveraging geometrical cues. If similarity is found, an instance of the object in the scene is considered to have been identified (Aldoma et al. 2012, pp. 85–87).

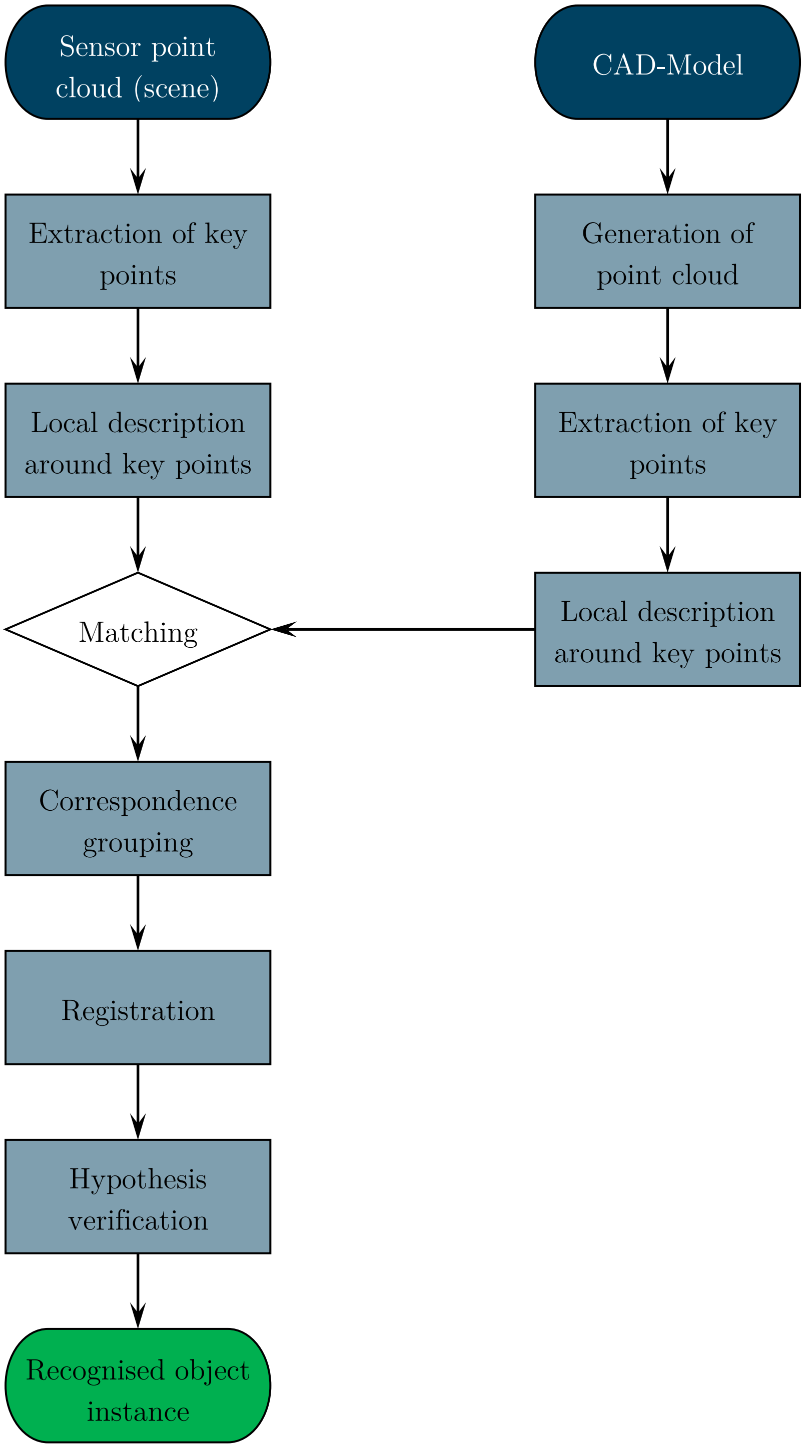


Figure 2.26: Local recognition pipeline, according to (Aldoma et al. 2012, pp. 85–87)

2. <u>Global recognition pipelines</u> make use of global descriptors. Global descriptors encode the properties of all points within a point cloud for which they are calculated. Figure 2.27 shows the process flow for a global recognition pipeline. The right branch again shows the offline training process. An important difference to the procedure for a local pipeline is the generation of only partial point clouds based on the CAD-Model. This is necessary because global descriptors describe a point cloud in its entirety and the sensory point clouds do not necessarily describe the complete object. Matching a descriptor that describes the ideal complete point cloud of a CAD-Model to a more or less complete descriptor of the sensor point cloud would falsify the recognition. Generation of partial point clouds that simulate different views of the object counteract this issue. After the generation of a partial point cloud a global descriptor is computed and the offline process ends with the provision of this descriptor for the matching process. The left branch in Figure 2.27 describes the online matching process. The point cloud of the scene recorded by a sensor is first fed into a segmentation process. Segmentation isolates the object of interest from the irrelevant background, resulting in a point cloud describing only the object. In the following step this extracted point cloud is described by a global descriptor and made available to the matching process. During matching, the descriptors of the scene are compared with all descriptors of known objects from the offline process, similar to the procedure described for the local pipeline. A distance metric is used to quantify the agreement of the global descriptors. In the absolute ideal case, the descriptors of scene and model correspond completely. However, this does not occur in reality, as the sensor-based and segmented point cloud would have to be identical to the partial point cloud derived from the CAD-Model. A threshold is therefore used to ensure a minimum level of agreement and potential candidates are ranked according to their quality of agreement. Once again, recognition has essentially been completed and can be further refined. Performing a registration process, the matching partial point cloud describing the model is aligned with the scene point cloud segment. In this step ICP algorithms are used to refine the registration and improve the alignment so that there is minimal error between the point clouds. Finally, the actual agreement between the point clouds is examined by means of hypothesis testing, as was already done with the local pipeline. By carrying out this step it can be finally determined whether an instance of the object was found in the scene (Aldoma et al. 2012, pp. 85–87).

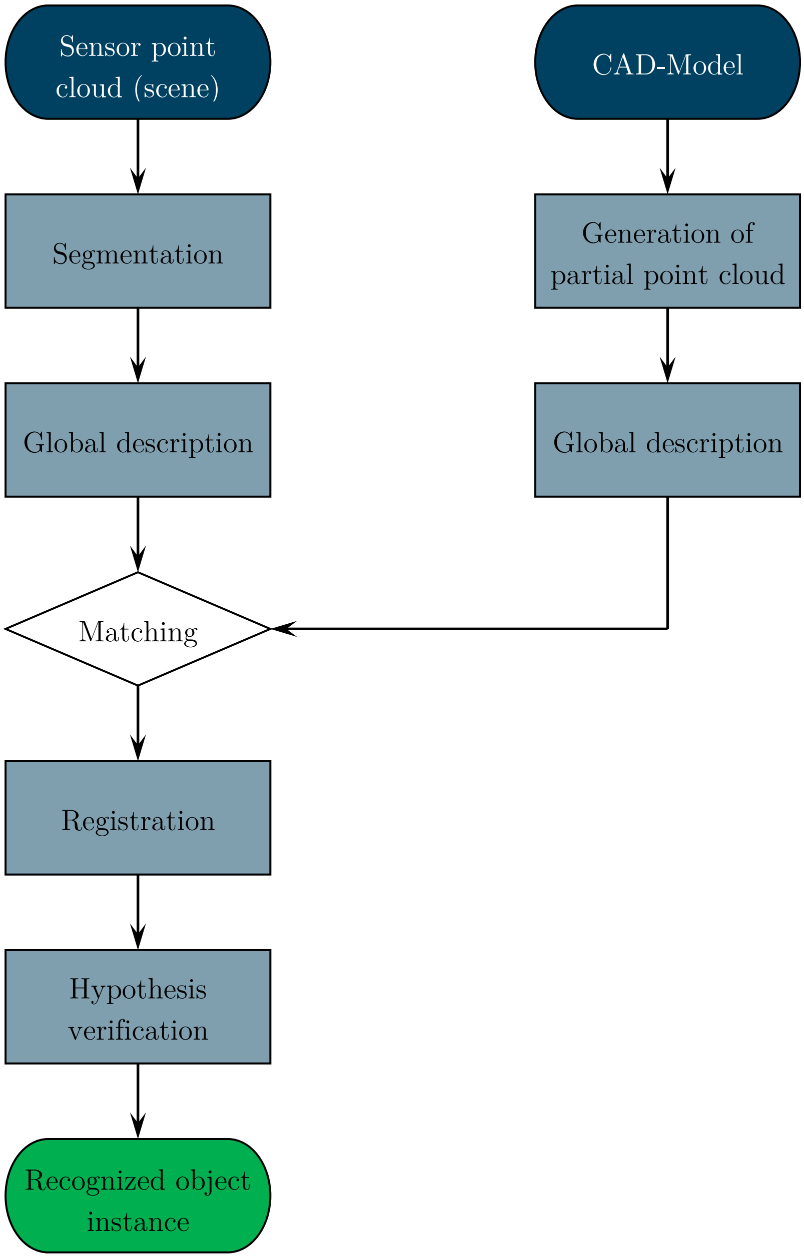


Figure 2.27: Global recognition pipeline, according to (Aldoma et al. 2012, pp. 85–87)

2.2.4 Descriptors for point cloud features

The integral meaning of point cloud descriptors for 3D-Object recognition can already be guessed from the previous section. Descriptors are the representations of objects for the matching process within 3D-Object recognition and encode the geometry and or appearance of known objects within the knowledge base or sensor data (Mateo et al. 2014, pp. 428–429). Descriptors for point cloud features must have the following essential characteristics (Hänsch et al. 2014, p. 59; Han et al. 2018, p. 2):

- <u>Robustness against noise</u> like sensor measurement errors
- <u>Computational efficiency</u> for creation and matching of descriptors
- <u>Invariance against rigid transformation</u> like non-scaled rotation or translation
- <u>Invariance against point cloud resolution</u> in order to enable reliable matching of point clouds with different resolutions

Since different descriptors encode different properties of point clouds, they also have different strengths and weaknesses. For this reason, descriptors are always selected for specific applications (Han et al. 2018, pp. 26–30).

The conducted systematic literature review (see Section 2.2.2) shows that only descriptors encoding geometry are used for 3D-Object recognition based on CAD-Data. In particular, so-called surface normals-based descriptors are used for the recognition (Mateo et al. 2014, pp. 428–429). Two specific descriptors, one local as well as one global, were selected due to their superior performance and the findings of the systematic literature review and are explained in the following (Mateo et al. 2014, pp. 433–434):

1. <u>Signature of histogram of orientation (SHOT)</u> is a local surface normals-based descriptor originally proposed by (Tombari et al. 2010). This descriptor encodes histograms of basic first-order differential entities, which are the surface-related normals of points within the spherical neighbourhood (support) of a given key point (see Figure 2.28). For this purpose, the isotropic spherical support is divided into 32 individual volumes, with 8 divisions along the azimuth, 2 along the elevation and 2 along the radius. A separate histogram is calculated for each volume, which describes the angular deviation of the normal of each point within the volume from the normal of the key point as a cosine value. This approach enhances the discriminative power of the descriptor by storing information concerning the location of points within the support, while also being robust to noise using histograms. The final descriptor is composed of all local histograms in the reference frame, which makes it invariant to rotation (Tombari et al. 2010, pp. 362–364).

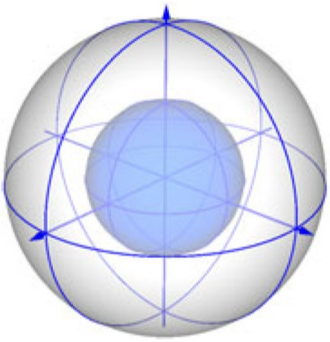
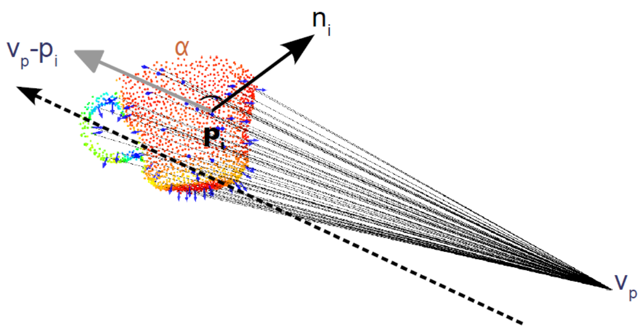


Figure 2.28: SHOT signature structure (Tombari et al. 2010, p. 363)

2. Clustered viewpoint feature histogram (CVFH) is a global surface normals-based descriptor originally presented by (Aldoma et al. 2011). It is an extension of the formerly presented viewpoint feature histogram (VFH), which was derived from a descriptor called a fast point feature histogram (FPFH) adding additional viewpoint invariance. The VFH descriptor consists of two components, one viewpoint direction component and a surface shape component (see Figure 2.29 a). The viewpoint component is a histogram of the angle between central viewpoint direction and each point's surface normal. As for the shape component, three angular properties between each query point and its neighbours in the form of histograms are computed and binned in 45 bins from a spherical support region. CVFH descriptors now in addition take advantage from stable object regions by applying a region growing algorithm after removing points with high curvature. For each stable region, VFH is computed and clustered with an additional shape distribution component (SDC) encoding information about the distribution of points around a region's centroid (see Figure 2.29 b). The SDC allows the differentiation of objects with similar characteristics, like size and normal distribution, from each other (Rusu et al. 2010, pp. 2158–2159; Aldoma et al. 2011, pp. 586–587).

a)



b)

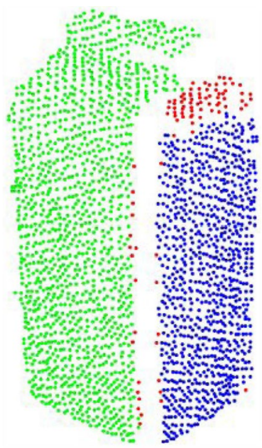


Figure 2.29: CVFH descriptors: a) VFH components (Rusu et al. 2010, p. 2158), b) Stable regions clustering (Aldoma et al. 2011, p. 588)

Once descriptors have been computed for both the scene and each model, the matching steps aim to yield corresponding ones. Due to the different nature of descriptors, different matching strategies are pursued. The recommended matching strategies of the two descriptors described above are explained (Aldoma et al. 2012, pp. 85–87):

1. **Nearest neighbour search (NNS)** structures like k-dimensional trees (k-d trees) are used for matching local descriptors. K-d trees are binary space partitioning trees with the aim of finding an element within the tree that is closest to a given input element. The element for descriptor matching is a multi-dimensional vector. For every scene key point descriptor, the entire set of key point descriptors of each known object is searched and the similarity is checked using similarity measures, which are distance metrics. A threshold determines whether the similarity is sufficient or not.
2. **Brute force search** is used for comparison of global descriptors. This type of search compares all possible combinations of scene descriptor and descriptors of all known objects to find the best match. Again, the comparison is evaluated using similarity measures or distance metrics. The candidate with the best matching descriptor, has the highest probability to be the object in the scene.

The distance metrics recommended for determining the similarity of SHOT and CVFH descriptors are the Euclidean distance d_{L2} or Chi-squared distance d_{χ^2}, which are defined as follows, between two n-dimensional vectors p and q (Mateo et al. 2014, p. 430; Aldoma et al. 2012, p. 86):

$$d_{L2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.1}$$

$$d_{\chi^2} = \sum_{i=1}^{n}\frac{(p_i - q_i)}{(p_i + q_i)} \tag{2.2}$$

2.2.5 Optical 3D-Sensing technology for vision systems

In three-dimensional machine vision, various optical sensors are used to perceive the physical world. Sensors for spatial perception are based on two fundamental physical principles (Blais 2004, pp. 231–236):

1. **Triangulating methods** make use of the trigonometric properties of triangles. Using two known points in space, whose distance is known, the position of any other point can be determined by means of angle measurement. Figure 2.30 shows the principle of triangulation. Starting from two stations S_1 and S_2 for which the position and thus distance c is known, the angles to an object point P are determined. Applying trigonometry, the length of the sides a and b of the triangle can now be calculated and thus the position of the object point P can be determined (Suk and Bhandarkar 1992, p. 25).

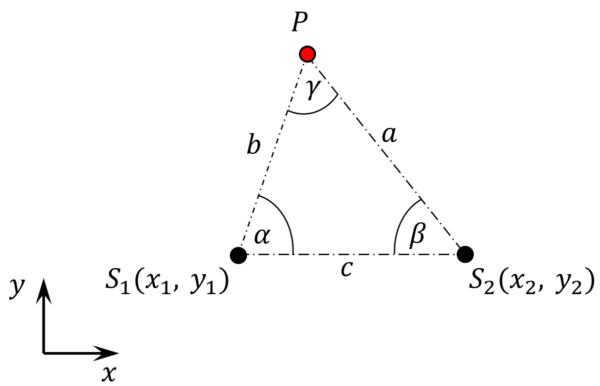


Figure 2.30: Principle of triangulation

2. **Time-of-flight (ToF) methods** use the velocity of propagation of light or sound waves and measure the time between the emission of the wave and the return of its reflection. The laws of kinematics can subsequently be applied to determine the distance that the light and the sound wave have travelled respectively. Half of this distance thus corresponds to the distance to a measuring point. Figure 2.31 shows the underlying principle of ToF procedures. Starting from a station S_1 a wave is emitted in a known direction α. The time period t between the emission of the wave from S_1 and the return of the reflection from P to S_1 is determined exactly. Time period t is measured either from the phase shifting of the waves or by emitting single pulses. Using the propagation velocity of light c_0, the distance d between S_1 and P can now be calculated. With the knowledge of the wave direction α and the distance d, the position of the point P can now be determined (Suk and Bhandarkar 1992, p. 21).

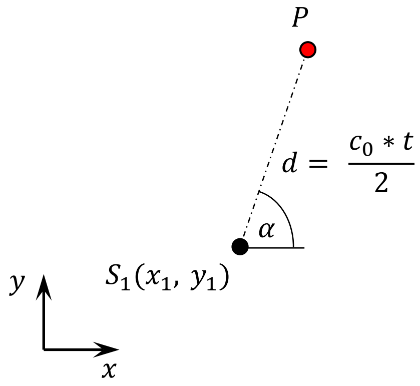


Figure 2.31: Time-of-flight principle

2.2.5.1 Stereo vision sensors based on triangulation

Stereo vision describes the acquisition of depth information by comparing and analysing corresponding (matching) features from two or more camera images (Bradski and Kaehler 2011, p. 405). Features can be prominent pixels, contours or textures. In traditional or passive stereovision, images are captured with two fixed, synchronised cameras. In order to obtain depth information from the image data (pixels), four essential steps are performed (Bradski and Kaehler 2011, p. 415):

1. Undistortion describes the mathematical elimination of the radial and tangential lens distortion caused by the image sensors. The result of this step is undistorted image data.
2. Rectification is the process that adjusts the angles and distances between cameras. Afterwards the image data is row-aligned (y-direction) and rectified, which means the image planes are coplanar and the pixel-rows are aligned.
3. Correspondences in the format of corresponding features between the image data are searched for and a so-called disparity map is created. The disparity map contains the distance differences in x-direction (difference of x-coordinates) of all corresponding features.
4. Reprojection describes the transformation of the disparity map into distance or depth information by triangulation.

Steps 1 and 2 are standard mathematical procedures and will not be explained further at this point, as they do not contribute much to understanding. Figure 2.32 shows an idealised setup for stereovision. The optical axes (grey) of the two image sensors are ideally parallel and the image planes (blue) are coplanar. In this so-called frontal parallel arrangement, a physical point P will produce an image view on the left p_l and right p_r image planes. The difference of the x-coordinates x_l^p and x_r^p of the two images p_l and p_r of the point P on the images represents the disparity. Mathematically, the disparity is defined as follows:

$$d = x_l^p - x_r^p \tag{2.3}$$

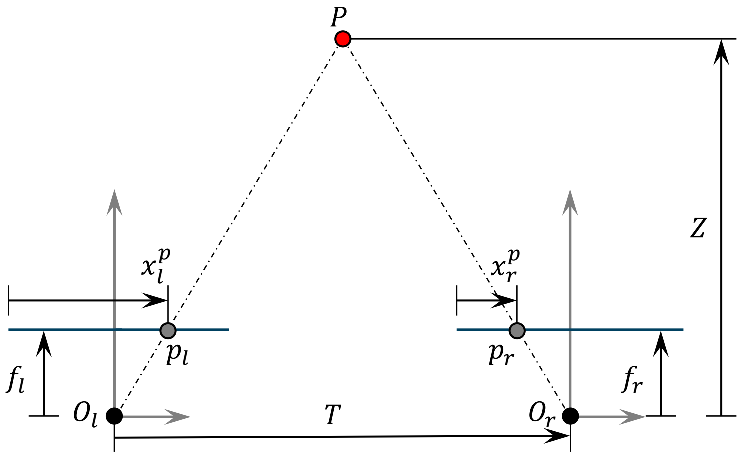


Figure 2.32: Principle of stereovision with ideally aligned cameras, adapted from (Bradski and Kaehler 2011, p. 414)

Figure 2.32 also shows that for large Z the disparity d tends towards zero and for small Z it tends towards infinity. Assuming that the focal length $f = f_l = f_r$ and the distance between the image sensors T is known, the following relationship between depth Z and disparity d can be formulated by triangulation:

$$Z = \frac{f * T}{d} \tag{2.4}$$

The depth Z is calculated in relation to the centres of projection O_l and O_r, as Equation (2.4) suggests the depth Z and the disparity d have an inversely proportional relationship. This non-linear behaviour leads to the consequence that stereo vision has a high depth resolution only in the relatively close range, where the function $Z(d)$ is almost linear (see Figure 2.33). If the disparities for all corresponding features in the images are determined and compiled in the form of a disparity map, a so-called depth map containing all depth information can be derived using Equation (2.4).

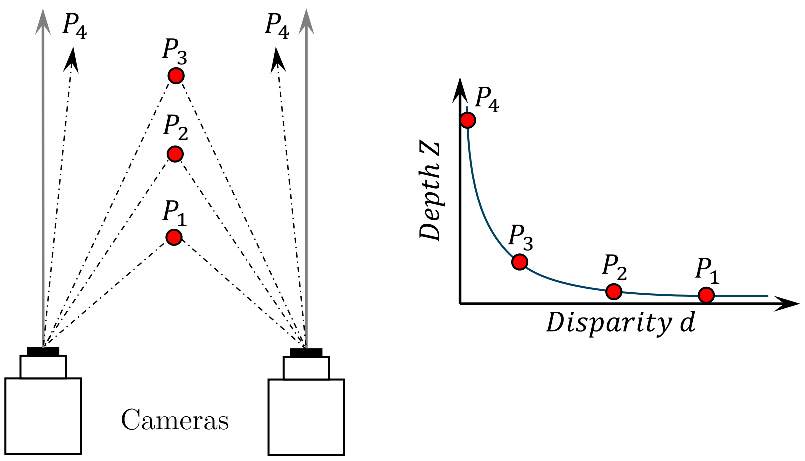


Figure 2.33: Relationship of depth and disparity in stereo vision, adapted from (Bradski and Kaehler 2011, p. 417)

For the measurement of static objects or scenes, the images can also be captured by moving a single camera, which is called "mono-stereo vision" or "motion-stereo vision" (Suk and Bhandarkar 1992, p. 27; Bradski and Kaehler 2011, pp. 453–454). This technique is nowadays widely used for 3D-Scanning with mobile devices.

The passive form of stereo vision explained so far has the disadvantage that only naturally existing features in the scene can be used to gain depth information by matching corresponding ones. This matching process in particular is still difficult today and is therefore an ongoing object of research in 3D-Sensing technology (Lazaros et al. 2008, p. 458). This limits the perception of passive stereo vision to objects that have clearly perceptible contours or textures on their surface (Jang et al. 2013, p. 1255). In industrial applications, object surfaces only rarely show such features. This disadvantage can be compensated for by artificially applying a texture to the surfaces of objects. So that objects do not have to be physically adapted, textures can be applied using a projector, which is then called active stereo vision (Je et al. 2004, p. 95; Jang et al. 2013, p. 1255). Figure 2.34 shows a setup for active stereo vision, with two cameras and one projector.

A variant of the active stereo vision uses the structured light approach (see Section 2.2.5.2) and is therefore called structured-light stereo. The combination of the two basic approaches creates synergies that lead to higher accuracy, a larger field of view and lower sensitivity to environmental influences (Jang et al. 2013, p. 1264).

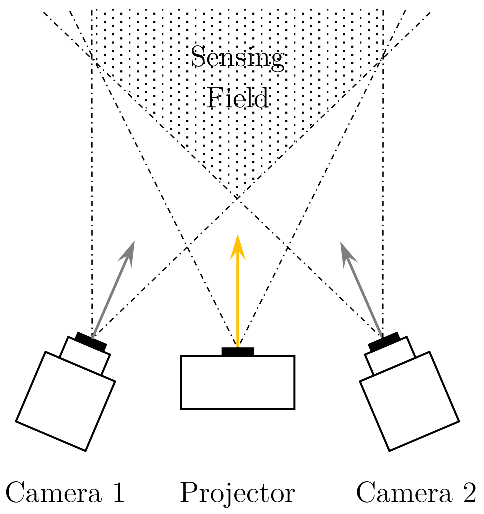


Figure 2.34: Setup for active stereo vision

2.2.5.2 Structured light sensors based on triangulation

Another option for obtaining depth information from a scene is the structured light method. Structured light is based on the projection of one or more special light patterns onto a scene. Figure 2.37 shows a setup for structured light depth sensing. In contrast to stereovision, conventional structured light approaches use only a single camera and projector to directly gather a depth map from a scene (Scharstein and Szeliski, 195). Figure 2.35 shows the underlying principle of structured light depth acquisition. Light source E emits a point beam. An image sensor is oriented so that its optical axis (grey) intersects the beam of E at a point P_T at the angle α. This configuration creates an image p_T of the point P_T in the middle of the imager (blue), with pixel-coordinate x_T. Assuming that focal length f, distance T between the calibrated point P_T, the centre of projection O as well as the angle α are known, the depth Z can be determined for a measuring point P_m. Point P_m creates an image p_m on the imager whose pixel-coordinate x_m has a shift s against x_T:

$$s = x_m - x_T \tag{2.5}$$

From this shift s the depth Z can be determined as follows:

$$Z = \frac{T}{\left(\frac{s}{f \ * tan\,\alpha} + 1\right)} \tag{2.6}$$

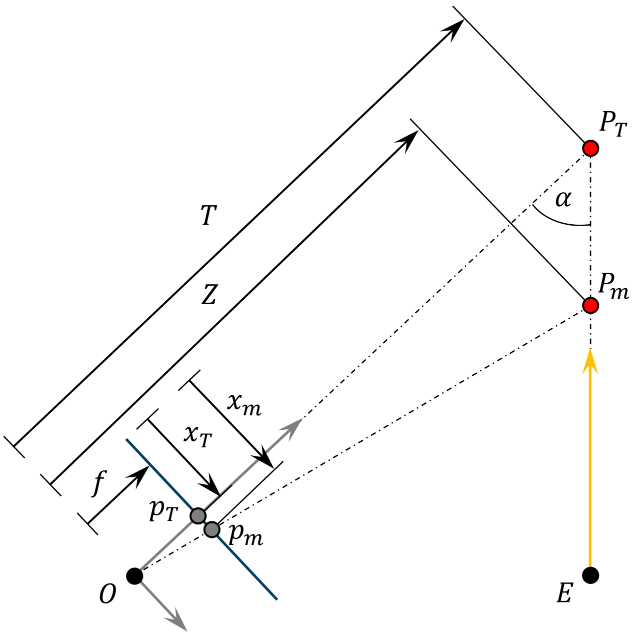


Figure 2.35: Principle of structured light depth acquisition, adapted from (Batchelor 2012a, p. 1525)

Equation (2.6) shows that the sensitivity of the measuring arrangement can be influenced by clever adjustment of angle α in relation to the parameters f and T. This makes it possible to achieve an optimal resolution over a variety of measuring ranges, which is an advantage over the classic stereo vision described in the previous section (Besl 1988, p. 135).

The principle described so far is based on a point-shaped light source and thus uses simple single-point triangulation. It would therefore be necessary to perform spherical scanning movements with the measuring device in order to capture entire scenes point by point. Therefore, further approaches were developed, which use lines or line patterns as projections and thus can capture large areas without making scanning movements (Blais 2004, pp. 232–235). However, the declared basic principle remains the same for all procedures based on structured light.

Light-sectioning methods calculate a depth profile along a single line or a line pattern projected onto an object's surface (Blais 2004, pp. 233–234). This is done by analysing the distortion of a line caused by observing it from a different perspective than that of the projector (see Figure 2.36).

Coded-light approaches are an extension of the light sectioning methods. For the dense measurement of surfaces, a large number of lines have to be projected in dense distances. In order to avoid confusion of these lines, binary patterns are projected in time sequences. A frequently used binary pattern is the Gray-code (Horn and Kiryati 1999, pp. 87–89).

Phase shifting methods (also called dynamic structured lighting) project a sinusoidal brightness modulated signal onto a surface to be measured. This brightness modulation is detected by an image sensor. If the signal emitted by the projector is subsequently cosine modulated, a height profile can be calculated from the phase shift.

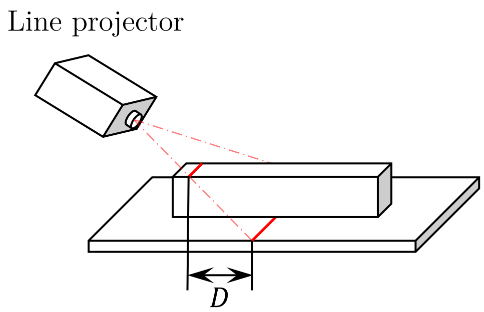


Figure 2.36: Distortion of projection due to light section

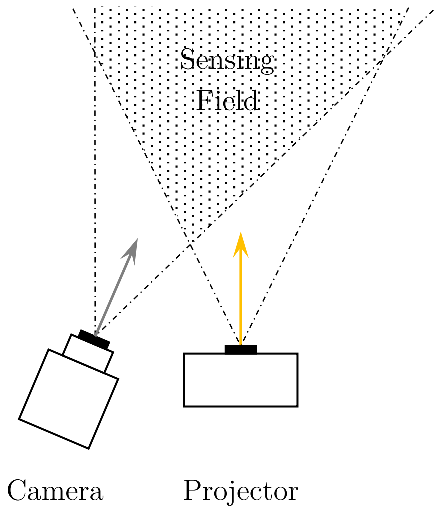


Figure 2.37: Setup for conventional structured light vision, adapted from (Jang et al. 2013, p. 1257)

2.2.5.3 Scanning and non-scanning sensors based on time-of-flight

As already mentioned, ToF methods can use either sound or light waves for measuring depth within scenes. For 3D measurement tasks in industry, laser-based technologies are heavily used (Hansard et al. 2013, pp. 1–2). A fundamental distinction is made between scanning and non-scanning ToF procedures (Ailisto et al. 2001, p. 2; Zhao 2012, pp. 627–630).

In classic ToF scanning, a light beam is deflected by means of an optical system and thus enables the measurement of points in space. Depending on the nature of the optics and its rotational degrees of freedom, the scan field can be circular or spherical. Sensors that follow this principle are also referred to as light detection and ranging (LIDAR) sensors (Fang et al. 2018, pp. 1–2).

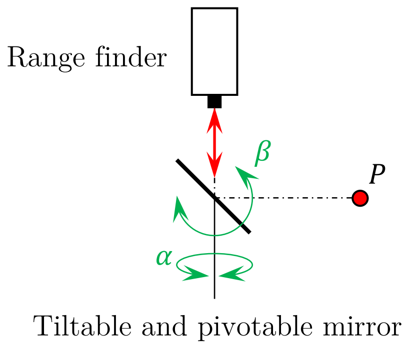


Figure 2.38: Principle of scanning time-of-flight depth acquisition, adapted from (Zhao 2012, p. 628)

Non-scanning ToF approaches are still the subject of research and are relatively new. Depth information is collected from the scene using so-called ToF-Cameras. ToF-Cameras are very similar to ordinary cameras and consist of a light source, optical components, control circuitry, processing circuitry and functional units. However, there is a key component that replaces the image sensor of conventional cameras which is called a ToF-Chip. The ToF-Chip incorporates active light detection, allowing one to detect the phase shifts of emitted and reflected light waves to be detected for individual pixels. According to the time-of-flight principle, depth can thus be obtained for each of these pixels (He and Chen 2019, p. 12495; Zhao 2012, 631-630).

As of today, ToF cameras with their unique sensor architecture still have some weaknesses such as low spatial resolution, low depth precision, distortions due to geometric, radiometric and illumination variations (Hansard et al. 2013, pp. 4–12; He and Chen 2019, p. 12496). Due to their advantages such as near real-time depth acquisition in video rate and compact design, the cameras are nevertheless used in industrial applications (He and Chen 2019, p. 12496).

2.3 Product data management

Product data management describes the storage and management of product-related data originating from product development. These product-related data can be product-defining, presenting or representing. Within the larger framework of product life cycle management (PLM), PDM provides and manages product data and thus becomes the informational backbone for all downstream phases after development (see Figure 2.39). The foundation for this data provision along the entire product life cycle is an integrated product model (Eigner 2014a, pp. 268–271; Wiendahl 2019, pp. 143–145).

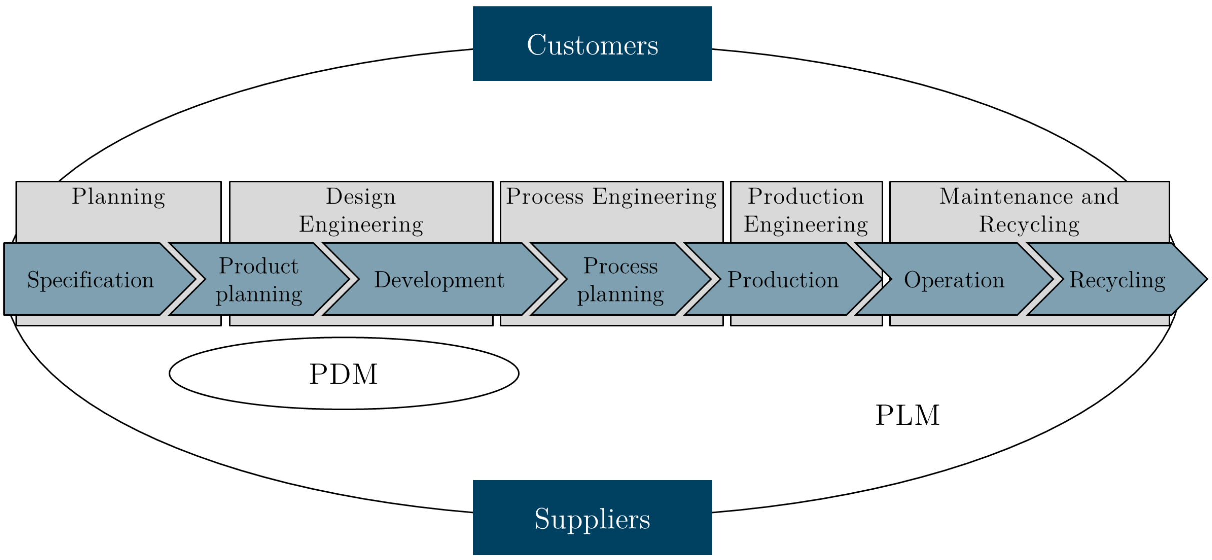


Figure 2.39: PDM within the framework of PLM, adapted from (Eigner 2014a, p. 270)

2.3.1 Product models

Product models have the objective of virtually representing products with their information relevant to the entire life cycle. Product models consist of two components (Eigner 2014a, p. 272):

1. Product master records and product structures describe the allocation of product components to each other. These components can be raw materials, semi-finished products, individual parts, sub-assemblies or other products. The overall structure is expressed in the form of an engineering bill of materials (EBOM).
2. Documents and document structures are closely related to the product master records and structures. In this context, a document is a summary or compilation of information, handled as a unit, which is stored in a non-volatile form on an information carrier (DIN 6789:2013-10, p. 5). In particular, "technical documents shall be of the type and completeness necessary for technical purposes" (DIN 6789:2013-10, p. 6). Document structures provide the uniform formulation and allocation of documents (Eigner 2014b, pp. 249–253).

Each node of the product structure or bill of materials item can be assigned any number of descriptive documents respectively, in order to provide product information. Product information is divided into technical, commercial and quality information (see Figure 2.40). Product-defining technical information is of particular importance for this thesis. Technical product information is again differentiated into geometrical, technological, system and organisational information.

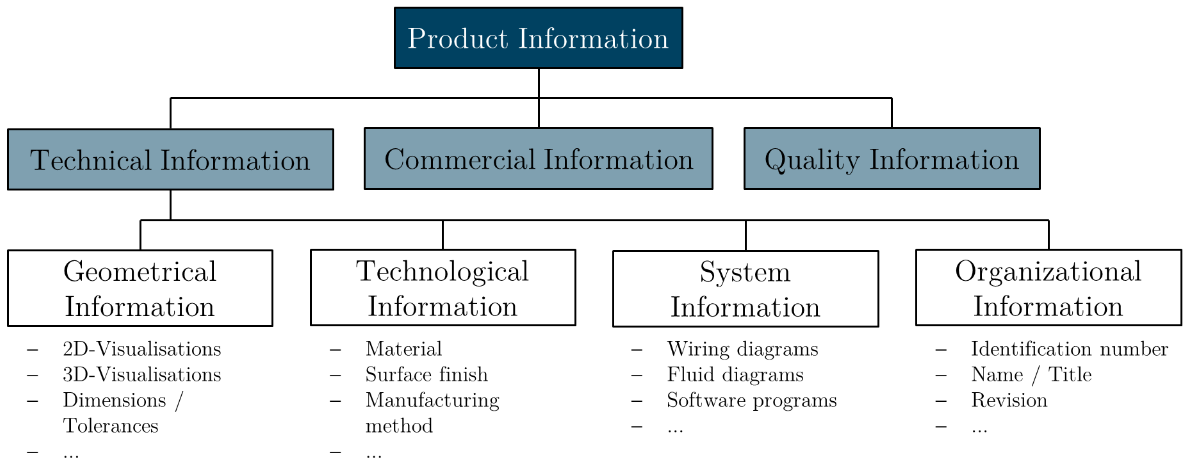


Figure 2.40: Product information structure, adapted from (Eigner 2014b, p. 250)

The central industry standard for the cross-system and cross-company exchange of product data respectively is called standard for the exchange of product model data (STEP), whose latest protocol is defined in (ISO 10303-242:2020-04). In particular, STEP enables the exchange of a wide range of computer-aided (CAx) data, such as geometric data, product structure or organisational data to describe products throughout their life cycle. The STEP standard focuses on product data representation to support CAx- and PDM-Data exchange, system integration, visualisation and long-term storage of product information (Rosche et al. 2009, pp. 28–29). This is made possible by combining a number of partial models for different technical fields of application (electronic engineering, mechanical engineering, production engineering, etc.) as an integrated product model (Eigner 2014a, p. 272). This integrated resource model defines the following aspects (Hehenberger 2020, pp. 142–143):

- Basics of product specification and administration
- Geometry and topology
- Product structure representation
- Material data
- Visualisation
- Tolerance data
- Process structures

2.3.2 Three-dimensional CAD-Models

The explicit models that describe geometric and technological information are generally known as CAD-Models. Whereas the first CAD-Models were two-dimensional or two-and-a-half-dimensional models, today 3D models are almost exclusively used for product definition (Roubanov 2014, p. 118).

Three-dimensional CAD-Models can, depending to their representation of geometry be categorised in:

- Wire-frame models represent a physical object only by their edges and corners (see Figure 2.41 a). As the name suggests, this results in a three-dimensional frame, which is constructed from basic geometries such as lines, arcs, ellipses or free-form curves. This description is mathematically and computationally less demanding, but also has a low information content compared with other forms of representation
- Surface models describe the surface or hull of a physical object (see Figure 2.41 b). Surface modelling can be divided into two types, analytically describable and non-analytically describable geometries. The analytically describable geometries allow a curve or a surface to be described exactly, whereas the analytically non-describable geometries (so called free-form curves or free-form surfaces) only approximate or interpolate them.
- Solid or volumetric models describe physical objects by volume elements (see Figure 2.41 c). Two basic methods of description can be used for this purpose: Constructive solid geometry (CSG) and boundary representation (BREP). In CSG modelling, an object is created from volumetric primitives such as squares or cylinders by applying set-theoretical operations. BREP modelling describes the volume of physical objects by their boundary surfaces. The intersection lines of these boundary surfaces result in the object edges.

Although CAD-Models can serve as a starting point for complex rendering processes to create photorealistic images, the main focus is on the geometrically and technologically exact description of physical objects (Um 2018, pp. 2–5; La Rocca 2012, p. 173; Mustafa et al. 2017, p. 1; Kirkwood and Sherwood 2018, p. 1).

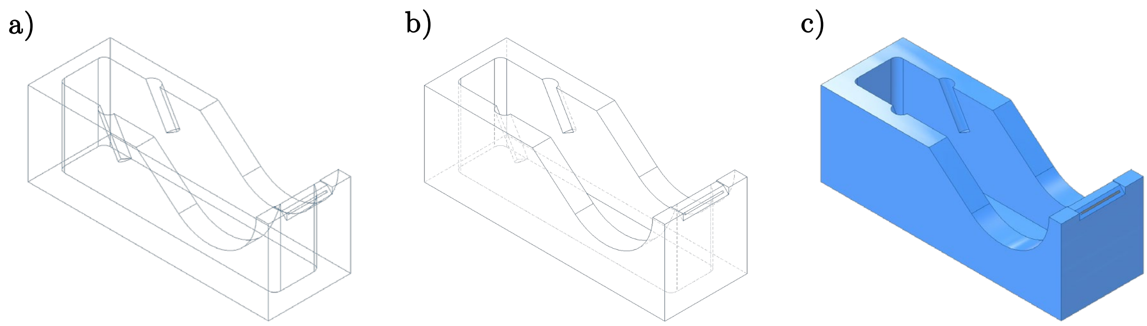


Figure 2.41: CAD geometry representations: a) Wire frame model, b) Surface model, c) Solid model

The geometric forms of representation explained above are used in explicit CAD-Data formats for the representation of physical objects. These data formats appear both as software manufacturer-specific (native) formats and as software manufacturer-neutral formats (Kirkwood and Sherwood 2018, p. 2). The latter is also called exchange format, because these formats are widely used for cross-system and cross-company data exchange by exporting these formats from native formats (Mustafa et al. 2017, p. 1). Common to all data formats is that additional metadata are stored, which provide further information in addition to the geometric information. In the literature, almost no information is given about the actual information available in the metadata of CAD-Data formats. However, a well-known issue is that the export of native to exchange formats leads to significant loss of information, both metadata and geometric data (Mustafa et al. 2017, pp. 3–4; Kirkwood and Sherwood 2018, pp. 2–3). The included metadata of common native CAD-Formats are (Hehenberger 2020, pp. 142–143) :

- Descriptions
- Identification numbers
- Material properties (Description, mechanical characteristics)
- Surface properties
- Mass properties (Weight, centre of gravity position, moments of inertia)
- Tolerances

2.4 Conclusion of literature review

In order to critically examine the knowledge found in the literature and to answer SRQ1, a conclusion is drawn at this stage. As of today, it is impossible to imagine industrial value-added systems without automatic identification technology. Since the introduction of the barcode in the 1950s, automatic identification technology has undergone constant development. In the future, automatic identification systems will continue to play an important role and become a fundamental component of cyber-physical systems and digital factories, as they are the linking technology between the virtual and the physical world. A wide variety of different automatic identification technologies have emerged which use different, mostly codified, artificial identifiers to enable identification. As shown in Section 2.1.6, there is no common understanding of the classification of automatic identification systems. Most of the approaches found distinguish automatic identification technologies on the basis of the physical principle of data acquisition. One approach first distinguishes between automatic identification technologies that contain information directly or indirectly, but this only describes whether information is directly interpretable or whether the identifiers merely provide a link to the actual information. Moreover, these classifications only refer to indirect automatic identification technologies, i.e. those technologies that use artificial identifiers for identification purposes. This is probably due to the fact that direct identification on the basis of natural object characteristics has played a subordinate role up to now and that there are hardly any approaches for this type of identification (see Section 2.1.7). Another very important discovery in the literature related to automatic identification technology is the fact that a multitude of meanings are attributed to the concept of identity. Most authors, especially in the context of Digitalisation and Industry 4.0, speak of unique identification by means of RFID-Technology without explaining this further. In fact, there are different types of identity, which are expressed in particular through industrial numbering systems (see Section 2.1.3). The first type of identity enables the unique identification of a specific object within an object group, which also means the assignment of one specific set of information to this object. The second type, on the other hand, makes it possible to identify an individual object as a member of a group of objects or a subgroup of objects, thus making it possible to assign a set of information written for this group or subgroup. The type or level of identity is always chosen to suit the application and thus the desired accuracy of identification. It is very important to recognise that an identity only exists within an associated reference system which knows its definition. For example, the mere knowledge of an identification number of the object is useless if it cannot be assigned to a defining set of information. In the industrial environment the objects that identified are all of the types of goods that occur in the industrial material flow. In particular, identification is carried out for piece goods, packaged goods, load carriers and loading units. The identification of these objects within the material flow systems serves the synchronisation of material flow and information flow and thus supports all related business processes.

Machine vision techniques are already being used in a variety of different applications in the industrial environment. In the sense of automatic identification, mainly 2D-Image data from conventional cameras are processed to detect artificial identifiers like visual codes. However, machine vision offers much more potential, which can be exploited for the purpose of automatic identification. Specifically, the methods of 3D-Object recognition allow the identification of objects by their natural characteristics, which are available without additional costs or efforts. These methods have not yet been considered in the literature on automatic identification technology, although they have great potential for the realisation of direct identification systems. There are various approaches available, which differ in particular in the type of sensor used. Approaches using 3D sensors are considered in the literature to be more powerful for this application, as they provide additional depth information for evaluation, which conventional 2D-Cameras are not capable of. The increasing affordability and availability of depth sensors further supports this reasoning. One basic prerequisite for automatic identification based on 3D-Object recognition is, of course, that the objects to be identified are distinguishable at all on the basis of their 3D-Geometry. Unfortunately, this requirement does not necessarily apply to packaged goods, load carriers and loading units, which are typically of a more uniform nature. In the case of the unpackaged piece goods, the possibility of identification on the basis of their geometry does indeed exist, but here too, only the external surface features are perceived, which is why the mere use of optical features limits the possible distinctiveness. At this point, the use of appearance information, such as the texture or colour of objects, would be of interest in order to provide a further visual distinguishing feature. Although machine vision provides methods for this purpose, these are not considered in the literature to be very precise or suitable for application in an industrial environment. This is due to the high susceptibility of such methods to lighting influences and colour deviations, as well as the fact that many industrial objects have no texture or are even reflective. The use of mainly geometric features of objects for recognition has also led to the development of CAD-based approaches. For this purpose, CAD-Models are used to generate the knowledge base of known objects for the recognition itself. With regard to the industrial product life cycle, these CAD-Models are created during the development and the design engineering phases respectively, and serve as a representative and defining basis for all further steps.

The storage and management of these product models containing all the product information falls within the scope of product data management, which takes place in the superordinate framework of product life cycle management. The geometrical and technological information from the category of technical product information is of particular interest for this work. In the literature there are no direct specifications for the necessary information in the technical documentation, presumably because this can vary greatly depending on the product and therefore relevant standards must be observed. However, the CAD-Models used for product documentation contain, in addition to the geometry, further metadata, which include information on materials, surface properties, mass properties and tolerances. Unfortunately, there is no openly accessible information in the literature on the structure of software manufacturer-specific CAD-Formats, which is why an extraction of the geometrical and metadata is not easy. Not least for this reason, various standards for neutral data exchange formats have developed, although in most cases there is also no uniform definition of the metadata to be carried along. A positive exception here is the STEP-Format, which is based on various partial models in the form of protocols that serve to define a structure for various technical product information. This format at least offers the possibility of a standardised, software manufacturer-independent definition of CAD-Models, but the actual information content remains dependent on the export interface written by CAD-Software manufacturers.

After the analysis of the best available in all fields involved in the research problem, the development of an automatic identification system for the direct identification of components on an article number level seems possible. In the literature on automatic identification technology there are only few attempts to implement such a system. These attempts are limited to the analysis of local surface features and not to the overall geometry. In the literature on 3D-Object recognition, there are approaches that enable the identification of objects on the basis of CAD-Models. However, since 3D-Object recognition only refers to optically perceivable surface geometries, the differentiation of objects during identification is also limited. This obstacle can be removed by using the technical product information from PDM, which is anyway contained in CAD-Models. Material properties and mass properties seem to be of particular interest here. Appearance properties present some difficulties due to the technical possibilities with CAD formats and optical recognition, which have already been described above. Nevertheless, a colour differentiation feature could be useful for identification. By recombining and expanding existing approaches, an automatic identification system for the direct identification of components can be developed, which at the same time can be integrated into existing data structures of product lifecycle management.

The following chapter deals with the development of a multi-sensor AIS, which has the characteristics described above.

3 Development of multi-sensor automatic identification system

This chapter is dedicated to the development of the envisaged automatic identification system (AIS) for direct identification of unpackaged piece goods. For the development of such a mechatronic system consisting of hardware and software components, different engineering design models are available. The development procedure used in this work follows the waterfall model originally presented by (Royce 1970) and has since been successfully used for the development of mechatronic systems (Eigner 2014c, pp. 42–43). Figure 3.1 shows the model applied in this thesis with its consecutive phases. The choice of this specific model is based on its simplicity and the fact that going through several development cycles would be beyond the scope of this thesis. The following sections reflect the individual phases of the waterfall model, whereby the description of the problem is given by the research problem definition and the findings from the literature review

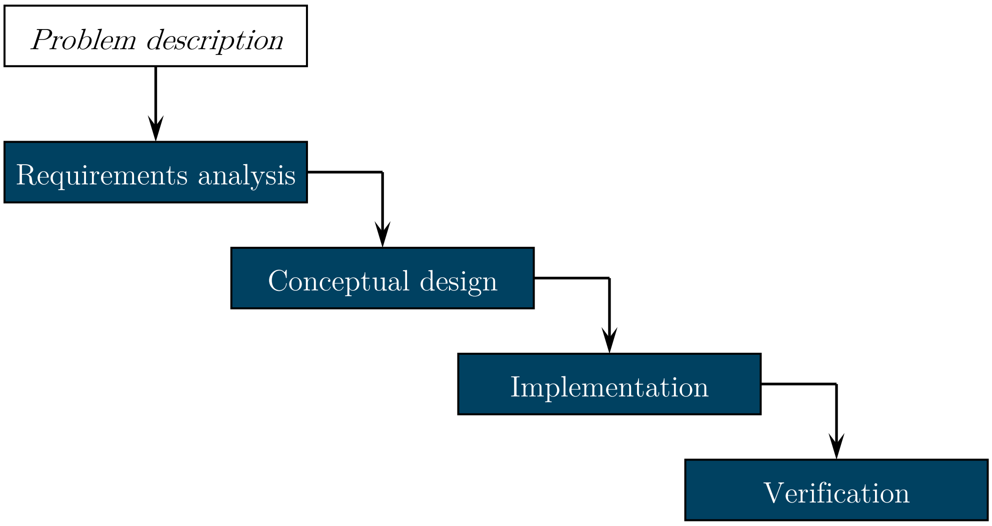


Figure 3.1: Waterfall model, according to (Royce 1970)

3.1 Requirements analysis

This section serves to specify the requirements the proposed automatic identification system must meet and at the same time is the first step of the waterfall model for development (see Figure 3.1). These properties should initially be kept relatively generic, since in the subsequent concept phase, solutions for the required properties should be developed. The definition of these requirements ensures the target-oriented course of the subsequent development process and at the same time determine the characteristics of the resulting artefact.

The literature review shows that various identification objects exist in the industrial material flow, which appear as piece goods, packaged goods or loading units (see Section 2.1.5). Since 3D-Object recognition methods from machine vision require sufficient geometric distinctiveness, which is not reliably the case for packaged goods and loading units, these are not considered. Another prerequisite for 3D-Object recognition from CAD-Data is that the identification objects must have a rigid structure, otherwise the models could deviate from the actual physical objects and matching would not be successful. The identification objects addressed by the identification system to be developed are therefore rigid piece goods. The natural features which make these objects distinguishable are based on the physical object features (see Section 2.1.4) as well as on the existing feature information from the CAD-Models (see Section 2.3.2). The most obvious natural distinguishing feature to be considered for the proposed automatic identification system is 3D-Geometry. Another possible natural feature is the mass characteristics of an identification object. The determination of the 6DoF-Pose of an object in space, which, when coupled to the 3D-Object recognition, makes it possible to call up the position of the centre of mass from the CAD-Model by drawing conclusions about the coordinate system of the object. The comparison of this centre of mass information with a centre of mass determined by means of a measuring plate on which the object rests provides valuable information for the purpose of identification. The use of this centre of mass information makes it possible, for example, to distinguish objects which are not distinguishable due to their external optical characteristics, as long as the centre of gravity position changes sufficiently due to their internal structure. The same applies to the actual weight information, which is obtained without further conclusions by weighing the physical object and comparing it with the metadata available in the CAD-Model. Material information can also be used as a distinguishing feature, as this information can be taken from the physical identification object and retrieved from the CAD-Model. Here, CAD-Models usually contain the description of the materials, which makes it necessary to translate them into sensorially measurable properties. It is important to note that identification objects can consist of several different materials and that only the material closest to the sensor is detected. For this reason, an exact differentiation of materials does not seem to make sense and the intended system should only differentiate between metallic and non-metallic objects. Lastly, the visual appearance of identification objects can also be used as a distinguishing feature. According to the literature review, this feature is not used for CAD-based 3D-Object recognition because colour information is not sufficiently defined by CAD formats and colour recognition is highly dependent on lighting conditions, reflectance of the objects and the quality of the sensor technology used. Nevertheless, the automatic identification system to be developed should be capable of distinguishing primary colours, since the use of this information will certainly improve distinguishability.

In summary, the automatic identification system to be developed in this thesis must enable the direct identification of rigid unpackaged piece goods and use the below-mentioned object characteristics as natural identification features:

- 3D-Geometry of the identification object (point cloud)
- Weight of the identification object
- Position of the centre of mass of the identification object
- Hull material of the identification object (metallic or non-metallic)
- Primary colour of the identification object

For this purpose, the proposed system should be able to store an object to be identified, record its identification features and compare these with a knowledge base built from CAD-Models. These CAD-Models should be used in the software manufacturer-neutral and well-defined STEP-Format. Once the identification object has been assigned to a corresponding CAD-Model, its identity information should be retrieved from the CAD-Model and passed on. Identification should only be partially unique at an article number level. The resulting system, consisting of hardware and software components, should be designed for indoor operation. Figure 3.2 shows the proposed automatic identification system in the form of a black box. Figure 3.3 and Figure 3.4 give a detailed view of the material flow and information flow. The illustration of the energy flow is neglected because the system does not perform any significant energy conversions. All the aforementioned figures follow the colour scheme explained in the labelling of Figure 3.2.

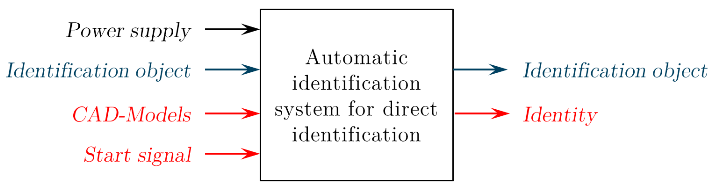


Figure 3.2: Black box of the proposed automatic identification system, with associated energy flow (black), material flow (blue), information flow (red)

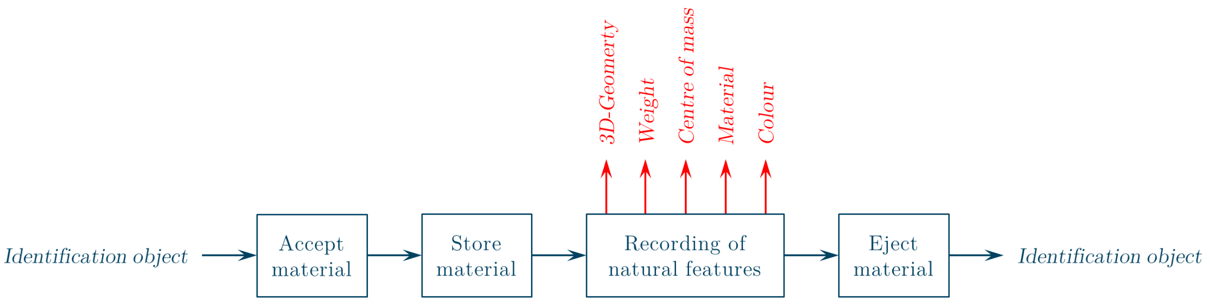


Figure 3.3: Material flow of the proposed automatic identification system

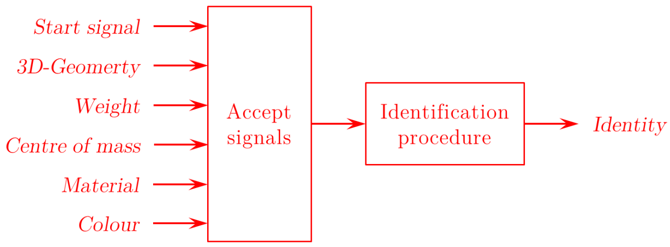


Figure 3.4: Information flow of the proposed automatic identification system

3.1.1 Hardware requirements

In this section, the requirements for the hardware of the mechatronic system to be developed are presented. The hardware consists of sensors and elements for handling the piece goods to be identified as well as of a 3D-Sensor for geometric acquisition. The hardware requirements (HR) to be met by the hardware of the Auto-ID system for multi-sensor identification are shown in Table 3.1.

Table 3.1: Summary of hardware requirements

Hardware requirement	Description
HR1	Hardware must allow the storage of an unpacked piece good for identification
HR2	Hardware must be able to capture the 3D geometry of a piece good as completely and accurately as possible in order to offer a good foundation for identification
HR3	Hardware must be able to measure weight
HR4	Hardware must be able to differentiate between metals and non-metals
HR5	Hardware must be able to recognise primary colours
HR6	Hardware must be able to be connected to and operated from a computer

3.1.2 Software requirements

In this section, the requirements for the software of the mechatronic system to be developed are illustrated. The software consists of different modules, which include the interfaces to the hardware and the necessary algorithm for identification. The software requirements (SR) presented in Table 3.2 refer to the software of the proposed Auto-ID system for multi-sensor identification.

Table 3.2: Summary of software requirements

Software requirement	Description
SR1	Software must offer interfaces for the communication with sensorial hardware
SR2	Software must accept CAD-Models for building identification knowledge base
SR3	Software must be able to handle and process multi-sensor data
SR4	Software must be able to execute 3D-Object recognition including 6-DoF pose estimation
SR5	Software must offer an algorithm for combining multi-sensor information for identification
SR6	Software must be able to communicate identity information to subsequent systems

3.2 Conceptual design

The second step of the waterfall model for development (see Figure 3.1) is the creation of a conceptual design. Based on the development task clarified in the previous section in the form of explicit requirements, principle solutions are now derived. Conceptual design describes the formulation of individual principal solutions, which are then merged into an overall concept (Pahl et al. 2007, p. 159). In order to proceed systematically, the steps shown in Figure 3.5 are followed. Based on the requirements of the proposed automatic identification system, individual subfunctions are first formulated in an abstraction step. These subfunctions together result in a function structure which aims to fulfil the requirements. After the formulation of the subfunctions, solution principles are searched for which fulfil them. From the solution principles found for each individual subfunction, a best suitable solution principle is now determined by selection using suitable criteria. The combination of these best suitable solution principles results in an overall functional structure, which is then regarded as a system variant. If multiple solution principles of equivalent quality are identified for individual subfunctions, several system variants may occur. In this case, a decision must be made on how to proceed, which may involve selecting one of the variants found. At the end of the conceptual design a tangible concept is available which can be realised in the following development step. In the following sections, the steps described are carried out separately for hardware and software, thus creating a hardware and software concept.

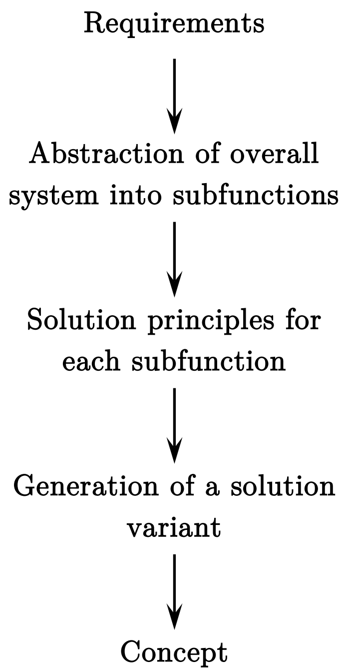


Figure 3.5: Steps of conceptual design, adapted from (Pahl et al. 2007, p. 160)

3.2.1 Hardware concept generation

As explained above, individual subfunctions are to be defined which reflect the functional structure of the overall automatic identification system. The hardware related subfunctions (HSF) are addressed in this section. The individual subfunctions are derived from the material flow (see Figure 3.3) and the requirements for the hardware (see Section 3.1.1). Table 3.3 shows the hardware subfunctions for which solution principles are elaborated below.

Table 3.3: Hardware-related subfunctions

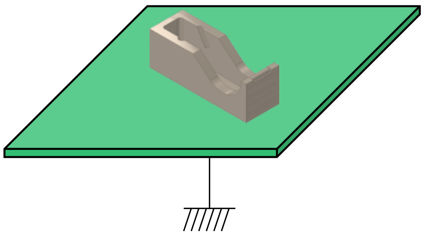
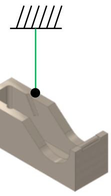
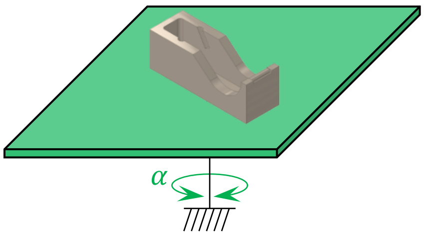
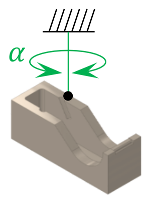
Subfunction	Description
HSF1	Store individual piece good
HSF2	Capture 3D-Geometry of stored piece good
HSF3	Measure weight of stored piece good
HSF4	Locate centre of mass of stored piece good
HSF5	Capture material properties of stored piece good
HSF6	Sense primary colour of stored piece good

3.2.1.1 Solution principles for hardware subfunctions

In this section hardware solution principles (HSP) are proposed, which aim to fulfil the required subfunctions. Finding such solution principles is a creative process, which is why all the solution approaches found will be presented below. The most suitable solution principles are then selected for the respective subfunction.

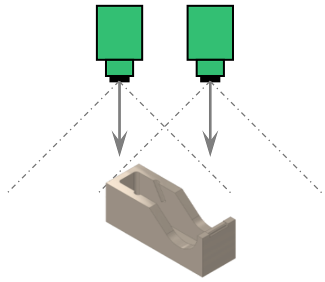
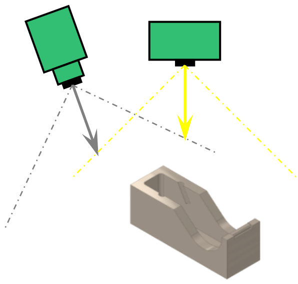
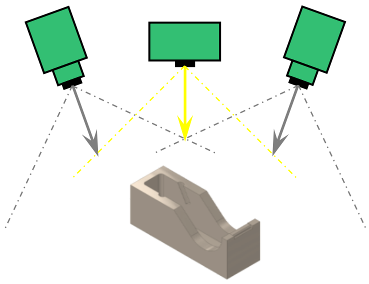
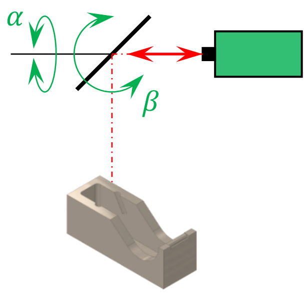
With regard to the first subfunction HSF1, there are essentially four possible solution variants, which are illustrated in Table 3.4. The storage of a piece of goods can be achieved either by placing it on a surface or by suspending it using a suitable fixture. These two variants can now form a fixed storage position (see HSP1-1) or a movable storage position (see HSP1-2), depending on the degrees of freedom allowed. Only rotational variants are considered for movable storage, since translational movement does not provide any advantage for optical sensing, while rotational movements are often used for complete scanning of objects.

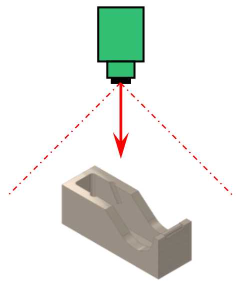
Table 3.4: Solution principles for HSF1

Label	Principle	Illustration
HSP1-1	**Fixed storage**	
HSP1-1-a	Placement on fixed surface	
HSP1-1-b	Fixed suspension	
HSP1-2	**Movable storage**	
HSP1-2-a	Placement on rotatable surface	
HSP1-2-b	Rotatable suspension	

Subfunction HSF2 can be solved using the 3D-Sensing technologies for vision systems explained in the literature review (see Section 2.2.5). Table 3.5 shows all variants of principles found for the solution of HSF2 in the literature. There are two broad categories of solution principles based either on triangulation (see HSP2-1) or on ToF (see HSP2-2), containing several explicit variants.

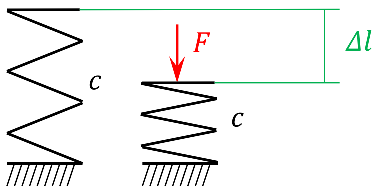
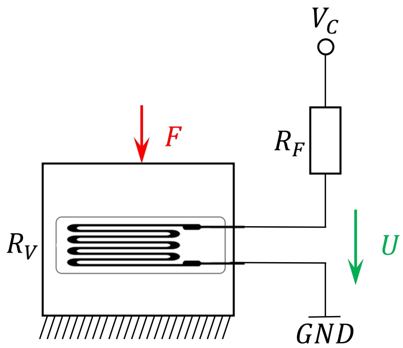
Table 3.5: Solution principles for HSF2

Label	Principle	Illustration
HSP2-1	**Triangulating sensor technology**	
HSP2-1-a	Passive stereo-vision sensing	
HSP2-1-b	Structured-light sensing	
HSP2-1-c	Active stereo-vision sensing	
HSP2-2	**ToF-Sensor technology**	
HSP2-2-a	Scanning ToF-Sensing	

HSP2-2-b	Non-scanning ToF-Sensing	

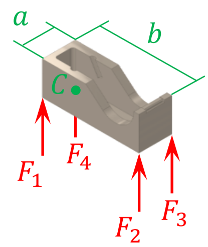
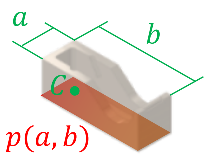
For the solution of the third subfunction (HSF3), there are two possible options, which are shown in Table 3.6. All methods for determining weights are based on measuring the deformation of elastic bodies caused by externally applied forces. There are two common approaches with regard to the measuring principle and the measured variables recorded with it. On the one hand, the deformation Δl of an elastic body of known stiffness c can be measured (HSP3-1-a) and thus the applied force or weight can be deduced applying Hooke's law. On the other hand, by attaching so-called strain gauges (HSP3-1-b) to an elastic body, its deformation can be translated into a voltage change ΔU. This principle is based on the change in resistance of metallic conductors by compressing or stretching them, giving them a variable resistance R_V. Other solution principles based on comparison with reference weights are not considered here as they are difficult to apply to automated systems.

Table 3.6: Solution principles for HSF3

Label	Principle	Illustration
HSP3-1	**Elastic deformation measurement**	
HSP3-1-a	Travel measurement	
HSP3-1-b	Strain gauge measurement	

The determination of the centre of mass for the fulfilment of subfunction HSF4 can also be done by several solution variants. Table 3.7 shows the solution approaches found, which are based on weighing or calculation operations using the point cloud. According to the principle of weighing, by measuring several bearing forces F_i holding a steady object, the centre of gravity can be determined by applying the laws of statics (HSP4-1-a). Instead of forces, the surface load p with which a body rests on a surface can also be used as a basis (HSP4-1-b). Both described solution variants are suitable for the exact determination of the centre of mass position for arbitrary bodies. For bodies with uniform material distribution, the centroid of an associated surface describing point cloud corresponds to a body's centre of mass (HSP4-2-a).

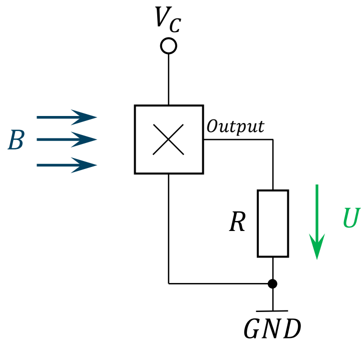
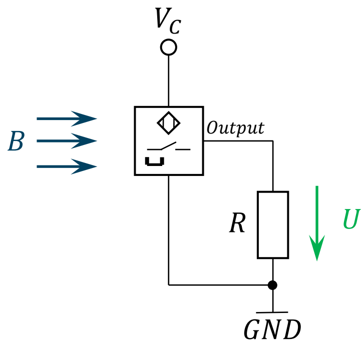
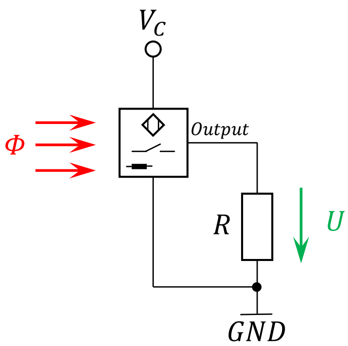
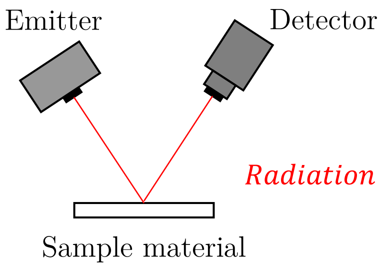
Table 3.7: Solution principles for HSF4

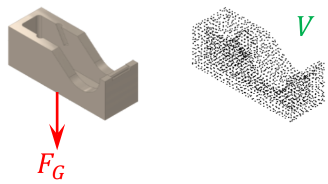
Label	Principle	Illustration
HSP4-1	**Localisation via weighing**	
HSP4-1-a	Multi bearing force localisation	
HSP4-1-b	Contact pressure localisation	
HSP4-2	**Localisation via point cloud**	
HSP4-2-a	Centroid calculation localisation	

Subfunction HSF5 requires the acquisition of material properties. Due to the scope of this work, only methods which can capture material properties in a non-destructive way are considered. The solution principles found which meet this precondition are shown in Table 3.8. Magnetic material properties can be detected by sensors without destruction (HSP5-1). For this purpose, sensors are available which make use of the Hall effect, which acts on conductors through which current flows in a magnetic field (HSP5-1-a). Alternatively, so-called reed switches can be used as magnetic field detectors, in which miniature contacts are switched under the influence of a magnetic field (HSP-5-1-b). A further magnetic property of materials is their permeability, which can be monitored using inductive switches (HSP5-1-c). The precise determination of the magnetic permeability of a material would be very helpful for use as an identifier, but this must be carried out under laboratory conditions and is therefore not considered here. The same applies to the dielectric constant of a material, which must also be determined under laboratory conditions.

Another possibility is to determine the exact spectroscopic composition of a material, which can be done in an industrial context with spectroscopes (HSP5-2-a). Lastly, it would be possible to include density as a material property by determining the weight using a scale and the volume of an object from the recorded point cloud (HSP5-3-a).

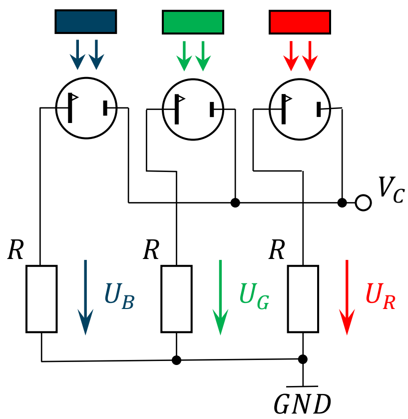
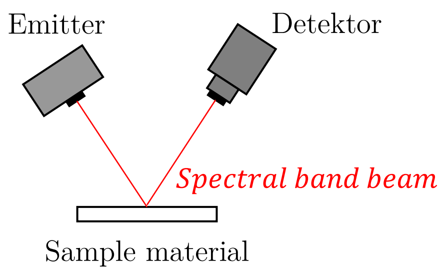
Table 3.8: Solution principles for HSF5

Label	Principle	Illustration
HSP5-1	**Sensing magnetic properties**	
HSP5-1-a	Hall effect sensor	
HSP5-1-b	Reed switch	
HSP5-1-c	Inductive switch	
HSP5-2	**Sensing spectrometric properties**	
HSP5-2-a	Spectroscope	

HSP5-3	**Sensing density properties**	
HSP5-3-a	Weight and point cloud volume	

Two methods can be used to determine the colour of an object, as required to fulfil subfunction HSF6. Tristimulus colourimetry (HSP6-1) describes the decomposition of a light spectrum into individual primary colour channels (red, green and blue) and the subsequent measurement of the individual intensities. So-called RGB sensors are used for this (HSP6-1-a). A more precise method is spectral colourimetry, in which a known spectral band is sent to a sample object. The reflected band is then detected by a detector and the wavelengths absorbed by the object are analysed (HSP6-2-a). This is shown below in Table 3.9.

Table 3.9: Solution principles for HSF6

Label	Principle	Illustration
HSP6-1	**Tristimulus colourimetry**	
HSP6-1-a	Red green blue (RGB) Sensor	
HSP6-2	**Spectral colourimetry**	
HSP6-2-a	Spectrophotometric colour sensing	

3.2.1.2 Selection of solution principles for hardware concept generation

From the solution principles presented for the individual hardware subfunctions, one or more suitable combination(s) must be selected in order to generate hardware solution variants (HSVA). To guarantee a comprehensible selection process, this is done by means of utility analysis. Therefore, criteria are first defined for each subfunction, which serve to evaluate the solution principles found. Using the method of pairwise comparison, weighting factors are determined which reflect the significance of a single criterion in the assessment process. Subsequently, each HSP is evaluated for every HSF according to the criteria by awarding points based on the five-level scale proposed by (VDI 2225-3:1998-11, p. 4). The detailed benefit analyses for each hardware subfunction can be found in Appendix A. In the following section the results of the utility analysis are presented and briefly discussed.

With regard to HSF1, two selection criteria were applied:

– The effort required for the storage of piece goods for the identification process should be minimal and universal for a variety of different objects
– The compatibility of the solution principle with the other HSFs, as one HSP should not have a negative impact on the performance of other HSFs

Both criteria are considered equally important and are therefore weighted equally. HSP1-1a can be seen as an ideal solution for the storage of piece goods, as they only need to be placed on the fixed surface. However, there is a disadvantage in the compatibility with HSF2, as the contact surface between the surface and the piece goods does not allow a complete recording of the 3D-Geometry. HSP1-1-b requires a great deal of effort in terms of suspension possibilities, which is likely to be difficult with a variety of piece goods. In return, a suspension would allow for a much lower level of coverage and the 3D-Geometry of a suspended object could be recorded to a greater extent compared with HSP1-1-a. Both solution principles HSP1-1-a and HSP1-1-b have the disadvantage that a complete recording of the 3D-Geometry can only be achieved by different perspectives of a 3D-Sensor. These perspectives can be obtained either by moving one sensor around the object to be scanned or by arranging several sensors around it. Moving a sensor around the object would require an exact determination of the 3D-Sensor's position and orientation in space to enable precise scanning. The use of several 3D-Sensors generates additional costs, which increase with the number of perspectives and the resulting completeness of the scan. HSP1-2-a and HSP1-2-b therefore suggest a rotatable storage as an extension of HSP1-1-a and HSP1-1-b. The rotation allows for a very complete scan using a single 3D-Sensor. HSP1-2-a is chosen as the solution principle for HSF1 because of the significantly lower effort required for the storage and the good compatibility with the other HSFs.

Three criteria were considered to select a solution principle for HSF2:

– The absolute accuracy with regard to the dimensions which a solution principle such as 3D-Sensing technology offers when recording a 3D-Geometry.
– The robustness of the method in terms of illumination and surface structure of scan objects.
– The area of perception of a technology, which also affects the time needed for a scan.

The comparison carried out in pairs shows that the first two criteria are equally important and together account for about 83 % of the total weight of the evaluation. The latter criterion plays a relatively minor role. An evaluation of the costs of the respective solution principle was deliberately omitted, since literature often mentions the increasing affordability of 3D-Sensing technologies. Similarly, due to the complexity and application dependence of such technologies, no reliable general cost projections can be made. Solution principle HSP2-1-a as a triangulating technology can provide comparatively limited accuracy and robustness through the use of passive stereo vision (see Section 2.2.5.1). The most serious disadvantage, which often occurs in industrial environments in terms of robustness, is that texture-free surfaces cannot be captured. The field of perception is also relatively small with such passive 3D-Sensors. HSP2-1-b, as an active triangulating technology, can ensure ideal accuracy and robustness by using a projector. Using only one camera, occlusions can occur and the perception area is smaller than with an active stereo vision approach as suggested by HSP2-1-c. Solution principle HSP2-1-c is therefore assessed as an ideal solution principle in terms of accuracy, robustness and perception area. HSP2-2-a, which is based on the ToF-Principle and scans the 3D-Geometry of an object point by point, is also ideal in terms of accuracy and robustness. A major drawback, however, is the perception area, which in this context is understood as a single scan point. Compared to the other HSPs, the acquisition of an entire 3D-Geometry takes considerably longer and is therefore just barely acceptable. This point-by-point geometry acquisition does not apply to HSP2-2-b, which is also based on the ToF principle, but has a perceptual area comparable with that of the triangulating technologies. Unfortunately, these ToF-Cameras only have limited accuracy and robustness according to the current literature (see Section 2.2.5.3). Regarding the selection of an HSP for HSF2, HSP2-1-b and HSP2-1-c are particularly suitable for use within the proposed automatic identification system. Although HSP2-1-c has a slightly better field of view, HSP2-1-b is an equivalent solution, especially if the scan object is rotated for a full scan.

The following criteria have been used to assess the HSPs to solve HSF3:

- <u>The absolute measuring accuracy</u> that a solution principle offers.
- <u>The effort of measurement</u> required to obtain a machine-readable signal using the principle.
- <u>The compatibility</u> of the solution principle with the other HSFs

Solution principle HSP3-1-a offers good accuracy and the measurement effort to determine the travel is moderate. In fact, the accuracy of this method depends on the magnitude of the travel. Preferably a small change in force such as weight should result in a large, easily measurable displacement. This fact makes the use of HSP3-1-a in combination with HSF4 difficult. The displacement of a weighing plate would mean that the calibration of a 3D-Sensor would no longer be accurate with regard to its position in relation to a coordinate system situated on the plate. HSP3-1-b does not have this disadvantage, because minimal deformations can already be measured with a strain gauge. Both the accuracy and the measuring effort are superior with this principle, as the hysteresis of a mechanical spring does not occur and the deformation results directly in an easy-to-measure resistance change. This leads to the selection of HSP3-1-b as the solution principle for HSF3.

To solve the subfunction HSF4, the following criteria for the selection of an HSP were applied:

– **The achievable localisation accuracy** of the centre of mass using the suggested principle.
– **The technical feasibility** of a solution principle with regard to the current state of the art.

Both criteria receive the same weighting in the assessment, as they are basic prerequisites. The solution principles HSP4-1-a and HSP4-1-b, which are based on the principle of weighing, theoretically offer the possibility of an exact localisation of the centre of mass of an object. HSP4-1a in particular is very accurate, as the bearing forces and distances between bearing points can be measured very accurately. No difficulties are expected in terms of technical feasibility. The accuracy of HSP4-1-b is more limited compared with the HSP4-1-a, as the determination of the surface pressure is much more demanding and less accurate. Common sensor mats for solving such a measuring task offer a limited size and resolution, which is why the localisation of the centre of mass of an object is also only of limited accuracy. This also leads to problems of technical feasibility, which is why this solution is rated worse than HSP4-1-a. Solution principle HSP4-2-a follows a different approach and determines the centre of mass as the centre of the recorded point cloud. Solution principle HSP4-2-a follows a different approach and equates the centroid of the recorded point cloud of an object with its centre of gravity. For objects with uniform mass distribution, this approach can be used without any problems, since in this case the centroid and centre of gravity coincide. However, this condition is by no means always met, so the accuracy of this method is only barely acceptable. In return, the technical implementation of this solution principle would be very simple and no further sensors would be required apart from the already existing optical sensor technology. As a result of the utility value analysis, solution principle HSP4-1-a is considered ideal for the solution of HSF4 and is therefore selected.

The selection criteria for selecting a solution principle for HSF 5 are:

– **The distinctiveness of different materials** that a solution principle provides
– **The technical feasibility** of a respective solution principle with regard to the state of the art

In fact, only the spectroscopy proposed by HSP5-2-a allows a detailed differentiation of materials according to their chemical composition. Depending on the material to be examined, however, different spectroscopy methods have to be applied, which are very complex and can only be used to a limited extent in an industrial environment. The use of passive magnetic field detecting sensors, as is the case with HSP5-1a- and HSP5-1-b, would only make it possible to distinguish between permanent ferromagnetic materials and non-magnetic materials. This distinguishing feature for an automatic identification system is unlikely to be significant in an industrial environment. HSP5-1-c allows the differentiation of metals and non-metals, which means limited distinctiveness analogous to HSP5-1-a and HSP5-1-b. However, the distinction between non-metals and metals as an identification feature seems to be of much greater interest than the detection of magnetism. With regard to technical feasibility, none of the solution principles mentioned under HSP5-1 are in great demand and there are a large number of existing sensors which are used for industrial applications. HSP5-3-a proposes the determination of density as a material property of an object, which could be used for identification purposes. These density properties can be determined without additional technical effort using the 3D-Sensor technology for volume measurement and the weighing system for weight determination. However, the use of density for material identification is only accurate if the measured object consists entirely of one material. Since 3D-Geometry and weight are used for identification anyway, an average density is implicitly considered. No solution principle emerges from the utility analysis as being most suitable. For this reason, HSP5-1-c is chosen as the solution principle for HSF5, as the distinction between metals and non-metals is considered the most advantageous for identification purposes.

Three evaluation criteria were applied to find a solution principle for the final subfunction HSF6:

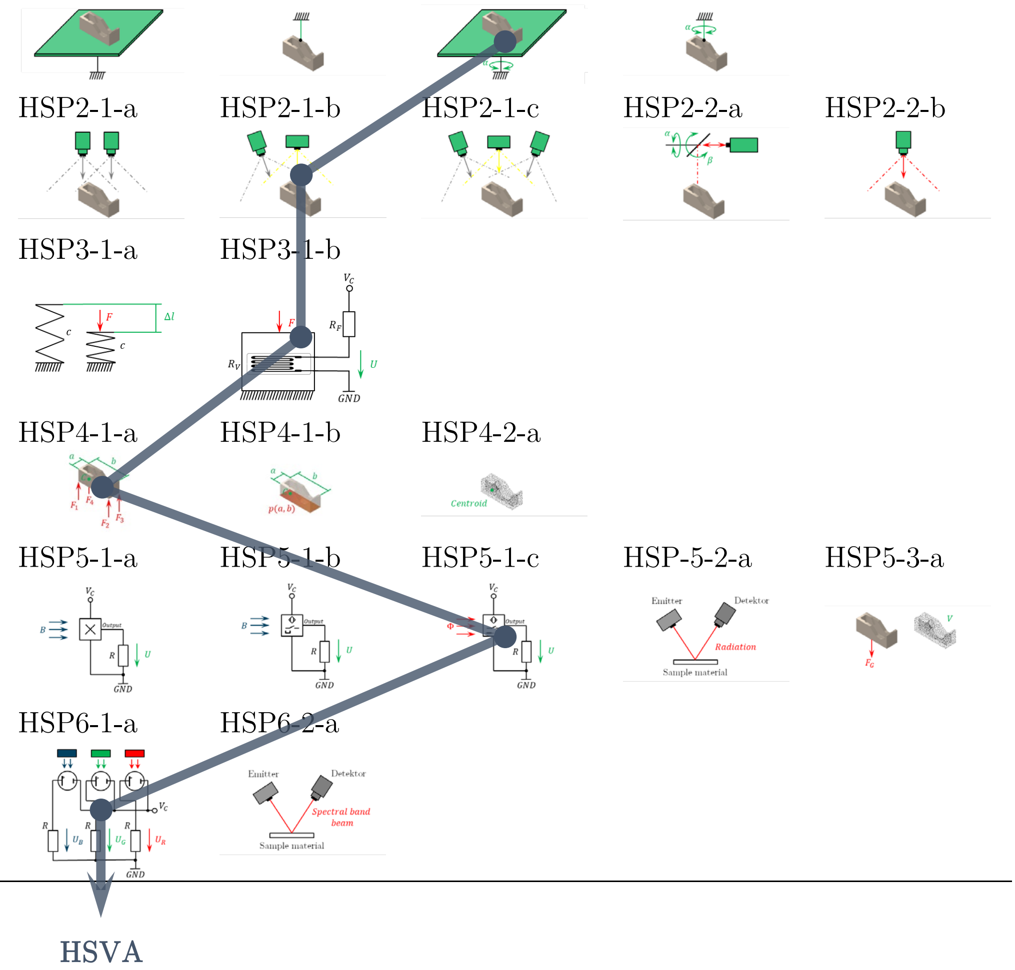
- **The accuracy** the solution principle offers in terms of colour determination
- **The effort** for the colour determination using the respective principle
- **The compatibility** of the solution principle with the other HSFs

The weighting factors determined by pairwise comparison show that the highest weighting is given to compatibility followed by the effort for colour determination. Less attention is paid to the accuracy of colour determination, mainly because it is not possible to retrieve exact colour information from CAD-Models, which mainly describe geometry (see Section 2.3.2). HSP6-1-a provides reasonable accuracy, but has drawbacks under inconsistent illumination conditions. With regard to the effort required to determine the colour, HSP6-1-a can be regarded as an ideal solution, as all colour cameras use this principle in the form of Beyer filter photosensors. The existing camera used for the 3D-Sensing technology can therefore be used for colour perception without additional effort. This and the fact that the camera takes a global view of the scene and thus globally predominant colours can be determined instead of local colours also makes HSF6-1-a the ideal solution in terms of compatibility with other HSFs. The solution principle HSP6-2-a offers an ideal accuracy for colour determination, but it is, however, limited to a small area and single colour. The measurement effort is quite large and time-consuming, since a large number of spectral bands are emitted and their reflection from the target is measured. The installation of a colorimeter is also necessary, which means additional hardware effort. As far as compatibility to other HSFs is concerned, HSP6-2-a is only a barely acceptable solution because colour perception is limited to a small area and the accuracy of colour determination provided by the solution principle cannot be used because this information is not provided by the CAD-Models. The choice of the solution principle for HSF6 thus falls on HSP6-1-a, since no additional hardware is required and the accuracy of the colour determination is considered sufficient.

Table 3.10 shows the selected hardware solution principles resulting in the hardware solution variant as a morphological box.

Table 3.10: Morphological box for hardware concept

Subfunction	Hardware solution principles				
HSF1	HSP1-1-a	HSP1-1-b	HSP1-2-a	HSP1-2-b	
HSF2	HSP2-1-a	HSP2-1-b	HSP2-1-c	HSP2-2-a	HSP2-2-b
HSF3	HSP3-1-a	HSP3-1-b			
HSF4	HSP4-1-a	HSP4-1-b	HSP4-2-a		
HSF5	HSP5-1-a	HSP5-1-b	HSP5-1-c	HSP-5-2-a	HSP5-3-a
HSF6	HSP6-1-a	HSP6-2-a			



3.2.1.3 Centre of mass localisation principle for weighing plate

Figure 3.6 shows the free body diagram of a weighing plate with four bearing points, based on HSP4-1-a. The bearing forces F_1, F_2, F_3 and F_4 support an idealised massless plate in positive z-direction, on which a weight force F_m acts in negative z-direction. The weight force F_m represents the force which a spatially extended body induces when placed on the plate according to the centre of mass principle. When the system is at rest, all forces cancel each other out and Equation (3.1) applies. This fact makes it possible to determine the mass of the body on the plate by adding up the bearing forces F_1, F_2, F_3 and F_4 and dividing their sum by the gravitational acceleration g as formulated in (3.4). Since the point of attack of F_m on the plate is shifted by the amounts x_m in x-direction and y_m in y-direction from the bearing point indexed with 1, moments arise around the x-axis and y-axis respectively. Similarly, bearing forces F_2, F_3 and F_4 generate moments due to their distance l_x and l_y from bearing point 1, which counteract those generated by the weight force F_m. Applying static laws leads to Equations (3.2) and (3.3). If the distance between the bearing points l_x and l_y is known and if the magnitude of the bearing forces F_1 to F_4 acting there is known, the displacement x_m and y_m can be deduced. Equations

(3.5) and (3.6) thus describe the location of the point of attack of the weight force of an object placed on the surface. The force's point of attack can be interpreted as a projection of the object's centre of mass on the plate.

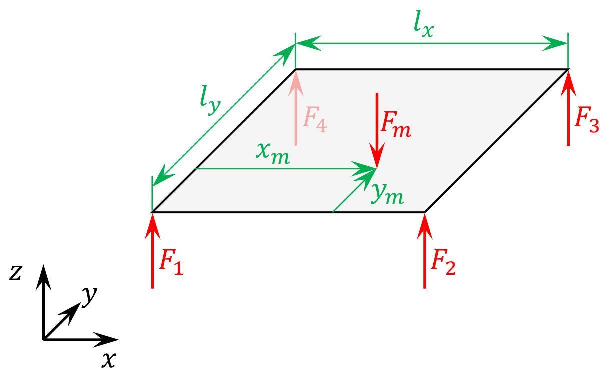


Figure 3.6: Static free body diagram of weighing plate

$$\sum F_z = F_1 + F_2 + F_3 + F_4 - F_m = 0 \tag{3.1}$$

$$\sum M_{1,x} = F_3 \times l_y + F_4 \times l_y - F_m \times y_m = 0 \tag{3.2}$$

$$\sum M_{1,y} = F_m \times x_m - F_2 \times l_x - F_3 \times l_x = 0 \tag{3.3}$$

$$m = \frac{(F_1 + F_2 + F_3 + F_4)}{g} \tag{3.4}$$

$$x_m = \frac{(F_2 + F_3)}{(F_1 + F_2 + F_3 + F_4)} \times l_x \tag{3.5}$$

$$y_m = \frac{(F_3 + F_4)}{(F_1 + F_2 + F_3 + F_4)} \times l_y \tag{3.6}$$

3.2.2 Software concept generation

In order to create a software concept for the Auto-ID system to be developed, the generic identification process to be followed by the software is first defined. Then an overall software structure is developed, which serves to fulfil the software subfunctions. The software subfunctions (SSF) result from the intended information flow (see Figure 3.4), the software requirements (see Section 3.1.2) as well as the closely related hardware concept developed above (see Section 3.2.1.2). Table 3.11 shows the SSF to be covered by the software.

Table 3.11: Software subfunctions

Subfunction	Description
SSF1	Build identification knowledge base from CAD-Models
SSF2	Acquire sensor data from sensorial hardware and transform to multi-sensor information
SSF3	Perform identification with multi-sensor information
SSF4	Communicate identity information

3.2.2.1 Generic process of identification

The three-stage identification process found in the literature (see Section 2.1.4) is rather practice-oriented and therefore offers little overview of the actual operations involved in finding an identity. Furthermore, current Auto-ID systems usually work with only one identifier that is evaluated for identification purposes and there is no generic approach to identification based on multiple identifiers. For this reason, a generic identification process is formulated below, which includes the recording of several features for identification purposes and describes the precise process of determining an identity. The formulated identification process is used for the further development of the software concept within this thesis.

The fundamental prerequisite for identification is always an identification knowledge base, which links the identity information of objects with their identification features. Generically formulated, this knowledge base K contains j ordered pairs P_j each consisting of a set S_{F_j} containing k identification features F_k and a set of identity information S_{I_j} of size l, describing known objects:

$$K = \{P_1, P_2, \dots, P_j\} \tag{3.7}$$

$$P_j = \left(S_{F_j}, S_{I_j} \right) \tag{3.8}$$

$$S_{F_j} = \{F_1, F_2, \dots, F_k\}, \tag{3.9}$$

$$S_{I_j} = \{I_1, I_2, \dots, I_l\}, \tag{3.10}$$

This formulation is generic and applies both to direct identification based on natural object features and to indirect identification based on artificial identification features. This is due to the fact that the identification features F_k contained in set S_{F_j} can include natural features, artificial features or a combination of both. An example of a natural identification feature F_k would be the weight of an object, whereas a barcode would be an example of an artificial identification feature F_k. The identity information I_l, which is contained in S_{I_j}, defines the actual identity of an object within the framework of the identification knowledge base. Two examples of identity information that may appear as I_l are the article number or the name of an object. It is self-explanatory that an identification can only take place if an identification knowledge base with the declared structure is available and an object to be identified is defined within this framework. The creation of such an identification knowledge base therefore takes place in an offline process before an online identification system can use it to identify explicit objects. This online process follows the formulation found in the literature (see Section 2.1.4) and begins with data collection. More precisely, the data is collected from a physical scene in a real-world environment containing an object to be identified. By means of sensor systems and appropriate processing methods, it is therefore possible to acquire a set of identification features describing the object contained therein. The generic formulation of such a set S_f consisting of n identification features f_n describing an identification object is as follows:

$$S_f = \{f_1, f_2, \dots, f_n\} \tag{3.11}$$

Analogous to the counterpart from the knowledge base, natural and/or artificial identification features can be acquired from the scene and therefore be contained as an f_n within S_f. After the collection of data from the scene and the availability of an S_f, the next step is the extraction of the identity information from the identification knowledge base. This is done by comparing the identification feature set S_f collected from the scene with the identification feature set S_{F_j} of each P_j describing known objects contained in the identification knowledge base K. The aim of this comparison is to find the explicit P_j whose S_{F_j} corresponds exactly to S_f and therefore ideally $S_f = P_j(S_{F_j})$ applies. If this P_j is found within K, the set of identity information S_{I_j} can be easily accessed and identification is complete. Since data collection and formulation is in reality burdened with imperfections, it makes sense to allow a certain degree of imperfection. Accordingly, the search explained above is for the pair P_j of the knowledge base K where S_f is approximately S_{F_j} and thus $S_f \approx P_j(S_{F_j})$ applies. In generic terms, the extraction of the set of identity information S_{I_j} from the identification knowledge base K can be formulated as follows:

$$S_f \approx P_j(S_{F_j}) \xrightarrow{yields} S_{I_j} \tag{3.12}$$

The desired approximate equality of S_{F_j} and S_f must be established in advance with respect to the capabilities of the system that acquires the identification features from the scene. For example, the sensor-based acquisition of an object's geometry for 3D-Object recognition can result in larger deviations, as would be the case when reading a barcode. Hence, the required approximate equality should be formulated by specifically defining suitable tolerances for determining when a single identification feature $f_n \approx F_k$.

3.2.2.2 Software structure

The overall structure of the software is conceptualised using unified modelling language (UML). UML is a modelling language that supports software development through graphical illustration and is standardised according to (ISO/IEC 19505-2:2012). Within UML there are different types of diagrams, which are selected according to the desired purpose. For the development of the software concept within this thesis, component diagrams are used as representatives of the implementation structure diagrams of UML. Component diagrams are used to visualise the relationship between different components in a software system, where components represent class modules from an informational point of view. The subdivision of software systems into individual components originates from the paradigm of component-based software engineering (CBSE), which emphasises the separation of the overall functionality of a software into individual functional modules. The essential advantage of such a functional structuring during the planning of the software is the fact that individual components can be replaced or reused modularly.

Figure 3.7 shows the UML component diagram that has been developed for the overall software structure. As shown in the diagram, the software concept consists of three core components. These three components represent all functionalities of the automatic identification system for direct identification to be developed. The CAD-Models identity information and the sensor data supplied by the hardware of the Auto-ID system serve respectively as input variables. In the UML component diagram this is illustrated by the two non-connected required interfaces. The output parameters of the software are the identity information of an identified object, which can be passed on to downstream information systems via the provided interface depicted in the UML component diagram. Due to the interfaces described, the whole structure diagram in Figure 3.7 can also be formulated as a component, which, however, is not included here for reasons of simplicity.

The component "Multi-sensor identification" is the core component of the software concept. Within this component, identification is carried out in the actual sense, which is why the knowledge base, which is indispensable for identification, is also managed by this component. The identification knowledge base is created from the CAD-Models and the associated identity information received via the associated interface. For this purpose, the CAD-Models are internally converted into the form of a point cloud and transferred to the component "3D-Object recognition" via the "Point cloud" interface. The latter component returns individual descriptors for 3D-Object recognition for each of the transmitted point clouds to the component "Multi-sensor identification", which are received via the interface "Descriptor". Together with the identity information these descriptors form an individual data set for each known object within the identification knowledge base. The described procedure for the generation of the identification knowledge base is an offline process which has to be carried out before an actual online identification process can take place. When carrying out an online identification process, the multi-sensor information from the “Multi-sensor data processing” component is first retrieved via the “Other information” interface. Using this information on all identifiers acquired by the hardware from the physical object, except the acquired point cloud, matching records can be extracted from the identification knowledge base. On the one hand, this pre-filtering of candidates has the advantage that the number of descriptors to be compared during 3D-Object recognition is reduced and thus less computing time is required. On the other hand, false recognitions are reduced, which could result from objects that are difficult or impossible to distinguish visually. The 3D-Object recognition descriptors of the filtered data sets generated during the offline process are then made available to the component “3D-Object recognition” via the interface “Descriptors”. As return of the 3D-Object recognition the component “Multi-sensor identification” receives the result of the recognition process via the interface “Recognition result”. In the case of successful recognition, the descriptor describing the point cloud from the sensor information matches one of the candidate descriptors within the filtered set from the identification knowledge base. The identity information can therefore be retrieved from the identification knowledge base via the record with the matching descriptor. Finally, the “Multi-sensor identification” component can supply this identity information to downstream information systems via the corresponding interface illustrated in the UML component diagram.

The component “Multi-sensor data processing” forms the interface to the sensor hardware of the Auto-ID system to be developed. This component transforms the raw sensor data acquired from the scene into multi-sensor information, which provides the basis for the further identification process. Depending on the nature of the sensor data, appropriate methods have to be used within this component to obtain high quality information. The “Sensor information” interface provided by this component is used by both the “Multi-sensor identification” component and the “3D-Object recognition” component. Here the geometric information from the scene in the form of the scene point cloud is provided directly to the component “3D-Object recognition”, while the remaining sensor information is being passed to the component “Multi-sensor identification”.

The component “3D-Object recognition” executes the essential operations which are necessary for optical recognition (see Section 2.2.1). Consequently, the component takes over both the extraction of point cloud features and their description by descriptors as well as the recognition itself by matching these descriptors. In cooperation with the component “Multi-sensor identification”, descriptors are generated for provided point clouds of CAD-Models, which are subsequently used to generate the identification knowledge base. Through the interface “point cloud”, however, the point clouds of the identification object from the scene provided by the component “Multi-sensor data processing” can also be received. After describing this identification object related point cloud, its descriptor is available for matching within the component. For the matching itself, candidate descriptors are provided by the component “Multi-sensor identification”, as previously described. By matching this set with the descriptor of the scene point cloud a recognition result is obtained. As described in Section 2.2.1, this recognition result includes both the conformity of geometric features and information regarding the 6-DoF pose in relation to the optical sensor. This 6-DoF information is the basis for the comparison of the centre of mass information between the CAD-Model and the identification object within the scene, which is carried out in the component “Multi-sensor identification” by applying complex geometric transformations.

The superordinate concept shown in Figure 3.7 forms the framework for further developments, whereby each of the components illustrated encapsulates subcomponents, so-called ‘classes’. These classes actually handle the specific tasks of processing and transforming data and information respectively and are therefore referred to as ‘realisations’ in UML. The detailed definition of each component with its interfaces and implementations using UML are given in Figure 3.8. At this point, no further explanation of the implementations is given, as their functions are described above. The generic formulation of the overall concept and its individual components was deliberately chosen to obtain universality.

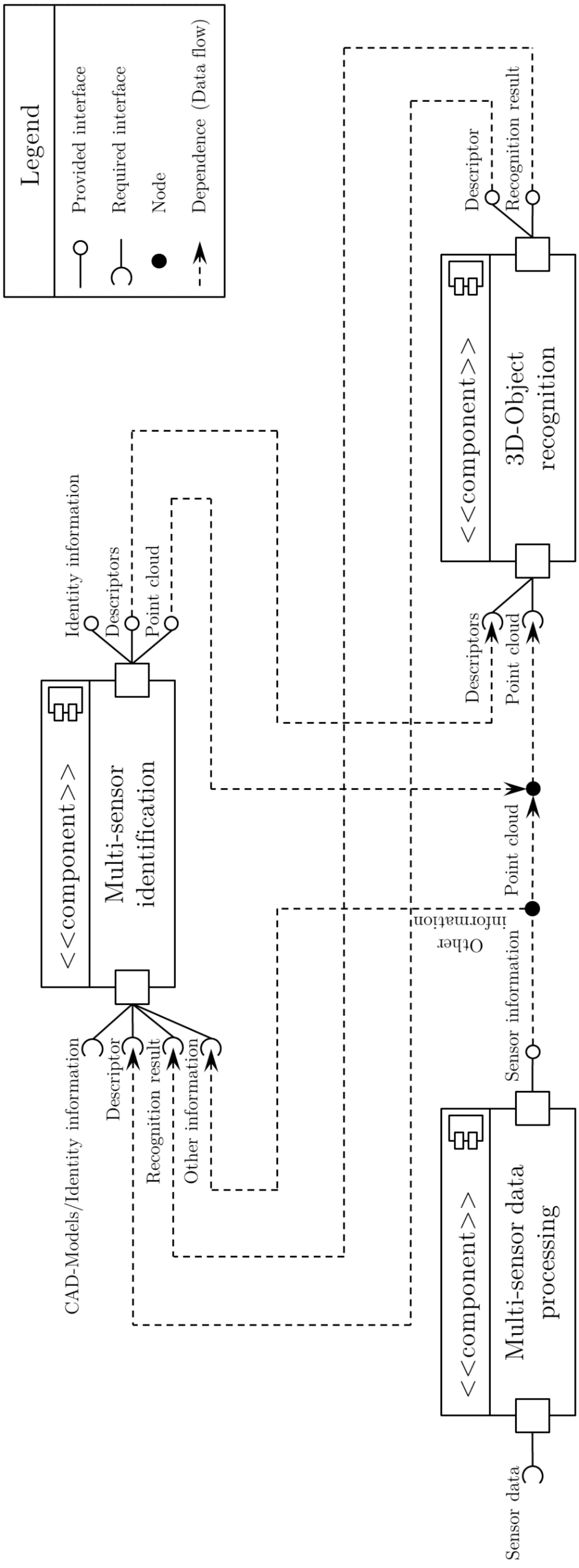


Figure 3.7: UML component diagram for overall software concept

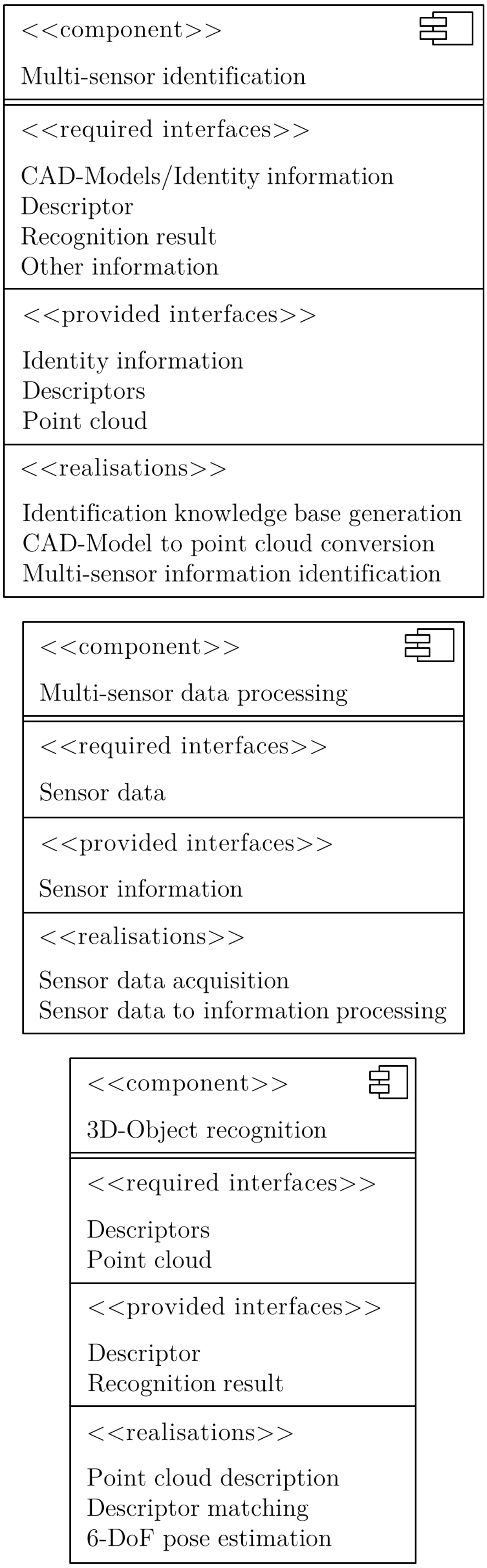


Figure 3.8: UML component definition for overall software concept

3.3 Proposed concept for multi-sensor AIS

For the solution of HSF1, HSF2 and HSF6 with HSP1-2-a, HSP2-1-b and HSP6-1-a a scan system is proposed consisting of a structured-light sensor and a rotary device. Since the rotary movement is indispensable for the complete recording of a scan, it makes sense to assign this rotary device to the scan system. In fact, some commercially available scanning systems include a turntable for this reason. Structured-light sensors basically consist of a projector and a camera (see Section 2.2.5.2), the image information of the camera can therefore also be used to extract the colour of an identification object. The image data of the camera's colour sensor are analysed by computer vision and machine vision methods respectively, in order to differentiate primary colours.

In order to jointly solve HSF3, HSF4 and HSF5 by the selected solution principles HSP3-1-b, HSP4-1-a and HSP5-1-c, a sensor platform is proposed. This sensor platform carries the rotating device on which the object to be identified is placed and it also enables the acquisition of multi-sensor information.

Figure 3.9 shows a schematic view of the developed concept. It shows the overall arrangement of the structured-light sensor as well as the sensor platform consisting of inductive material sensor, rotating device and force sensors. An important aspect that emerges from the figure as well are the different coordinate systems that occur. The coordinate system indexed with "S" belongs to the structured light sensor, a point cloud recorded by the sensor thus refers to this coordinate system. The coordinate system of the sensor platform, indexed "SP", serves to determine the location of the projected centre of mass of an identification object placed on the rotary device. The coordinate system of the identification object, indexed with "O", can have any orientation.

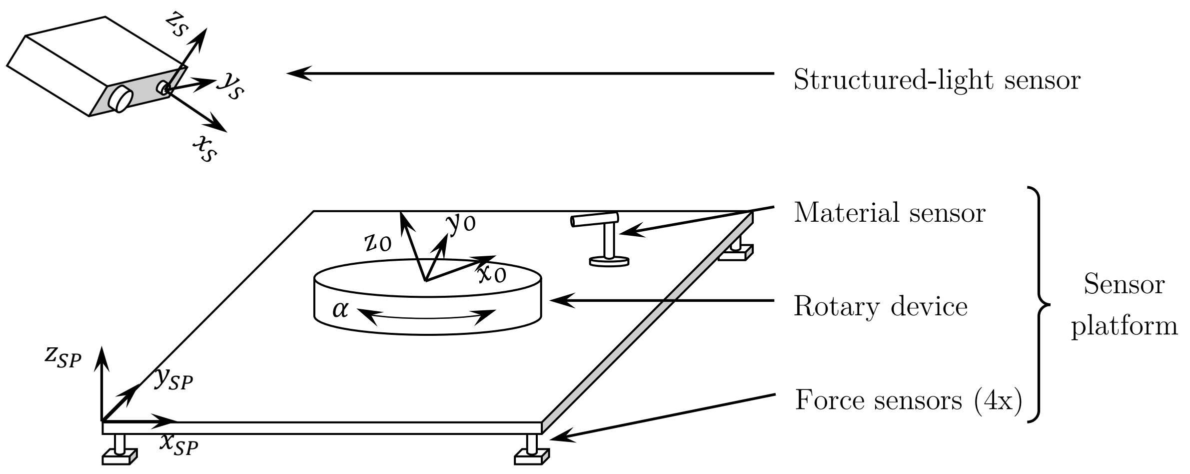


Figure 3.9: Proposed concept for multi-sensor AIS

4 Implementation of prototype system

This chapter focuses on the implementation of a first prototype of the conceptualised automatic identification system (AIS) for the direct identification of piece goods. The implementation is described separately for hardware and software in the following sections. Section 4.1 deals in particular with the mechatronic implementation of the 3D-Sensing technology and the sensor platform, according to the concept proposed in Section 3.3. The actual core of the Auto-ID system in terms of software is implemented and explained in detail within Section 4.2.

4.1 Hardware implementation

The structure followed for implementing the hardware is shown in Figure 4.1. Since the 3D-Scanning process and all algorithms for 3D-Object recognition are computationally intensive, a personal computer (PC) is used as the higher-level computing and communication unit. The 3D-Scanners and turntables available on the market generally provide a universal serial bus (USB) interface for direct connection to computers. In order to connect the sensors of the sensor-platform to the computer, a microcontroller, which can also be connected via a USB, is interposed. Microcontrollers are often used for hardware-related computing tasks and offer great possibilities for operating sensors. Such systems based on microcontrollers are often referred to as embedded systems, which bridge the gap between sensor or actuator hardware and computers. The sensor platform in combination with the microcontroller therefore represents an embedded system. The prototype for the Auto-ID system to be realised in the context of this thesis will be implemented by means of the described structure.

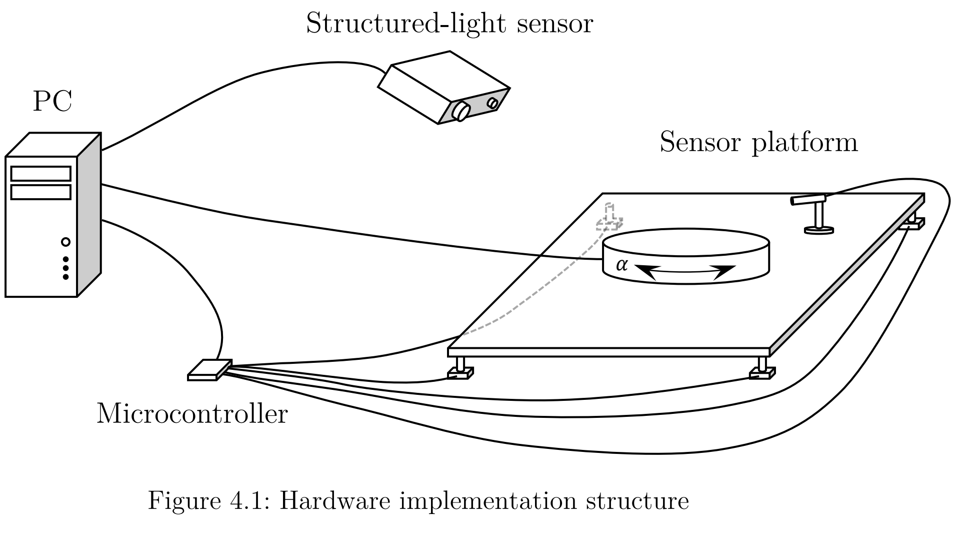


Figure 4.1: Hardware implementation structure

4.1.1 Structured-light sensor, rotary device and scanning software

Due to its availability at Reutlingen University's Werk150, a 3D-Scanning system of Hewlett-Packard's 3D-Scan series is used. The individual components of the available 3D-Scanning system that are used within this thesis are described in the following sections.

4.1.1.1 HP 3D Structured-light Scanner Pro S3

The HP 3D Structured-light Scanner Pro S3 consists essentially of an industrial camera and a light-emitting diode (LED) projector mounted on a tripod (see Figure 4.2). As explained in Section 2.2.5.2, the projector beams light patterns that are captured by the camera onto a scanning object. Depth is calculated from the distortion of these patterns in the image data. Since the industrial camera supplies colour image data, the colour texture of a scan object can also be captured. As can be seen from Table 4.1, the system offers excellent precision for a scan area of up to 500 x 500 mm. With the included calibration panel and the adjustable geometries between projector and camera, scans of the highest accuracy can be made using this structured-light sensor. The projector is connected directly to a PC via a high definition multimedia interface (HDMI) and the industrial camera via USB port.

Figure 4.2: HP 3D Structured-light Scanner Pro S3 (Hewlett-Packard 2017b, p. 1)

Table 4.1: Selected characteristics of HP 3D Structured-light Scanner Pro S3 (Hewlett-Packard 2017b, pp. 1–2)

Characteristics	Unit	Value
Weight	[kg]	8.8
Dimensions	[mm^3]	320 x 610 x 210
Scan area	[mm^2]	max. 500 x 500
Resolution	[%]	0.05 of scan size
Precision	[mm]	max. 0.05

4.1.1.2 HP Automatic Turntable Pro

An optional accessory to the HP 3D-Structured-light Scanner Pro S3 is the HP 3D Automatic Turntable Pro shown in Figure 4.3. The turntable rotates objects placed on it while scanning by 360 degrees to capture each side, thus enabling a complete scan of its surface. Specifications of the turntable are given in Table 4.2.

Figure 4.3: HP Automatic Turntable Pro (Hewlett-Packard 2017a)

Table 4.2: Selected characteristics of HP 3D Automatic Turntable Pro (Hewlett-Packard 2017a, p. 2)

Characteristics	Unit	Value
Weight	[kg]	1.6
Dimensions	[mm]	Ø180 x 50
Object weight	[kg]	max. 5
Turning angle	[°]	unlimited

4.1.1.3 HP 3D Scan Software Pro 5

The HP 3D Scan Software Pro 5, which also belongs to the HP 3D Structured-light Scanner Pro S3, offers all functions necessary to scan an object. As a first step, the software enables the adjustment and calibration of the structured-light sensor, which is the basis for the acquisition of precise scans. In the following stages the software related to the HP Automatic Turntable Pro offers the possibility to scan an object with any number of scans from different views and thus to collect surface fragments. In the final step, these fragments can be post-processed and merged with the software to eliminate any interference and thus obtain complete scans. The software also offers the export of the scanning results in standard tessellation language (STL) as well as polygon file format (PLY), which are the common formats used for 3D-Object recognition.

4.1.2 Sensor platform

The sensor platform for collecting further multi-sensor information consists of a microcontroller, four force sensors and an inductive sensor. The sensor platform is a completely new development within the scope of this thesis and uses individual components freely available on the commercial market. In the following, the individual components of the sensor platform which are used for implementation are shown.

4.1.2.1 Microcontroller

The Arduino Nano V3 board shown in Figure 4.4, based on the Atmel chipset ATmega328P, was chosen for the implementation of the microcontroller due to the author's previous experience with it. The ATmega328P chip is an 8-bit single-chip microcontroller with 32 kilobytes of programmable flash operating at a maximum frequency of 20 Megahertz. Further selected technical data of the Arduino Nano are shown in Table 4.3. Despite its small dimensions, this board and the microcontroller respectively provide more than sufficient computing power to read the four force sensors and the inductive sensor of the sensor platform and also to serially communicate with the PC via the integrated USB-Port. Beyond the hardware, the development of the embedded software for the microcontroller can be done with the Arduino integrated development environment (IDE).

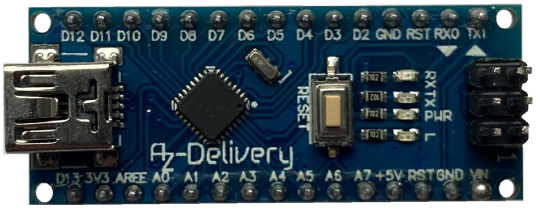


Figure 4.4: Arduino Nano V3 board

Table 4.3: Selected characteristics of Arduino Nano board

Characteristics	Unit	Value
Operating Voltage	[V]	5
Current Consumption	[mA]	19
Flash Memory	[KB]	32
Clock Speed	[MHz]	16
Analog IN Pins	[-]	8
Digital I/O Pins	[-]	22
Dimensions	[mm^2]	18 x 45
Weight	[g]	7

4.1.2.2 Force sensors and amplifiers

During the selection of the load cells, attention was paid to ensure that they could be easily mounted underneath a plate and that their dimensions were as flat as possible, while the measuring range was still large enough to accommodate the rotary device and the mass of the plate. A flat weighing plate design has the advantage that the platform does not stand out significantly from the background, which offers advantages for scanning and 3D-Object recognition. Figure 4.5 shows the selected force sensor, which can cover a measuring range up to 50 kg. Since a total of four force sensors are used, weights of up to 200kg could be measured, which is not a requirement for the prototype of this thesis. The sensor itself consists of a metal sensor body, whose strain is converted into a resistance change by means of strain gauges. By adapting the geometry of the sensor body, the weighing range can be adjusted with the same accuracy given by the strain gauge. By calculating and simulating the strains occurring on the sensor body, the necessary adjustment was calculated to adjust the measuring range to a maximum of 10 kg. All force sensors used within this thesis were adapted by milling, resulting in the characteristic presented in Table 4.4. A technical drawing outlining the exact dimensions can be found in Appendix B. Potential manufacturing inaccuracies are compensated for by calibration.

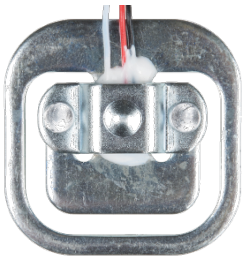


Figure 4.5: Original force sensor (SparkFun Electronics, p. 1)

Table 4.4: Selected characteristics of adapted force sensor based on (SparkFun Electronics, p. 1)

Characteristics	**Unit**	**Value**
Measuring range	[kg]	max. 10
Nonlinearity	[%FS]	0.03
Repeatability	[%FS]	0.03
Hysteresis	[%FS]	0.03

As can be seen in Figure 4.5 the selected force sensor with its three output cables forms one half of a Wheatstone bridge. Wheatstone bridges are often used in measurement engineering to precisely convert resistance changes into voltage changes. So-called analogue-to-digital converters (ADC) are used for the precise measurement of voltages in connection with microcontrollers, which convert analogue signals like voltages into digital signals. The HX711 from AVIA Semiconductor is a high precision ADC specially designed for force sensors with Wheatstone bridge circuitry. Figure 4.6 shows the HX711 board used in this thesis. The HX711 chip converts the measurement voltage of the Wheatstone bridge into a 24-bit value and provides this value to the microcontroller via a serial communication port. Table 4.5 shows selected characteristics extracted from the HX711 data sheet.

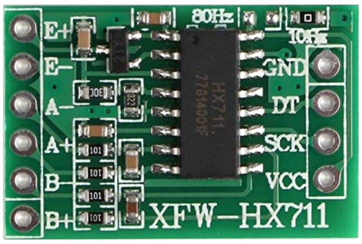


Figure 4.6: Force sensor amplifier HX711 board

Table 4.5: Selected characteristics of HX711 board (Avia Semiconductor, p. 1)

Characteristics	Unit	Value
Operating Voltage	[V]	5
Current consumption	[mA]	1.5
Resolution	[bit]	24
Sampling rate	[Hz]	10 or 80

4.1.2.3 Material sensor

An inductive proximity switch is used as a material sensor, which enables the differentiation between metals and non-metals. This type of sensor is widely used in industrial machines and can easily be used in combination with a microcontroller. Figure 4.8 shows the selected sensor Henschen LJ12A3-4-Z/BX, which is an NPN-type inductive proximity switch. NPN-type means that the electrical ground of a consumer is switched through, that is connected to the signal cable of the sensor when the sensor detects metal. By means of a so-called pull-up resistor circuit the sensor can easily be connected to the digital I/O pins of the microcontroller. Table 4.6 shows selected characteristics of the chosen inductive proximity switch.

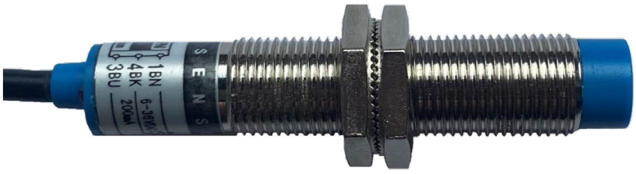


Figure 4.7: Inductive proximity switch Henschen LJ12A3-4-Z/BX

Table 4.6: Selected characteristics of Henschen LJ12A3-4-Z/BX

Characteristics	Unit	Value
Operating Voltage	[V]	6-12
Operating Current	[mA]	max. 200
Sensing range	[mm]	max. 5

4.1.2.4 Electronic circuit design

For the use of the force sensor in conjunction with the HX711 a Wheatstone bridge is used according to the deflection method. For this purpose, a bridge circuit as shown in Figure 4.8 is implemented and connected to a supply voltage U_0. The resistors R_1 and R_3 in the left branch represent the strain gauge's variable resistors of the force sensor. In the right branch, there are two fixed resistors R_2 and R_4. In the unloaded state of the force sensor all resistors have the same value and the bridge is therefore completely symmetrical. Equation (4.1) describes the bridge voltage U_B for small changes in resistance ΔR_1 and ΔR_3. The variable resistances of the strain gauge R_1 and R_3 actually undergo an opposite resistance change when deformed, which leads to a stronger influence on the bridge voltage U_B.

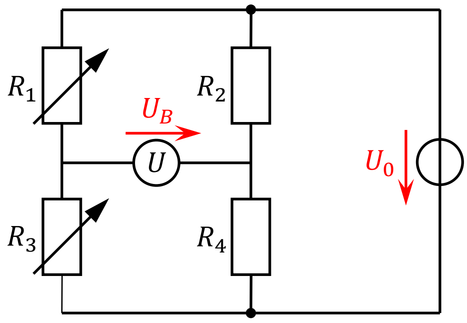


Figure 4.8: Deflection method Wheatstone bridge

$$U_B = \frac{U_0}{4 \times R} \times (\Delta R_1 - \Delta R_3), \qquad with\ R = R_1 = R_2 = R_3 = R_4 \tag{4.1}$$

The previously explained Wheatstone bridge is connected to the HX711 board as shown in Figure 4.9. In the unloaded state, the two resistors R1 and R3 of the strain gauge of the force sensors have a resistance of 1k Ohm, which is why R2 and R4 are also selected as 1k Ohm resistors. The pins E+ and E- of the HX711 board provide the supply voltage and are therefore connected to the two voltage dividing branches as illustrated. For measuring the bridge voltage, the measuring inputs A+ and A- of the HX711 board are connected between the resistors of the two branches. The resulting subcircuit is supplied with +5V supply voltage via the VCC/VIN and GND pins. The pins DT and SCK required for serial communication are connected directly to the digital pins of the microcontroller.

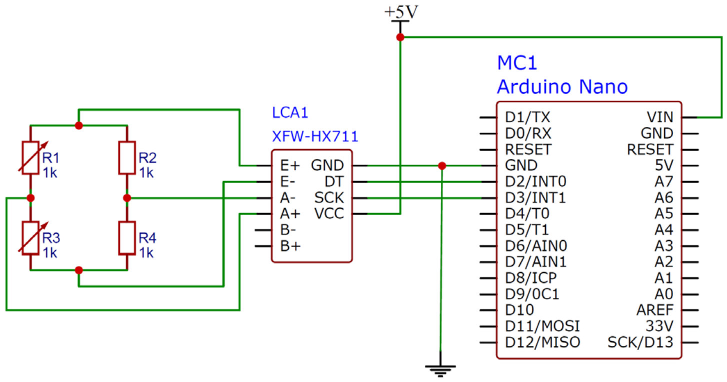


Figure 4.9: Force sensor and amplifier sub-circuit

The switching state of the inductive proximity switch is determined using a pull-up resistor circuit connected to the microcontroller. Figure 4.10 shows the underlying principle of pull-up resistor circuits, which are used to correctly bias the inputs of digital gates to stop them from floating about randomly when there is no input condition. When the switch S is open, almost no current flows through the large ohmic resistor R, so the gate voltage U_G is practically equal to the supply voltage U_0. When the switch S is closed, a current $I = U_0 / R$ limited by the resistance flows. In this case the gate voltage U_G is securely pulled to zero potential.

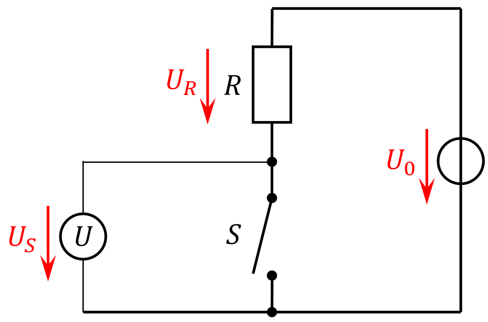


Figure 4.10: Pull-up resistor principle

The circuit used in this thesis, based on this principle, is shown in Figure 4.11. The pull-up resistor R1 is connected between a +5V supply voltage and the ground-switching Signal pin of the inductive proximity switch S1. The digital gate pin D11 of the microcontroller MC1 is also connected to the pull-up resistor and thus only detects the states low (0V) or high (+5V).

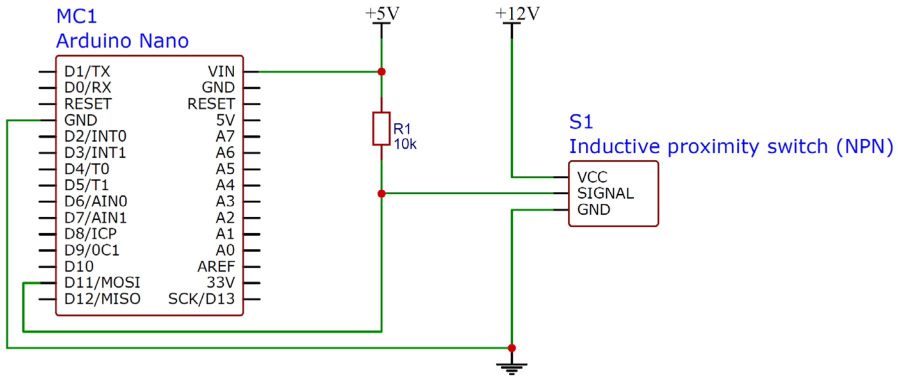


Figure 4.11: Inductive proximity switch sub-circuit

The complete design of the electronic circuit for the sensor platform is shown in Figure 4.12. For simplification, Figure 4.9 shows only one force sensor and amplifier, in fact four of these combinations are used within the overall circuit. So as to keep the cable length between force sensor and amplifier as short as possible, only one force sensor per HX711 board was purposely used. This reduces the disturbing influence of external inductions on the analogue voltage signals and thus significantly improves the accuracy of the measurement. Each HX711 board is connected by two cables to the Arduino Nano for serial communication, which leads to the occupancy of pins D2 to D9. The subcircuit for detecting the stator of the inductive proximity switch is connected via digital pin D11 of the Arduino Nano. Additionally, an LED is attached to pin D10 of the Arduino Nano board in order to visualise the status of its USB connection to the PC. For the power supply of the entire circuit a +5V as well as a +12V power source is required.

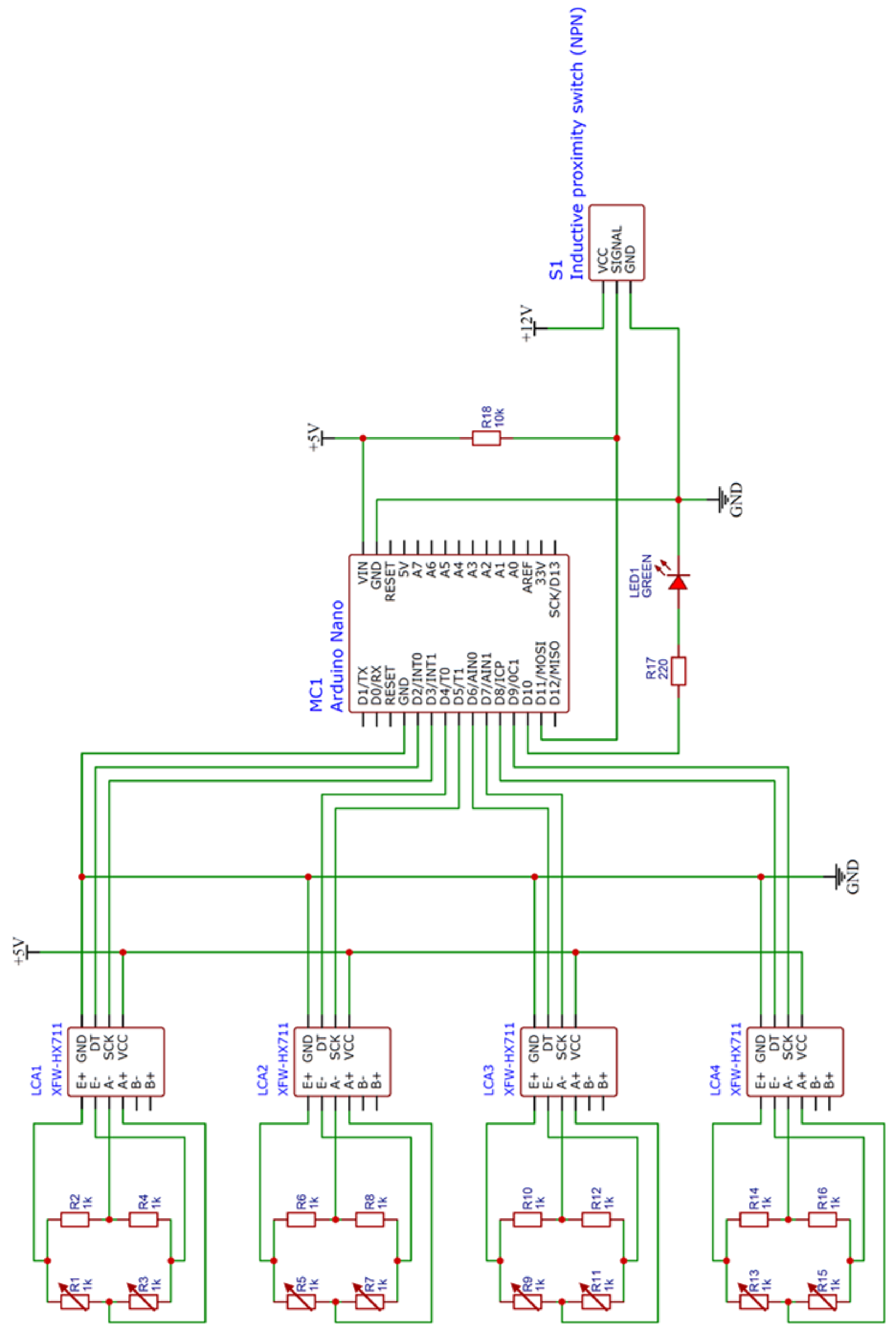


Figure 4.12: Complete electronic circuit for sensor platform

4.1.2.5 Weighing plate assembly design

The weighing plate assembly was designed using CAD tools. In order to ensure sufficient stability for the accommodation of the HP Automatic Turn Table Pro rotary device and an identification object placed on it, a 300x300x5 mm steel plate is used as a base. For attaching the force sensors to the corners of this steel base plate, a 3D-printable two-piece custom mount has been designed. After inserting a sensor into the two-piece mount, it is attached to the steel plate using two M3 countersunk screws. Figure 4.13 shows the assembly of the weighing plate in an exploded view. Detailed drawings of all components and subassemblies are given in Appendix B.

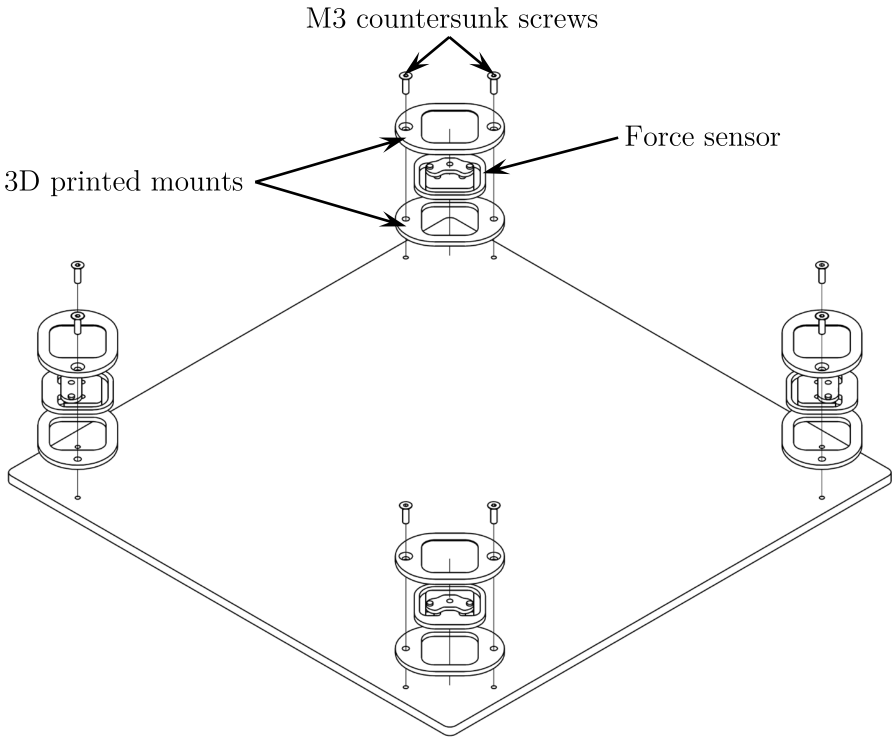


Figure 4.13: Exploded view of the weighing plate assembly (bottom view)

4.1.2.6 Material sensor assembly design

According to the proposed concept (see Section 3.3) the material sensor will be movably mounted on the weighing plate. Since the weighing plate is made of steel, magnets can be used to create a sliding mount that fixes the inductive proximity switch. The material sensor assembly thus consists of a mount designed for 3D-Printing, the inductive proximity switch and two magnets. Figure 4.14 shows an exploded view of the designed assembly, with the magnets bonded to the mount and the sensor fastened by screws. All detail drawings of the components are given in Appendix B.

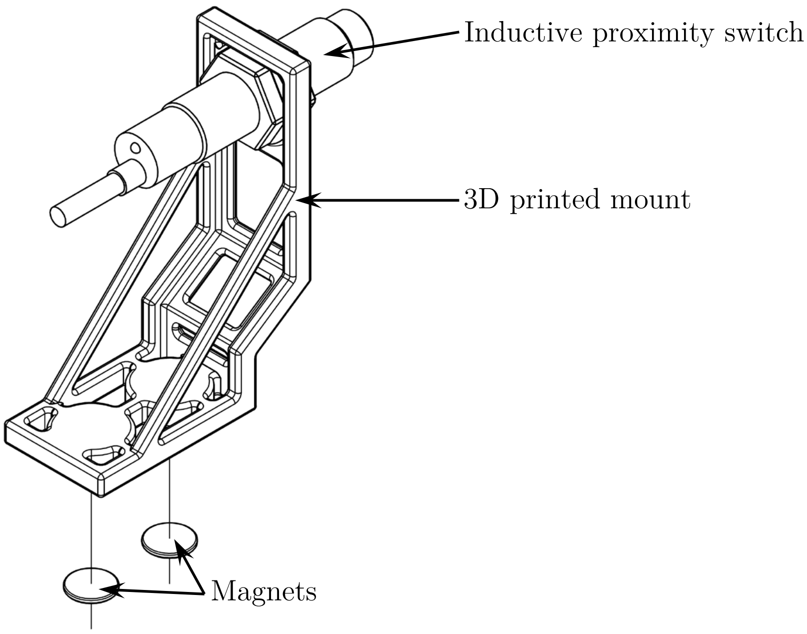


Figure 4.14: Exploded view of material sensor assembly

4.1.3 Results of hardware implementation

On the basis of the weighing plate assembly and material sensor assembly previously specified using CAD tools, a hardware prototype of the sensor platform was implemented. Figure 4.15 shows the prototype of the sensor platform carrying the rotary device and an identification object. The electronic circuit designed in Section 4.1.2.4 using the components presented in Section 4.1.2.1, Section 4.1.2.2 and Section 4.1.2.3 was realised by means of custom-soldered circuit boards for the force sensor and amplifier subcircuits as well as a breadboard setup. The force sensor mounts and the material sensor mount, which were specially designed for 3D-printing, were manufactured with a fused deposition modelling (FDM) 3D-Printer at the Reutlingen University's Werk150. The eight metric M3x0.5 threads of the steel plate were added manually by drilling and tapping based on the drawing (see Appendix B). By mounting the force sensors using the brackets and M3 countersunk screws, the distance between the measuring points of the bearing force is 255 mm. Within this square area of 255x255 mm^2 defined by these measuring points a plane coordinate system results, which represents the reference system for determining the centre of mass as explained in Section 3.2.1.3.

Figure 4.16 shows the complete setup of the hardware prototype including the structured-light sensor, while scanning. By calibrating the scanner and performing a scan without an identification object in a first step, the background can simply be subtracted from a scan that contains the object. In combination with the flat design of the sensor platform this procedure leads to excellent scans, which only describe the identification object.

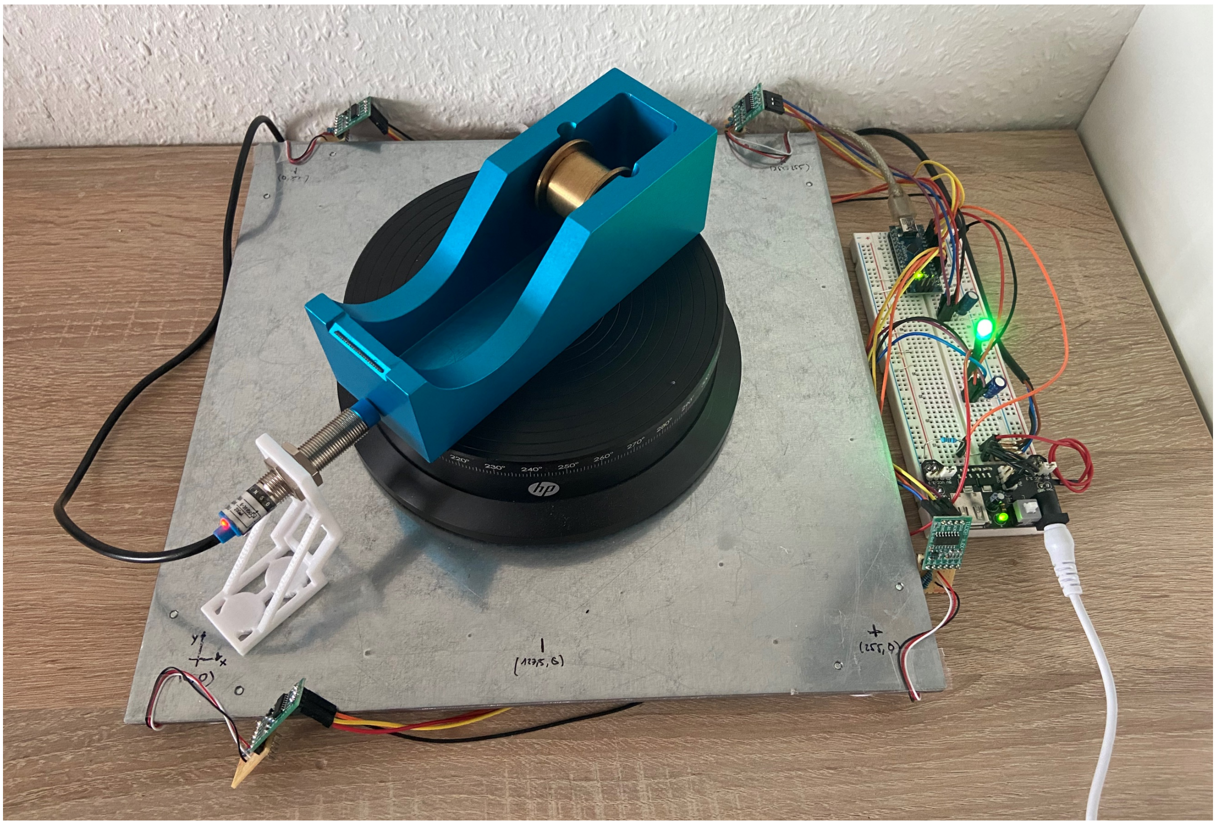


Figure 4.15: Sensor platform prototype with rotary device carrying identification object

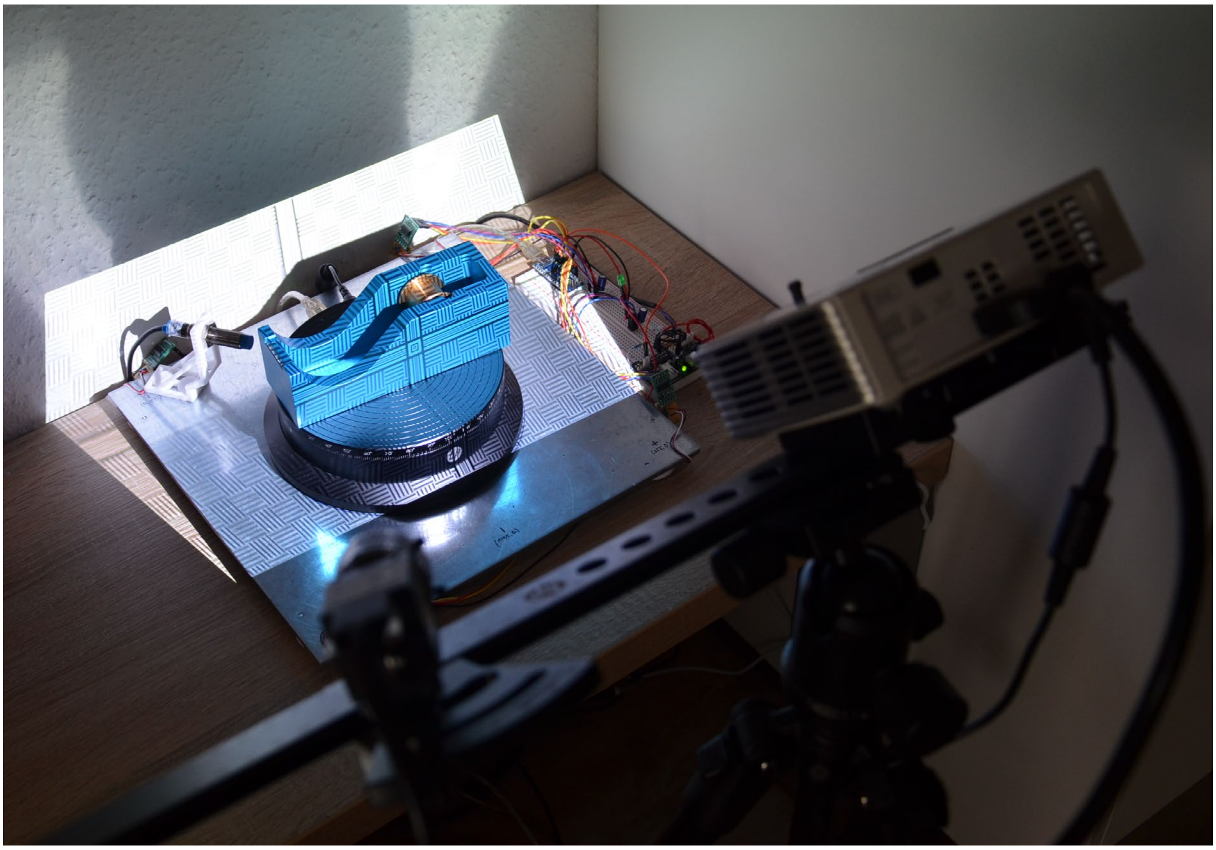


Figure 4.16: Complete hardware prototype while scanning

4.2 Software implementation

In this section the implementation of all the software enabling multi-sensor identification is described, which corresponds to the component structure of the software concept developed in Section 3.2.2.2. Additionally, the software of the sensor platforms microcontroller is implemented.

4.2.1 Process structure for multi-sensor identification

On the basis of the proposed generic identification process (see Section 3.2.2.1) and the functional structure of the 3D-Object recognition pipelines (see Section 2.2.3), the process structure shown in Figure 4.18 is proposed and implemented.

In the offline process an identification knowledge base is created on the basis of CAD-Models. For this purpose, all CAD-Models are first converted into point clouds, which are then described using descriptors for 3D-Object recognition. In addition, the identity information is extracted from the CAD-Model and stored together with the associated “CAD point cloud” and its “CAD descriptor” as a data record in the identification knowledge base. The execution of the last-mentioned steps in an offline process is very useful, as all steps are computationally intensive and therefore take a lot of time. Similarly, the generation of the identification knowledge base only needs to be performed when new identification objects in the form of new CAD-Models are added. The offline process ends with the provision of the identification knowledge base to the online process.

The online process starts by retrieving the multi-sensor information: Weight, colour, material and sensor point cloud. Using the weight, colour and material information, a pre-filtering step takes place which filters out the data records of the identification knowledge base corresponding to the sensor information. The resulting prefiltered identification knowledge base is then provided for 3D-Object recognition. As already mentioned in Section 3.2.2.2, this prefiltering offers two basic advantages: Saving time and differentiating between objects that are visually indistinguishable. For 3D-Object recognition in the online process, the descriptor for the sensor’s point cloud is required in addition to the pre-filtered identification knowledge base. This descriptor is generated by an intermediate step from the sensor point cloud. After carrying out the 3D-Object recognition step, the data set from the identification knowledge base is available that best matches the sensor point cloud acquired from the scene. Knowing this set of data, an optional validation step can be performed by comparing the position of the centre of mass between the identification object and the CAD-Model, which further improves the identification reliability. Finally, the identity of the recognised object can be retrieved. The identity information is available at the end of the entire process structure.

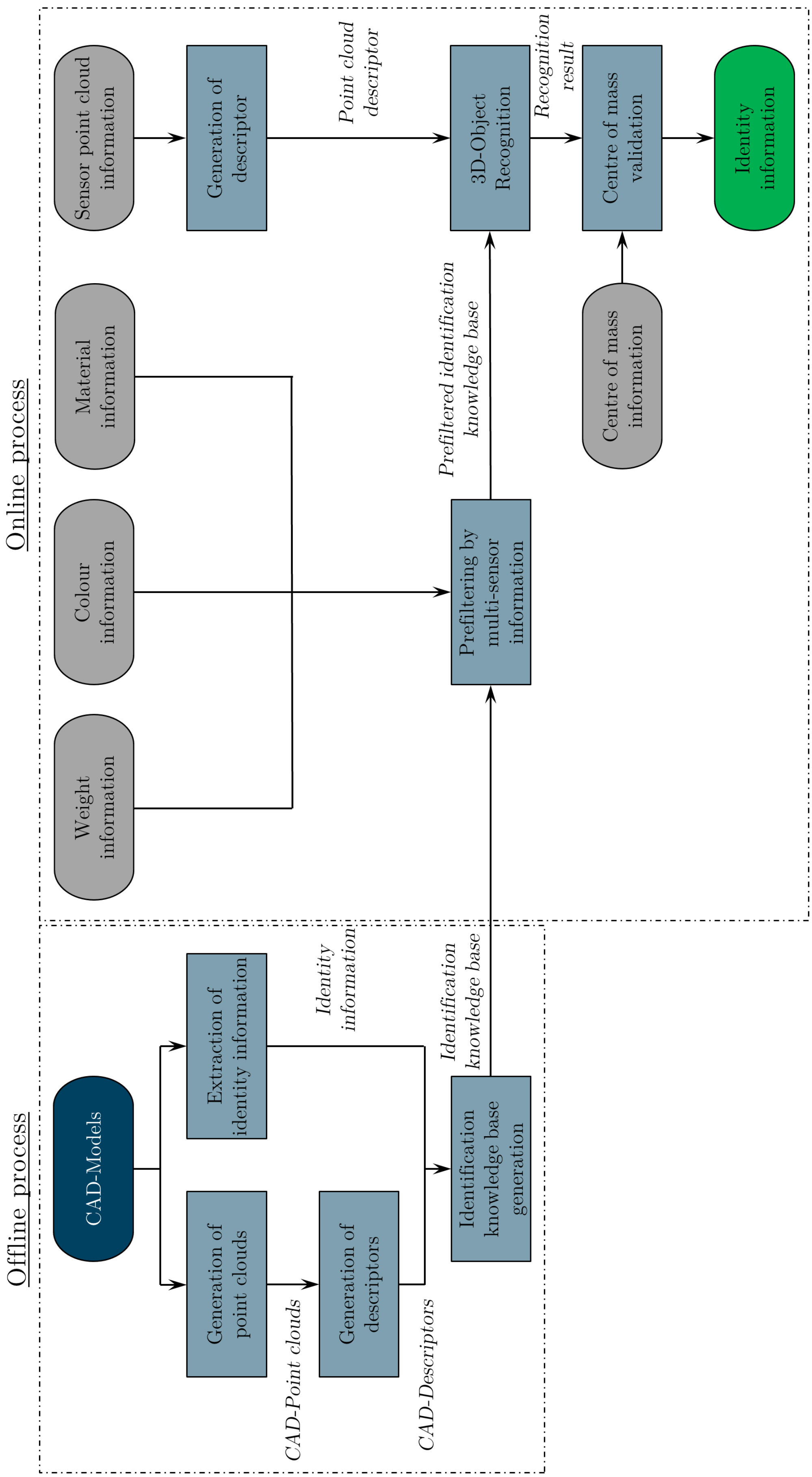


Figure 4.17: Process structure for multi-sensor identification software

4.2.2 Centre of mass validation for identification accuracy improvement

The aim of mass centre validation is to compare the location of the centre of masses projection of an identification object onto the sensor platform with the location detected by sensors. As shown in Figure 3.9, three different coordinate systems describing three-dimensional space $\mathbb{R}^3$ have to be considered for this purpose. The geometrical relations between the three coordinate systems with respect to the centre of mass projection of the Auto-ID system developed in this work are shown in Figure 4.18. By fundamental three-dimensional transformation by translation (Equation (4.2)) and rotations around the x-axis (Equation (4.3)), y-axis (Equation (4.4)) and z-axis (Equation (4.5)), point clouds within one coordinate system can be arbitrarily transferred into other coordinate systems. Scaling of coordinate systems is not considered, since exact dimensions are necessary for identification.

For the centre of mass validation, it is reasonable to choose the coordinate system of the sensor platform (Index SP) as the fundamental reference. This is due to the fact that the centre of mass position detected by the sensors of the weighing plate can be directly compared with the calculated position of the centre of mass projection point C'. The point cloud of the scanned identification object is primarily determined in relation to the coordinate system of the structured-light sensor (Index S). Calibrating the structured-light sensor is necessary to transform the point clouds recorded by it into the coordinate system of the sensor platform. If a point cloud is now detected by the sensor, its points are given in relation to the coordinate system of the sensor plate. After performing 3D-Object recognition including 6-DoF pose estimation, the geometric transformation is known, which transfers the coordinate system of the CAD point cloud (Index O) to the sensor plate coordinate system. If the CAD-Models centre of mass C is known in relation to the coordinate system of the CAD point cloud (Vector $\overrightarrow{O_O C}$), this point can be transferred to the coordinate system of the sensor platform. The projection C' of the centre of mass C on the xy-plane of the sensor platforms coordinate system can then easily be gathered by zeroing the z-coordinate of point C. Through comparison of the $x_{c\prime}$ and $y_{c\prime}$ coordinates with the coordinates determined by the weighing plate, centre of mass validation would be enabled.

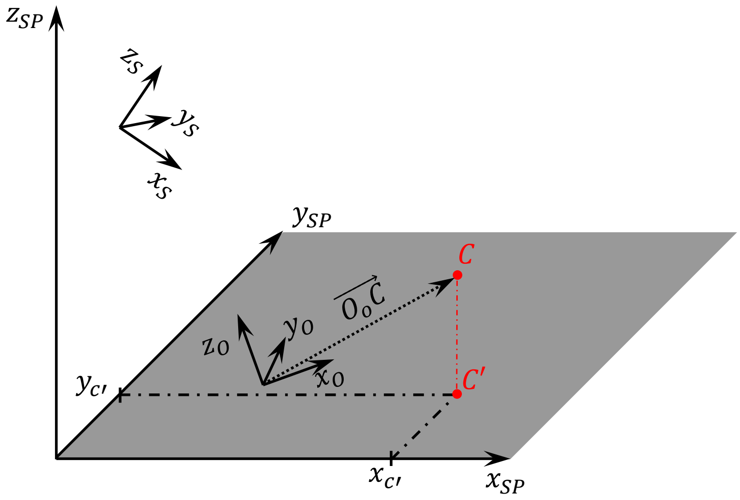


Figure 4.18: Geometric considerations for centre of mass validation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.5}$$

4.2.3 Programming language and software libraries

According to IEEE Spectrum's sixth annual interactive ranking of the top programming languages Python is currently the most popular programming language (Cass 2019). Due to its popularity, the experience of the author as well as the excellent suitability for software development, the high-level programming language Python is mainly used for implementation of the multi-sensor identification software in this thesis.

The software library Point cloud library (PCL) identified in the systematic literature review (see Section 2.2.2.5), is an open-source library that offers algorithms for three-dimensional machine vision. The native PCL is written in the programming language C++. Existing bindings, which allow a linkage between Python and PCL written in C++, are incomplete and poorly documented. For this reason, a custom software module for 3D-Object Recognition is implemented for this thesis using C++, and then integrated into Python using the binding library Pybind11. The PCL is used in particular to describe point clouds using CVFH descriptors and to perform recognition.

Two further libraries used for generating and processing point clouds from CAD-Models within this work are Open3D and Open CASCADE. Open3D is an open source library available in the programming language Python as well as C++ for rapid development of software that deals with three-dimensional data. Open CASCADE is an open source software development kit for the development of applications related to CAD, also available in Python. In the context of the software in this thesis, Open CASCADE is used to preprocess CAD-Models for use with Open3D. Using the uniform sampling and voxel-grid algorithms (see Section 2.2.3) offered by Open3D, the CAD-Models are then transformed into point clouds. Furthermore, Open3D is used to perform the 6-DoF pose estimation by means of global and local registration.

The library OpenCV is used to convert the RGB image data of the optical sensor into colour information. OpenCV is an open source library available in Python and C++ for the development of computer vision applications. In particular, the algorithms provided by OpenCV for masking image data as well as for edge detection were used to obtain colour information from the sensor data.

4.2.4 Component implementation according to the software concept

This section describes the implementation of the 'realisation' required by the software concept for each component as stated in Section 3.2.2.2. These component-related realisations are implemented by means of object-oriented programming. For this purpose, so-called classes are defined, which represent a construction plan for the derivation of explicit instances. Each class consists of a name, attributes and methods. The attributes describe the properties one instance can have, where the values of these attributes can vary from instance to instance. The methods describe the behaviour or actions that each instance of a class can perform. This abstract approach of programming empowers the modularity and expandability of each component and thus leads to a flexible overall software.

In the following sections, the classes implemented for each component are explained based on their methods. For reasons of simplicity, only public classes that can be called from outside an object and are necessary for understanding, are explained. The exact implementation can be found in the source codes presented in Appendix C.

4.2.4.1 3D-Object recognition component

For the realisation of the 3D-Object recognition component two classes are implemented. The first class enables 3D-Object recognition using the CVFH descriptors selected in the systematic literature review (see 2.2.2.5) and is therefore called "CVFHDescriptorRecognition". This class only covers the generation and matching of descriptors, which is why a further class called "SixDoFPoseEstimation" is implemented for the 6-DoF pose estimation. Table 4.7 provides an overview of the methods of the two classes, which are explained below. All the methods available through the classes serve to cover the component realisations required by the software concept (see Section 3.2.2.2).

The class "CVFHDescriptorRecognition" establishes the connection between Python and the PCL C++ module developed especially for this thesis. As the name suggests, its method "generate_descriptor(*args)" generates CFVH descriptors for a point cloud passed to it in the form of a system path. After transferring this path to the C++ module, the module initially reads in the point cloud file. Subsequently, the surface normals for each point within the point cloud are estimated by means of NNS. By determining the nearest neighbours of a point, conclusions can be drawn about the local topology of the surface around this point and thus a geometric normal can be estimated. The search radius which is used to determine the nearest neighbours must be defined in relation to the resolution of the sample point cloud. After estimating all normals for all points, the actual CVFH descriptor is calculated for the entire point cloud. This descriptor is then stored to another system path, given through the arguments of the "generate_descriptor(*args)" method.

The "perform_training(*args)" method of the class "CVFHDescriptorRecognition" trains the knowledge base required for 3D-Object recognition. For this purpose, a list of CVFH descriptor system paths of objects to be recognised is passed to the method. Based on these descriptors the method creates a k-d-tree search structure, which is the foundation for matching. This search structure defining the recognition knowledge base is then exported to a file and stored to a predefined system path.

The last method of the class "CVFHDescriptorRecognition" to be described is the matching method "perform_matching(*args)". By calling up this method the actual 3D-Object recognition takes place by matching. A descriptor passed to the method is compared with all the descriptors stored in the recognition knowledge base. The quality of a comparison is determined by the Chi-squared distance metric. If the descriptors match exactly, the Chi-squared distance metric has a value of zero. In reality this does not happen because a descriptor of a sensor point cloud is never as ideal as the descriptor of a point cloud derived from a CAD-Model. The matching descriptor out of the recognition knowledge base is therefore the descriptor which has the smallest distance to the sensor point cloud descriptor. This matching descriptor against the descriptor passed to the function is returned by the function "perform_matching(*args)" together with the value of the Chi-squared distance metric.

Table 4.7: Methods for public use of class CVFHDescriptorRecognition

Method	Description
*generate_descriptor(*args)*	Generates a CVFH-Descriptor for a point cloud passed to the method
*perform_training(*args)*	Trains the knowledge base required for 3D-Object recognition
*perform_matching(*args)*	Matches a CVFH-Descriptor passed to the function to all elements of the recognition knowledge base

The only method of the class "SixDoFPoseEstimation" for public calling is "get_transformation(*args)". After transferring two system paths of point cloud files to the function, the function calculates the geometric transformation matrix for transferring the point clouds into each other. This matrix thus represents the connection between the two coordinate systems of the point clouds. Within this thesis the matrix can therefore be used to transfer the CAD point cloud into the sensor point cloud and thus calculate the projection of the centre of mass on the coordinate system of the sensor platform.

Table 4.8: Methods for public use of class SixDoFPoseEstimation

Method	Description
*get_transformation(*args)*	Calculates the transformation matrix in order to transform two point clouds into each other

4.2.4.2 Multi-sensor data processing component

The realisation of the component "Multi-sensor data processing" is achieved by implementing two classes. The class named "StructuredLightSensor" is responsible for processing and providing the data of the structured-light sensor, which acquires the geometry of an identification object as a point cloud. For processing the data of the multiple sensors of the sensor platform, the class named "SensorPlatform" is implemented. Table 4.9 and Table 4.10 summarise the methods of these two classes.

The class "StructuredLightSensor" has two methods which on the one hand enable the point cloud to be retrieved and on the other hand extract the colour of the identification object from the sensor data. By calling the method "get_point_cloud()" the data of the structured-light sensor are read in and preprocessed for 3D-Object recognition. This preprocessing essentially consists of downsampling the sensor point cloud to the same resolution as chosen for the CAD point clouds using voxel downsampling. The benefit of this is that the descriptors of the CAD point cloud and sensor point cloud are more similar, which improves the overall quality of recognition. The second method "get_color()" of the class "StructuredLightSensor" reads the RGB-Image data available from the structured-light sensor in a first step. By applying a so-called colour mask to the RGB image data, areas can be extracted that correspond to the colour of the colour mask. Using colour masks in all primary colours, image areas with known colours can be extracted. The largest of these different coloured areas is determined as the predominant colour.

Table 4.9: Methods for public use of class StructuredLightSensor

Method	Description
get_point_cloud()	Retrieves the point cloud of the structured-light sensor and makes it available for 3D-Object recognition
get_color()	Processes the RGB-Image data of the structured-light sensor and extracts the predominant primary colour

The "SensorPlatform" class consists of three methods that are used to acquire material information, weight information and centre of mass location from an identification object placed on the sensor platform. All three methods communicate via a serial connection with the microcontroller software. When a method is called up, a command is sent to the microcontroller, which directly responds with the required information. Table 4.10 gives a further description of the individual methods, which are not very complex.

Table 4.10: Methods for public use of class SensorPlatform

Method	Description
get_material()	Retrieves the identification objects material information (metal or non-metal) from the sensor platform
get_weight()	Retrieves information about the total weight of an identification object placed on the sensor platform
get_centreofmass()	Gets the centre of mass position in relation to the coordinate system of the sensor platform

4.2.4.3 Multi-sensor identification component

The core component "Multi-sensor identification" consists of only one class with two methods. The method "generate_identification_knowledge_base()" triggers the offline process illustrated in Figure 4.17, which generates the identification knowledge base using the CAD-Models. For this purpose, each CAD-Model is first converted into a point cloud by sampling points into its surfaces. By internally initialising an object of class "CVFHDescriptorRecognition" from the "3D-Object recognition component", a descriptor is then created for each CAD point cloud by calling its function "generate_descriptor(*args)". The CAD point cloud, descriptor and identity information are stored together as an identification knowledge base, which is then available for the online identification process.

The online identification process is executed when the function “perform_identification()” is called. Internally, objects of all classes of the components “3D-Object recognition” and “Multi-sensor data processing” are initialised. By calling up the previously explained methods of these objects, the online multi-sensor identification process is performed as shown in Figure 4.17 and explained in Section 4.2.1.

Table 4.11: Methods of class MultiSensorIdentification

Method	**Description**
generate_identification_knowledge_base()	Generates the identification knowledge base from CAD-Models (offline process)
perform_identification()	Performs the multi-sensor identification (online process)

4.2.5 Microcontroller software

The software implementation for the selected microcontroller in form of an Arduino Nano is performed using the corresponding Arduino IDE. Since microcontrollers usually possess very limited on-chip memory, compilers are used to convert the programming-friendly high-level languages into machine code executable by the microcontroller. The Arduino IDE essentially consists of a high-level language editor and a compiler that loads the compiled machine code directly to an Arduino board connected to the PC via USB. The high-level programming language used by the Arduino IDE is the C programming language.

In order to provide the multi-sensor data on request of the multi-sensor data processing component (see Section 3.2.2.2) of the main software, the microcontroller software must perform the following tasks:

- Read sensors raw value (24-bit) from force sensor amplifier HX711
- Calibration and scaling of raw sensor values
- Conversion of raw sensor values into weight data
- Conversion of raw sensor values into centre of mass position data
- Read inductive sensor value and conversion to material data
- Serial communication with PC via USB

The software developed and implemented in the form of the source code can be found in Appendix C.1. The source code is commented and allows an easy understanding of the program flow. At this point no further explanation is given, as this would exceed the scope of this thesis.

4.2.6 Results of software implementation

The software was implemented according to the software concept (see Section 3.2.2) and the described process structure for multi-sensor identification (see Figure 4.17). The process step "Centre of mass validation" (see Figure 4.17) could not be implemented as two facts found inhibit this:

1. The conversion of the CAD-Models into point clouds by sampling points in their surfaces with the software library Open3D shifts the coordinate system in an unknown way and the centre of mass information available from the CAD-Models can thus no longer be associated.
2. The HP 3D Structured-light Scanner Pro S3 used in combination with the corresponding HP 3D Scan Software Pro 5 does not allow calibration to a specific coordinate system. Although the scanner is calibrated using a calibration panel, the point cloud generated by the software has a random origin. According to the manual and consultation with HP, there is no possibility to achieve such a calibration with this scanning system.

Since the process step "Centre of mass validation" within the framework of the multi-sensor identification only allows an additional validation of the identification result, this only leads to a further improvement of the identification accuracy. Both facts can only be solved by considerable time and development effort, which would go beyond the scope of this thesis and are therefore proposed for follow-up work. The software implementation of this thesis thus ends with the determination of the 6-DoF pose of the CAD point cloud in relation to the point cloud captured by the 3D sensor. In addition, the position of the centre of mass measured by the sensor platform can already be retrieved by the software.

The result of the software implementation is nevertheless a functioning software for identification by means of multi-sensor information. Both the offline process and the online process can be carried out using the implemented software, which in combination with the implemented hardware (see Section 4.1.3) results in a working prototype multi-sensor AIS for direct identification of unpackaged piece goods.

The verification of this multi-sensor AIS prototype is the subject of the verification performed in the following chapter.

5 Verification of prototype system

This chapter is dedicated to the verification of the multi-sensor information identification system, which was developed and implemented in the previous chapters. In the first step the implemented system is verified against the software and hardware requirement defined in Section 3.1. In the second step, the ability of the system to identify several different components based on their natural identification features will be verified.

5.1 Verification against hardware and software requirements

The result of the hardware implementation is described in Section 4.1.3 in the form of the prototype. At the beginning of the development, the requirements were defined, which the hardware of the multi-sensor Auto-ID system to be developed for the identification of unpackaged piece goods on the basis of their natural characteristics must fulfil. The specific hardware requirements were defined in Table 3.1 in Section 3.1.1. In the following paragraphs these requirements are verified against the actually implemented hardware prototypes.

HR1 requires that the hardware allows the storage of an unpackaged piece good for identification. The hardware prototype makes this possible by simply placing such an unpackaged piece good on the rotary device as can be seen from Figure 4.15. During development, other storage options were also considered, but these did not prevail over the selected solution (see Section 3.2.1.2). Placing cylindrical parts on the rotary device is also possible, but requires a suitable support. As the storage of such parts is generally associated with this disadvantage, this is not considered to be a disadvantage of the proposed solution. HR1 is therefore considered to be fully complied with.

HR2 requires the ability of the hardware to scan a stored piece of good as completely and accurately as possible to enable an exact identification. The rotary device in the form of HP Automatic Turn Table Pro in combination with the very precise HP 3D Structured-light Scanner Pro S3 allows identification objects to be captured fairly completely and very accurately (see Section 4.1.1). In fact, by scanning several times from different angles and rotating the identification object, the scan can be performed in its entirety, thus creating a point cloud that almost fully describes the model. Deep holes and shiny spots in the surface are the only elements that cannot be scanned sufficiently. However, this disadvantage exists with all 3D-Sensing technologies and can be partially eliminated. Since the scan is not complete without scanning several times and from different perspectives this hardware requirement is not fully complied with. The use of multiple or non-stationary 3D scanners could remedy this and lead to more complete scans.

HR3 requires the measurement of the weight of an identification object by the hardware. Four force sensors are used in the hardware prototype for this purpose, which can determine the weight of the object placed on the sensor platform very precisely (see Section 4.1.2.2). This hardware function is therefore considered to be completely fulfilled.

HR4 requires making the distinction between metals and non-metals, which is related to the hull material of an object to be identified. With the realised material sensor mount, this difference can be fulfilled by the hardware prototype with the help of an inductive proximity switch. With the realised material sensor mount (see Section 4.1.2.6), this differentiation can be accomplished using the inductive proximity switch of the hardware prototype. One drawback of this design is that the material sensor mount must be manually guided to the identification object by an operator. At the same time, only external materials can be perceived through the inductive sensor. If an identification object consists of several materials and the operator guides the sensor to the wrong surface spot, the identification might fail. As all materials are defined by the CAD-Model, the material information is not fully exploited. In fact, the possibilities of differentiating between materials by means of sensor technology are very limited, which is why technology sets a limit here. It would, however, be possible to inspect the material at various points on the object after optical recognition using a robotic arm. HR4 could therefore have been better implemented.

HR5 requires the recognition of primary colours by the hardware. For this purpose, RGB image data is provided by the HP 3D Structured-light Scanner Pro S3 in the implemented prototype. The RGB image data is collected by the structured light scanner by exposing the identification object through the projector in red, green and blue. The camera of the scanner is a monochrome camera, which therefore can only measure the intensity of the reflected light. For each exposed colour, the colour profile of the scene can be calculated and converted to an RGB image. The described method offers good precision and is above all less sensitive to reflective surfaces than an RGB camera. From the image data provided by the hardware, the primary colours can be determined by appropriate methods of computer vision. HR5 is therefore considered to be completely fulfilled.

HR6 requires the hardware to be connected and operated by a computer. All implemented hardware components can be easily connected to a computer using USB ports and an HDMI port. Through these standardised connection methods, a secure connection between hardware prototype and computer can be established, which simultaneously guarantees operability. HR6 is therefore completely complied with.

In summary, the implementation of the hardware through the hardware prototype meets the requirements well. Some points for improvement were mentioned in the previous paragraphs and offer potential for follow-up work. Table 5.1 summarises the verification of the implemented hardware and determines the degree of compliance according to (VDI 2225-3:1998-11).

Table 5.1: Hardware verification results

Hardware requirement	Degree of compliance	Rating (VDI 2225)
HR1	ideal	4
HR2	sufficient	2
HR3	ideal	4
HR4	good	3
HR5	ideal	4
HR6	ideal	4
		21/24 (87,5 %)

The requirements for the software of the multi-sensor Auto-ID system to be developed were defined in Table 3.2 presented in Section 3.1.2. Analogous to the procedure for the verification of the hardware implementation, the following paragraphs serve the verification of the software implantation against the requirements placed on it.

SR1 requires interfaces of the software to communicate with the sensor hardware. These interfaces to the sensor hardware were already considered in the software concept (see Section 3.2.2.2) and were completely implemented in the prototype software (see Section 4.2.4.2). By calling up the methods given by the software implementation, the communication to retrieve the information from the sensory hardware occurs automatically. SR1 is therefore considered to be fully complied with.

SR2 demands the development of the identification knowledge base of the multi-sensor Auto-ID system on the basis of CAD-Models. The implemented software enables the import of CAD-Models in STEP and STL format, where the identity information is encoded via the file name. The STEP format is a widely used and well-standardised format and can be exported by all common CAD applications (see Section 2.3.1). The implemented software thus follows a general approach and is not limited to specific software manufacturers. SR2 is therefore considered fully satisfied.

SR3 requires the handling and processing of multi-sensor data through the implemented software. The implemented software prototype enables this through the software of the microcontroller and the “Multi-sensor data processing” component. In fact, much of the processing of sensor platforms data into information is already done by the microcontroller software, which converts the sensor data streams into information and provides it to the “Multi-sensor data processing” component. The data of the structured light sensors is processed completely in the latter component. The software implementation thus completely fulfils SR3.

According to SR4 the software implementation must enable 3D-Object recognition including 6-DoF pose estimation. The 3D-Object recognition is enabled by a specifically developed C++ software module integrated in Python. This module allows recognition through a global recognition pipeline based on CVFH descriptors, following the approach selected through the systematic literature review (see Section 2.2.2). The 3D-Object recognition is thus implemented according to the latest literature. For the estimation of the 6-DoF pose of a CAD point cloud in relation to the sensorially recorded point cloud, a Python implementation is used. Using the methods of the two classes "CVFHDescriptorRecognition" and "SixDoFPoseEstimation" (see Section 4.2.4.1), SR4 can be fully met through the implemented software.

SR5 requires an algorithm which enables identification based on the combination of multi-sensor information. This combination of multi-sensor information for identification purposes is illustrated by the online identification process, which is shown in the overall process structure illustrated in Figure 4.17. As explained in Section 4.2.6, the centre of mass validation could not yet be implemented within this thesis, which excludes this information for use. However, identification is carried out beforehand and would only be further improved by this step. As this work is the first approach to the identification of piece goods on the basis of their natural features through multi-sensor identification and the process mentioned above still enables this, SR5 is considered to be fulfilled well.

The communication of the identity information of an identified piece good to downstream systems is required by SR6. The implemented software can retrieve all identity information stored in the identification knowledge base and transfer it to downstream systems. Depending on the type of interface to the downstream systems, additional communication modules must be implemented. SR6 is fully compliant and Python makes it easy to implement a variety of interfaces to downstream information systems.

The software implementation also fulfils the requirements placed on the software well. Only the implementation of the centre of mass validation is lacking due to the problems described in Section 4.2.6 which leads to a non-ideal compliance with regard to SR5. Table 5.2 summarises the above verification of the software implementation and makes and assesses the degree of compliance based on the scale according to (VDI 2225-3:1998-11).

Table 5.2: Software verification results

Software requirement	Degree of compliance	Rating (VDI 2225)
SR1	ideal	4
SR2	ideal	2
SR3	ideal	4
SR4	ideal	4
SR5	good	3
SR6	ideal	4
		23/24 (95,8 %)

5.2 Experimental verification

For the experimental verification of the implemented multi-sensor information Auto-ID system, a selection of objects is tested to ensure that they are correctly identified. The identification knowledge base is first established on the basis of the CAD-Models of the objects to be identified by means of the procedure described in Appendix D.1. The online identification process is then carried out three times for all identification objects in random order. The procedures and necessary steps for carrying out the online identification process are outlined in Appendix D.2.

5.2.1 Identification objects for experiment

A set of identification objects was compiled for verification purposes. When selecting the objects, attention was paid to the fact that they vary in geometry, colour, weight and material. Figure 5.1 shows the entire set of identification objects, which can be assigned to the descriptions in Table 5.3 using the illustrated identification number (ID). The full and empty soda cans (ID 1 to 16) vary in colour, geometry and weight where some are only distinguishable by one identification feature (e.g. Colour for ID 1 to 3; Weight for ID 1 and 9). The remaining identification objects for the experiment (ID 17 to 21), were selected for their non-cylindrical shape and differ in geometry, colour, weight and material.

It is obvious that the full or empty soda cans especially, as well as the full or empty toolbox would not be distinguishable by means of purely optical recognition and can only be distinguished using multi-sensor information. Also, the optical differentiation of the tape dispenser assembly (ID 17) and body (ID 18) by 3D-Object recognition is associated with a high degree of uncertainty. This is due to the fact that matching the CAD point cloud of the body to a sensor point cloud of the assembly can also lead to recognition of the body, especially if the sensor point cloud is cluttered or occluded.

Figure 5.1: Identification objects for experimental verification

Table 5.3: Description of identification objects for experimental verification

ID	Description	Dimensions	Weight	Material	Colour
01	SchwipSchwapFull	Ø67 x115	360	metal	red
02	PepsiFull	Ø67 x115	360	metal	blue
03	LiptonFull	Ø67 x115	360	metal	yellow
04	RedBullWatermelonFull	Ø53 x135	270	metal	red
05	RedBullBlueberryFull	Ø53 x135	270	metal	blue
06	FantaFull	Ø58 x146	360	metal	orange
07	SpriteFull	Ø58 x146	360	metal	green
08	CocaColaFull	Ø58 x146	360	metal	red
09	SchwipSchwapEmpty	Ø67 x115	25	metal	red
10	PepsiEmpty	Ø67 x115	25	metal	blue
11	LiptonEmpty	Ø67 x115	25	metal	yellow
12	RedBullWatermelonEmpty	Ø53 x135	15	metal	red
13	RedBullBlueberryEmpty	Ø53 x135	15	metal	blue
14	FantaEmpty	Ø58 x146	25	metal	orange
15	SpriteEmpty	Ø58 x146	25	metal	green
16	CocaColaEmpty	Ø58 x146	25	metal	red
17	TapeDispenserASM	55x55x157	710	metal	blue
18	TapeDispenserBody	55x55x157	585	metal	blue
19	PressureControlValve	50x50x77	575	metal	grey
20	ToolboxFull	50x105155	535	non-metal	grey
21	ToolboxEmpty	50x105x155	160	non-metal	grey

5.2.2 Experimental results

During the experiment, the identification objects were identified three times each in random order, with a weight tolerance of ±10 % of the CAD weight to the sensor measured weight. All objects were scanned by means of 6 rotations of 60° each at an angle of about 45° to the turntable surface. The complete protocol of the experiment can be found in Appendix D.3.

Out of the 63 multi-sensor information identifications carried out, all were successful. The recording of the weight information by means of the weighing plate took place without any problems and with high accuracy. The detection of metallic and non-metallic surfaces of the identification objects by the inductive sensor succeeded. Both metallic and non-metallic surfaces of the identification objects were reliably detected by the inductive sensor. Furthermore, the recognition of the primary colours of the identification objects based on the RGB-Image data of the structured-light sensor worked successfully. The prefiltering carried out on the basis of this multi-sensor information thus resulted each time in the expected number of candidate objects for 3D-Object recognition. Based on this, the appropriate object identity information was found each time.

As a result of the investigation, it can be stated that the implemented prototype of the multi-sensor information Auto-ID system for the identification of unpackaged piece goods on the basis of their natural features works well under the described conditions. The perception of the identifying features of geometry, colour, weight and material by means of the selected sensors was reliable and allowed for the differentiation of objects which are not distinguishable optically. In addition, there was not a single case of false positive detection, which can be explained by the exclusion from the recognition knowledge base of those objects which are not compliant with the multi-sensor information. The results of the verification show that this pre-filtering has worked successfully. Generating the identification knowledge base from CAD-Models through the offline process took about 25 seconds for all 21 objects. On average, it took 3:08 minutes to identify one object (online process); the limiting factors here are the time needed for manual scanning and post-processing (approximately 80 % of the time span) and the computing power of the PC. By using a different structured-light sensor or several sensors arranged in perspective, the required scan time would be considerably reduced. The experiment was conducted using a Microsoft Surface Pro 4 with an Intel i5 processor (2,4 GHz) and 8GB RAM, thus it is possible that the identification time could be reduced by using a more powerful PC. There is also still potential for more efficient software implementation, which could also lead to a further improvement in identification time.

In the following chapter, conclusions are drawn from the overall research and the results of this verification.

6 Conclusions and recommendations

This chapter aims to summarise the findings of this research and thus to draw a conclusion. Furthermore, the theoretical and practical contributions through this thesis are discussed and recommendations are made for future research.

6.1 Summary of the research

The results of the research are summarised according to the sequence of the subordinate research questions (SRQ) raised in Section 1.2, which were formulated with the intention of supporting the overall research process. The following sections will answer the SRQs and provide further information which leads to the key findings.

6.1.1 Automatic identification systems in industry and research

As of today, automatic identification systems (AIS) are used in industry for a wide range of applications requiring the coupling of material flow and information flow. In more or less hierarchically organised automation structures following the automation pyramid, AIS can be assigned to the field level where they are mainly used for data collection (see Section 2.1.1). With the transition from the automation pyramid to decentralised CPSs triggered by digitalisation, AIS will continue to be used as data acquisition systems, but in this context AIS in particular represents the binding nodes between the real and the virtual world.

In literature there are divided views on the concept of identity or the process of identification in general (see Section 2.1.2). Research in the field of digitalisation, more precisely IoT, understands identity as a unique set of identity information that is assigned to a single thing or object. The relevant practical guidelines, which have been formulated for industrial applications, distinguish between different levels of identity and thus remain more general and applicable (see Section 2.1.3). An identity is always linked to the framework intended to determine the identity and is only known within this framework. The design of the framework and the associated definition of the identity are therefore variable.

For identification purposes, the AISs that are predominantly used today, are based on artificial identification features, while there are only few approaches regarding direct identification based on natural identification features (see Section 2.1.4, Section 2.1.6 and Section 2.1.7). Existing approaches for direct identification are more the subject of research than industrially proven concepts. Last but not least, this can be explained by the fact that the packaged identification objects occurring in industrial material flow often have few non-uniform identification features, resulting in the need for artificial differentiation (see Section 2.1.5).

6.1.2 Machine vision in industry and research

Machine vision (MV) is a very extensive subject area, with a wide range of possible industrial applications. With the implementation of vision-related tasks by machines, an attempt is made to completely or partly solve tasks previously carried out by humans. For this purpose, the more practically oriented MV makes use of the methods of scientifically structured computer vision (CV) and also includes the aspect of hardware (see Section 2.2).

The area of MV that fits the subject of this thesis is 3D-Object recognition (see Section 2.2.1). Object recognition can take place at different levels, which are category-level (classification) and instance-level (identification). The latter level is especially important for this thesis and this can be accomplished via 3D-Object recognition. Besides the type of input data and the nature of the recognition pipeline, the features and descriptors used for object representation are of utmost importance for successful instance-

level 3D-Object recognition.

Due to the higher information content of point clouds compared to two-dimensional image data, there are more direct and hence more significant characteristics available for the purpose of identification. Pipelines based on both local and global descriptors are available for 3D-Object recognition based on CAD-Models (see Section 2.2.3). From the many approaches discussed in the literature, a systematic literature review was conducted to identify a practice that corresponds to the state of the art (see Section 2.2.2). This approach uses a recognition pipeline based on global CVFH-Descriptors (see Section 2.2.4) for three-dimensional point clouds.

There are two fundamental principles upon which 3D-Sensing technologies are based for the acquisition of point clouds: Triangulation and Time-of-flight (see Section 2.2.5). The sensors in these two categories have different benefits and drawbacks and should be selected specifically for the application (see Section 2.2.5.1, Section 2.2.5.2 and Section 2.2.5.3).

6.1.3 Product data management in industry

Product Data Management (PDM) describes the storage and management of product-related data originating from product development throughout the life cycle of products. With regard to the thesis, PDM can be considered as an informative backbone for AIS.

There are different product models within PDM, which serve to virtually represent products in the form of associated information (see Section 2.3.1). The technical information defining the product is valuable for the purpose of identification. Organisational information as one of the categories of technical information includes the set of identity information that defines the identity of an object. Geometric information as another subcategory of technical information defines the physical appearance or properties of products or objects respectively. Using these two information sources, the identity (ID, name, etc.) and the natural object features (geometry, appearance, mass characteristics and material properties) can be obtained for a multi-sensor AIS.

The international standard for the definition and exchange of product model data, called Standard for the exchange of product model data (STEP), offers universality and software manufacturer independence and thus provides an excellent information basis.

STEP also offers a CAD-Model format that follows an open standard. In particular, hardly any information is available in the literature on manufacturer-specific CAD-Formats. However, there are three basic types of models: Wire-frame, surface and solid (see Section 2.3.2). What they all have in common is that CAD-Models mainly define the geometric properties. Material information, mass information and appearance information are not necessarily integrated into CAD-Models. Native CAD-Formats contain significantly more information than comparable exchange formats, but specific literature on the exact metadata of specific formats could not be found.

In general, CAD formats contain the following information: Descriptions, identification numbers, material properties, surface properties, mass properties and tolerances.

6.1.4 Design of a multi-sensor system for direct identification

With regard to the design of a multi-sensor AIS for direct identification of unpackaged goods, the information from the literature review was merged and interpreted into requirements (see Section 3.1). The design of such a multi-sensor AIS must allow the use of an object's natural identification features in the form of 3D-Geometry, weight, centre of mass location, hull material and predominant colour, for identification purposes. As a basis for identification, the design must allow CAD-Models from the PDM to be used for the generation of the identification knowledge base. Since AIS act as data acquisition systems in an information network, the design must also enable the collected identity information to be communicated to downstream systems.

In the course of this research these general requirements were converted into specific hardware and software requirements the system aimed for in order to generate a basis for conceptual design (see Section 3.1).

For the hardware requirements, subfunctions were defined, which served to fulfil the requirements. After the creative finding of solution principles for each subfunction, an optimal solution was selected by means of utility analysis (see Section 3.2.1.2 and Appendix A). The combination of all dominant solution principles led to the emergence of an overall concept (see Section 3.3). In terms of design, the proposed multi-sensor AIS consists of a structured-light sensor and a sensor platform (see Figure 3.9). The sensor platform also comprises a weighing plate, the material sensor and a rotary device.

Subfunctions were also defined for the software requirements, which can be fulfilled by a component-based design (see Section 3.2.2.2). The subdivision into individual components guarantees a universally applicable concept which is easy to extend. The specification for the design of the individual components is given in Figure 3.8.

6.1.5 Practical implementation of multi-sensor AIS

In Chapter 4 of this thesis one multi-sensor AIS according to the developed design was implemented in terms of hardware and software.

The hardware prototype system is based on the high accuracy HP 3D Structured-light Scanner Pro S3 in combination with the HP Automatic Turntable Pro (see Section 4.1.1). The sensor platform excluding the rotary device was designed and implemented specifically for this thesis. The weighing plate has been realised with four modified force sensors, which are attached to a steel plate by means of a 3D-printed two-piece mount (see Section 4.1.2.2 and Section 4.1.2.5).

An inductive proximity switch is used to record the material information and can be moved variably by means of a mount that adheres magnetically to the steel weighing plate (see Section 4.1.2.3 and Section 4.1.2.6).

The electronic circuit for reading the sensors and preprocessing their data is based on an Arduino Nano V3 microcontroller board (see Section 4.1.2.1). The electronic circuit has been specifically developed (Section 4.1.2.4). Using custom-soldered circuit boards, the half-bridges of the force sensors were supplemented to form a full Wheatstone bridge and each connected to an HX711 force sensor amplifier. All four force sensor amplifier combinations were connected to the microcontroller via two cable connections for serial communication. The inductive proximity switch was connected to a digital pin of the microcontroller using a pull-up circuit.

The software implementation started with the definition of the process structure for multi-sensor identification (see Section 4.2.1), that the software prototype follows. Python was chosen as the main programming language, but because the PCL is only available in C++, a special integration module for Python based on C++ had to be developed (see Section 4.2.2). According to the component structure defined in the conceptual design, software classes were then implemented, which serve to execute the aforementioned process structure by calling their methods (see Section 4.2.4). Finally, a microcontroller software using the C programming language was implemented, which handles the sensor data and communicates with the main software.

6.1.6 Effects of using multi-sensor information in addition to optical recognition for identification purposes

Experimental verification has shown that the use of multi-sensor information can use optically undetectable features such as an object material and weight characteristics for the purpose of differentiation (see Section 5.2). Compared to purely optically operating systems for object recognition, a significantly increased perceptive faculty is thus achieved. As a result, considerably more demanding identification tasks in terms of distinctiveness can be solved by means of a multi-sensor AIS.

When objects are scanned in reality, incomplete point clouds often arise due to occlusion. Incomplete or cluttered point clouds, leading to unreliable results, especially if two very similar objects that differ only in very small details are to be differentiated. This results in so-called false positives, which incorrectly report an object as having been recognised. One example is the identification of an individual part of an assembly instead of the assembly itself. At this point, the use of multi-sensor information offers significant potential for increasing recognition accuracy. In the example given, the weight of the assembly compared to that of the individual components would be a clear indication. Due to the prefiltering of the identification knowledgebase based on the multi-sensor information proposed in this thesis for the generation of the recognition knowledge base, the recognition accuracy is inevitably superior to purely optical recognition.

Another effect, which becomes more and more apparent as the number of objects to be identified increases, is the time reduction compared to purely optical recognition. This effect occurs because in the matching step of 3D-Object recognition, descriptors are compared with each other through computationally intensive processes. The prefiltering of the recognition knowledge base on the basis of the multi-sensor information thus also results in a reduced number of descriptors to be matched. In the case of purely optical recognition, this reduction of the descriptors to be matched either does not take place or takes place only on the basis of the colour information, which leads unavoidably to more descriptors and thus more time expenditure.

In summary, it can be argued that the use of multi-sensor information leads to significantly higher distinctiveness, improved identification accuracy and is less time-consuming compared with purely optical recognition.

6.2 Contribution of the research

This research mainly contributes to the existing scholarship in the field of automatic identification technology. In particular, theoretical and practical contributions are made to the hitherto little studied field of direct identification on the basis of natural object features as outlined in the following sections.

6.2.1 Theoretical contributions

The main theoretical contribution to the body of knowledge through this research is combining elements from the fields of automatic identification technology, machine vision and product data management to achieve direct identification of unpacked piece goods using multi-sensor information. The author is not aware of any existing approach that combines these three fields for this purpose. The theoretical contributions listed below are made in particular by this research:

1. The contextual understanding of the concept of identity and the process of identification in industrial material flow systems is enriched. For this purpose, different formulations and understandings of identity from the literature are compiled and compared. Furthermore, established numbering systems and existing Auto-ID technologies used for direct and indirect identification are summarised.
2. The currently existing methods for 3D-Object recognition based on CAD-Data from the field of machine vision are summarised and critically examined by means of a systematic literature review. This provides an overview of the current state of the art in terms of the algorithms and sensor technology used.
3. The information usable for the purpose of direct identification, which is provided by CAD-Data, is presented. This enhances the possible level of data integration, as much of the existing information is not used yet.
4. A generic process of identification is introduced, which covers the possibility of using several identification features for identification. This extends existing formulations from the literature and forms the theoretical foundation for the new approach of multi-sensor identification.
5. An overall concept for a multi-sensor identification system consisting of hardware and software components is developed based on the findings from the latest literature. This leads

to a novel approach for an Auto-ID technology, in the little-explored branch of direct identification based on natural object features.

6. The underlying process structure for identification by means of multi-sensor information is developed and described, which represents the theoretical basis for such an identification technology.
7. The functionality of the developed multi-sensor AIS based on the novel concept is proven and the effects of using multi-sensor information for identification are explored for the first time.

6.2.2 Practical contributions

The main practical contribution of this research is the identification of unpackaged piece goods using their CAD-Models without the need for any artificial identification features. The author is not aware of any existing approach that builds on the combination of multi-sensor information for direct identification. In particular, the following practical contributions are made through this research:

1. The identification of unpackaged piece goods at item number level is enabled, without the need for artificial object features. This avoids the additional efforts and issues with application usually involved in using artificial identifiers (see Section 1.2).
2. The defining data master from product development is used as a basis for identification and is available through PDM over the entire product life cycle, enhancing the overall data integration.

6.3 Key findings

This section summarises the key findings for each of the SRQs asked in Section 1.2.

SRQ 1:	What is the state of the art in automatic identification technology, machine vision and product data management in industry and can a system using multi-sensor information for direct identification be built on it?

In contrast to the broad field of indirect automatic identification, there are only a few technologies for direct identification that are still emerging from research. Nevertheless, existing concepts such as the concept of identity and the process of identification can be used as a basis for building a multi-sensor identification system. From the huge field of machine vision, numerous methods for 3D-Object recognition are available, which must be selected specifically for an application. Recognition pipelines based on three-dimensional point cloud descriptors are particularly suitable for the realisation of a multi-sensor AIS. All data or information required for 3D-Object recognition and direct identification can be retrieved from the product models provided by product data management. By combining and expanding existing approaches for each of the fields, a multi-sensor system for direct identification can be achieved.

SRQ 2:	How must an automated identification system be designed to allow direct identification based on multi-sensor information?

The overall design of a system for direct identification based on multi-sensor information must consist of a hardware design and a software design. The design of the hardware must enable the acquisition of an object's natural identification features (3D-Geometry, appearance, mass properties, material properties) using suitable sensors. The design of the software must be able to convert the sensor data into information, which can then be matched against the technical product information (CAD-Models) defining known objects.

SRQ 3:	What does a practical implementation of an automated multi-sensor system for direct identification look like?

One first practical implementation of the hardware consists of a structured-light 3D-Scanner and a sensor platform. The 3D-Scanner acquires the 3D-Geometry and appearance of an object placed on the sensor platform. The sensor platform allows the collection of sensor data regarding the mass properties and material properties of an object and also enables its rotation for complete scanning. The associated software implementation is based on a special process structure for multi-sensor identification, which uses various software components or modules mapping the actual identification algorithm.

SRQ 4:	What are the effects of using multi-sensor information in addition to optical 3D-Object recognition for identification purposes?

The additional use of multi-sensor information improves the accuracy and distinctiveness of identification compared to purely optical 3D-Object recognition. The basis for these effects is the prefiltering of known objects using the multi-sensor information regarding their natural features, which leads to a reduced number of possible candidates for 3D-Object recognition. Subsequently, prefiltering by means of multi-sensor information also reduces the required computing effort and thus the required computing time for instance-level 3D-Object recognition.

6.4 Conclusions

The primary research question (PRQ) (see Section 1.2) asks how an automated system for direct identification of industrial components using multi-sensor information can be accomplished. The answer to this question developed through the overall research carried out and through the answers given to the SRQs in Section 6.1. An automated system for the direct identification of objects occurring in industrial material flow can be realised by acquiring multi-sensor information on their natural identification features, including 3D-Geometry, appearance, mass properties and material properties. The characteristics of such a multi-sensor AIS for direct identification make it particularly suitable for identification of unpackaged piece goods occurring in industrial material flow, as they provide more unique identification features compared with uniformly packed goods.

Regarding the research problem statement (RPS) (see Section 1.2) such a multi-sensor AIS allows the identification of objects without any additional effort. This is due to the fact that no artificial identification features are used. Process steps for attaching or removing, as well as the maintenance of artificial identification features are thus completely eliminated. Furthermore, as the object is identified on the basis of its natural features, artificial identifiers cannot get lost or destroyed and do not have to be integrated on objects. The same applies to issues concerning the applicability of artificial identifiers. Object geometries do not have to be adapted for the purpose of identification, but in combination with other physical properties become unmistakable identifying features, not requiring an emergency strategy to compensate for damaged artificial identifiers. Difficulties in the application of indirect identification caused by production steps are also solved by a multi-sensor AIS, since additional features added to an object can be used for the purpose of identification.

The primary research objective (PRO) of this thesis was to develop an automatic system capable of directly identifying objects by matching information from CAD-Data with multi-sensor information. This objective was attained through the successful implementation of the prototype system (see Chapter 4). The ability of the system to directly identify unmarked objects was proven by theoretical and experimental verification (see Chapter 5).

6.5 Limitations and recommendations for further research and development

In this section the limitations of the research carried out are outlined. These limitations then serve as a basis for the recommendation of subsequent research.

The research carried out within this thesis follows the DSR methodology as outlined in Section 1.4. In particular during the phases “Suggestion of possible solution” and “Development” creative thinking processes take place which strongly depend on the knowledge and skills of the researcher. In order to compensate for this bias, several cycles of the DSR methodology can be performed, enhancing the experience of the researcher. Due to the limited time frame of this research, only one cycle of the DSR methodology was carried out. The solution presented is therefore a first approach and can be further improved.

With regard to the developed multi-sensor AIS prototype, the following recommendations for further research and development are made on the basis of the knowledge gained during the research:

1. Implementation of the centre of mass validation, which was not pursued further as outlined in Section 4.2.6. The centre of mass validation would be an additional step in the process structure of multi-sensor identification and could further improve the accuracy of the identification. Further research is needed to investigate the exact effects of this additional validation step.
2. The rotation of an identification object through the rotary device involves forces or moments, which counteract the inertia of the object. By measuring the applied moment and resulting acceleration, the moment of inertia of the identification object can be determined on the basis of 6-DoF pose estimation. The comparison of this mass information with the CAD-Models properties offers a further identification feature. Additional research and development is needed to investigate the feasibility and effects of this approach.
3. The dimensions of a bounding box, which contains the identification object within the 3D-Sensor data, can be used to prefilter the identification knowledge base in addition to the already-used multi-sensor information. Such a bounding box can be calculated with little computational effort for CAD point cloud and sensor point cloud and may be a valuable feature for the prefiltering step. The effects of this approach are to be investigated through subsequent research and development.
4. The developed prototype enables the partially unique identification of unpackaged piece goods on an article number level. As there are existing approaches for unique direct identification based on grinding imprints or natural textures (see Section 2.1.7), these might be combined with the approach of this research in order to create an enhanced multi-sensor AIS for identification on serial number level. In fact, the existing approaches for unique direct identification require an image of the small surface patch representing the 'fingerprint'. Without knowing the location of this surface patch on the object, it is therefore not possible to gather the 'fingerprint' for identification. However, CAD-Models could be used for the exact definition of these surface patches on an identification objects surfaces. The multi-sensor AIS developed within this thesis could thus be used to retrieve this location information from the CAD-Model in order to extract the image of the 'fingerprint' from the image data of the 3D-Sensor. Additional research and development is needed to investigate the feasibility and effects of this combination.
5. The ability of the developed multi-sensor AIS to completely scan the geometry of an identification object is merely sufficient (see Section 5.1). The use of additional 3D-Scanners or the manipulation of the identification object by a robot in front of one or more 3D-Sensors could drastically improve this. In particular, the integration of the different sensors used within this research into the gripping system of a robot might offer potential. Further research and development activities are necessary to explore these approaches.

Another limitation of this research is the focus on the technical development of a multi-sensor AIS for the direct identification of unpackaged goods. The actual applications or use of such a system in the industrial environment has not yet been examined. In order to address this limitation further research is needed to investigate the applications and use of such a system within the industry.

References

[1] Ahola, Jari M.; Heikkila, Tapio; Koskinen, Jukka; Seppala, Tuomas; Tamminen, Tarmo (2016): A configurable CAD-based object recognition system for varying resolution 3D-sensors. In : 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. Piscataway (NJ), USA: IEEE, pp. 1–6.

[2] Ailisto, Heikki; Heikkinen, Veli; Mitikka, R.; Myllylä, Risto; Kostamovaara, J.; Mäntyniemi, A. et al. (2001): 3-D imaging with scannerless LIDAR. In : ODIMAP III - 3rd Topical Meeting on Optoelectronic Distance Measurement and Applications. Pavia, Italy.

[3] Aldoma, Aitor; Marton, Zoltan-Csaba; Tombari, Federico; Wohlkinger, Walter; Potthast, Christian; Zeisl, Bernhard et al. (2012): Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation. In *IEEE Robot. Automat. Mag.* (3), pp. 80–91. DOI: 10.1109/MRA.2012.2206675.

[4] Aldoma, Aitor; Vincze, Markus; Blodow, Nico; Gossow, David; Gedikli, Suat; Rusu, Radu Bogdan; Bradski, Gary (2011): CAD-model recognition and 6DOF pose estimation using 3D cues. In : IEEE International Conference on Computer Vision workshops (ICCV workshops), 2011. Barcelona, Spain. Piscataway (NJ), USA: IEEE, pp. 585–592.

[5] Alhamzi, Khaled; Elmogy, Mohammed; Barakat, Sherif (2014): 3D Object Recognition Based on Image Features: A Survey. In *International Journal of Computer and Information Technology (IJCIT)* (3), pp. 651–660.

[6] Andreopoulos, Alexander; Tsotsos, John K. (2013): 50 Years of object recognition: Directions forward. In *Computer Vision and Image Understanding* (117), pp. 827–891. DOI: 10.1016/j.cviu.2013.04.005.

[7] Arman, Farshid; Aggarwal, J. K. (1993): Model-based object recognition in dense-range images—a review. In *ACM Computing Surveys (CSUR)* (1), pp. 5–43. DOI: 10.1145/151254.151255.

[8] Arnold, Dieter; Furmans, Kai (2019): Materialfluss in Logistiksystemen. 7th ed. Berlin, Germany: Springer.

[9] Arnold, Dieter; Isermann, Heinz; Kuhn, Axel; Tempelmeier, Horst; Furmans, Kai (Eds.) (2008): Handbuch Logistik. 3rd ed. Berlin, Germany: Springer.

[10] Avia Semiconductor: HX711. 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales. Available online at https://www.mouser.com/datasheet/2/813/hx711_english-1022875.pdf, checked on 11/20/2020.

[11] Batchelor, Bruce G. (2012a): Lighting-Viewing Methods. In Bruce G. Batchelor (Ed.): Machine Vision Handbook. London, UK: Springer, pp. 1345–1560.

[12] Batchelor, Bruce G. (2012b): Machine Vision for Industrial Applications. In Bruce G. Batchelor (Ed.): Machine Vision Handbook. London, UK: Springer, pp. 1–59.

[13] Bauernhansl, Thomas (2014): Die Vierte Industrielle Revolution. Der Weg in ein wertschaffendes Produktionsparadigma. In Thomas Bauernhansl, Michael Ten Hompel, Birgit Vogel-Heuser (Eds.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Anwendung, Technologien, Migration. Wiesbaden, Germany: Springer, pp. 5–35.

[14] Bay, Herbert; Ess, Andreas; Tuytelaars, Tinne; van Gool, Luc (2008): Speeded-Up Robust Features (SURF). In *Computer Vision and Image Understanding* 110 (3), pp. 346–359.

[15] Besl, Paul J. (1988): Active, optical range imaging sensors. In *Machine Vision and Applications* 1 (2), pp. 127–152. DOI: 10.1007/BF01212277.

[16] Beyerer, Jürgen; Puente León, Fernando; Frese, Christian (2016): Machine vision. Automated Visual Inspection: Theory, Practice and Applications. 1st ed. Berlin, Germany: Springer.

[17] Blais, Francois (2004): Review of 20 years of range sensor development. In *Journal of Electronic Imaging* 13 (1), p. 231. DOI: 10.1117/1.1631921.

[18] Bradski, Gary; Kaehler, Adrian (2011): Learning OpenCV. Computer vision with the OpenCV library. 1st ed. Sebastopol (CA), USA: O'Reilly Media.

[19] Breidenstein, Bernd; Mörke, Tobias; Denkena Berend; Hockauf, Rolf (2016): Markierungsfreie Bauteil-Identifikation. Potentiale von Oberflächenmerkmalen zur Bauteil-Identifikation in Fertigung und Nutzungsphase. In *wt Werkstattstechnik online* 106 (6), pp. 412–415. Available online at https://www.werkstattstechnik.de/wt/get_article.php?data[article_id]=85934, checked on 10/23/2019.

[20] Buchanan, James D. R.; Cowburn, Russell; Jausovec, Ana-Vanessa; Petit, Dorothée; Seem, Peter; Xiong, Gang et al. (2005): 'Fingerprinting' documents and packaging. Unique surface imperfections serve as an easily identifiable feature in the fight against fraud. In *Nature* 436 (7050), p. 475. DOI: 10.1038/436475a.

[21] Byne, J.H.M.; Anderson, J.A.D.W. (1998): A CAD-based computer vision system. In *Image and Vision Computing* 16 (8), pp. 533–539. DOI: 10.1016/S0262-8856(98)00100-0.

[22] Cass, Stephen (2019): The Top Programming Languages 2019. Python remains the big kahuna, but specialist languages hold their own. Available online at https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019, updated on 9/6/2019, checked on 10/9/2020.

[23] Cognex (2016): Introduction to machine vision. A guide to automating process & quality improvement. Available online at https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf, checked on 9/19/2020.

[24] Corsini, Massimo; Cignoni, Paolo; Scopigno, Roberto (2012): Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. In *IEEE Transactions on Visualization and Computer Graphics* 18 (6), pp. 914–924. DOI: 10.1109/TVCG.2012.34.

[25] Cowburn, Russell (2008): Laser surface authentication – reading Nature's own security code. In *Contemporary Physics* 49 (5), pp. 331–342. DOI: 10.1080/00107510802583948.

[26] Creswell, John W.; Creswell, J. David (2018): Research design. Qualitative, Quantitative, and Mixed Methods Approaches. 5th ed. Los Angeles (CA),USA: SAGE Publications.

[27] Dangelmaier, Wilhelm (2001): Fertigungsplanung. Planung von Aufbau und Ablauf der Fertigung Grundlagen, Algorithmen und Beispiele. 2. Auflage. Berlin, Germany: Springer.

[28] Deckert, Ronald (2019): Digitalisierung und Industrie 4.0. Technologischer Wandel und individuelle Weiterentwicklung. Wiesbaden, Germany: Springer Gabler.

[29] VDI 2225-3:1998-11, 1998: Design engineering methodics.

[30] DIN 6789:2013-10, 2013: Dokumentationssystematik - Verfälschungssicherheit und Qualitätskriterien für die Freigabe digitaler Produktdaten.

[31] Dong, Tianzhen; Qi, Xiao; Zhang, Qing; Li, Wenju; Xiong, Liang (2019): Overview on Vision-Based 3D Object Recognition Methods. In Yao Zhao, Nick Barnes, Baoquan Chen, Rüdiger Westermann, Xiangwei Kong, Chunyu Lin (Eds.): Image and Graphics. Cham, 2019. Cham, Switzerland: Springer, pp. 243–254.

[32] Dragon, Ralf; Mörke, Tobias; Rosenhahn, Bodo; Ostermann, Jörn (2011): Fingerprints for Machines - Characterization and Optical Identification of Grinding Imprints. In Rudolf Mester, Michael Felsberg (Eds.): Pattern Recognition. 33rd DAGM Symposium Frankfurt/Main, Germany, August 31 – September 2, 2011 Proceedings. Berlin, Germany: Springer, pp. 276–285.

[33] Dresch, Aline; Pacheco Lacerda, Daniel; Valle Antunes Jr, José Antônio (2015): Design Science Research. A Method for Science and Technology Advancement. Cham, Switzerland: Springer.

[34] Eigner, Martin (2014a): Product Lifecycle Management (PLM). In Martin Eigner, Daniil Roubanov, Radoslav Zafirov (Eds.): Modellbasierte virtuelle Produktentwicklung. Berlin, Germany: Springer, pp. 267–300.

[35] Eigner, Martin (2014b): Technische Organisation des Produktentwicklungsprozesses. In Martin Eigner, Daniil Roubanov, Radoslav Zafirov (Eds.): Modellbasierte virtuelle Produktentwicklung. Berlin, Germany: Springer, pp. 227–266.

[36] Eigner, Martin (2014c): Überblick Disziplin-spezifische und -übergreifende Vorgehensmodelle. In Martin Eigner, Daniil Roubanov, Radoslav Zafirov (Eds.): Modellbasierte virtuelle Produktentwicklung. Berlin, Germany: Springer, pp. 15–52.

[37] Fang, Zheng; Zhao, Shibo; Wen, Shiguang; Zhang, Yu (2018): A Real-Time 3D Perception and Reconstruction System Based on a 2D Laser Scanner. In *Journal of Sensors*, pp. 1–14. DOI: 10.1155/2018/2937694.

[38] Finkenzeller, Klaus (2015): RFID-Handbuch. Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC. 7th ed. München, Germany: Hanser.

[39] Fischer, Peter; Hofer, Peter (2011): Lexikon der Informatik. 15th ed. Heidelberg, Germany: Springer.

[40] Forstner, Lisa; Dümmler, Mathias (2014): Integrierte Wertschöpfungsnetzwerke. Chancen und Potenziale durch Industrie 4.0. In *Elektrotechnik & Informationstechnik* 131 (7), pp. 199–201. DOI: 10.1007/s00502-014-0224-y.

[41] Frauenhofer IPM (2017): Track & Trace per Fingerabdruck. Edited by Frauenhofer-Institut für pyhsikalische Messtechnik. Available online at https://www.ipm.fraunhofer.de/content/dam/ipm/de/PDFs/produktblaetter/PK/IVS/Track-trace-FINGERPRINT-de.pdf, checked on 10/23/2019.

[42] Han, Pengfei; Zhao, Gang (2015): CAD-based 3D objects recognition in monocular images for mobile augmented reality. In *Computers & Graphics* 50, pp. 36–46. DOI: 10.1016/j.cag.2015.05.021.

[43] Han, Xian-Feng; Jin, Jesse S.; Xie, Juan; Wang, Ming-Jie; Jiang, Wei (2018): A comprehensive review of 3D point cloud descriptors. Available online at https://www.researchgate.net/publication/323003836_A_comprehensive_review_of_3D_point_cloud_descriptors, checked on 9/3/2020.

[44] Hansard, Miles; Lee, Seungkyu; Choi, Ouk; Horaud, Radu (2013): Time-of-flight cameras. Principles, methods and applications. London, UK: Springer.

[45] Hänsch, R.; Weber, T.; Hellwich, O. (2014): Comparison of 3D interest point detectors and descriptors for point cloud fusion. In *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (3), pp. 57–64. DOI: 10.5194/isprsannals-II-3-57-2014.

[46] Hashimoto, Manabu; Akizuki, Shuichi; Takei, Shoichi (2017): A Survey and Technology Trends of 3D Features for Object Recognition. In *Electronics and Communications in Japan* 100 (11), pp. 31–42. DOI: 10.1002/ecj.11992.

[47] He, Yu; Chen, Shengyong (2019): Recent Advances in 3D Data Acquisition and Processing by Time-of-Flight Camera. In *IEEE Access* 7, pp. 12495–12510. DOI: 10.1109/ACCESS.2019.2891693.

[48] Hehenberger, Peter (2020): Computerunterstützte Produktion. Eine kompakte Einführung. 2nd ed. Berlin, Germany: Springer.

[49] Helmus, Manfred; Meins-Becker, Anica; Laußrat, Lars; Kelm, Agnes (Eds.) (2009): RFID in der Baulogistik. Forschungsbericht zum Projekt "Integriertes Wertschöpfungsmodell mit RFID in der Bau- und Immobilienwirtschaft". 1st ed. Wiesbaden, Germany: Vieweg+Teubner.

[50] Hewlett-Packard (2017a): HP 3D Automatic Turntable Pro. Quick Setup Guide. Available online at http://h10032.www1.hp.com/ctg/Manual/c05323258.

[51] Hewlett-Packard (2017b): HP 3D Scan. HP 3D Structured Light Scanner Pro S3. Available online at https://www8.hp.com/h20195/v2/GetDocument.aspx?docname=4AA6-9360ENW, checked on 10/5/2020.

[52] Hippenmeyer, Heinrich; Moosmann, Thomas (2016): Automatische Identifikation für Industrie 4.0. Berlin, Germany: Springer.

[53] Horn, Eli; Kiryati, Nahum (1999): Toward optimal structured light patterns. In *Image and Vision Computing* 17 (2), pp. 87–97. DOI: 10.1016/S0262-8856(98)00113-9.

[54] ISO 10303-242:2020-04, 2020: Industrial automation systems and integration - Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering.

[55] ISO/IEC 19505-2:2012, 2012: Information technology - Object Management Group Unified Modeling Language (OMG UML) - Part 2: Superstructure.

[56] Jain, Ramesh; Kasturi, Rangachar; Schnuck, Brian G. (1995): Machine Vision. New York City (NY),USA: McGraw-Hill.

[57] Jang, Wonkwi; Je, Changsoo; Seo, Yongduek; Lee, Sang Wook (2013): Structured-light stereo. Comparative analysis and integration of structured-light and active stereo for measuring dynamic shape. In *Optics and Lasers in Engineering* 51 (11), pp. 1255–1264. DOI: 10.1016/j.optlaseng.2013.05.001.

[58] Japs, Simon (2007): Marktübersicht und politisches Umfeld. In Hans-Jörg Bullinger, Michael Ten Hompel (Eds.): Internet der Dinge. Berlin, Germany: Springer, pp. 1–17.

[59] Je, Changsoo; Lee, Sang Wook; Park, Rae-Hong (2004): High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging. In Tomás Pajdla, Jiří Matas (Eds.): Computer Vision - ECCV 2004. Berlin, Germany: Springer, pp. 95–107.

[60] Jünemann, Reinhardt; Beyer, Andreas (1998): Steuerung von Materialfluß- und Logistiksystemen. Informations- und Steuerungssysteme, Automatisierungstechnik. Berlin, Heidelberg: Springer (Logistik in Industrie, Handel und Dienstleistungen). Available online at http://dx.doi.org/10.1007/978-3-642-72225-7.

[61] Kärkkäinen, Mikko; Ala-Risku, Timo (2003): Automatic identification. Applications and technologies. Available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.575.7551&rep=rep1&type=pdf, checked on 10/21/2019.

[62] Kirkwood, Robert; Sherwood, James A. (2018): Sustained CAD/CAE integration: integrating with successive versions of step or IGES files. In *Engineering with Computers* 34 (1), pp. 1–13. DOI: 10.1007/s00366-017-0516-z.

[63] Kitchenham, Barbara (2007): Guidelines for performing Systematic Literature Reviews in Software Engineering. Available online at http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9B3FE0164BBCC85C944B22BDB7AE2BAA?doi=10.1.1.117.471&rep=rep1&type=pdf, checked on 8/10/2019.

[64] Koch, Oliver; Deiters, Wolfgang (2007): RFID im Gesundheitswesen. Nutzenpotentiale und Stolpersteine auf dem Weg zu einer erfolgreichen Anwendung. In Hans-Jörg Bullinger, Michael Ten Hompel (Eds.): Internet der Dinge. Berlin, Germany: Springer, pp. 191–201.

[65] Krämer, Klaus (2002): Automatisierung in Materialfluss und Logistik. Ebenen, Informationslogistik, Identifikationssysteme, intelligente Geräte. Wiesbaden, Germany: Deutscher Universitätsverlag.

[66] Kropik, Markus (2009): Produktionsleitsysteme in der Automobilfertigung. Heidelberg, Germany: Springer.

[67] Kurbel, Karl (2016): Enterprise Resource Planning und Supply Chain Management in der Industrie. Von MRP bis Industrie 4.0. 8th ed. Berlin, Germany: De Gruyter.

[68] La Rocca, Gianfranco (2012): Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. In *Advanced Engineering Informatics* 26 (2), pp. 159–179. DOI: 10.1016/j.aei.2012.02.002.

[69] Lampe, Matthias; Flörkemeier, Christian (2005): Einführung in die RFID-Technologie. In Elgar Fleisch, Friedemann Mattern (Eds.): Das Internet der Dinge. Ubiquitous Computing und RFID in der Praxis. Berlin, Germany: Springer Verlag, pp. 69–86.

[70] Lazaros, Nalpantidis; Sirakoulis, Georgios C.; Gasteratos, Antonios (2008): Review of Stereo Vision Algorithms: From Software to Hardware. In *International Journal of Optomechatronics* 2 (4), pp. 435–462. DOI: 10.1080/15599610802438680.

[71] Leavy, Patricia (2017): Research design. Quantitative, Qualitative, Mixed Methods, Arts-Based, and Community-Based Participatory Research Approaches. New York City (NY),USA: The Guilford Press.

[72] Lee, Jay; Bagheri, Behrad; Kao, Hung-An (2015): A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. In *Manufacturing Letters* 3, pp. 18–23. DOI: 10.1016/j.mfglet.2014.12.001.

[73] Lee, Sukhan; Kim, Jaewoong; Lee, Moonju; Yoo, Kyeongdae; Barajas, Leandro G.; Menassa, Roland: 3D visual perception system for bin picking in automotive sub-assembly automation. In : IEEE International Conference on Automation Science and Engineering (CASE). Piscataway (NJ), USA: IEEE, pp. 706–713.

[74] Liu, Hongsen; Cong, Yang; Yang, Chenguang; Tang, Yandong (2019): Efficient 3D object recognition via geometric information preservation. In *Pattern Recognition* 92, pp. 135–145. DOI: 10.1016/j.patcog.2019.03.025.

[75] Lolling, Andreas (2003): Identifikationssysteme in der Logistik. Übersicht und praxisorientierte Auswahl. München, Germany: Huss Verlag.

[76] Lowe, David G. (2004): Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision* 60 (2), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

[77] Luo, Ren C.; Kuo, Chia-Wen (2015): A scalable modular architecture of 3D object acquisition for manufacturing automation. In : IEEE 13th International Conference on Industrial Informatics (INDIN). Piscataway (NJ), USA: IEEE, pp. 269–274.

[78] Martin, Heinrich (2014): Transport- und Lagerlogistik. Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik. 9th ed. Wiesbaden, Germany: Springer.

[79] Mateo, Carlos M.; Gil, Pablo; Torres, Fernando (2014): A Performance Evaluation of Surface Normals-based Descriptors for Recognition of Objects Using CAD-Models. In : 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO). Piscataway (NJ), USA: IEEE, pp. 428–435.

[80] Meudt, Tobias; Pohl, Malte; Metternich, Joachim (2017): Die Automatisierungspyramide - Ein Literaturüberblick. Technische Universität Darmstadt. Darmstadt. Available online at https://tuprints.ulb.tu-darmstadt.de/6298/1/2017%20-%20Die%20Automatisierungspyramide%20-%20Ein%20Literatur%C3%BCberblick-2.pdf, checked on 4/23/2020.

[81] Milenkovic, Milan (2020): Internet of Things: Concepts and System Design. Cham, Switzerland: Springer. Available online at https://doi.org/10.1007/978-3-030-41346-0.

[82] Müller, Jürgen (2018): Auto-ID-Verfahren im Kontext allgegenwärtiger Datenverarbeitung. Datenschutzrechtliche Betrachtung des Einsatzes von RFID-Systemen. Wiesbaden, Germany: Springer.

[83] Mustafa, Faiz F.; Al-Ashaab, Ahmed; Al-Amili, Hussein A. (2017): A Comparative Study of Product Data Exchange among CAD Systems. In : 2017 IEEE First International Conference on Recent Trends of Engineering Science and Sustainability. Piscataway (NJ), USA: IEEE, pp. 1–6.

[84] DIN 6763:1985-12, 1985: Nummerung - Grundbegriffe.

[85] Pahl, Gerhard; Beitz, Wolfgang; Feldhusen, Jörg; Grote, Karl-Heinrich (Eds.) (2007): Engineering Design. A Systematic Approach. London, UK: Springer.

[86] Pan, Wang; Zhu, Feng; Hao, Yingming; Zhang, Limin (2017): 6D Pose Estimation Based on Multiple Appearance Features from Single Color Image. In : IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). Piscataway (NJ), USA: IEEE, pp. 406–411.

[87] Pfohl, Hans-Christian (2018): Logistiksysteme. Betriebswirtschaftliche Grundlagen. 9th ed. Berlin, Germany: Springer.

[88] Poggio, Tomaso; Ullman, Shimon (2014): Machine Recognition of Objects. In Katsushi Ikeuchi (Ed.): Computer vision. A reference guide. New York City (NY), USA: Springer.

[89] Pretto, Alberto; Tonello, Stefano; Menegatti, Emanuele (2013): Flexible 3D localization of planar objects for industrial bin-picking with monocamera vision system. In : IEEE International Conference on Automation Science and Engineering (CASE). Piscataway (NJ), USA: IEEE, pp. 168–175.

[90] Ravari, Alireza Norouzzadeh; Taghirad, Hamid D. (2014): Transformation invariant 3D object recognition based on information complexity. In : Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM). Tehran, Iran. Piscataway (NJ), USA: IEEE, pp. 902–907.

[91] Rosche, Phil; Delaunay, Jean-Yves; Ungerer, Max (2009): Whitepaper Development of a Convergent Modular STEP Application Protocol Based on AP 203 and AP 214. STEP AP 242 - Managed Model Based 3D Engineering. Available online at http://www.ap242.org/c/document_library/get_file?uuid=157a9a8f-6888-4589-906d-71d09e75878c&groupId=52520, checked on 9/5/2020.

[92] Roubanov, Daniil (2014): Mechanikkonstruktion (M-CAD). In Martin Eigner, Daniil Roubanov, Radoslav Zafirov (Eds.): Modellbasierte virtuelle Produktentwicklung. Berlin, Germany: Springer, pp. 115–136.

[93] Royce, Winston W. (1970): Managing the Development of Large Software Systems. In : Proceedings IEEE Wescon. Piscataway (NJ), USA: IEEE, pp. 1–9.

[94] Rusu, Radu B.; Bradski, Gary; Thibaux, Romain; Hsu, John (2010): Fast 3D recognition and pose using the Viewpoint Feature Histogram. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010). Piscataway (NJ), USA: IEEE, pp. 2155–2162.

[95] Santucci, Gérald (2010): The Internet of Things. Between the Revolution of the Internet and the Metamorphosis of Objects. Available online at https://theinternetofthings.report/whitepapers/the-internet-of-things-between-the-revolution-of-the-internet-and-the-metamorphosis-of-objects/1007, checked on 6/10/2020.

[96] Scharstein, Daniel; Szeliski, Richard: High-accuracy stereo depth maps using structured light. In : Proceedings. CVPR 2003: Computer Vision and Pattern Recognition Conference. Piscataway (NJ), USA: IEEE, 195-202.

[97] Schöning, Harald; Dorchain, Marc (2014): Data Mining und Analyse. In Thomas Bauernhansl, Michael Ten Hompel, Birgit Vogel-Heuser (Eds.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Anwendung, Technologien, Migration. Wiesbaden, Germany: Springer, pp. 543–554.

[98] Siepmann, David; Graef, Norbert (2016): Industrie 4.0. Grundlagen und Gesamtzusammenhang. In Armin Roth (Ed.): Einführung und Umsetzung von Industrie 4.0. Grundlagen, Vorgehensmodell und Use Cases aus der Praxis. Berlin, Germany: Springer, pp. 17–82.

[99] Snyder, Hannah (2019): Literature review as a research methodology: An overview and guidelines. In *Journal of Business Research* 104, pp. 333–339. DOI: 10.1016/j.jbusres.2019.07.039.

[100] Song, Kai-Tai; Wu, Cheng-Hei; Jiang, Sin-Yi (2017): CAD-based Pose Estimation Design for Random Bin Picking using a RGB-D Camera. In *Journal of Intelligent & Robotic Systems* (87), pp. 455–470. DOI: 10.1007/s10846-017-0501-1.

[101] Šonka, Milan; Hlaváč, Václav; Boyle, Roger (2015): Image processing, analysis, and machine vision. 4th ed. Stamford (CT), USA: Cengage Learning.

[102] SparkFun Electronics: Datasheet SEN-10245. Available online at https://www.sparkfun.com/datasheets/Sensors/loadsensor.pdf, checked on 10/4/2020.

[103] Suk, Minsoo; Bhandarkar, Suchendra M. (1992): Three-Dimensional Object Recognition from Range Images. Tokyo, Japan: Springer.

[104] Ten Hompel, Michael; Büchter, Hubert; Franzke, Ulrich (2008): Identifikationssysteme und Automatisierung. Berlin, Germany: Springer.

[105] Ten Hompel, Michael; Heidenblut, Volker (2011): Taschenlexikon Logistik. Abkürzungen, Definitionen und Erläuterungen der wichtigsten Begriffe aus Materialfluss und Logistik. 3rd ed. Berlin, Germany: Springer.

[106] Ten Hompel, Michael; Schmidt, Thorsten (2007): Warehouse Management. Automation and Organisation of Warehouse and Order Picking Systems. Berlin, Germany: Springer.

[107] Ten Hompel, Michael; Schmidt, Thorsten; Dregger, Johannes (2018): Materialflusssysteme. Förder- und Lagertechnik. 42018th ed. Berlin, Germany: Springer.

[108] Tombari, Federico; Salti, Samuele; Di Stefano, Luigi (2010): Unique Signatures of Histograms for Local Surface Description. In : European Conference on Computer Vision (ECCV). Berlin, Germany: Springer, pp. 356–369.

[109] DIN 30781:1989-05, 1989: Transportkette - Grundbegriffe.

[110] Trappey, Amy J. C.; Trappey, Charles V.; Govindarajan, Usharani Hareesh; Sun, John J.; Chuang, Allen C. (2016): A Review of Technology Standards and Patent Portfolios for Enabling Cyber-Physical Systems in Advanced Manufacturing. In *IEEE Access* 4, pp. 7356–7382. DOI: 10.1109/ACCESS.2016.2619360.

[111] Treiber, Marco (2013): An introduction to object recognition. Selected algorithms for a wide variety of applications. London, UK: Springer.

[112] Tsarouchi, Panagiota; Matthaiakis, Stereos-Alexandros; Michalos, George; Makris, Sotiris; Chryssolouris, George (2016): A method for detection of randomly placed objects for robotic handling. In *CIRP Journal of Manufacturing Science and Technology* (14), pp. 20–27. DOI: 10.1016/j.cirpj.2016.04.005.

[113] Uckelmann, Dieter; Harrison, Mark; Michahelles, Florian (2011): An Architectural Approach Towards the Future Internet of Things. In Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.): Architecting the Internet of Things. Heidelberg, Germany: Springer, pp. 1–24.

[114] Ulrich, Markus; Wiedemann, Christian; Steger, Carsten (2012): Combining scale-space and similarity-based aspect graphs for fast 3D object recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (10), pp. 1902–1914. DOI: 10.1109/TPAMI.2011.266.

[115] Um, Dugan (2018): Solid modeling and applications. Rapid prototyping, CAD and CAE theory. 2nd ed. Cham, Switzerland: Springer.

[116] Vaishnavi, Vijay; Kuechler, William (2008): Design Science Research Methods and Patterns. Innovating Information and Communication Technology. New York City (NY), USA: Auerbach Publications.

[117] VDI/VDE-Gesellschaft (2013): Cyber-Physical Systems. Chancen und Nutzen aus Sicht der Automation. Available online at https://www.vdi.de/ueber-uns/presse/publikationen/details/cyber-physical-systems-chancen-und-nutzen-aus-sicht-der-automation, checked on 4/23/2020.

[118] DIN 55405:2014-12, 2014: Verpackung - Terminologie - Begriffe.

[119] VFC Research (2018): Share of data capture market by technology worldwide in 2015 and 2016. Includes hand-held scanners, industrial stationary scanners, and point-of-sale check out. Available online at https://www.statista.com/statistics/712201/worldwide-data-capture-technology-market-by-technology-type/, checked on 10/23/2019.

[120] Wannenwetsch, Helmut (2014): Integrierte Materialwirtschaft, Logistik und Beschaffung. 5th ed. Berlin, Germany: Springer.

[121] Weißflog, Luise; Wilsky, Philipp; Markert, Tobias; Riedel, Ralph (2019): Technologieauswahl zur automatischen Identifikation. Ein systematischer Ansatz zur Technologieauswahl für die automatische Identifikation von Werkstücken und

Produkten. In *Industrie 4.0 Management - Gegenwart und Zukunft industrieller Geschäftsprozesse* (5), pp. 55–58.

[122] Wiendahl, Hans-Peter (2019): Betriebsorganisation für Ingenieure. 9th ed. München, Germany: Hanser.

[123] Zhao, Dongming (2012): Imaging and Range Image Processing. In Bruce G. Batchelor (Ed.): Machine Vision Handbook. London, UK: Springer, pp. 625–663.

Appendix A - Utility analyses for hardware selection

This appendix contains the documents describing the utility analyses carried out in order to select the solution variants for the hardware solution principles in Section 3.2.1.2.

Criteria hardware subfunction 1 (HSF1)

Store individual piece good

Pairwise comparison for weighting factor calculation

			A	***B***		
i	***Criteria***		HSF1-C1	HSF1-C2	***Sum***	***Weighting factor w_i***
1	Effort for storage	HSF1-C1	*NaN*	*2*	*2*	*0,500*
2	Compatibility to other HSF	HSF1-C2	*2*	*NaN*	*2*	*0,500*
					4	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Effort for storage	*The effort which a solution principle for the storage of piece goods demands*
Compatibility to other HSF	*Compatibility of solution principle with other hardware subfunctions demands*

Statement of reasons

A2 Equally important since solution must offer easy handling and compatibility to other HSF at the same time

Utility analysis hardware subfunction 1 (HSF1)

Store individual piece good

Utility analysis

i	Criteria		Weighting factor w_i	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$
			Solution principle	HSP1-1-a		HSP1-1-b		HSP1-2-a		HSP1-2-b	
			j	1		2		3		4	
1	Effort for storage	HSF1-C1	0,500	4	2,00	1	0,50	4	2,00	1	0,50
2	Compatibility to other HSF	HSF1-C2	0,500	2	1,00	3	1,50	3	1,50	4	2,00

	HSP1-1-a	HSP1-1-b	HSP1-2-a	HSP1-2-b
$G_{w,j}$	3,00	2,00	3,50	2,50
$G_{wg,j}$	0,75	0,67	0,88	0,63

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Ideal, since placing a piece good on a flat surface is not complex at all

g_{12} Sufficient support of other HSF, since contact surface is not scannable and several perspectives necessary for complete scan

g_{21} Barely acceptable since there is no universal way of suspending all sorts of piece goods

g_{22} Good support of other HSF, since complete scanning would be possible from different perspectives

g_{31} Ideal, since placing a piece good on a flat surface is not complex at all

g_{32} Good support of other HSF, since contact surface is not scannable while roation enables different perspectives

g_{41} Barely acceptable since there is no universal way of suspending all sorts of piece goods

g_{42} Ideal support of other HSF, since complete scanning would be possible through rotation from one perspective

Criteria hardware subfunction 2 (HSF2)

Capture 3D-Geometry of stored piece good

Pairwise comparison for weighting factor calculation

			A	B	C		
i	***Criteria***		HSF2-C1	HSF2-C2	HSF2-C3	***Sum***	***Weighting factor w_i***
1	Accurancy	HSF2-C1	*NaN*	*2*	*3*	5	*0,417*
2	Robustness	HSF2-C2	*2*	*NaN*	*3*	5	*0,417*
3	Perception area	HSF2-C3	*1*	*1*	*NaN*	2	*0,167*
						12	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Accurancy	*The accuracy that a scan can achieve with the respective technology in terms of dimensional accuracy*
Robustness	*Robustness of the technology in terms of illumination and material surfaces*
Perception area	*Size of perception area which strongly relates to the necessary scanning time*

Statement of reasons

A2 Equally important since optimal solution would be accurate and robust at the same time
A3 Perception area is less important than accuracy, since the geometry gets matched and must be as accurate as possible
B3 Perception area is less important than robustness, since robustness is an absolute requirement

Utility analysis hardware subfunction 2 (HSF2)

Capture 3D-Geometry of stored piece good

Utility analysis

			Solution principle	HSP2-1-a		HSP2-1-b		HSP2-1-c		HSP2-2-a		HSP2-2-b	
			j	*1*		*2*		*3*		*4*		*5*	
i	***Criteria***		***Weighting factor*** w_i	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$
1	Accurancy	HSF2-C1	0,417	2	0,83	4	1,67	4	1,67	4	1,67	2	0,83
2	Robustness	HSF2-C2	0,417	1	0,42	4	1,67	4	1,67	4	1,67	2	0,83
3	Perception area	HSF2-C3	0,167	2	0,33	3	0,50	4	0,67	1	0,17	3	0,50

	HSP2-1-a	HSP2-1-b	HSP2-1-c	HSP2-2-a	HSP2-2-b
$G_{w,j}$	1,58	3,83	4,00	3,50	2,17
$G_{wg,j}$	0,79	0,96	1,00	0,88	0,72

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Sufficient accurancy, since there is a shift in disparity and only natural features are used for alginment of stereo photos

g_{12} Barely acceptable, because not applicable to uniform surface structures since surface features are used for alignment of stereo photos

g_{13} Sufficient area of perception, that is lower than the other variants

g_{21} Ideal accurancy possible by using a projected pattern which theoretically is not limited

g_{22} Ideal robustness due to the projected pattern which is used for calculation of 3D-Geometry

g_{23} Good perception area, but lower than active stereo vision

g_{31} Ideal accurancy possible by using a projected pattern which theoretically is not limited

g_{32} Ideal robustness due to the projected pattern which is used for calculation of 3D-Geometry

g_{33} Ideal perception area, that is furthermore minimizing occlusions

g_{41} Ideal accurancy through scanning ToF-Principle

g_{42} Ideal robustness due to scanning individual points and the ToF-Principle

g_{43} Barely acceptable perception area, since pointwise scanning is time consuming

g_{51} Sufficient accurancy, that is strongly dependent on the sensor quality and the current state of the art

g_{52} Sufficient robustness, because the surface texture of the scanned object has strong influence on scan quality

g_{53} Good perception area which is compareble to structured light 3D-Sensing

Criteria hardware subfunction 3 (HSF3)

Measure weight of stored piece good

Pairwise comparison for weighting factor calculation

i	*Criteria*		*A* HSF3-C1	*B* HSF3-C2	*C* HSF3-C3	*Sum*	*Weighting factor w_i*
1	Accurancy	HSF3-C1	*NaN*	*3*	*2*	*5*	*0,417*
2	Measuring effort	HSF3-C2	*1*	*NaN*	*1*	*2*	*0,167*
3	Compatibility to other HSF	HSF3-C3	*2*	*3*	*NaN*	*5*	*0,417*
						12	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Accurancy — *Absoulte accuancy the solution offers in terms of weight measurement*
Measuring effort — *Effort for measuring the weight and converting it to a machine understandable format*
Compatibility to other HSF — *Compatibility to the other HSF*

Statement of reasons

A2 Accurancy is more important than the effort for the measurement, since weight is an important identificaiton feature
A3 Accurancy is as important as compatibility to other HSF, because both are prerequisites for a successful identification
B3 Compatibility to other HSF is more important than measureing effort, since other HSF shouldn't be effekted by the solution

Utility analysis hardware subfunction 3 (HSF3)

Measure weight of stored piece good

Utility analysis

				HSP3-1-a		HSP3-2-a	
			Solution principle	HSP3-1-a		HSP3-2-a	
			j	*1*		*2*	
i	***Criteria***		***Weighting factor*** w_i	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$
1	Accurancy	HSF3-C1	0,417	3	1,25	4	1,67
2	Measuring effort	HSF3-C2	0,167	3	0,50	4	0,67
3	Compatibility to other HSF	HSF3-C3	0,417	1	0,42	4	1,67

$G_{w,j}$	2,17	4,00
$G_{wg,j}$	0,72	1,00

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Good accurancy, but comperatively high hysteresis due to the need of large deformation values and use of mechanical springs

g_{12} Good measuring effort, but need to measure travel which is then converted to an electrical signal which is machine readable

g_{13} Barely acceptable compatibility, since the displacement of the component due to the spring defomation affects optical recognition

g_{21} Ideal accurancy only limited available sensor hardware

g_{22} Ideal measuring effort, since the physical principle directly changes voltage that can easily be translated to a machine readable format

g_{23} Ideal compatibility, because the displacement is extremely low and no other HSF is affected

Criteria hardware subfunction 4 (HSF4)

Locate centre of mass of stored piece good

Pairwise comparison for weighting factor calculation

			A	B		
i	***Criteria***		HSF4-C1	HSF4-C2	***Sum***	***Weighting factor w_i***
1	Accurancy	HSF4-C1	*NaN*	*2*	*2*	*0,500*
2	Technical feasibility	HSF4-C2	*2*	*NaN*	*2*	*0,500*
					4	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Accurancy — *The absolute accurancy of the localisation of the centre of mass*

Technical feasibility — *The feasibility of technically implementing the solution principle*

Statement of reasons

A2 Equally important since solution must be technically feasible and accurate

Utility analysis hardware subfunction 4 (HSF4)

Locate centre of mass of stored piece good

Utility analysis

i	*Criteria*		*Solution principle*	HSP4-1-a		HSP4-1-b		HSP4-2-a	
			j	*1*		*2*		*3*	
			Weighting factor w_i	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$	g_{ji}	$w_i * g_{ji}$
1	Accurancy	HSF4-C1	0,500	4	2,00	3	1,50	1	0,50
2	Technical feasibility	HSF4-C2	0,500	4	2,00	2	1,00	4	2,00

	HSP4-1-a	HSP4-1-b	HSP4-2-a
$G_{w,j}$	4,00	2,50	2,50
$G_{wg,j}$	1,00	0,83	0,63

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Ideal accurancy only depending on the accurancy of force mesaurement and distances between bearing points

g_{12} Ideal technical feasibility, since force sensors are easy to use and deliver machine readable signals

g_{21} Good accurancy, since sensors for contace pressure are limited in resolution which directly limits accurancy of localisation

g_{22} Sufficient technical feasibility, since there are contact pressure sensors. These sensors are not as common as force sensors

g_{31} Barely acceptable, because centroid of point cloud only equals centre of mass if mass distribution of object is uniform

g_{32} Ideal feasibility, since centroid of point cloud is easy to calculate

Criteria hardware subfunction 5 (HSF5)

Capture material properties of stored piece good

Pairwise comparison for weighting factor calculation

			A	*B*		
	Criteria		HSF5-C1	HSF5-C2	***Sum***	***Weighting factor w_i***
1	Distinctiveness	HSF5-C1	*NaN*	*2*	*2*	*0,500*
2	Technical feasibility	HSF5-C2	*2*	*NaN*	*2*	*0,500*
					4	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Distinctiveness — *The degree of materials differentiation offered by the solution principle*
Technical feasibility — *The feasibility of technically implementing the solution principle*

Statement of reasons

A2 Distinctiveness and technical feasability are equally important, since are basic prerequisites

Utility analysis hardware subfunction 5 (HSF5)

Capture material properties of stored piece good

Utility analysis

			Solution principle	HSP5-1-a		HSP5-1-b		HSP5-1-c		HSP5-2-a		HSP5-3-a	
			j	*1*		*2*		*3*		*4*		*5*	
i	***Criteria***		***Weighting factor w_i***	***g_{ji}***	***w_i *g_{ji}***	***g_{ji}***	***w_i *g_{ji}***	***g_{ji}***	***w_i *g_{ji}***	***g_{ji}***	***w_i *g_{ji}***	***g_{ji}***	***w_i *g_{ji}***
1	Distinctiveness	HSF5-C1	0,500	2	1,00	2	1,00	2	1,00	3	1,50	2	1,00
2	Technical feasibility	HSF5-C2	0,500	4	2,00	4	2,00	4	2,00	1	0,50	4	2,00

	HSP5-1-a	HSP5-1-b	HSP5-1-c	HSP5-2-a	HSP5-3-a
$G_{w,j}$	3,00	3,00	3,00	2,00	3,00
$G_{wg,j}$	0,75	0,75	0,75	0,67	0,75

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Sufficient distinctiveness of materials, only magnetic and nonmagnetic materials can be distinguished
g_{12} Ideal feasibility through available and commonly used sensors
g_{21} Sufficient distinctiveness of materials, only magnetic and nonmagnetic materials can be distinguished
g_{22} Ideal feasibility through available and commonly used sensors
g_{31} Sufficient distinctiveness of materials, only metal and nonmetals can be distinguished
g_{32} Ideal feasibility through available and commonly used sensors
g_{41} Good distinctiveness of materials, since the material is analysed in detail. There are different spectroscopes for different material categories
g_{42} Spectroscopes are available for industrial usage but are not a heavily used technology in that field which limits availability
g_{51} Sufficient distinctiveness of materials, because object must consist of one material only and point cloud must be complete
g_{52} Ideal feasibility, because the volume of the point cloud can be easily calculated and weight properties can easily be acquired

Criteria hardware subfunction 6 (HSF6)

Sense primary colour of stored piece good

Pairwise comparison for weighting factor calculation

i	*Criteria*		A HSF6-C1	B HSF6-C2	C HSF6-C3	*Sum*	*Weighting factor w_i*
1	Accurancy	HSF6-C1	*NaN*	*2*	*1*	*3*	*0,250*
2	Measuring effort	HSF6-C2	*2*	*NaN*	*2*	*4*	*0,333*
3	Compatibility to other HSF	HSF6-C3	*3*	*2*	*NaN*	*5*	*0,417*
						12	

Scale for pairwise comparision

1 Less important
2 Equally important
3 More important

Description of criteria

Accurancy	*Absoulte accuancy the solution offers in terms of colour determination*
Measuring effort	*Effort for determining the colour*
Compatibility to other HSF	*Compatibility to the other HSF*

Statement of reasons

A2 Measuring effort is equally important as accurancy, since both are prerequisites for an successful identification
A3 Compatibility to other HSF is more important than accurancy since the CAD-Models don't give an excat colour description
B3 Compatibility to other HSF is equally important as measuring effort, since both are important prerequisites for the system

Utility analysis hardware subfunction 6 (HSF6)

Sense primary colour of stored piece good

Utility analysis

			Solution principle	HSP6-1-a		HSP6-2-a	
			j	*1*		*2*	
i	*Criteria*		*Weighting factor w_i*	*g_{ji}*	*$w_i * g_{ji}$*	*g_{ji}*	*$w_i * g_{ji}$*
1	Accurancy	HSF6-C1	0,250	3	0,75	4	1,00
2	Measuring effort	HSF6-C2	0,333	4	1,33	2	0,67
3	Compatibility to other HSF	HSF6-C3	0,417	4	1,67	1	0,42

$G_{w,j}$	3,75	2,08
$G_{wg,j}$	0,94	0,52

Scale for g_{ji} according to VDI 2225

0 unstatisfactory
1 barely acceptable
2 sufficient
3 good
4 ideal

Calculations

$$G_{w,j} = \sum_{i=0}^{i} w_i * g_{ji}$$

$$G_{wg,j} = \frac{\sum_{i=0}^{i} w_i * g_{ji}}{g_{ji,max}}$$

Statement of reasons

g_{11} Good accurancy, which is only limited by illumination conditions and quality of sensors

g_{12} Ideal measuring effort, since the camera for 3D-Scanning already delivers RGB-Values

g_{13} Ideal compatiblity, sice hardware is already available through 3D-Sensing technology

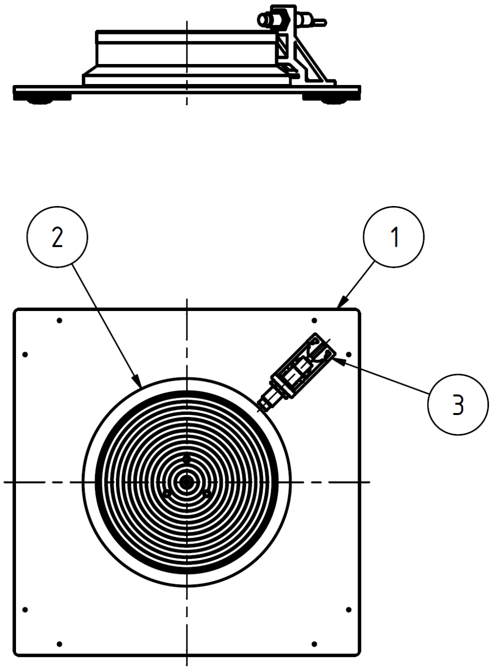
g_{21} Ideal accurancy, due to the measuring principle and percise sensors

g_{22} Sufficient measuring effort, since the colour determination takes additional time and additional hardware is necessary

g_{23} Barley acceptable, since the colour information from the CAD-Model is sparse which makes percise colour determination not useful at all

Appendix B - Technical documents

This appendix contains the technical documents, which were created as a basis for the implementation of the multi-sensor AIS prototype.



Bill of materials		
Position	Amount	Description
1	1	Weightening plate asm
2	1	Turn table
3	1	Material sensor asm

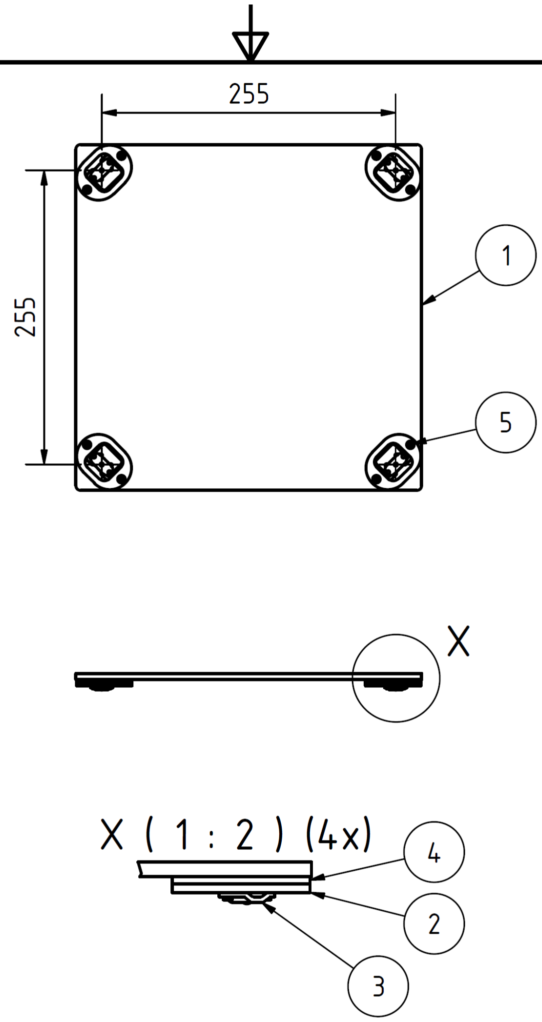
Material
Tolerances

	Date	Name
Drawn	17.09.2020	Marc Ungen
Checked	17.09.2020	Marc Ungen
Scale	1 : 5	

Sensor platform

1
A4

Rev.	Modification	Date	Name
00	Created	17.09.2020	



Bill of materials		
Position	Amount	Description
1	1	Weighing plate
2	4	Lower load cell housing
3	4	Load cell
4	4	Upper load cell housing
5	8	DIN 7991 – M3x10

Material | Tolerances

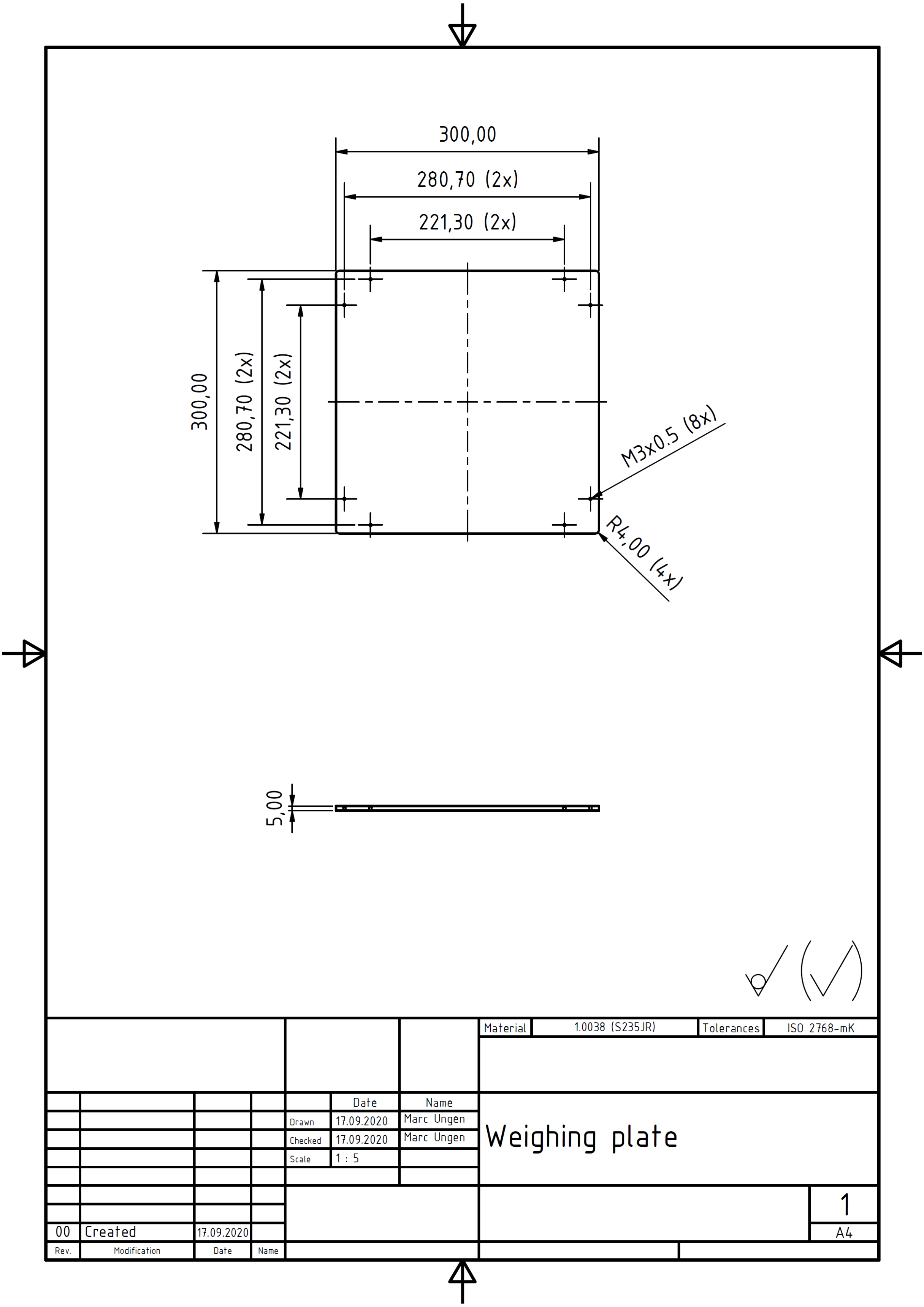
	Date	Name
Drawn	17.09.2020	Marc Ungen
Checked	17.09.2020	Marc Ungen
Scale	1 : 5	

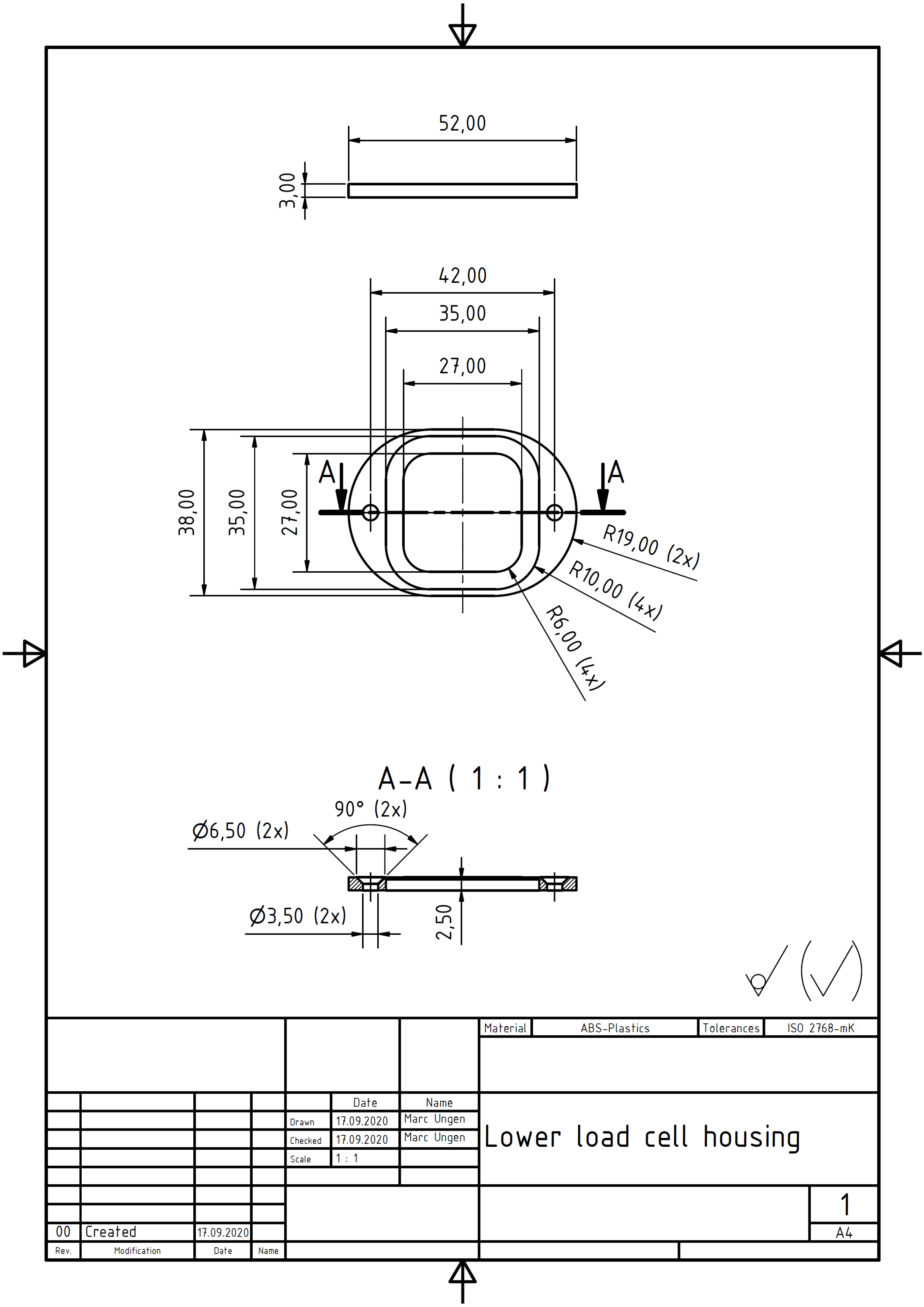
Weighing plate assembly

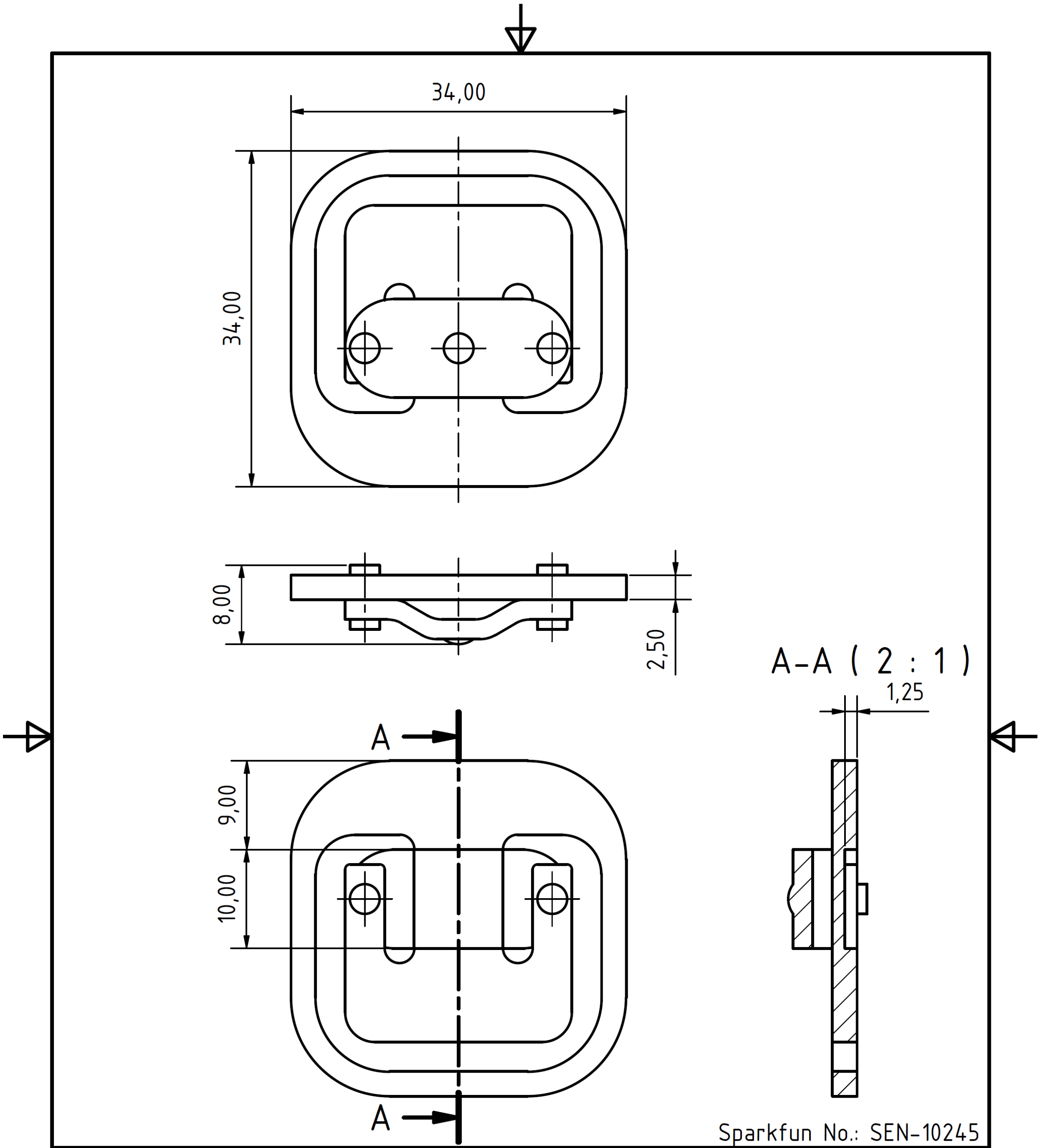
1

A4

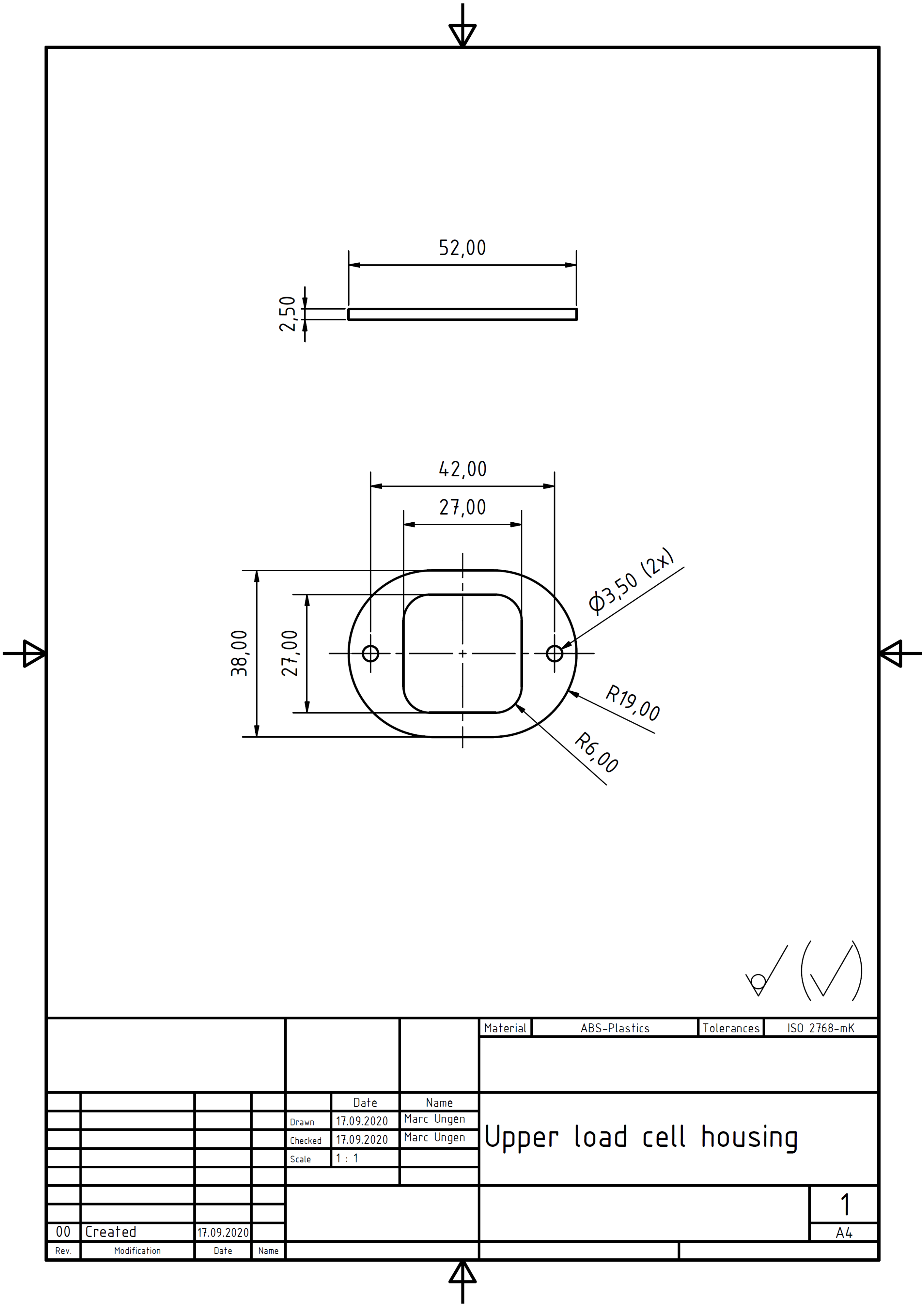
Rev.	Modification	Date	Name
00	Created	17.09.2020	

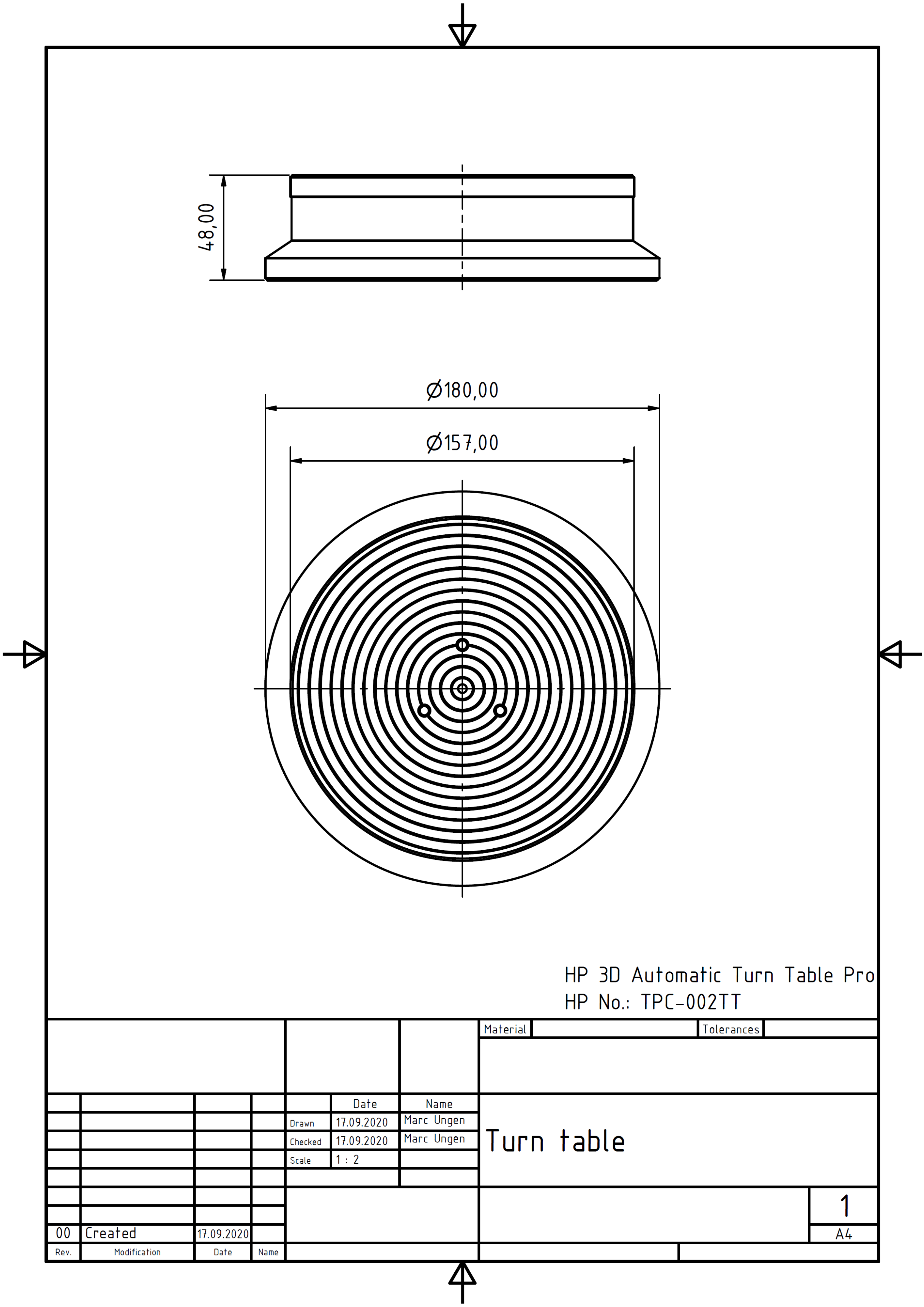

300,00
280,70 (2x)
221,30 (2x)
300,00
280,70 (2x)
221,30 (2x)
M3x0.5 (8x)
R4,00 (4x)
5,00
Material
1.0038 (S235JR)
Tolerances
ISO 2768-mK
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 5
Weighing plate
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name

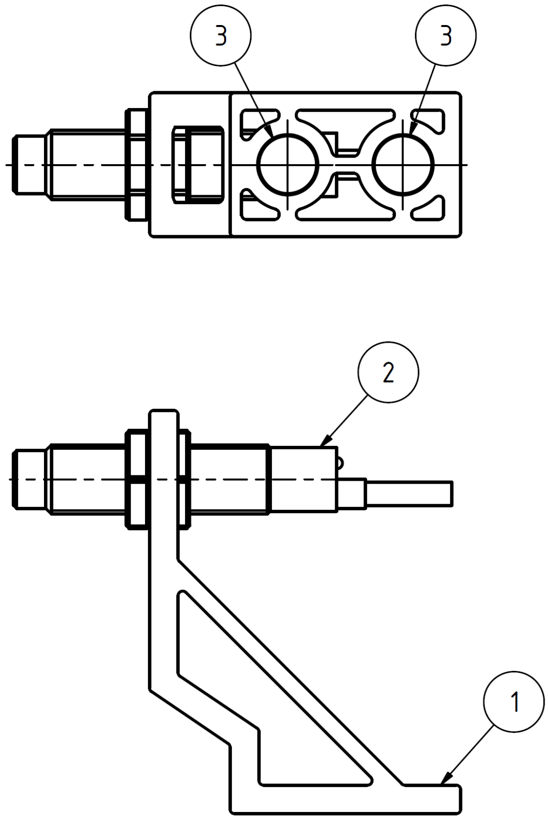

52,00
3,00
42,00
35,00
27,00
38,00
35,00
27,00
A
A
R19,00 (2x)
R10,00 (4x)
R6,00 (4x)
A-A (1 : 1)
90° (2x)
Ø6,50 (2x)
Ø3,50 (2x)
2,50
Material
ABS-Plastics
Tolerances
ISO 2768-mK
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 1
Lower load cell housing
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name



						Material	Tolerances ISO 2768-mk
					Date	Name	
				Drawn	17.09.2020	Marc Ungen	Load cell (adapted)
				Checked	17.09.2020	Marc Ungen	
				Scale	2 : 1		
							1
00	Created	17.09.2020					A4
Rev.	Modification	Date	Name				


52,00
2,50
42,00
27,00
38,00
27,00
⌀3,50 (2x)
R19,00
R6,00
Material
ABS-Plastics
Tolerances
ISO 2768-mK
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 1
Upper load cell housing
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name


48,00
Ø180,00
Ø157,00
HP 3D Automatic Turn Table Pro
HP No.: TPC-002TT
Material
Tolerances
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 2
Turn table
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name



Bill of materials		
Posititon	Amount	Description
1	1	Sensor mount
2	1	Inductive sensor
3	2	Magnet

Material | Tolerances

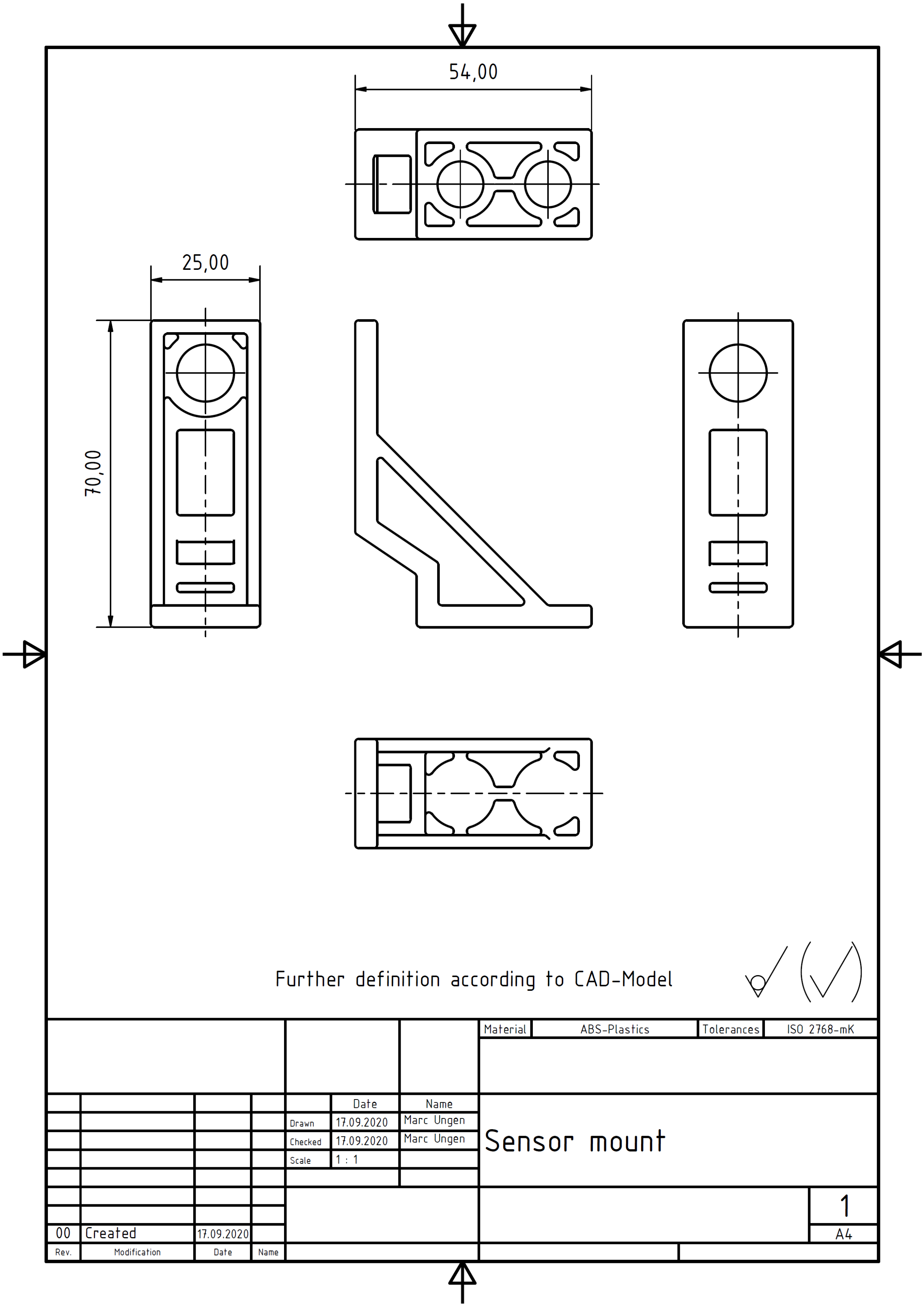
	Date	Name
Drawn	17.09.2020	Marc Ungen
Checked	17.09.2020	Marc Ungen
Scale	1 : 1	

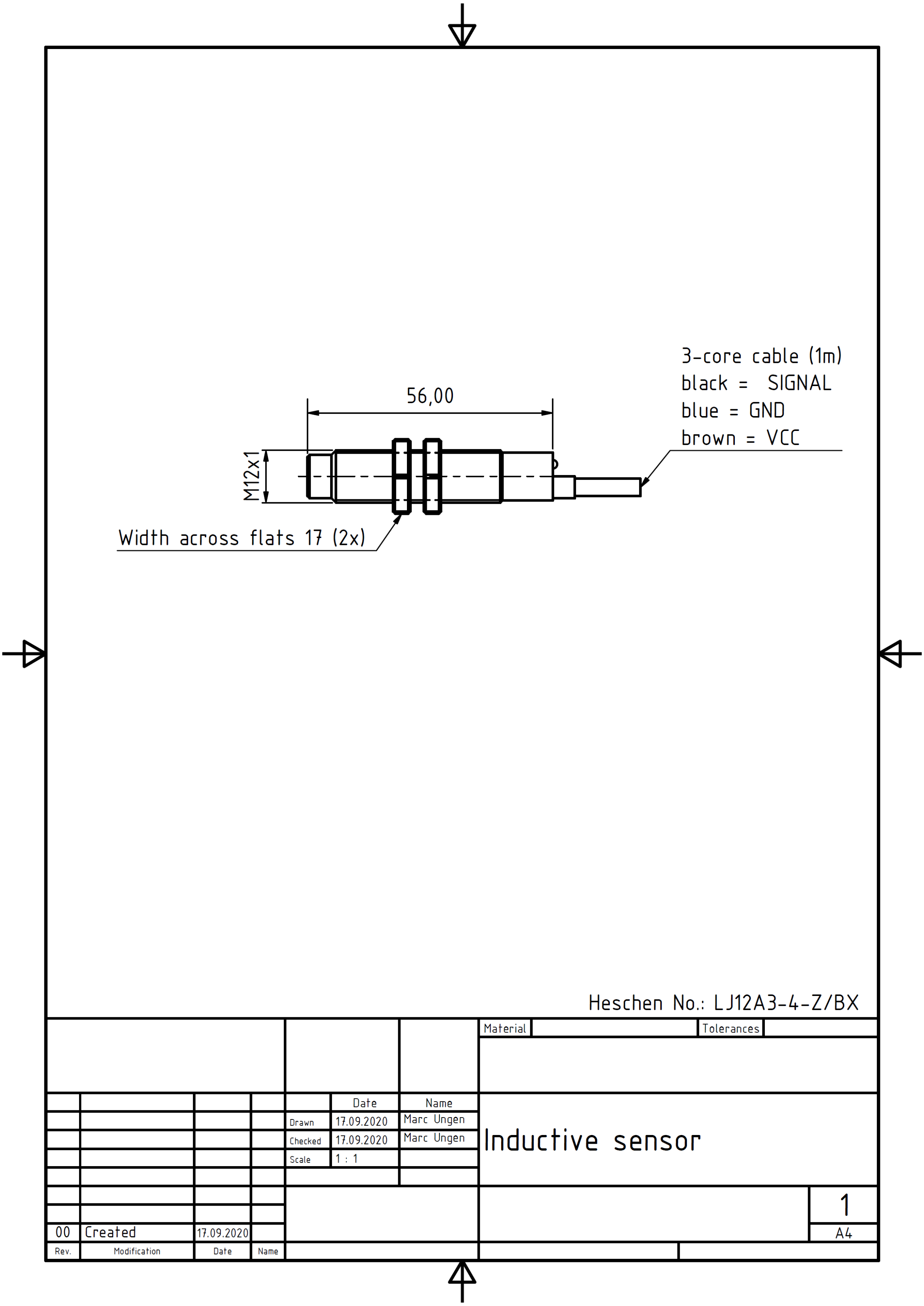
Material sensor assembly

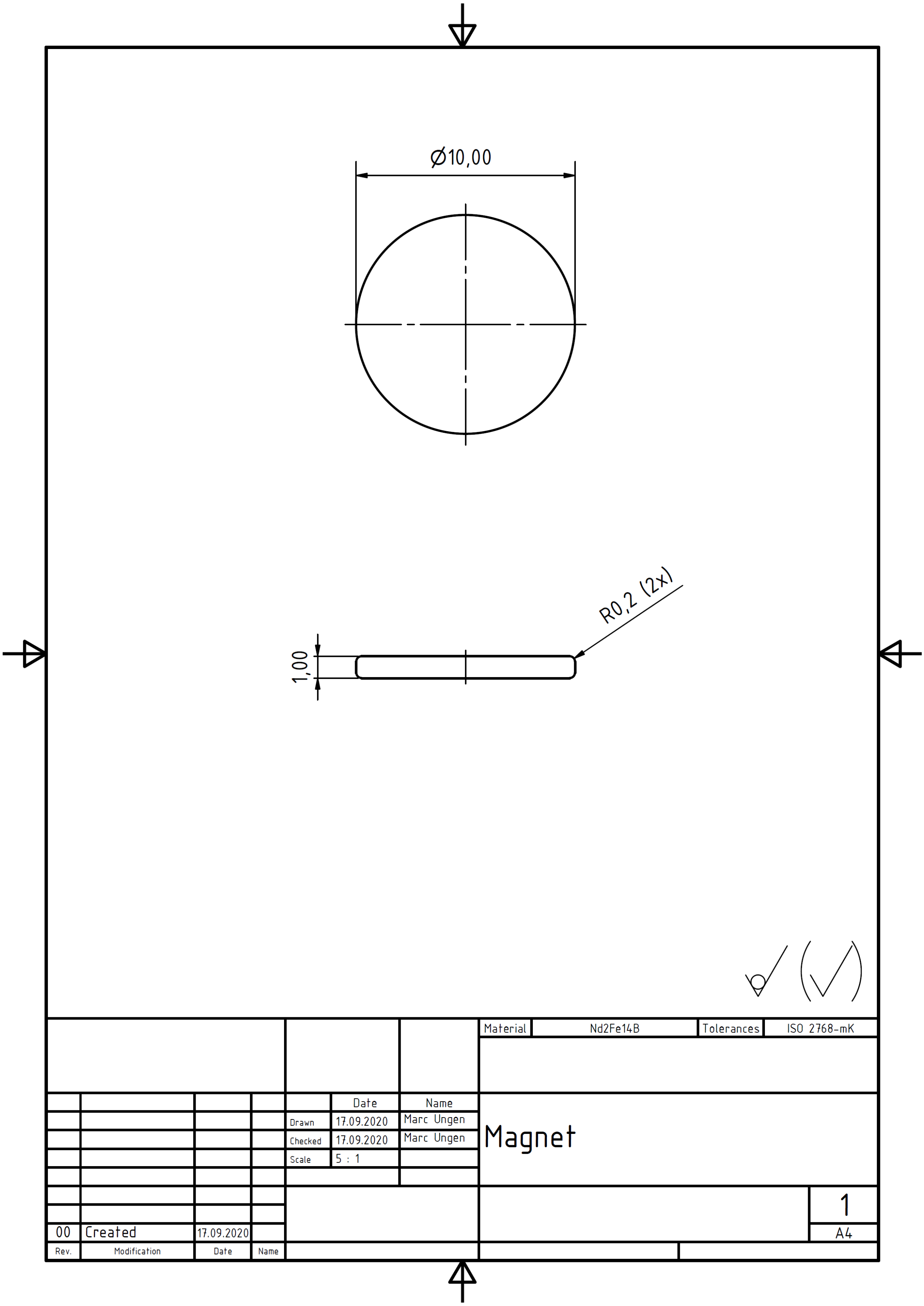
1

A4

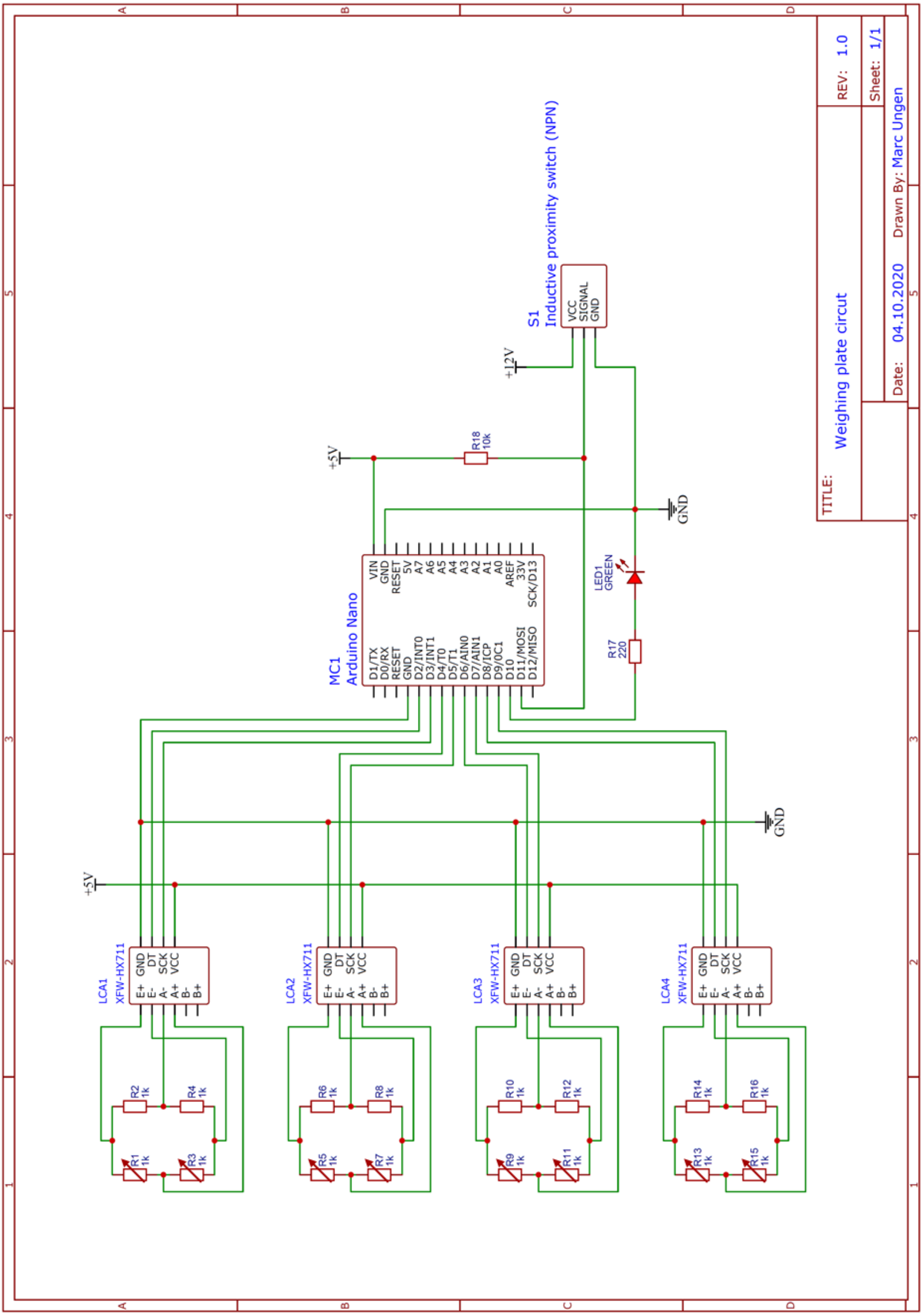
Rev.	Modification	Date	Name
00	Created	17.09.2020	


54,00
25,00
70,00
Further definition according to CAD-Model
Material
ABS-Plastics
Tolerances
ISO 2768-mK
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 1
Sensor mount
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name


3-core cable (1m)
black = SIGNAL
blue = GND
brown = VCC
56,00
M12x1
Width across flats 17 (2x)
Heschen No.: LJ12A3-4-Z/BX
Material
Tolerances
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
1 : 1
Inductive sensor
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name



Ø10,00
R0,2 (2x)
1,00
Material
Nd2Fe14B
Tolerances
ISO 2768-mK
Date
Name
Drawn
17.09.2020
Marc Ungen
Checked
17.09.2020
Marc Ungen
Scale
5 : 1
Magnet
1
A4
00
Created
17.09.2020
Rev.
Modification
Date
Name


TITLE:
Weighing plate circut
REV: 1.0
Sheet: 1/1
Date: 04.10.2020
Drawn By: Marc Ungen
S1
Inductive proximity switch (NPN)
VCC
SIGNAL
GND
+12V
+5V
R18
10k
GND
LED1
GREEN
R17
220
MC1
Arduino Nano
VIN
GND
RESET
5V
A7
A6
A5
A4
A3
A2
A1
A0
AREF
33V
SCK/D13
D1/TX
D0/RX
RESET
GND
D2/INT0
D3/INT1
D4/T0
D5/T1
D6/AIN0
D7/AIN1
D8/ICP
D9/OC1
D10
D11/MOSI
D12/MISO
LCA1
LCA2
LCA3
LCA4
XFW-HX711
GND
DT
SCK
VCC
E+
E-
A-
A+
B-
B+
R1 1k
R2 1k
R3 1k
R4 1k
R5 1k
R6 1k
R7 1k
R8 1k
R9 1k
R10 1k
R11 1k
R12 1k
R13 1k
R14 1k
R15 1k
R16 1k

Appendix C – Source codes

This appendix contains the source codes of all the software components developed for the implementation of the multi-sensor AIS prototype.

C.1 PCL binding module for Python (source.cpp - C++)

```cpp
// Libraries

#include <pybind11/pybind11.h>
#include <pybind11/stl.h>
#include <pcl/point_types.h>
#include <pcl/point_cloud.h>
#include <pcl/common/common.h>
#include <pcl/common/transforms.h>
#include <pcl/console/print.h>
#include <pcl/io/pcd_io.h>
#include <iostream>
#include <flann/flann.h>
#include <flann/io/hdf5.h>
#include <boost/filesystem.hpp>
#include <fstream>
#include <pcl/features/cvfh.h>
#include <pcl/features/normal_3d_omp.h>

// Definitions

namespace py = pybind11;

typedef pcl::PointXYZ PointType;
typedef pcl::Normal NormalType;
typedef pcl::VFHSignature308 DescriptorType;
typedef std::pair<std::string, std::vector<float> > cvfh_model;
typedef std::pair<std::string, double > result;

// Declarations of classes

class CVFHFeature
{

public:

    CVFHFeature()
    {
        normal_radius_ = 0.00f;
        normalize_bins_ = true;
        eps_angle_thresh_ = 5.00f;
        curvature_thresh_ = 1.00f;
        debug_ = false;
    }

    ~CVFHFeature()
    {
    }


    void setNormalEstimationRadius(float f)
    {
        normal_radius_ = f;
        return;
    }

    void normalizeBins(bool b)
    {
        normalize_bins_ = b;
        return;
    }

    void setEPSAngleThreshold(float f)
    {
        eps_angle_thresh_ = f;
        return;
    }

    void setCurvatureThreshold(float f)
    {
        curvature_thresh_ = f;
        return;
    }
```

```
    void setPCDSourceFile(std::string s)
    {
        source_file_ = s;
        return;
    }

    void setNormalsTargetFile(std::string s)
    {
        normals_file_ = s;
        return;
    }

    void setDescriptorTargetFile(std::string s)
    {
        descriptor_file_ = s;
        return;
    }

    void debug(bool b)
    {
        debug_ = b;
        return;
    }

    void describe()
    {
        run();
        return;
    }

private:

    bool debug_;
    bool normalize_bins_;
    float normal_radius_;
    float eps_angle_thresh_;
    float curvature_thresh_;
    std::string source_file_;
    std::string normals_file_;
    std::string descriptor_file_;

    void run()
    {
        pcl::PointCloud<PointType>::Ptr cloud(new pcl::PointCloud<PointType>);
        pcl::PointCloud<NormalType>::Ptr cloud_normals(new pcl::PointCloud<NormalType>());
        pcl::PointCloud<DescriptorType>::Ptr cvfhs(new pcl::PointCloud<DescriptorType>());

        // Import point cloud

        if (pcl::io::loadPCDFile(source_file_, *cloud) < 0)
        {
            std::cout << "[MODULE] Loading point cloud for featuring failed" << std::endl;
            return;
        }

        // Compute normals

        pcl::NormalEstimation<PointType, NormalType> normalEstimation;
        normalEstimation.setInputCloud(cloud);
        normalEstimation.setRadiusSearch(normal_radius_);
        pcl::search::KdTree<PointType>::Ptr kdtree(new pcl::search::KdTree<PointType>);
        normalEstimation.setSearchMethod(kdtree);
        normalEstimation.compute(*cloud_normals);

        // CVFH estimation
        pcl::CVFHEstimation<PointType, NormalType, DescriptorType> cvfh;

        cvfh.setInputCloud(cloud);
        cvfh.setInputNormals(cloud_normals);
        cvfh.setSearchMethod(kdtree);
        cvfh.setEPSAngleThreshold(eps_angle_thresh_ / 180.0 * M_PI);
        cvfh.setCurvatureThreshold(curvature_thresh_);
        cvfh.setNormalizeBins(normalize_bins_);
        cvfh.compute(*cvfhs);
```

```
        if (cvfhs->points.size() != 1)
        {
            std::cout << "[MODULE] Featuring failed " << std::endl;
            return;
        }

        //  Save files
        pcl::io::savePCDFile(normals_file_, *cloud_normals, true);
        pcl::io::savePCDFile(descriptor_file_, *cvfhs, true);
        if (debug_)
        {
            std::cout << "[MODULE] Files have been written" << std::endl;
            std::cout << "[MODULE] CVFH Descriptor has been computed" << std::endl;
        }
        return;
    }
};


class CVFHMatcher
{

public:

    CVFHMatcher()
    {
        kdtree_idx_file_path_ = "kdtree.idx";
        training_data_h5_file_path_ = "training_data.h5";
        training_data_list_file_path_ = "training_data.list";
        k_ = 1;
        debug_ = false;
    }

    ~CVFHMatcher() {}

    void setKdTreeIdxFilePath(std::string s)
    {
        kdtree_idx_file_path_ = s;
        return;
    }

    void setTrainingDataH5FilePath(std::string s)
    {
        training_data_h5_file_path_ = s;
        return;
    }

    void setTrainingDataListFilePath(std::string s)
    {
        training_data_list_file_path_ = s;
        return;
    }

    void setDatabasePath(std::string s)
    {
       database_path_ = s;
       return;
    }

    void setNearestNeighbors(int i)
    {
        k_ = i;
        return;
    }

    void debug(bool b)
    {
        debug_ = b;
        return;
    }
```

```
void addModel(const std::string model_path_)
{
    cvfh_model cvfh;
    pcl::PCLPointCloud2 cloud;
    int version;
    Eigen::Vector4f origin;
    Eigen::Quaternionf orientation;
    pcl::PCDReader r;
    int type; unsigned int idx;
    r.readHeader(model_path_, cloud, origin, orientation, version, type, idx);
    int cvfh_idx = pcl::getFieldIndex(cloud, "vfh");
    if (cvfh_idx == -1)
    {
        std::cout << "[MODULE] Failed loading model descriptor" << std::endl;
    }
    if ((int)cloud.width * cloud.height != 1)
    {
        std::cout << "[MODULE] Failed loading model descriptor" << std::endl;
    }

    // Load CVFH and add to training data
    pcl::PointCloud <DescriptorType> point;
    pcl::io::loadPCDFile(model_path_, point);
    cvfh.second.resize(308);

    std::vector <pcl::PCLPointField> fields;
    pcl::getFieldIndex<DescriptorType>("cvfh", fields);

    for (std::size_t i = 0; i < fields[cvfh_idx].count; ++i)
    {
        cvfh.second[i] = point.points[0].histogram[i];
    }
    cvfh.first = model_path_;
    models.push_back(cvfh);
    return;
}

void addScene(const std::string scene_path_)
{
    int cvfh_idx;
    pcl::PCLPointCloud2 cloud;
    int version;
    Eigen::Vector4f origin;
    Eigen::Quaternionf orientation;
    pcl::PCDReader r;
    int type; unsigned int idx;
    r.readHeader(scene_path_, cloud, origin, orientation, version, type, idx);

    cvfh_idx = pcl::getFieldIndex(cloud, "vfh");
    if (cvfh_idx == -1)
    {
        std::cout << "[MODULE] Failed loading scene descriptor" << std::endl;
    }

    if ((int)cloud.width * cloud.height != 1)
    {
        std::cout << "[MODULE] Failed loading scene descriptor" << std::endl;
    }

    // Load CVFH
    pcl::PointCloud <DescriptorType> point;
    pcl::io::loadPCDFile(scene_path_, point);
    scene_cvfh_.second.resize(308);

    std::vector <pcl::PCLPointField> fields;
    pcl::getFieldIndex<DescriptorType>("cvfh", fields);

    for (std::size_t i = 0; i < fields[cvfh_idx].count; ++i)
    {
        scene_cvfh_.second[i] = point.points[0].histogram[i];
    }
    scene_cvfh_.first = scene_path_;
    return;
}
```

```
    void train()
    {
        run_training();
        return;
    }

    std::vector<std::pair<std::string, double >> match()
    {
        run_matching();
        return results_;
    }

private:

    std::string kdtree_idx_file_path_;
    std::string training_data_h5_file_path_;
    std::string training_data_list_file_path_;
    std::string database_path_;
    std::vector<cvfh_model> models;
    cvfh_model scene_cvfh_;
    bool debug_;
    int k_;
    std::vector<result> results_;

    void run_training()
    {

        flann::Matrix<float> data(new float[models.size() * models[0].second.size()],
models.size(), models[0].second.size());

        for (std::size_t i = 0; i < data.rows; ++i)
            for (std::size_t j = 0; j < data.cols; ++j)
                data[i][j] = models[i].second[j];

        // Save training data
        flann::save_to_file(data, training_data_h5_file_path_, "training_data");
        std::ofstream fs;
        fs.open(training_data_list_file_path_.c_str());
        for (std::size_t i = 0; i < models.size(); ++i)
            fs << models[i].first << "\n";
        fs.close();

        if (debug_)
        {
            std::cout << "[MODULE] Building KDTree index for " << data.rows << " models" <<
std::endl;
        }

        // Generate and save tree index
        flann::Index<flann::ChiSquareDistance<float> > index(data,flann::LinearIndexParams());
        //flann::Index<flann::ChiSquareDistance<float> > index (data, flann::KDTreeIndexParams
(4));
        index.buildIndex();
        index.save(kdtree_idx_file_path_);
        delete[] data.ptr();

        return;
    }

    inline void nearestKSearch(flann::Index<flann::ChiSquareDistance<float> >& index, const
cvfh_model& model,
            int k, flann::Matrix<int>& indices, flann::Matrix<float>& distances)
    {
        flann::Matrix<float> p = flann::Matrix<float>(new float[model.second.size()], 1,
model.second.size());
        memcpy(&p.ptr()[0], &model.second[0], p.cols * p.rows * sizeof(float));

        indices = flann::Matrix<int>(new int[k], 1, k);
        distances = flann::Matrix<float>(new float[k], 1, k);
        index.knnSearch(p, indices, distances, k, flann::SearchParams(512));
        delete[] p.ptr();

        return;
    }
```

```
bool loadFileList(std::vector<cvfh_model>& models, const std::string& filename)
{
    std::fstream fs;
    fs.open(filename.c_str());
    if (!fs.is_open() || fs.fail())
        return (false);

    std::string line;
    while (!fs.eof())
    {
        getline(fs, line);
        if (line.empty())
            continue;
        cvfh_model m;
        m.first = line;
        models.push_back(m);
    }
    fs.close();
    return (true);
}

void run_matching()
{

    std::string extension(".pcd");
    transform(extension.begin(), extension.end(), extension.begin(),(int(*)(int))tolower);

    std::vector<cvfh_model> models;
    flann::Matrix<int> k_indices;
    flann::Matrix<float> k_distances;
    flann::Matrix<float> data;

    if (!boost::filesystem::exists(training_data_h5_file_path_) ||
    !boost::filesystem::exists(training_data_list_file_path_))
    {
        std::cout << "[MODULE] Loading training data for matching failed" << std::endl;
        return;
    }
    else
    {
        loadFileList(models, training_data_list_file_path_);
        flann::load_from_file(data, training_data_h5_file_path_, "training_data");
        if (data.rows < k_)
        {
            std::cout << "[MODULE] Not enough models in training data for " << k_ <<
            "nearest neighbors" << std::endl;
            return;
        }

        if (debug_)
        {
            std::cout << "[MODULE] Found training data for matching (" << data.rows <<
            "models loaded)" << std::endl;
        }
    }

    if (!boost::filesystem::exists(kdtree_idx_file_path_))
    {
        std::cout << "[MODULE] Loading KD-Tree index file failed" << std::endl;
        return;
    }
    else
    {
        flann::Index<flann::ChiSquareDistance<float> >
         index(data,flann::SavedIndexParams(kdtree_idx_file_path_));

        index.buildIndex();
        nearestKSearch(index, scene_cvfh_, k_, k_indices, k_distances);
    }

    if (debug_)
    {
        std::cout << "[MODULE] The closest " << k_ << " neighbors are:" << std::endl;
    }

    results_.clear();

    for (int i = 0; i < k_; ++i)
```

```
        {
            std::string path = models.at(k_indices[0][i]).first.c_str();
            double distance = k_distances[0][i];
            result r;
            r.first = path;
            r.second = distance;
            results_.push_back(r);

            if (debug_)
            {
                std::cout << "[MODULE] " << path << "with distance: " << distance <<std::endl;
            }
        }

        return;
    }
};


PYBIND11_MODULE(CVFHRecognition, m) {

    m.doc() = "Python CAD Recognition Module by Marc Ungen";
    py::class_<CVFHFeature>(m, "CVFHFeature")
        .def(py::init<>())
        .def("setNormalEstimationRadius", &CVFHFeature::setNormalEstimationRadius)
        .def("setEPSAngleThreshold", &CVFHFeature::setEPSAngleThreshold)
        .def("setCurvatureThreshold", &CVFHFeature::setCurvatureThreshold)
        .def("normalizeBins", &CVFHFeature::normalizeBins)
        .def("setPCDSourceFile", &CVFHFeature::setPCDSourceFile)
        .def("setNormalsTargetFile", &CVFHFeature::setNormalsTargetFile)
        .def("setDescriptorTargetFile", &CVFHFeature::setDescriptorTargetFile)
        .def("describe", &CVFHFeature::describe)
        ;

    py::class_<CVFHMatcher>(m, "CVHFMatcher")
        .def(py::init<>())
        .def("setKdTreeIdxFilePath", &CVFHMatcher::setKdTreeIdxFilePath)
        .def("setTrainingDataH5FilePath", &CVFHMatcher::setTrainingDataH5FilePath)
        .def("setTrainingDataListFilePath", &CVFHMatcher::setTrainingDataListFilePath)
        .def("setDatabasePath", &CVFHMatcher::setDatabasePath)
        .def("addModel", &CVFHMatcher::addModel)
        .def("addScene", &CVFHMatcher::addScene)
        .def("debug", &CVFHMatcher::debug)
        .def("setNearestNeighbors", &CVFHMatcher::setNearestNeighbors)
        .def("train", &CVFHMatcher::train)
        .def("match", &CVFHMatcher::match)
        ;

}
```

C.2 Microcontroller software (main.ino - C)

```
/* Libraries */

#include "HX711.h"

/* Variables */

String cmd; // Variable for received command via serial

const int pin_led = 10; // Serial status LED pin
const int pin_ind_sensor = 11; // Inductive sensor pin
const int pin_lc1_dout = 2; // Loadcell 1 HX711 DT pin
const int pin_lc1_sck = 3; // Loadcell 1 HX711 SCK pin
const int pin_lc2_dout = 4; // Loadcell 2 HX711 DT pin
const int pin_lc2_sck = 5; // Loadcell 2 HX711 SCK pin
const int pin_lc3_dout = 6; // Loadcell 3 HX711 DT pin
const int pin_lc3_sck = 7; // Loadcell 3 HX711 SCK pin
const int pin_lc4_dout = 8; // Loadcell 4 HX711 DT pin
const int pin_lc4_sck = 9; // Loadcell 4 HX711 SCK pin

long lc1_tare = 0; // Loadcell 1 tare value
long lc2_tare = 0; // Loadcell 2 tare value
long lc3_tare = 0; // Loadcell 3 tare value
long lc4_tare = 0; // Loadcell 4 tare value

double lc1_scale = 0.986; // Loadcell 1 scale value
double lc2_scale = 0.992; // Loadcell 2 scale value
double lc3_scale = 1.000; // Loadcell 3 scale value
double lc4_scale = 0.975; // Loadcell 4 scale value

double lc1_val = 0; // Loadcell 1 value scaled and tared
double lc2_val = 0; // Loadcell 2 value scaled and tared
double lc3_val = 0; // Loadcell 3 value scaled and tared
double lc4_val = 0; // Loadcell 4 value scaled and tared
double sum_val = 0; // Sum of all scaled and tared loadcell values


double weight_factor = 0.0065; // Conversion factor of scaled loadcell value to weight
double weight = 0.00; // Weight value in gram

double x_ratio = 0; // Ratio of x-coordinate in terms of l, between 0 to 1
double y_ratio = 0; // Ratio of y-coordinate in terms of l, from 0 to 1
const double l = 255.00; // Distance between bearing points of sensor platform in millimetre
double x = 0; // Centre of masses x-coordinate in millimetre with regard to platforms
coordinate system
double y = 0; // Centre of masses y-coordinate in millimetre with regard to platforms
coordinate system

/* Objects */

HX711 lc1;  //Loadcell 1 Object
HX711 lc2;  //Loadcell 2 Object
HX711 lc3;  //Loadcell 3 Object
HX711 lc4;  //Loadcell 4 Object

/* Initialisation */

void setup() {
  pinMode(pin_led, OUTPUT); // Set digital pin as an output
  pinMode(pin_ind_sensor, INPUT); // Set digital pin as an input

  Serial.begin(250000); // Begin serial communication and wait for connection
  while (!Serial)
  {
  }

  Serial.println("sensorplatform"); // Send whois feedback

  while (true){
    if (Serial.available() > 0)
    {// Case serial command available
      cmd = Serial.readString();
      cmd.trim();
```

```
      if (cmd == "init")  // Initialisation of sensors on serial command
      {
        lc1.begin(pin_lc1_dout, pin_lc1_sck); // Begin communication with HX711 of loadcell
        lc1.set_gain(128);   // Set amplifier gain of loadcell
        lc1_tare = lc1.read();  // Read value as tare value from loadcell

        lc2.begin(pin_lc2_dout, pin_lc2_sck);
        lc2.set_gain(128);
        lc2_tare = lc2.read();

        lc3.begin(pin_lc3_dout, pin_lc3_sck);
        lc3.set_gain(128);
        lc3_tare = lc3.read();


        lc4.begin(pin_lc4_dout, pin_lc4_sck);
        lc4.set_gain(128);
        lc4_tare = lc4.read();

        digitalWrite(pin_led, HIGH); // Turn on status LED for active serial communication

        Serial.println("success");  // Report successful initialisation via serial

        break;
      }
    }
  }
}

/* Main loop */

void loop() {

  // Read calibrated and scaled values of all loadcells
  lc1_val = (double(lc1.read() - lc1_tare) * lc1_scale);
  lc2_val = (double(lc2.read() - lc2_tare) * lc2_scale);
  lc3_val = (double(lc3.read() - lc3_tare) * lc3_scale);
  lc4_val = (double(lc4.read() - lc4_tare) * lc4_scale);
  sum_val = lc1_val + lc2_val + lc3_val + lc4_val;

  // Calculate weight
  weight = sum_val * weight_factor;

  if (weight > 50)
  {// Case object is settled on sensor platform
    // Calculate centre of mass coordinate ratios from bearing forces and limit values
    x_ratio = (lc2_val + lc3_val) / sum_val;
    y_ratio = (lc3_val + lc4_val) / sum_val;

    if(x_ratio < 0.00)
    {
      x_ratio = 0.00;
    }
    else if (x_ratio > 1.00)
    {
      x_ratio = 1.00;
    }

    if(y_ratio < 0.00)
    {
      y_ratio = 0.00;
    }
    else if (y_ratio > 1.00)
    {
      y_ratio = 1.00;
    }

    // Calculate centre of mass coordinates in mm
    x = x_ratio * l;
    y = y_ratio * l;
    }
  else
  {// Case no object settles on sensor platform
      // Reset all values
      weight = 0.00;
      x = 0.00;
      y = 0.00;
  }
```

```
  if (Serial.available() > 0)
  {// Case serial command available
    // Read command from serial
    cmd = Serial.readString();
    cmd.trim();

    if (cmd == "material")
    {// Case serial command requests material information
        if (!digitalRead(pin_ind_sensor) == 1)
        {// Case inductive switch detects metal
          Serial.println("metal");  // Report metal via serial
        }
        else
        {// Case inductive switch detects mommetal
          Serial.println("nonmetal"); // Report nonmetal via serial
        }
    }
    else if (cmd == "tare")
    {// Case serial command requests taring of loadcells
      lc1_tare = lc1.read();  // Read value as new tare value from loadcell
      lc2_tare = lc2.read();
      lc3_tare = lc3.read();
      lc4_tare = lc4.read();
    }
    else if (cmd == "centreofmass")
    {// Case serial command requests centre of mass coordinates
      Serial.println(String(x, 2) + " " + String(y, 2)); // Report coordinates via serial
    }
    else if (cmd == "weight")
    {// Case serial command requests weight
      Serial.println(String(weight, 2));  // Report weight via serial
    }
  }
}
```

C.3 3D-Object recognition component (three_dimensional_object_recognition.py - Python)

```python
# Imports and libraries

import copy
from modules.CVFH import CVFHRecognition
import numpy as np
import open3d as o3d
import os


# Declarations of classes

class CVFHDescriptorRecognition:

  def __init__(self, config):

      self.normal_estimation_radius = float(config["normal_estimation_radius"])
      self.eps_angle_threshold = float(config["eps_angle_threshold"])
      self.curvature_threshold = float(config["curvature_threshold"])
      self.normalize_bins = bool(config["normalize_bins"])
      self.recognition_knowledge_base_path = os.getcwd().replace("\\", "/") +
        str(config["recognition_knowledge_base_path"])

      self.feat = CVFHRecognition.CVFHFeature()
      self.feat.setNormalEstimationRadius(self.normal_estimation_radius)
      self.feat.setEPSAngleThreshold(self.eps_angle_threshold)
      self.feat.setCurvatureThreshold(self.curvature_threshold)
      self.feat.normalizeBins(self.normalize_bins)

      self.matcher = CVFHRecognition.CVHFMatcher()
      self.matcher.setKdTreeIdxFilePath(self.recognition_knowledge_base_path + "/kdtree.idx")
      self.matcher.setTrainingDataH5FilePath(self.recognition_knowledge_base_path +
        "/training_data.h5")
      self.matcher.setTrainingDataListFilePath(self.recognition_knowledge_base_path +
        "/training_data.list")
      self.matcher.setNearestNeighbors(1)

      return

  def generate_descriptor(self, pcd_file_path, normals_path, descr_path):

      self.feat.setPCDSourceFile(pcd_file_path)
      self.feat.setNormalsTargetFile(normals_path)
      self.feat.setDescriptorTargetFile(descr_path)
      self.feat.describe()

      return

  def perform_training(self, cvfh_file_paths):

      for file in os.listdir(self.recognition_knowledge_base_path):
          os.remove(self.recognition_knowledge_base_path + "/" + file)

      trainer = CVFHRecognition.CVHFMatcher()
      trainer.debug(False)
      trainer.setKdTreeIdxFilePath(self.recognition_knowledge_base_path + "/kdtree.idx")
      trainer.setTrainingDataH5FilePath(self.recognition_knowledge_base_path +
        "/training_data.h5")
      trainer.setTrainingDataListFilePath(self.recognition_knowledge_base_path +
        "/training_data.list")

      for cvfh in cvfh_file_paths:
          trainer.addModel(cvfh)

      trainer.train()

      return
```

```python
  def perform_matching(self, cvfh_file_path):

      self.matcher.addScene(cvfh_file_path)

      return self.matcher.match()


class SixDoFPoseEstimation:

  def __init__(self, config):

      self.voxel_size = float(config["voxel_size"])

      return

  def __feature_pc(self, pc):

      pc_down = pc.voxel_down_sample(self.voxel_size * 2)

      radius_normal = self.voxel_size * 5
      pc_down.estimate_normals(o3d.geometry.KDTreeSearchParamHybrid(radius=radius_normal,
        max_nn=20))

      radius_feature = self.voxel_size * 12
      pc_fpfh = o3d.registration.compute_fpfh_feature(pc_down,
        o3d.geometry.KDTreeSearchParamHybrid(radius=radius_feature, max_nn=500))

      return pc_down, pc_fpfh

  def __fast_registration(self, source_down, target_down, source_fpfh, target_fpfh):

      distance_threshold = self.voxel_size * 1.0

      result = o3d.registration.registration_fast_based_on_feature_matching(
          source_down, target_down, source_fpfh, target_fpfh,
          o3d.registration.FastGlobalRegistrationOption(
              maximum_correspondence_distance=distance_threshold))

      return result

  def __icp_registration(self, source, target, global_transformation):

      distance_threshold = self.voxel_size * 0.5
      radius_normal = self.voxel_size * 2

      source.estimate_normals(o3d.geometry.KDTreeSearchParamHybrid(radius=radius_normal,
        max_nn=20))
      target.estimate_normals(o3d.geometry.KDTreeSearchParamHybrid(radius=radius_normal,
        max_nn=20))

      result = o3d.registration.registration_icp(
          source, target, distance_threshold, global_transformation,
          o3d.registration.TransformationEstimationPointToPlane())

      return result

  def get_transformation(self, source_pcd_path, target_pcd_path):

      pc_source = o3d.io.read_point_cloud(source_pcd_path, format='pcd')
      pc_target = o3d.io.read_point_cloud(target_pcd_path, format='pcd')

      pc_source_down, pc_source_down_fpfh = self.__feature_pc(pc_source)
      pc_target_down, pc_target_down_fpfh = self.__feature_pc(pc_target)

      global_registration_result = self.__fast_registration(pc_source_down, pc_target_down,
        pc_source_down_fpfh, pc_target_down_fpfh)

      icp_registration_result = self.__icp_registration(pc_source, pc_target,
        global_registration_result.transformation)

      return icp_registration_result.transformation
```

C.4 Multi-sensor data processing component (multi_sensor_data_processing.py - Python)

```python
# Imports and libraries

import serial.tools.list_ports
import serial
import time
import open3d as o3d
import os
import cv2
import numpy as np


# Declarations of classes

class StructuredLightSensor:

    def __init__(self, config):

        self.scene_pcd_path = os.getcwd().replace("\\", "/") + str(config["scene_pcd_path"])
        self.scan_data_path = os.getcwd().replace("\\", "/") + str(config["scan_data_path"])

        return

    def __stl_to_pcd(self, stl_file_path, pcd_file_path, voxel_size=0.50):

        msh = o3d.io.read_triangle_mesh(stl_file_path)
        pcl = msh.sample_points_uniformly(float(config["sample_points"]))
        pcl.paint_uniform_color([0, 0, 0])
        pcl_down = pcl.voxel_down_sample(voxel_size)
        o3d.io.write_point_cloud(pcd_file_path, pcl_down, write_ascii=True)

        return

    def get_point_cloud(self):

        files = os.listdir(self.scan_data_path)
        scan_stl_path = None

        for f in files:
            if ".stl" in f.lower():
                scan_stl_path = self.scan_data_path + "/" + f
                break

        if not scan_stl_path:
            raise Warning("[MULTISENSORDATAPROCESSING] No stl data from scan")

        self.__stl_to_pcd(scan_stl_path, self.scene_pcd_path + "/scene_pc.pcd")

        return

    def get_color(self):

        files = os.listdir(self.scan_data_path)
        scan_jpg_path = None

        for f in files:
            if ".jpg" in f.lower():
                scan_jpg_path = self.scan_data_path + "/" + f
                break

        if not scan_jpg_path:
            raise Warning("[MULTISENSORDATAPROCESSING] No color image data from scan")

        colors = [["red", np.array([0, 180, 0]), np.array([5, 255, 255])],
                  ["yellow", np.array([10, 150, 190]), np.array([35, 255, 255])],
                  ["orange", np.array([0, 200, 210]), np.array([15, 255, 255])],
                  ["green", np.array([50, 160, 170]), np.array([70, 255, 255])],
                  ["blue", np.array([95, 115, 40]), np.array([140, 255, 255])],
                  ["grey", np.array([0, 0, 25]), np.array([95, 35, 255])],
                  ]

        img = cv2.imread(scan_jpg_path)
        img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

```
        detected_colors = []

        for i in colors:
            mask = cv2.inRange(img_hsv, i[1], i[2])
            max_area = 0
            cnt, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
            if len(cnt) != 0:
                for c in cnt:
                    area = cv2.contourArea(c)
                    if area > 100:
                        if area > max_area:
                            max_area = area


                if max_area != 0:
                    detected_colors.append([i[0], max_area])

        max_area = 0
        predominant_color = None
        for i in detected_colors:
            if i[1] > max_area:
                predominant_color = i[0]
                max_area = i[1]

        return predominant_color


class SensorPlatform:

  def __init__(self):

    self.s = serial.Serial(timeout=0.01)
    active_ports = [comport.device for comport in serial.tools.list_ports.comports()]

    if len(active_ports) > 0:
        for port in active_ports:
            try:
                self.s = serial.Serial(port, 250000)

            except:
                continue
            time.sleep(2)
            data = self.s.readline().rstrip().lstrip().decode("utf-8")
            if data == "sensorplatform":
                self.s.write(b'init')
                data = self.s.readline().rstrip().lstrip().decode("utf-8")
                if data == "success":
                    print("[SENSORPLATFORM] Initialisation successful on port " + port)
                    return
                raise Exception("[SENSORPLATFORM] Initialisation failed")
            else:
                continue

            raise Exception("[SENSORPLATFORM] Not found")
    return

  def get_material(self):

      self.s.write(b'material')
      time.sleep(0.01)
      data = self.s.readline().rstrip().lstrip().decode("utf-8")

      if data == "metal":
          return "metal"

      elif data == "nonmetal":
          return "nonmetal"

      return
```

```
def get_centreofmass(self):

    self.s.write(b'centreofmass')
    time.sleep(0.01)
    data = self.s.readline().rstrip().lstrip().decode("utf-8")
    values = list(map(float, data.split()))

    return values

def get_weight(self):

    self.s.write(b'weight')
    time.sleep(0.01)
    data = self.s.readline().rstrip().lstrip().decode("utf-8")

    return float(data)

def __delete__(self):

    self.s.close()

    return
```

C.5 Multi-sensor identification component (multi_sensor_identification.py - Python)

```
# Imports and libraries

from components.three_dimensional_object_recognition import CVFHDescriptorRecognition
from components.three_dimensional_object_recognition import SixDoFPoseEstimation
from components.multi_sensor_data_processing import SensorPlatform
from components.multi_sensor_data_processing import StructuredLightSensor
from OCC.Core.STEPControl import STEPControl_Reader
from OCC.Core.IFSelect import IFSelect_RetDone
from OCC.Core.StlAPI import StlAPI_Writer
from OCC.Core.BRepMesh import BRepMesh_IncrementalMesh
import os
import open3d as o3d
import json
import time


# Declarations of classes

class MultiSensorIdentification:

  def __init__(self):

      with open(os.getcwd().replace("\\", "/") + "/config/config.json") as json_file:
          config = json.load(json_file)

      self.ident_knowledge_base_path = os.getcwd().replace("\\", "/") +
        str(config["identification_knowledge_base_path"])
      self.cad_to_pcd_path = os.getcwd().replace("\\", "/") + str(config["cad_to_pcd_path"])
      self.scene_pcd_path = os.getcwd().replace("\\", "/") + str(config["scene_pcd_path"])

      if os.path.exists(self.ident_knowledge_base_path +
        "/identification_knowledge_base.json"):
          with open(self.ident_knowledge_base_path + "/identification_knowledge_base.json") as
json_file:
              self.identification_knowledge_base = json.load(json_file)
      else:
          self.identification_knowledge_base = []

      self.prefiltered_identification_knowledge_base = []

      self.sensor_platform = SensorPlatform()
      self.weight_tolerance = float(config["weight_tolerance"])
      self.voxel_size = float(config["voxel_size"])
      self.sample_points = int(config["sample_points"])
      self.rec = CVFHDescriptorRecognition(config)
      self.optical_sensor = StructuredLightSensor(config)
      self.pose_estimator = SixDoFPoseEstimation(config)

      return

  def __cad_to_pcd(self, cad_file_path, pcd_file_path):

      split = cad_file_path.split(".")

      if split[-1] == "stp":

          for file in os.listdir(self.cad_to_pcd_path):
              os.remove(self.cad_to_pcd_path + "/" + file)

          name = split[0].split("/")[-1]
          stl_file_path = self.cad_to_pcd_path + "/" + name + ".stl"

          step_reader = STEPControl_Reader()

          if step_reader.ReadFile(cad_file_path) == IFSelect_RetDone:
              step_reader.TransferRoot(1)
              _nbs = step_reader.NbShapes()
              shape = step_reader.Shape(1)

              mesh = BRepMesh_IncrementalMesh(shape, 0.01)
              mesh.Perform()

          stl_writer = StlAPI_Writer()
```

```
            stl_writer.SetASCIIMode(False)
            stl_writer.Write(shape, stl_file_path)

        elif split[-1] == "stl":
            stl_file_path = cad_file_path

        else:
            print("[MultiSensorIdentificaiton] CAD-Format not supported:", cad_file_path)

            return

        msh = o3d.io.read_triangle_mesh(stl_file_path)
        pcl = msh.sample_points_uniformly(self.sample_points)
        pcl.paint_uniform_color([0, 0, 0])
        pcl_down = pcl.voxel_down_sample(self.voxel_size)
        o3d.io.write_point_cloud(pcd_file_path, pcl_down, write_ascii=True)

        return

    def generate_identification_knowledge_base(self, cad_models_folder):

        print("[MULTISENSORIDENTIFICATION] Generating identification knowledge base")
        start_time = time.time()

        for file in os.listdir(self.ident_knowledge_base_path):
            os.remove(self.ident_knowledge_base_path + "/" + file)

        cad_file_paths = []
        for file in os.listdir(cad_models_folder):
            cad_file_paths.append(cad_models_folder + "/" + file)

        ident_kb = []

        for file in cad_file_paths:

            split = file.split(".")

            if split[1] not in ["stl", "stp"]:
                print("[MultiSensorIdentification] File format not compatible, will be
                 skipped:", file)
                continue

            split = split[0].split("/")[-1].split("_")

            dict = {
                "id": split[0],
                "description": split[1],
                "weight": split[2],
                "material": split[3],
                "color": split[4],
                "cad_path": file,
                "pcd_path": self.ident_knowledge_base_path + "/" + split[0] + "_pc.pcd",
                "normals_path": self.ident_knowledge_base_path + "/" + split[0] +
                 "_normals.pcd",
                "cvfh_path": self.ident_knowledge_base_path + "/" + split[0] + "_cvfh.pcd"
            }

            ident_kb.append(dict)

        self.identification_knowledge_base = ident_kb

        for model in ident_kb:
            self.__cad_to_pcd(model["cad_path"], model["pcd_path"])
            self.rec.generate_descriptor(model["pcd_path"], model["normals_path"],
                 model["cvfh_path"])

        with open(self.ident_knowledge_base_path + "/identification_knowledge_base.json", "w")
         as outfile:
            json.dump(ident_kb, outfile)

        print("[MULTISENSORIDENTIFICATION] Identification knowledge base generation took:",
          (time.time() - start_time), "seconds for ", len(ident_kb), "models")

        return
```

```
def __prefilter_identification_knowledge_base(self, weight, material, color):

    prefiltered_kb = []

    for model in self.identification_knowledge_base:

        w_low = int(model["weight"]) * (1 - self.weight_tolerance)
        w_high = int(model["weight"]) * (1 + self.weight_tolerance)

        if w_low <= weight <= w_high and model["material"] == material and model["color"] ==
             color:
            prefiltered_kb.append(model)

    if(len(prefiltered_kb) == 0):
        raise Warning("[MULTISENSORIDENTIFICATION] No matching object in identification
             knowledgebase")
    self.prefiltered_identification_knowledge_base = prefiltered_kb

def perform_identification(self):

    start_time = time.time()

    for file in os.listdir(self.scene_pcd_path):
        os.remove(self.scene_pcd_path + "/" + file)

    self.optical_sensor.get_point_cloud()

    weight = self.sensor_platform.get_weight()

    if weight == 0:
        raise Warning("[MULTISENSORIDENTIFICATION] No object on sensorplaform")

    material = self.sensor_platform.get_material()
    centre_of_mass = self.sensor_platform.get_centreofmass()

    color = self.optical_sensor.get_color()

    print("[MULTISENSORIDENTIFICATION] Weight information [g]: ", weight)
    print("[MULTISENSORIDENTIFICATION] Centre of mass location [mm]: x=", centre_of_mass[0],
      " y=", centre_of_mass[1])
    print("[MULTISENSORIDENTIFICATION] Material information: ", material)
    print("[MULTISENSORIDENTIFICATION] Colour information: ", color)

    self.rec.generate_descriptor(self.scene_pcd_path + "/scene_pc.pcd",
                                 self.scene_pcd_path + "/scene_normals.pcd",
                                 self.scene_pcd_path + "/scene_cvfh.pcd")

    self.__prefilter_identification_knowledge_base(weight, material, color)

    cvfh_paths = []

    for model in self.prefiltered_identification_knowledge_base:
        cvfh_paths.append(model["cvfh_path"])

    print("[MULTISENSORIDENTIFICATION] Candidates after prefiltering identification
      knowledge base: ", len(cvfh_paths))
    print("[MULTISENSORIDENTIFICATION] Training of recognition knowledgebase")

    self.rec.perform_training(cvfh_paths)

    print("[MULTISENSORIDENTIFICATION] Matching of descriptors")

    rec_res = self.rec.perform_matching(self.scene_pcd_path  + "/scene_cvfh.pcd")

    if rec_res[0][1] < 500:
        id = rec_res[0][0].split(".")[0].split("/")[-1].split("_")[0]
        identity_info = (next(item for item in
             self.prefiltered_identification_knowledge_base if item["id"] == id))

    else:
        raise Warning("[MULTISENSORIDENTIFICATION] Identification failed due to optical
             matching result")

    print("[MULTISENSORIDENTIFICATION] 6-DoF Pose estimation")

    trans = self.pose_estimator.get_transformation(identity_info["pcd_path"],
      self.scene_pcd_path + "/scene_pc.pcd")
    print("[MULTISENSORIDENTIFICATION] Transformation:")
```

```
print(trans)

print("[MULTISENSORIDENTIFICATION] Identity information:", identity_info)
print("[MULTISENSORIDENTIFICATION] Identification took:", (time.time() - start_time),
 "seconds")

return identity_info
```

C.6 Multi-sensor identification config file (config.json - JSON)

```json
{
    "weight_tolerance": "0.1",
    "sample_points": "100000",
    "voxel_size": "0.50",
    "normal_estimation_radius": "8.00",
    "eps_angle_threshold": "5.00",
    "curvature_threshold": "1.00",
    "normalize_bins": "True",
    "identification_knowledge_base_path":
"/temp/identification_knowledge_base",
    "recognition_knowledge_base_path": "/temp/recognition_knowledge_base",
    "cad_to_pcd_path": "/temp/cad_to_pcd",
    "scene_pcd_path": "/temp/scene_pcd",
    "scan_data_path": "/data/scan"
}
```

C.7 Multi-sensor identification main program (main.py - Python)

```python
# Imports and libraries

from components.multi_sensor_identification import MultiSensorIdentification


if __name__ == "__main__":

    msi = MultiSensorIdentification()

    if input("[MULTISENSORIDENTIFICATION] Generate identification knowledgebase?
        [Y/N]").upper() == "Y":

        msi.generate_identification_knowledge_base("C:/Users/marcu/PycharmProjects/ThesisMarc
            Ungen/data/stl")

    while True:

        input("[MULTISENSORIDENTIFICATION] Perform scanning and hit ENTER")

        try:

            identity_information = msi.perform_identification()

        except Exception as e:

            print(e)

            continue
```

Appendix D – Experimental verification

For the experimental verification of the multi-sensor Auto-ID system prototype developed in this thesis, different unpackaged piece goods are identified. In the following paragraphs, the procedure followed for the experimental verification is described on the basis of one exemplary piece good. In the following sections, the procedure for identification that was run through in the experimental review is described using an exemplary identification object.

D.1 Identification knowledge base generation (offline process)

The generation of the identification knowledge base is based on CAD-Models. Figure D.1 shows the CAD-Model of a tape dispenser, which consists of a body (blue) and a roll holder (gold). The model is exported to STEP format through the CAD application used to create it. The file name is used to encrypt identity information and is constructed according to the following scheme: "ID_NAME_WEIGHT_MATERIAL_COLOUR". The finished file must be stored in a folder specified to the multi-sensor identification software via its configuration. By calling the method "generate_identification_knowledge_base()" of the class "MultiSensorIdentification" using Python, the console output shown in Figure D.2 is obtained. The method reads all CAD files in the specified folder, converts them into a point cloud and generates a CVFH descriptor. Furthermore, a Java Script Object Notation (JSON) file is created, which contains the identity information of all known objects. The identification knowledge base is now prepared for the online identification process.

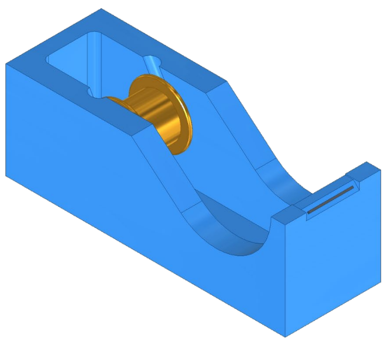


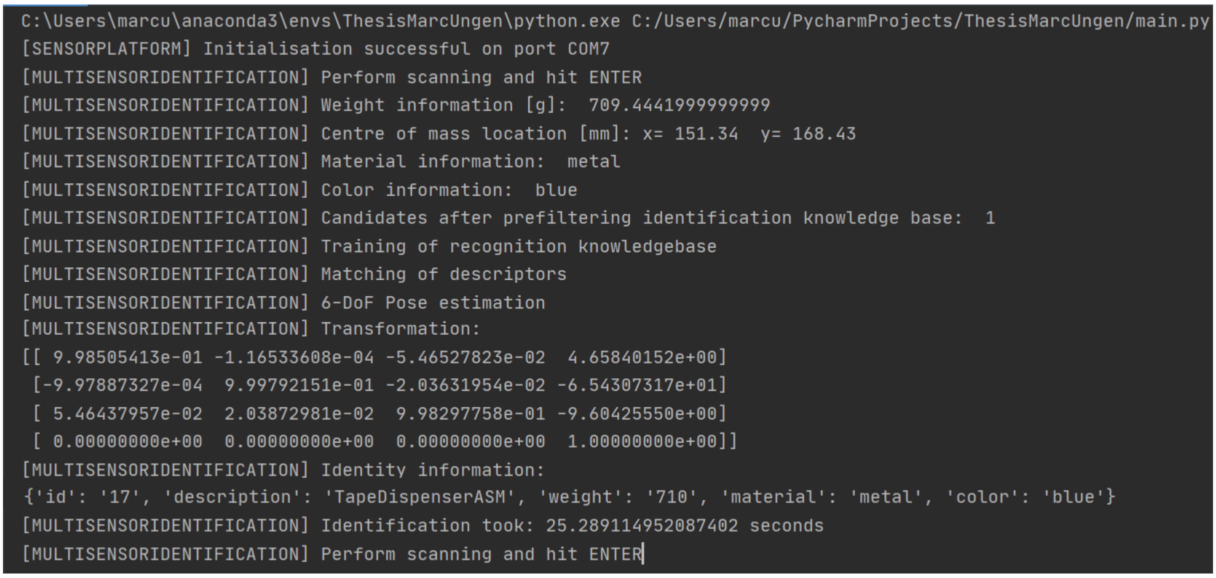
Figure D.1: Example of CAD-Model for knowledge base generation
"17_TapeDispenserASM_710_metal_blue.stp"

```
C:\Users\marcu\anaconda3\envs\ThesisMarcUngen\python.exe C:/Users/marcu/PycharmProjects/ThesisMarcUngen/main.py
[MULTISENSORIDENTIFICATION] Generating identification knowledge base
[MULTISENSORIDENTIFICATION] Identification knowledge base generation took: 81.6206579208374 seconds for  16 models
```

Figure D.2: Console output while knowledge base generation

D.2 Multi-sensor information identification (online process)

The online identification process is performed by calling the method "perform_identification()" of class "MultiSensorIdentification". The console output for the whole process after calling the method is shown in Figure D.3. First the software prompts the user to perform the 3D-Scan. The scan is acquired and post-processed using the HP 3D Scan Software Pro 5 (see Section 4.1.1.3). Once the scanner has been calibrated, the identification object is placed on the rotary device (see Figure D.4, left). By scanning and rotating several times (see Figure D.4, right), an almost complete scan of the identification object is recorded. The raw result of the scanning, containing background clutter, is shown in Figure D.5. The removal of the background clutter is done in one manual step using the tools of the HP 3D Scan Software Pro 5. Figure D.6 shows the result after the manual processing. In the last processing step the point cloud is statistically smoothed again by various post-processing algorithms, the file ready for export is shown in Figure D.7. Once the scan is complete, the enter key in the main program is pressed, which continues the identification process. The program continues by requesting the weight information, the centre of mass location and the material information from the sensor platform (see Figure D.3). Subsequently, the colour information describing the identification object is extracted from the structured-light sensor data and the point cloud is read in and described using a CVFH descriptor. In order to carry out 3D-Oject recognition, the training of the recognition knowledge base is still missing, but only the prefiltered candidates are considered, which is why the prefiltering step is carried out based on the multi-sensor information. Having extracted these candidate objects out of the identification knowledge base, the recognition knowledge base gets trained. After successful training the descriptor of the sensory point cloud is matched against all descriptors of point clouds of CAD-Models contained in the recognition knowledge base. After matching, the identification is done and the 6-DoF pose estimation is performed for validation of the centre of mass location. This step was not fully implemented for the reasons mentioned in Section 4.2.6, but the software already determines the transformation matrix (see Figure D.3) necessary to transfer the CAD point cloud into the sensor point cloud (see Figure D.8). Finally, the identity information is displayed through the console.



```
C:\Users\marcu\anaconda3\envs\ThesisMarcUngen\python.exe C:/Users/marcu/PycharmProjects/ThesisMarcUngen/main.py
[SENSORPLATFORM] Initialisation successful on port COM7
[MULTISENSORIDENTIFICATION] Perform scanning and hit ENTER
[MULTISENSORIDENTIFICATION] Weight information [g]:  709.4441999999999
[MULTISENSORIDENTIFICATION] Centre of mass location [mm]: x= 151.34  y= 168.43
[MULTISENSORIDENTIFICATION] Material information:  metal
[MULTISENSORIDENTIFICATION] Color information:  blue
[MULTISENSORIDENTIFICATION] Candidates after prefiltering identification knowledge base:  1
[MULTISENSORIDENTIFICATION] Training of recognition knowledgebase
[MULTISENSORIDENTIFICATION] Matching of descriptors
[MULTISENSORIDENTIFICATION] 6-DoF Pose estimation
[MULTISENSORIDENTIFICATION] Transformation:
[[ 9.98505413e-01 -1.16533608e-04 -5.46527823e-02  4.65840152e+00]
 [-9.97887327e-04  9.99792151e-01 -2.03631954e-02 -6.54307317e+01]
 [ 5.46437957e-02  2.03872981e-02  9.98297758e-01 -9.60425550e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
[MULTISENSORIDENTIFICATION] Identity information:
{'id': '17', 'description': 'TapeDispenserASM', 'weight': '710', 'material': 'metal', 'color': 'blue'}
[MULTISENSORIDENTIFICATION] Identification took: 25.289114952087402 seconds
[MULTISENSORIDENTIFICATION] Perform scanning and hit ENTER
```

Figure D.3: Console output for complete online identification process

a) b)

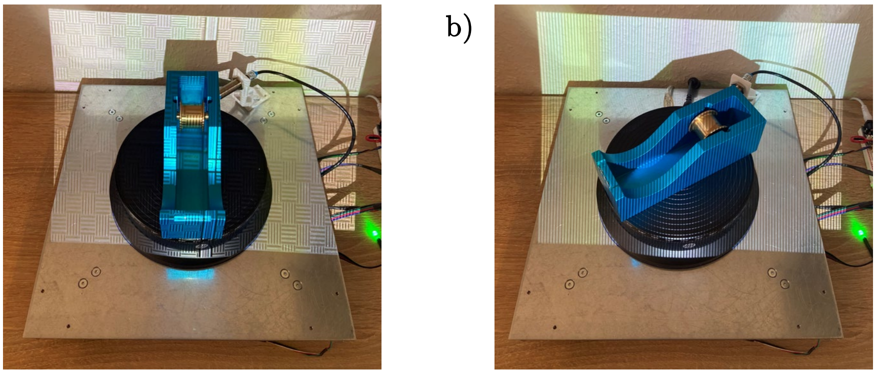


Figure D.4: Identification object settled on rotary device: a) initial pose, b) rotated while scanning

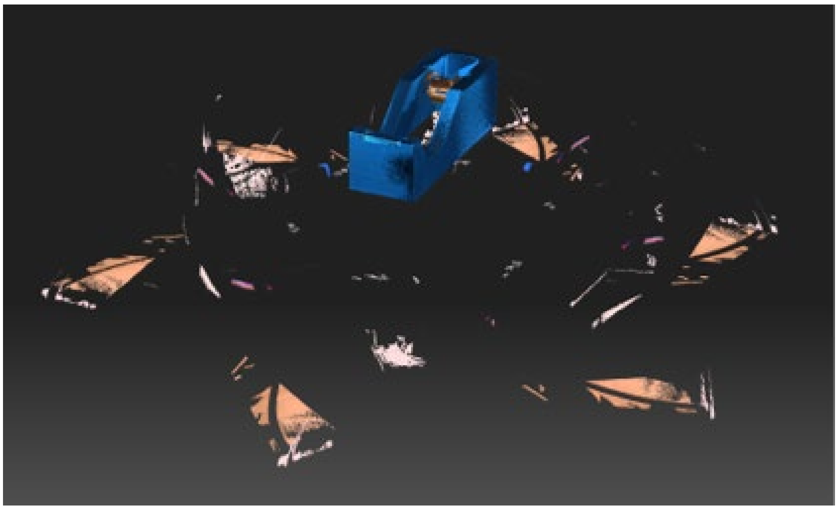


Figure D.5: Raw 3D-Scan result containing background clutter

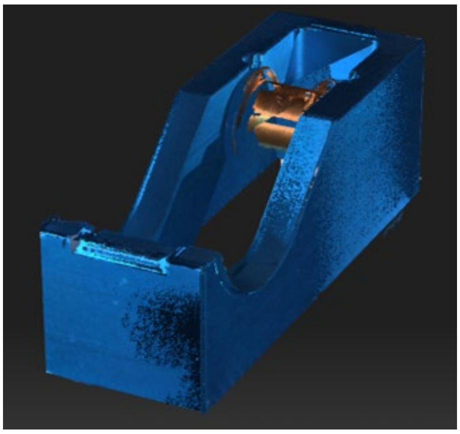


Figure D.6: Manually processed 3D-Scan

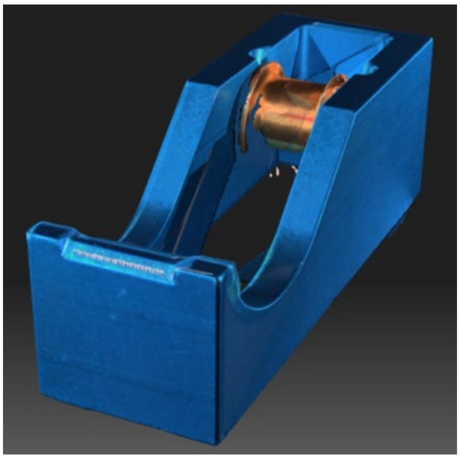


Figure D.7: Finished post-processed 3D-Scan

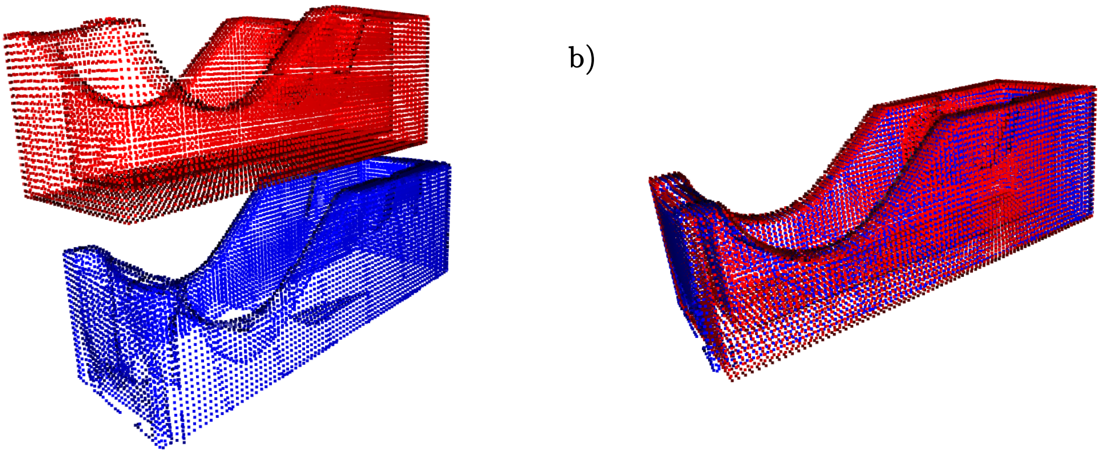


Figure D.8: 6-DoF pose estimation: a) Before transformation b) After transformation - CAD point cloud (red) and sensor point cloud (blue)

D.3 Experiment protocol

Experiment protocol
PC: Surface Pro 4 I5, 8GB RAM
Weight Tolerance: +-10% from CAD

ID	Description	Identification 1 values					
		Weight [g]	Material	Colour	Prefiltered	Correct	Time [min:s]
1	SchwipSchwapFull	355,39	metal	red	2	Yes	03:06
2	PepsiFull	364,65	metal	blue	1	Yes	03:03
3	LiptonFull	350,03	metal	yellow	1	Yes	03:02
4	RedBullWatermelonFull	275,42	metal	red	1	Yes	03:15
5	RedBullBlueberryFull	274,95	metal	blue	1	Yes	03:12
6	FantaFull	363,66	metal	orange	1	Yes	03:16
7	SpriteFull	365,5	metal	green	1	Yes	03:08
8	CocaColaFull	362,52	metal	red	2	Yes	03:12
9	SchwipSchwapEmpty	26,33	metal	red	2	Yes	03:11
10	PepsiEmpty	24,61	metal	blue	1	Yes	03:00
11	LiptonEmpty	25,15	metal	yellow	1	Yes	03:14
12	RedBullWatermelonEmpty	15,71	metal	red	1	Yes	03:00
13	RedBullBlueberryEmpty	15,41	metal	blue	1	Yes	03:15
14	FantaEmpty	24,6	metal	orange	1	Yes	03:10
15	SpriteEmpty	24,99	metal	green	1	Yes	03:03
16	CocaColaEmpty	24,18	metal	red	2	Yes	03:10
17	TapeDispenserASM	713,4	metal	blue	1	Yes	03:12
18	TapeDispenserBody	587,94	metal	blue	1	Yes	03:11
19	PressureControlValve	580,83	metal	grey	1	Yes	03:24
20	ToolboxFull	539,79	non-metal	grey	1	Yes	02:44
21	ToolboxEmpty	156,89	non-metal	grey	1	Yes	02:54

ID	Description	Identification 2 values					
		Weight [g]	Material	Colour	Prefiltered	Correct	Time [min:s]
1	SchwipSchwapFull	356,94	metal	red	2	Yes	03:13
2	PepsiFull	366,67	metal	blue	1	Yes	03:04
3	LiptonFull	349,85	metal	yellow	1	Yes	03:05
4	RedBullWatermelonFull	276,61	metal	red	1	Yes	03:00
5	RedBullBlueberryFull	272,97	metal	blue	1	Yes	03:08
6	FantaFull	362,53	metal	orange	1	Yes	03:10
7	SpriteFull	364,17	metal	green	1	Yes	03:11
8	CocaColaFull	360,67	metal	red	2	Yes	03:16
9	SchwipSchwapEmpty	24,87	metal	red	2	Yes	03:06
10	PepsiEmpty	24,76	metal	blue	1	Yes	03:02
11	LiptonEmpty	24,03	metal	yellow	1	Yes	03:11
12	RedBullWatermelonEmpty	16,8	metal	red	1	Yes	03:13
13	RedBullBlueberryEmpty	14,22	metal	blue	1	Yes	03:12
14	FantaEmpty	25,91	metal	orange	1	Yes	03:06
15	SpriteEmpty	25,25	metal	green	1	Yes	03:14
16	CocaColaEmpty	26,67	metal	red	2	Yes	03:04
17	TapeDispenserASM	714,74	metal	blue	1	Yes	03:12
18	TapeDispenserBody	587,23	metal	blue	1	Yes	03:17
19	PressureControlValve	581,08	metal	grey	1	Yes	03:19
20	ToolboxFull	540,05	non-metal	grey	1	Yes	02:40
21	ToolboxEmpty	156,31	non-metal	grey	1	Yes	02:47

Marc Ungen
16.10.2020

Experiment protocol
PC: Surface Pro 4 I5, 8GB RAM
Weight Tolerance: +-10% from CAD

ID	Description	Identification 3 values					
		Weight [g]	Material	Color	Prefiltered	Correct	Time [min:s]
1	SchwipSchwapFull	355,65	metal	red	2	Yes	03:16
2	PepsiFull	364,83	metal	blue	1	Yes	03:05
3	LiptonFull	350,87	metal	yellow	1	Yes	03:15
4	RedBullWatermelonFull	275,29	metal	red	1	Yes	03:10
5	RedBullBlueberryFull	274,07	metal	blue	1	Yes	03:13
6	FantaFull	364,03	metal	orange	1	Yes	03:15
7	SpriteFull	366,32	metal	green	1	Yes	03:12
8	CocaColaFull	362,57	metal	red	2	Yes	03:02
9	SchwipSchwapEmpty	24,73	metal	red	2	Yes	03:13
10	PepsiEmpty	25,72	metal	blue	1	Yes	03:01
11	LiptonEmpty	24,2	metal	yellow	1	Yes	03:02
12	RedBullWatermelonEmpty	16,69	metal	red	1	Yes	03:16
13	RedBullBlueberryEmpty	16,43	metal	blue	1	Yes	03:15
14	FantaEmpty	25,3	metal	orange	1	Yes	03:03
15	SpriteEmpty	26,78	metal	green	1	Yes	03:14
16	CocaColaEmpty	24,66	metal	red	2	Yes	03:14
17	TapeDispenserASM	713,33	metal	blue	1	Yes	03:14
18	TapeDispenserBody	586,21	metal	blue	1	Yes	03:11
19	PressureControlValve	580,71	metal	grey	1	Yes	03:23
20	ToolboxFull	539,72	non-metal	grey	1	Yes	02:50
21	ToolboxEmpty	156,47	non-metal	grey	1	Yes	02:44

Marc Ungen
16.10.2020