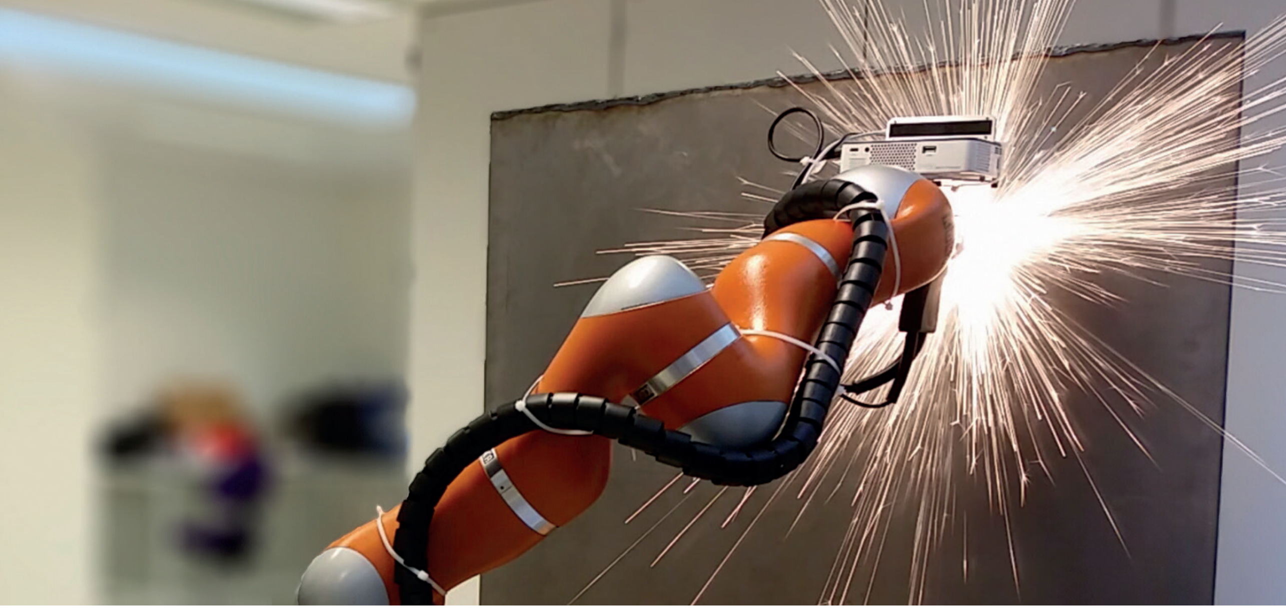
Facilitating Programming of Vision-Equipped Robots through Robotic Skills and Projection Mapping

Andersen, Rasmus Skovgaard

DOI (link to publication from Publisher):
10.5278/vbn.phd.engsci.00062

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record



FACILITATING PROGRAMMING OF VISION-EQUIPPED ROBOTS THROUGH ROBOTIC SKILLS AND PROJECTION MAPPING

BY
RASMUS SKOVGAARD ANDERSEN

DISSERTATION SUBMITTED 2016

AALBORG UNIVERSITY
DENMARK

Facilitating Programming of Vision-Equipped Robots through Robotic Skills and Projection Mapping

PhD Dissertation

Rasmus Skovgaard Andersen

Dissertation submitted January 29, 2016

Dissertation submitted: January 29, 2016

PhD supervisor: Prof. Ole Madsen
Aalborg University

Assistant PhD supervisor: Prof. Thomas B. Moeslund
Aalborg University

PhD committee: Associate Professor Morten Kristiansen (chairman)
Aalborg University, Denmark

Associate Professor Markus Vincze
Vienna University of Technology, Austria

Professor Henrik G. Petersen
University of Southern Denmark, Denmark

PhD Series: Faculty of Engineering and Science, Aalborg University

ISSN (online): 2246-1248
ISBN (online): 978-87-7112-498-9

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2016

Curriculum Vitae

Rasmus Skovgaard Andersen

Rasmus S. Andersen received his M.Sc. in Engineering (Informatics with specialization in Vision, Graphics and Interactive Systems) from Aalborg University, Denmark in 2010. He worked with teaching at Aalborg Studenterkursus and as a programmer at Lyngsoe Systems until 2012, where he started his PhD study in November 2012 with the Robotics and Automation group at the Department of Mechanical and Manufacturing Engineering, Aalborg University. During the PhD studies, he had a four month research stay at the Institute for Robotics and Intelligent Machines (IRIM) at the Georgia Institute of Technology in Atlanta, Georgia, where he was invited by Henrik I. Christensen. He has been involved in lecturing and supervision of undergraduate and graduate students in topics of robotics and machine vision. His main research interests lie within computer vision for robotics and human-robot interaction for industrial, collaborative robots.

Abstract

The field of collaborative industrial robots is currently developing fast both in the industry and in the scientific community. Where industrial robots traditionally have been placed behind security fences and programmed to perform simple, repetitive tasks, this next generation of robots will be able to work side-by-side with humans and collaborate on completing common tasks. This poses requirements to the robots within sensing, intelligence, safety, and human-robot interfaces.

The focus of this thesis is on smart interfaces for such collaborative, industrial robots with advanced sensing capabilities. The work is divided into two areas: Robot vision for robotic skills and projection mapping interfaces.

Robotic skills are a basis of task-oriented programming. The goal of the skill concept is to allow humans to instruct new tasks while focusing his or her attention towards the task at hand rather than on the robot's capabilities. In this thesis, it is investigated how a skill based architecture can incorporate advanced robot vision capabilities while keeping the robot programming fast and intuitive. A number of skills are developed for object detection, quality control, etc., and the skills are tested both in laboratories and industrial settings.

Projection mapping is the technique to project information into the real world. In contrast to projecting from a static projector onto a static screen, projection mapping warps projected graphics in such a way that it is displayed geometrically correct in modeled environments, including on walls, ceilings, objects, etc. It is in this thesis investigated how projection mapping can be applied as part of human-robot interfaces to simplify and improve human-robot interaction in scenarios involving robot programming as well as human-robot cooperation. By projecting text and graphics into the working area, or *task space*, humans have access to the relevant information when and where it is needed. In both programming and cooperation scenarios, projection mapping is used to project the state and intentions of cooperative robots onto both environments and detected and tracked objects. Through a number of user studies it is shown that it is possible to use projection mapping in these scenarios for seamless human-robot interaction, and that it in some cases provide significant advantages when compared to traditional monitor based interfaces.

Resumé

I disse år er samarbejdende industrirobotter i hastig udvikling såvel i industrien som i forskningsverdenen. Mens industrirobotter traditionelt skal placeres bag sikkerhedshegn og typisk programmeres til at udføre simple, repetitive opgaver, så vil den næste generation af robotter være i stand til at arbejde side om side med mennesker og samarbejde om fælles opgaver. Dette stiller krav til robotterne indenfor områderne sansning, intelligens, sikkerhed, og menneske-robot grænseflader.

Fokus i denne afhandling er på smarte grænseflader for sådanne samarbejdende industrirobotter med avancerede sansningsevner. Afhandlingen er opdelt i to områder: Robot vision til *robot-skills* og grænseflader baseret på projection mapping.

Robot-skills er en basis for task-orienteret programmering. Målet med skill-konceptet er at gøre det muligt for mennesker at instruere nye opgaver ved at fokusere på den aktuelle opgave i modsætning til robottens evner og funktioner. I denne afhandling undersøges det, hvordan en skill-baseret arkitektur kan inkorporere avancerede funktioner indenfor robot vision, og på samme tid holde fast i, at skill-baseret robotprogrammering skal være hurtig og intuitiv. Et antal skills udvikles, som inkorporerer detektion af objekter, kvalitetskontrol, mm., og disse skills testes i såvel laboratorie- som industriomgivelser.

Projektion mapping er teknikken at projektere information ind i den virkelige verden. I modsætning til projektering fra en statisk projektor op på et statisk lærred, så forvrænger man med projection mapping den projekterede grafik på en sådan måde, at den bliver vist geometrisk korrekt på modellerede omgivelser, inklusiv vægge, lofter, objekter, osv. Det undersøges i denne afhandling, hvordan projection mapping kan anvendes som et delelement i menneske-robot grænseflader til at simplificere og forbedre menneske-robot interaktionen i situationer der involverer robotprogrammering og menneske-robot samarbejde. Ved at projektere tekst og grafik ind i arbejdsområdet, også kaldet *task space*, kan mennesker få adgang til relevant information når og hvor det er påkrævet. I såvel programmerings- som samarbejdssituationer, anvendes projection mapping til at projektere en samarbejdende robots tilstand og intentioner på både statiske omgivelser og detekterede og trackede objekter. Gennem et antal brugerstudier vises det, at det er muligt at anvende projection mapping i sådanne situationer til ukompliceret menneske-robot interaktion, samt at teknikken i visse tilfælde giver signifikante fordele, når den sammenlignes med traditionelle skærmbaserede grænseflader.

Contents

Thesis Details

Thesis Title: Facilitating Programming of Vision-Equipped Robots through Robotic Skills and Projection Mapping
PhD Student: Rasmus Skovgaard Andersen
Supervisors: Prof. Ole Madsen, Aalborg University
Prof. Thomas B. Moeslund, Aalborg University

The main body of this thesis consists of the following papers and technical reports:

A Ole Madsen, Simon Bøgh, Casper Schou, Rasmus S. Andersen, Jens S. Damgaard, Mikkel R. Pedersen, and Volker Krüger, "Integration of Mobile Manipulators in an Industrial Production," *Industrial Robot: An International Journal*, vol. 42, No. 1, pp. 11-18, 2015.

B Rasmus S. Andersen, Casper Schou, Jens S. Damgaard, Ole Madsen, and Thomas B. Moeslund, "Human Assisted Computer Vision on Industrial Mobile Robots," *Proceedings of the 1'st AAU Workshop on Human-Centered Robotics*, pp. 15–21, AAU, 2013.

C Rasmus S. Andersen, Jens S. Damgaard, Ole Madsen, and Thomas B. Moeslund, "Fast calibration of industrial mobile robots to workstations using QR Codes," *Proceedings of the 44'th International Symposium on Robotics (ISR)*, pp. 257–262, 2013. Won the *Best Paper* award.

D Rasmus S. Andersen, Lazaros Nalpantidis, Volker Krüger, Ole Madsen, and Thomas B. Moeslund, "Using Robot Skills for Flexible Reprogramming of Pick Operations in Industrial Scenarios," *Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, pp. 678–685, 2014.

E Rasmus S. Andersen, Ole Madsen, and Thomas B. Moeslund, "Adaptive Model Based Quality Inspection", technical report synthesized from deliverable 3.8 in the TAPAS project.

F Rasmus S. Andersen, Ole Madsen, and Thomas B. Moeslund, "Hand-eye Calibration of Depth Cameras based on Planar Surfaces," extended abstract presented at the 1'st International Workshop on Intelligent Robot

Assistants (IRAS) at the 13'th International Conference on Intelligent Autonomous Systems (IAS), 2014.

G Rasmus S. Andersen, Thomas B. Moeslund, Ole Madsen, and Heni B. Amor, "Projecting Robot Intentions into Human Environments," submitted for publication at RO-MAN 2016.

H Rasmus S. Andersen, Simon Bøgh, Thomas B. Moeslund, and Ole Madsen, "Task Space HRI for Cooperative Mobile Robots in Fit-out Operations inside Ship Superstructures," submitted for publication at ICRA 2016.

I Rasmus S. Andersen, "Teaching Robotic Skills by Projecting into Task Space", technical report intended for later publication.

In addition to the main papers, the following publications have also been made during the PhD study:

[Rovida et al., 2014] Fransesco Rovida, Dimitris Chrysostomou, Casper Schou, Simon Bøgh, Ole Madsen, Volker Krüger, Rasmus S. Andersen, Mikkel R. Pedersen, Bjarne Grossmann, and Jens S. Damgaard, "SkiROS: A Four Tiered Architecture for Task-Level Programming of Industrial Mobile Manipulators," extended abstract presented at the 1'st International Workshop on Intelligent Robot Assistants (IRAS) at the 13'th International Conference on Intelligent Autonomous Systems (IAS), 2014.

[Pedersen et al., 2015] Mikkel R. Pedersen, Lazaros Nalpantidis, Rasmus S. Andersen, Casper Schou, Simon Bøgh, Volker Krüger, and Ole Madsen, "Robot Skills for Manufacturing: From Concept to Industrial Deployment," *Robotics and Computer-Integrated Manufacturing*, 2014.

[Andersen et al., 2015] Rasmus S. Andersen, Simon Bøgh, Thomas B. Moeslund, and Ole Madsen, "Intuitive Task Programming of Stud Welding Robots for Ship Construction," *IEEE International Conference on Industrial Technology (ICIT)*, pp. 3302–3307, 2015.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers as well as technical reports which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This thesis is submitted as a collection of papers to the Faculty of Engineering and Science at Aalborg University in partial fulfillment of the requirements for the degree of Doctor of Philosophy. The thesis represents a culmination of work and learning that has taken place at the department of Mechanical and Manufacturing Engineering, Aalborg University during the period from August 2012 to December 2015. The thesis is conducted on the basis of scientific papers and under supervision of Prof. Ole Madsen and Prof. Thomas B. Moeslund.

The thesis contains studies within robotics, robot vision, and human-robot interaction. The purpose has been to explore and develop ways of interacting with and programming of collaborative, industrial robots; and particularly to enable non-experts in robotics to program robots to solve tasks which require robot vision.

The PhD project has been partly funded by the European Union under the Seventh Framework Programme projects TAPAS (260026) and CARLoS (606363).

Acknowledgements

During this PhD study, I have been cooperating with many researchers and practitioners, and I would like to take this opportunity to thank everyone who in one way or the other has shared this journey with me.

During the first part of the PhD study I worked closely with Christof Ebers and Ricardo Tornese from the private company CIT; particularly during the two weeks I stayed at the company's office in Haid, Austria. I would like to thank both of you for rewarding discussions and experiences and for an open company culture which made cooperation easy and enjoyable.

Later in the PhD study I spent four Months at the Institute for Robotics and Intelligent Machines (IRIM) at the Georgia Institute of Technology in Atlanta, Georgia, under the supervision of Prof. Henrik I. Christensen and Dr. Heni B. Amor. I would like to thank both of you for rewarding discussions and for giving me the opportunity to be part of such an inspiring research environment. Also, I would like to thank everybody at the IRIM lab for welcoming me with open arms.

I would also like to thank my closest colleagues in the Robotics and Automation group at Aalborg University. Such an open, encouraging, and positive

working environment will be difficult to find in most places. Particularly thanks to Casper Schou, Jens S. Damgaard, and Simon Bøgh for always positive attitudes. Without you the work presented in this thesis would not have been possible.

Finally, I would like to thank my supervisors, Ole Madsen and Thomas B. Moeslund, for your enthusiastic and patient supervision throughout the PhD project. Especially I have appreciated the unorthodox collaboration between different departments at Aalborg University that you made happen for this project.

Rasmus S. Andersen
Aalborg University, February 1, 2016

Part I

Introduction

Chapter 1

Introduction

1.1 Project Motivation

Globalization has for several decades moved manufacturing jobs from western countries to low-wage developing countries. This has put pressure on both wages and productivity in the manufacturing sectors in the industrialized countries. One efficient way of increasing productivity is to increase the level of automation. A major limitation for increased automation is, however, the scale of production. Construction of automated production lines are major investments, and configuration of robots to perform the required operations is a time-consuming task which must be performed by highly specialized technicians. Thus, installation of new and fully automated production lines can only be justified if the quantities of identical items to be produced are very large. Automation has therefore proven to be particularly useful in industries that produce large numbers of almost identical products. One of the most prominent of these industries is car manufacturing, which has been on the forefront of automation since the first industrial robot, Unimate, was introduced by General Motors in 1961. Today, car manufacturing plants are heavily automated and employs a large number of robots, as illustrated in Figure 1.1.

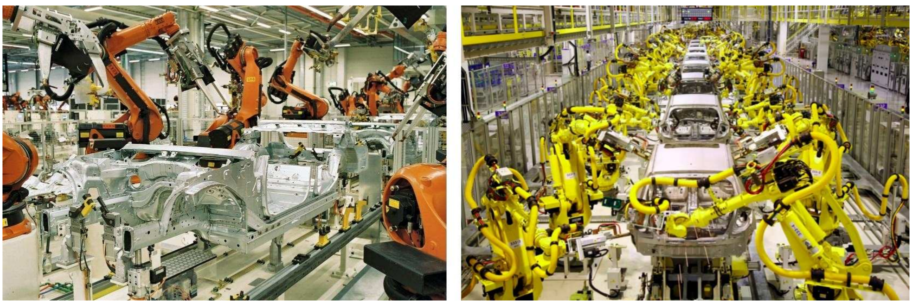


Fig. 1.1: Traditional industrial robots work at fully automated production lines. Human workers must not enter security zones around the robots during production. The left-most picture is from a BMW plant in Leipzig and the right-most is from a KIA Motors plant in Slovakia.

1.1.1 Challenges for Industrial Robots

Car manufacturing is well suited for automation using industrial robots because cars are traditionally produced in large quantities with only minor variations. For many kinds of production, the number of identical items is however not large enough to justify investment in automated robotic production lines. Much research has therefore been directed towards developing more flexible types of automation. In Europe, several *Strategic Research Agenda for European Robotics* have in recent years been laid forth in an attempt to preempt future developments and focus research and development in the most promising directions [EUROP, 2009, SPARK, 2014]. The latest SRA was made in 2014 by *SPARK*, which is a civilian robotics programme funded by the European Union. The SRA outlines manufacturing as one of the core application scenarios: "*Mid-to-low volume manufacturing requires low installation and running costs and a high degree of flexibility, which cannot typically be provided by traditional large scale manufacturing robotics. ... The merging of smart technologies, trainable systems, and intuitive user interfaces with compliant robot manipulators creates an opportunity to make a range of smart manufacturing robots that will enable automation in small and mid scale companies...*"

The SRA further lists a set of abilities, which are relevant for robots in the near future, including:

Configurability: "*The ability of the robot to be configured to perform a task or reconfigured to perform different tasks...*" It must be possible to (re-)configure robots fast and easy to prevent expensive idle time for long periods between production of (possibly small) production series.

Interaction Ability: "*The ability of the system to interact both cognitively and physically either with users, operators or other systems around it...*" Robots must be able to interact with other machines as well as humans. The communication between robots and human operators must be intuitive and to an increasing degree use modalities and interaction types that are natural to humans.

Perception Ability: "*The ability of the robot to perceive its environment*," including "*detecting objects, spaces, locations or items of interest in the vicinity of the system...*" Perception is a key element element in robotics and is a requirement for solving non-trivial tasks, including interacting with both machines and humans. Perception can include all senses, and among the most used is vision and force sensing. The SRA states that "what sets robots apart from other types of machines is their ability to sense their environment."

The goal of the European SRA's to develop robots that make production more flexible and at the same time highly efficient is visualized in Figure 1.2.

The figure shows the degree of efficiency versus flexibility in production. The goal can be approached in two different ways; by equipping human workers with better tools or by increasing the flexibility of automated production. These approaches end up in a similar vision for future production. If robots can reach a sufficient level within configuration, human-robot interaction, and autonomy, then they can both be regarded as flexible automation as well as highly advanced tools for human workers.

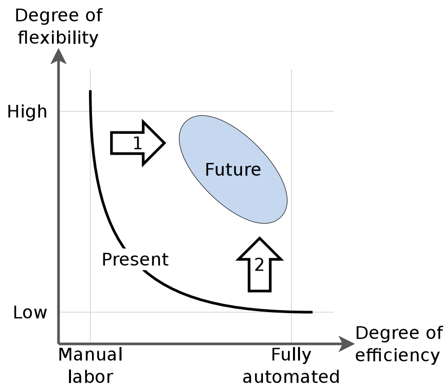


Fig. 1.2: The production of the future needs to be both very flexible to support smaller production series with high variability and efficient to keep the cost low. This goal can be formulated in two distinct ways: Workers can either be equipped with better tools; thereby increasing their productivity (1), or automated production lines can be made easier and faster to reconfigure (2).

1.1.2 Collaborative Robots for Future Production

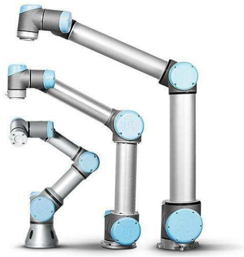
The robotics industry has within the last 5-7 years moved fast towards developing robots that possess more autonomy, provides better human-robot interaction and are easier to reconfigure. This development has spawned a whole new class of industrial robots, commonly know as *collaborative robots*. On the topic on human-robot interaction for collaborative robots, the European SRA by SPARK sets the following 2020-goals [SPARK, 2014]:

Human Machine Interfaces: "*To develop instructable interfaces. To develop physically interactive interfaces for collaborative working...*"

Human Robot Collaboration: "*To develop low cost safe dependable systems able to react and interact with people... To develop multi-modal collaboration.*"

Safety: "*To develop robust safety based design processes including inherent physical robot safety...*"

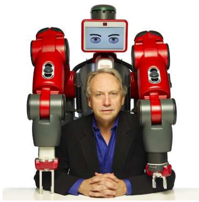
Some of the collaborative robots released to the market within the last few years are listed in Figure 1.3.



(a) UR 3-5-10 (2009-2015)

(b) KUKA iiwa (2013)

(c) Kawada Nextage (2009)



(d) Rethink Robotics Baxter (2012)

(e) Kinova Mico (2015)

(f) ABB YuMi (2015)

(g) KUKA KMRiiwa (2015)

Fig. 1.3: Collaborative robots on the market today.

There exist different types of collaborative robots, but there is at this point no universal consensus on how to classify them. The following classification is inspired and expanded from [ROBOTIQ, 2015]:

Force limited robots: Robots with simple interfaces and limited functionality, designed to be usable by non-experts. They distinguish themselves from traditional industrial robots by having the ability to sense and react to external forces. This allows them to be able to function safely in a workspace that is shared with human workers. The force limited robots are especially intended for small and mid-sized companies (SMEs) which have not previously been automating production because quantities have been too small to make traditional automation economically viable. To this group belongs all robots from the Danish robot manufacturer Universal Robots, the Kinova Mico, and the KUKA LWR and KUKA iiwa robots.

Cobots - sensor-equipped robots: These robots are, in addition to being force limited and therefore safe to work nearby, equipped with a multitude of sensors such as cameras and proximity sensors. The sensors are integrated directly in a user interface which allows an operator to program new robot tasks without first modifying hardware and software.

The sensor-equipped robots are both intended SMEs but also in some cases larger companies. SMEs use the robots to solve relatively advanced tasks without having to hire expensive robot integrators, while larger companies can use them for tasks which could previously not easily be automated. To this group of robots belong Rethink Robotics' Baxter, Kawada's Nextage and ABB's YuMi.

AIMMs: AIMM is short for Autonomous Industrial Mobile Manipulator. As the name suggests, an AIMM includes the ability to move and navigate autonomously in addition to the abilities of the other robot classes. Autonomous navigation adds a whole new layer of flexibility to the robots and potential tasks they can carry out. There have been much research on AIMMs starting in the 80'ths with MORO from Fraunhofer [Schuler, 1987] all the way up to present time [Hvilshøj et al., 2012], including Aalborg University's Little Helper series [Hvilshøj and Bøgh, 2011]. AIMMs have, however, not yet matured enough to be significantly represented on the market. Pure logistic robots which can navigate around humans in semi-structured environments do exists on the market, and examples include KIVA robots, which are used in Amazon warehouses, and the Danish startup MIR. Among AIMMs, which combines mobility and manipulation, KUKA perhaps made the to date most serious attempt at breaking into the market with their KMRiiawa from 2015 (see Figure 1.3(g)).

Common for all types of collaborative robots is that they are very new on the market. They are still trying to find their place on different markets, and much research and development is going into improving their interfaces, programming architectures, sensors and capabilities in general.

1.1.3 Task Level Programming and the Little Helper Project

At Aalborg University (AAU), research in collaborative industrial robots has especially been centered around the *Little Helper* project. This project was initiated in 2007 at the Department of Mechanical and Manufacturing Engineering at AAU, and it has since then been further developed through both national and European research projects, including TAPAS, ACAT, CARLoS, and CARMEN [TAPAS, 2014, ACAT, 2015, CARLoS, 2015]. The goal of the Little Helpers has since the beginning been to conceptualize and build flexible mobile robots, AIMMs, for industrial use [Hvilshøj et al., 2009]. The idea is to combine off-the-shelf building blocks including a mobile platform, a robot manipulator, one or more tools, and one or more cameras used for vision. The entire "family" of Little Helper robots is shown in Figure 1.4. All of the robots, except for Little Helper 1, is still used for robotics research today.

(a) LH1, 2008 (b) LH2, 2010 (c) LH3, 2012 (d) LH4, 2013 (e) Carmen platform, 2014

Fig. 1.4: The Little Helper family developed at Aalborg University.

An important focus with the Little Helper projects is to make the robots flexible; both with regards to being able to handle a large variety of tasks and also with regards to making it fast and easy to program the robots to perform new tasks [Hvilshøj and Bøgh, 2011]. To achieve this, a new programming paradigm has been developed based *robotic skills* [Bøgh et al., 2012b, Pedersen et al., 2015]. The programming paradigm is extensively described in Chapter 4. Briefly, a robotic skill describes and wraps the capabilities of a robot in a generic way and it can be parameterized to perform different tasks. Each skill has two distinct parts; a *teaching* part and an *execution* part. The teaching part makes it possible for an operator to specify all parameters of the skill which is then required to execute it subsequently. We denote programming with skills *task-level programming* because is can enable an operator to setup and program new tasks while focusing his attention on the task itself and the objects that need to be manipulated. Robot-specific parameters are hidden from the operator and set indirectly through the skills.

1.2 Initiating Problems

At the start of this PhD project, the skill based framework for the Little Helper robots were non-modular, hardware specific, and supported only relatively simple skills such as pick-from-location, place-onto, and place-into. Some of these skills relied on force-torque measurements, but no additional sensing and vision technologies were used. The initiating research problem of PhD project was formulated on this basis:

Initiating research problem
How can increasingly complex collaborative robots using advanced vision technologies be programmed fast, efficiently, and in a controllable manner?

This problem leads to the formulation of two main research objectives. The first main objective concerns vision-enabled skills primarily for object manipulation:

Main objective 1: Vision in a User Centered Skill Framework
Investigate if and how a skill-based framework can assist factory workers without expertise in robotics and computer vision to teach collaborative robots to recognize, pose estimate, grasp, and quality check objects, after only very limited training.

The second main objective concerns interaction with collaborative robots in general, including manual teaching of skills. When humans interact and collaborate with collaborative robots, their attention is mostly directed towards the robot and the task at hand. It is in many situations, however, still necessary to look at external interaction devices such as monitors or teach pendants to figure out the exact state and intentions of the robot. This is, for instance, the case during manual teaching of skills, where the operator has to read teaching instructions. It is hypothesized that the usability can be increased if information is instead projected directly into the working space; the *task space*:

Main objective 2: Projection Based Interfaces for Task Space Interaction
Investigate if and how the usability of interaction with collaborative robots can be increased by providing feedback to the operator directly in task space, when compared to traditional feedback methods (text sheets, teach pendants/tablets, and monitors).

State of the art on the areas of the two main objectives are presented in the following chapter. On this basis precise and concrete research objectives are stated in Section 3.1.

1.3 Reader's Guide

This thesis is a collection of papers. It consists of three parts. Part I is an introduction to the thesis with a general motivation, presentation of main research objectives. As part of the introduction, state-of-the-art related research is presented in Chapter 2. After this, the main research objectives are in Chapter 3 decomposed into a number of more specific objectives.

Part II is a summary report that presents the main results of the thesis. It starts in Chapter 4 with a general introduction to skills, task level programming, and the skill based system that has been developed and used for experiments throughout the duration of this PhD project. One paper is summarized in this chapter. Chapter 5 and 6 present the main results of the thesis; each covering one of the main research objectives. They summarize all remaining papers and technical reports that are part of the thesis. Chapter 7 briefly lists the main contributions of the thesis and recommends future work. The remaining part of the thesis, Part III, consists of the full papers and technical reports. Some of the papers have previously been published while others have been submitted shortly before this thesis was handed in.

Chapter 2

State of the Art

State-of-the-art research is presented in two related fields defined by the main research objectives. First in Section 2.1, research in robotic skills, their origins from related architectures, and their uses are reviewed. This relates to main objective 1. Then, human-robot interaction in *task space* applying projection mapping is reviewed in Section 2.2. This relates to main objective 2.

2.1 Robot Architectures and Skills

Related research within robotic skills is presented with focus on skills for industrial use. A background on programming architectures and how skill based programming relates to the alternatives is first provided, and on this background, current research in skills and their uses is presented.

2.1.1 Robot Control System Architectures

A traditional and still widely used way of programming robots is the *Sense-Plan-Act* (SPA) architecture [Gat, 1998, Nilsson, 1993]. Using this, the state of the robot changes between sense, plan, and act. In the sensing state, information from sensors is used to update and maintain a world model. In the planning state, high-level logic plans on basis of the world model what the robot has to do to achieve a specified goal. In the acting state, the plan is carried out, typically using control loops. The SPA architecture has, however, two fundamental limitations: Firstly, planning based on a detailed world is a very hard problem which in many cases will also be very time consuming. Secondly, the world may change during the planning process in a way that invalidates the resulting plan. Researchers will recognize this problem as the *running scientist syndrome*, where it becomes necessary to quickly stop the robot after one step in a larger plan fails.

The shortcomings of the SPA architecture can be addressed through an approach which is sometimes denoted *reactive planning* [Gat, 1998]. In reactive planning, part of the decision-making process is carried out on a lower level closer to real-time; thereby reducing the risk of acting based on obsolete information. The most well known of these approaches may be Rodney Brooks' *subsumption* architecture [Brooks, 1986]. In this approach, no time consuming upper planning layer exists. Instead, inputs are connected to outputs through a network on finite state machines, allowing the robot to act and react very fast on new sensor inputs.

Subsumption is well suited for relatively simple tasks such as navigating collision-free in a closed area, but has problems for more complicated tasks because it is not very modular: Different parts of the system are heavily interconnected, and changes to one subsystem may require other subsystems to be completely redesigned [Hartley and Pipitone, 1991]. In the 90'th, several researchers proposed three layered structures where all layers are executing in parallel [Bonasso, 1991, Connell, 1992, Gat, 1998]. In these approaches, the lowest level controls the behavior of the robot through real-time control loops. The behavior(s) is changed according to a plan or reactive in response to sensor inputs. The highest level contains functionality related to the overall task which can be executed without (hard) time constraints. Most notably the level includes a (slow) planner. The planner both makes initial plans and replans when the lower levels encounter situations which cannot be handled in real-time. The middle layer, sometimes denoted the *Sequencer*, links the other layers by continuously changing the behavior of the lower level both according to expected changes in the world and internal states and in respond to unexpected contingencies.

2.1.2 Skill Based Programming and Planning

Skill based paradigms apply a three layered architecture similar to Gat et. al. and attempt to wrap robot functionality inside skills with clearly defined purposes and interfaces. This serves two purposes: To facilitate modularization such that functionality from each layer can be reused across different robots and in different contexts, and to ease the job of planning; autonomous or from human programmers/instructors. Different definitions and naming conventions exist throughout literature, but in this thesis, the layers will be referred to as *device primitives*, *skills*, and *tasks*. A skill is a preprogrammed combination of device primitives and it requires typically a number of parameters to be

A skill-like concept was originally proposed in 1972 as building blocks for automatic planning of tasks using their now famous STRIPS planner [Fikes and Nilsson, 1972]. The STRIPS planner makes it possible to automatically combine a series of predefined actions (or skills) with formalized interfaces in order to get from a well-known initial state to a goal state. More recently, different researchers have used full skill architectures for autonomous planning by mapping individual skills to actions defined in *Planning Domain Definition Language* (PDDL) [Huckaby et al., 2013, Pedersen and Krüger, 2015]. In [Huckaby et al., 2013], skills are combined using *SysML*; an extension of the well-known modeling language UML. Skills are defined as *atomic operations* of a robot such as pick-up, drill, detect, transport, etc. The STRIPS planner is applied to generate assembly solution. The solution does not consider how skills are parameterized to actually perform the correct action, such as picking the correct object from the correct location. In [Pedersen and Krüger, 2015], the relevant parameters are provided automatically through a world model. This is possible because the scenarios are relatively simple. For more complicated scenarios and tasks, the world will either have to be extensively modeled, or the robot will have to have access to other sources of information for parameterization of the skills.

Different groups have proposed to let different robots assist each others in building comprehensible world models by uploading and downloading information to and from a common cloud service. The perhaps most comprehensive work in this field is the European project *RoboEarth* which attempts to build a "Wikipedia for robots" [Waibel et al., 2011, Tenorth et al., 2013]. The focus here is on sharing information that is learned by each robot connected to a central server. Several groups have proposed to use skills as a core element in the reuse of functionality [Bjorkelund et al., 2011, Mae et al., 2011]. Lund University's *Knowledge Integration Framework* (KIF) is targeted manufacturing environments, and the framework makes it possible to share programmed as well as learned information through a central server [Bjorkelund et al., 2011] similar to RoboEarth.

An alternative to automatic parameterization of skills is to provide an intuitive human-robot interface (HRI) that allow operators to supply the required information. Almost any HRI could in principle be used, including interfaces based GUI's, smart remotes, gesture recognition, speech recognition, and any combination of these methods. In [Archibald and Petriu, 1993], a graphical drag-and-drop programming interface is proposed for a *skills-oriented robot programming* (SKORP) framework, which enables operators to combine skills manually. This makes it possible to fast try a number different strategies for solving a task. However, only few "skills" are implemented, and their functionality is relatively simple such as *approach to touch* and *align normal*.

In [Guerin et al., 2015], a more generic skill-like programming architecture (CoSTAR) is proposed and tested on a collaborative UR5 robot in industrial settings. Tasks are programmed in a GUI by combining *behaviors* in behavior trees. The tree structure makes it possible to set up parallel behaviors and to select behavior(s) conditionally based on output of the previous behaviors. A similar approach is proposed in [Huckaby and Christensen, 2014], where SysML is used to manually setup assembly tasks. Common for both of these systems is, however, that no user tests are presented. Presumably, significant robot expertise is required to program non-trivial tasks.

The problem of either having limited functionality available or having to deal with a complicated programming system is in [Muszynski et al., 2012] proposed solved by letting the level of autonomy for the robot be adjustable. Adjustable autonomy was originally introduced by [Goodrich et al., 2001] for navigation and is in [Muszynski et al., 2012] extended to support simple manipulation. It is proposed to control a robot using a tablet GUI on either body, skill, or task level. Skills include here *navigation to goal* and *grasp object*, and task level control makes it possible to combine skills into sequences and specify parameters such as locations and objects. If a piece of information is missing, e.g. visual information on the object to pick, a lower level of autonomy can be set. Thus, an adjustable level of autonomy seems promising, but it has yet to be used for more complicated robot programming.

2.2 Interfaces using Projection Mapping

Traditionally, interfaces for industrial robots are based on either a teach pendant for online teaching or an interface to an external controller for motion planning. Collaborative robots have included force measurements which makes kinesthetic teaching possible; the operator can physically move the robot around. Typically, a teach pendant is still required for feedback though, as is for instance the case with Universal Robots. Rethink Robotics' robots are a notable exception. On these robots, feedback is given on a screen mounted on the robot. This frees the hands of the operator, but still forces his attention to be directed in one particular direction. This section explores research in human-computer and human-robot interaction which use projection mapping to enable interaction in *task space*. Task space denotes the space where a given task must be carried out.

2.2.1 Projection Mapping in Human-Computer Interfaces

Projection mapping means to pre-warp the projected graphics so that it looks geometrically correct on arbitrary surfaces in the environment. This makes it possible to use the real world instead of a specific flat surface as canvas for projection. Projection mapping can provide information to humans directly in the task space instead of having to rely in monitors.

In the human-computer interaction (HCI) community, projection mapping has been extensively studied as a means of providing natural interaction with objects as well as environments. Overhead projectors mounted in the ceiling combined with tracking of humans, objects, and/or IMU pointing devices can make it possible display useful information in a certain room, sometimes denoted a *smart space*. For instance in [Bandyopadhyay et al., 2001], projectors make it possible to create virtual drawings on objects as well as on the static environment. Objects are tracked using magnetic trackers fastened to each object and drawings will therefore remain on the objects as they are moved. Others have proposed to place steerable projectors in smart spaces, either by using pan-tilt units [Wilson and Pham, 2003,Ehnes et al., 2004] or mirrors [Pinhanez, 2001], in order to cover an entire floor and walls with a single projector. In [Ehnes et al., 2004], AR markers are tracked throughout a room and used for projecting interactive information such as GUI's and guidance for drilling. In [Wilson and Pham, 2003], Microsoft proposes a so called *WorldCursor*, in which a cursor can be projected anywhere in a room in line-of-sight from a projector. The idea is to extend the cursor of the Windows desktop to an entire room, and the WorldCursor is controlled using Microsoft's own IMU pointing device, the *XWand* [Wilson and Shafer, 2003].

Molyneaux et. al. adds in [Molyneaux and Gellersen, 2009] the the ability to detect and visually track so-called *smart objects*. A smart object is an object which stores information about itself such as current state and visual appearance, and which can communicate this knowledge to the smart space. Graphics can then be projected directly onto the tracked objects. This tracking-and-projection approach is comparable to the magnetic tracking proposed in [Bandyopadhyay et al., 2001]; however it is much more flexible because only a camera is used for detection and tracking.

After the launch of the Kinect in 2010, several groups have proposed to use their depth sensing capabilities to estimate the shape of the environment and use this for projection mapping. Researchers from Microsoft have presented two interaction systems themselves; the LightSpace [Wilson and Benko, 2010] and the RoomAlive [Jones et al., 2014]. LightSpace uses a number of pre-calibrated Kinect-projector pairs to give all flat surfaces the same interactive capabilities as Microsoft's own multimedia table, the Microsoft Surface (not to be confused with their later tablet computer of the same name). Using LightSpace, users can "pick" virtual objects from one surface and "place" them on another. Their RoomAlive can be considered a further development of LightSpace which is also based on multiple pre-calibrated Kinect-projector pairs. RoomAlive is, contrary to LightSpace, able to display graphics correctly on arbitrary shapes in the environment. Microsoft's primary use case for RoomAlive is real-time games, but the technology could be used for many other purposes as well.

Xiao et. al. proposes to use Kinect-projector pairs to make it possible to "draw" on surfaces similar to LightSpace in a system they denote *WorldKit* [Xiao et al., 2013]. Contrary to LightSpace, the WorldKit supports creating custom virtual interfaces to other devices. The system tracks users' hands and detects contact between hands and environment surfaces. Users can use this to draw messages, calendar displays controls for TV volume, etc.

A for mobile way of using projection mapping can be achieved by using hand-held instead of ceiling-mounted projectors [Cao and Balakrishnan, 2006, Cao et al., 2007, Molyneaux et al., 2012]. Molyneaux et. al. discriminates in [Molyneaux et al., 2012] between infrastructure-based smart spaces with ceiling-mounted projectors (such as LightSpace) and infrastructure-less systems (relying on hand-held projectors). They propose to use Kinects in both cases. In the smart space approach, Kinects are placed in the ceiling similar to Microsoft's systems. The Kinects are then used to build a model of the environment and to track the hand-held projector in 3D. In conjunction with an IMU device fixed to the projector, this allows the system to map graphics to any surface in a "flashlight" fashion. In the infrastructure-less system, a Kinect is fixed to the hand-held projector and a SLAM algorithm continuously builds a model of the room. This approach has the advantage that it enables projection mapping outside pre-calibrated smart spaces.

2.2.2 Projection Mapping for Human-Robot Interaction

Projection mapping used for human-robot interaction (HRI) is a significantly newer research area than for HCI. Similar to HCI, projection mapping inside smart spaces has also been proposed by several researchers for HRI. In [Ishii et al., 2009], a human-robot interface is proposed that allow humans to instruct simple logistic tasks to an autonomously moving robot. A human can select objects and locations by drawing circles around them using an IMU pointing device and the drawings are visualized by a projector. When the user accepts a task with the IMU device, the mobile robot drives to the marked object, picks it up, and transports it to the selected location. In 2008, a similar system was proposed, but replaced the projector with a laser pointer [Kemp et al., 2008]. The laser dot is detected with an omni-directional camera on the robot, and a pan-tilt stereo camera on the robot is rotated to accurately determine the location of the laser dot. The system is like the one from [Ishii et al., 2009] used for pick-and-place and transport operations. For these simple tasks the laser pointer works well, but a projector can of course display much richer information.

Schmidt et. al. also use projection interfaces to control robots, but propose to use a hand-held projector instead of a projector mounted on the robot [Schmidt et al., 2012]. Temporal codes are mixed into certain regions of the projected graphics and these make it possible to transmit data with the projector. The operator projects patters onto the robot which then receives and decodes the transmitted data. The principle is used to control moving robots as well as TVs and other devices. This system does however, as well as the ones previously described, not pre-warp projections, and larger pieces of graphics will therefore look distorted if projected onto non-orthogonal or non-flat surfaces.

In [Choi et al., 2013], pre-warping is used to project graphics so that a nearby human sees it as if it was projected onto a flat surface. A projector is mounted on a mobile robot, and (taking the position of the human into account) a chessboard pattern is iteratively adjusted to look correct from position of the human. While this is definitely an interesting problem, actual use cases for a system that compensates and attempts to hides structure in the environment seem limited; especially for typical industrial tasks.

More recent work explore how complex robot information can be visualized in human environments. Several research groups have designed smart spaces specifically for robotics research with a number of projectors mounted in the ceiling covering a large floor area [Omidshafiei et al., 2015, Ghiringhelli et al., 2014]. For instance, the MAR-CPS system presented in [Omidshafiei et al., 2015] combines a ceiling-to-floor projection system with a motion capture system that is capable of tracking flying drones as well as driving robots. This is used for different research areas, including surveillance coverage, path planning, and to identify robot states and error messages in general. Common for setups such as this is, however, that they are mainly intended for research purposes and not for end-users.

One type of robot intended for end-users for which projection mapping has been proposed is guide robots. In [Stricker et al., 2015], a robot guides a human by projecting arrows onto detected nearby walls. Both the position of the walls as well as the human into account. In [Coovert et al., 2014], it is proposed to use a projector mounted on a similar mobile robot to visualize its intended future movements. A comparative user study shows that arrows projected on the floor in front of the robot in general are understandable by humans and that they significantly improve the ability of humans to interpret the robot's intended actions. A similar user study is presented in [Chadalavada et al., 2015], where the robot's intended path is projected a few meters ahead. In this study, test persons are asked to walk towards and pass the robot with and without projection enabled. If the robot moves in a straight line, projection increases the average user rating 53%. If the robot takes a sudden turn, the rating increases by 65%.

The studies presented in [Coovert et al., 2014] and [Chadalavada et al., 2015] suggest that direct visual feedback in a common space in general can be advantageous - even though they concern very specific scenarios with limited scope and can not directly be generalized to other domains. Specifically for industrial robots, not much research exist in projection mapping interfaces. One exception is [Leutert et al., 2013], where it is proposed to use a projection based interface for assisting disabled people in handling and processing wooden pallets with industrial robots. An external projector and a flange-mounted projector are combined to project various data onto the static environment. The information that is projected includes floor images of the intended future pose of the robot as well as instructions given by the user. Instructions are given to the robot using a tracked pointing device and they are projected in real-time. The system seems promising, but it is difficult to conclude anything with regards to the usability of the different parts of the system since no user study is presented.

Common for existing work in projection mapping for robots is that the proposed systems are either only intended for research laboratory or that the projected information is simple and the interaction with the surrounding environment is very limited.

Chapter 3

Research Objectives and Methods

3.1 Specific Objectives

Two main research objectives were stated in Section 1.2 and are repeated below. The main objectives have been decomposed into a number of specific objectives which are also stated here. The research objectives provide the basis for the research contributions of this thesis.

Main objective 1: Vision in a User Centered Skill Framework
Investigate if and how a skill-based framework can assist factory workers without expertise in robotics and computer vision to teach collaborative robots to recognize, pose estimate, grasp, and quality check objects, after only very limited training.

Specific research objectives related to main objective 1:

1.1 Develop and implement a modular and flexible system for task level skill-programming using manual, kinesthetic teaching. Analyze how advanced vision capabilities can best be integrated into a skill based framework.

1.2 Develop skills that integrate vision capabilities such as recognition, pose estimation, and quality check, and which are easy and intuitive to program.

1.3 To support objective 1.2, evaluate the performance of affordable off-the-shelf depth cameras, which are compact enough to fit on industrial, collaborative robots. Evaluate and compare their performance specifically for reliable detection of small, industrial objects, which can be metallic as well as reflective.

1.4 Also to support objective 1.2, investigate fast and (to the operator) simple methods for calibrating depth cameras to robots.

Main objective 2: Teaching Skills in Task Space
Investigate if and how the usability of the teaching part of skills can be increased by providing feedback to the operator directly in task space during teaching, when compared to traditional feedback methods (text sheets, teach pendants/tablets, and monitors).

Specific research objectives related to main objective 2:

2.1 Investigate how projection mapping can be used to make the intentions and the internal state of robots clear to humans, and which advantages it provides w.r.t. usability when compared to traditional interfaces.

2.2 Implement task space interfaces for specific skills and evaluate their usability.

2.3 Implement a generic system for skill based programming with feedback in task space. Evaluate its usability when compared to traditional interfaces.

3.2 Research Methodologies

The methodology of this project is based on the critical rationalistic approach. In this approach, hypotheses are proposed and attempted proven or disproven. With each verification and falsification, additional knowledge is acquired, and this allows proposal of new and improved hypotheses in later iterations. As is often the case for robotics research, this project has required significant amounts of development and implementation. Proof-of-concept experiments have been used to test and verify the performance of major new developments.

The project is predominantly focused on human-robot interaction (HRI) and user experiments therefore play an essential role in evaluating research hypotheses. When applicable, the ISO standard on usability, 9241-11 (1998) [ISO, 1998], has been used to guide user experiments. Usability is here defined as a combination of three factors:

Effectiveness: The accuracy and completeness with which users achieve specified goals.

Efficiency: The spent resources in relation to the accuracy and completeness with which users achieve specified goals.

Satisfaction: The freedom from discomfort and positive attitudes towards the use of the product.

Effectiveness and efficiency can typically be evaluated from objective measures whereas satisfaction must be evaluated through qualitative and quantitative subjective methods such as questionnaires. For user experiments, a diverse user base has been used whenever possible to best reflect expected end-users.

The project has initially been coupled closely to European research projects including TAPAS [TAPAS, 2014] and CARLoS [CARLoS, 2015], and international cooperation has therefore been possible and necessary. The European projects have provided the opportunity to test functionality in actual industrial production environments in addition to the more traditional laboratory experiments. Tests in industrial scenarios are valuable because they provide a more realistic evaluation of robots intended for industrial use. Both for user experiments and proof-of-concept experiments, realistic settings are beneficial for getting reliable results.

Part II

Summary

Chapter 4

A Framework for Skill Based Programming

This chapter describes task level programming using robotic skills. First, robotic skills and skill based programming are introduced on a conceptual level in Section 4.1. Then, the skill based framework and implementation used for most of the scientific results in this thesis, Skill Based System (SBS), is presented in Section 4.2. On this basis, in Section 4.2.3 analyzed how vision capabilities are best integrated, in response to research objective 1.1. Finally, Section 4.3 summarizes a published journal paper which presents a proof-of-concept experiment in which SBS is used in a real-world industrial manufacturing setting.

The implementation of SBS is a joint work between the current author, Casper Schou, and Jens S. Damgaard.

4.1 Skill Based Programming

The conceptual skill based architecture is presented in the paper *Robot skills for manufacturing: From concept to industrial deployment* [Pedersen et al., 2015]. This section summarizes the central, conceptual parts of the paper. Additionally, the architecture is expanded with online manual teaching, which makes use of kinesthetic teaching to enable operators to intuitively program tasks while carrying them out.

4.1.1 Conceptual Architecture for Skill Based Programming

We propose a skill based architecture with three layers of abstraction: Tasks, skills, and device primitives. Device primitives are functions provided by a single device such as a robot arm, a gripper, a camera, or a force/torque sensor. In traditional programming for industrial robots, programs are created by combining multiple device primitives directly, and this requires a significant level of robot expertise. In skill based programming, device primitives are used by the *skill programmer* to design skills, which encapsulates robot knowledge and provides functionality to the end-user on a higher level. The purpose of skills is to enable shop floor workers on factories to fast and efficiently program robots to solve new tasks. Tasks can be programmed by sequencing and parameterizing a number of skills. The three layers can be defined as:

1. *Device primitives:* A robot is regarded as a composition of a number of devices; typically including a robot arm, a gripper, cameras, sensors, and possibly a mobile platform. Each of these devices provides functionality denoted *device primitives*. Device primitives are in most cases generic functions such as *move arm to Cartesian pose X*, *open gripper*, and *capture image*. This means that different devices often provide the same primitives. Programming based on device primitives can therefore be independent of specific hardware components.

2. *Skills:* A generic skill is a composition of sensing and manipulation primitives which together perform an object centered change to the world. A skill has a number of parameters which must be specified during programming. This parameterization makes the generic skill specific in the sense that it now performs one specific change to the world when executed. Skills depend on device primitives and not on specific devices. They are therefore decoupled from the hardware layer of the robot.

3. *Tasks:* A robot task can be specified as a desired transformation from an initial world state to a goal state. Tasks are set up and programmed by end-users such as shop floor workers by combining skills. Tasks are responsible for achieving the overall goals of the robot. The task layer is decoupled from the internals of the robot, and parts of the robot can therefore in principle be replaced without replacing the programmed tasks, as long as the new robot provides the same device primitives and therefore also skills.

Figure 4.1 illustrates functionality in the three layers. The “Pick object” skill is a sequence of device primitives including robot arm movements, gripper movements, and sensing. The sequence is constructed by a skill programmer. Contrary, the specific tasks such as “Fetch rotor cores” are programmed by end users by combining and parameterizing skills.

4.1.2 Definition of a Robotic Skill

Skills are high-level building blocks that can be combined to form a complete robot program given as one or more tasks. From the robot’s point of view, skills are “effectuating a change in a set of state variables, describing the knowledge

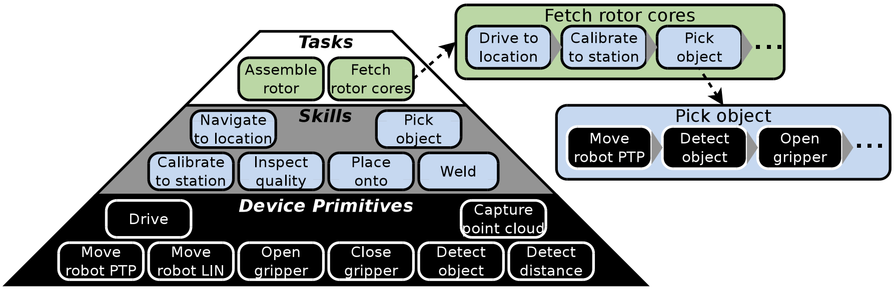


Fig. 4.1: The three abstraction layers: Task, skill, and device primitive.

the robot has of its surroundings" [Pedersen et al., 2015]. In the physical world, a skill constitutes an intuitive object-centered robot action which performs a state change to the world. This means that skills include *both* sensing *and* manipulation in the same skill. For instance, a pick skill must have a way of finding the object to pick; either from a world model stored by or available to the robot or through sensors from the external world. A "detect object"-function cannot be a skill because it does not perform a change to the world. Instead, this will be a device primitive or, if more devices are used for detection, a *service*. Services can be considered higher level device primitives, and are discussed further in Section 4.2.1.

In order to support intuitive task level programming, skills must be self-sustained. This means that they must hold all information necessary for them to be used. This includes:

- **Parameterization:** It must be possible to modify the action that a skill performs though simple parameters. However, no matter the parameters, the skill must perform the same type of action.
- **Preconditions:** A skill must be able to determine if it can be executed correctly and safely before execution starts.
- **Continuous evaluation:** During execution, the skill must be stopped if unexpected and/or dangerous situations occur.
- **Postconditions:** After execution, a skill must be able to determine whether it's operation has been completed as expected.

This leads to the generic skill model illustrated in Figure 4.2. The skill transforms the world from an initial state to a goal state based on parameters that have been specified previous to execution. The light green blocks ensure that the skills can only be executed in a safe and predictable manner. Execution together with the safety functions is denoted *operation*.

Similar to skills, all robot tasks can be specified as a transformation from an initial world state to a goal state. For non-trivial tasks, a single skill is not

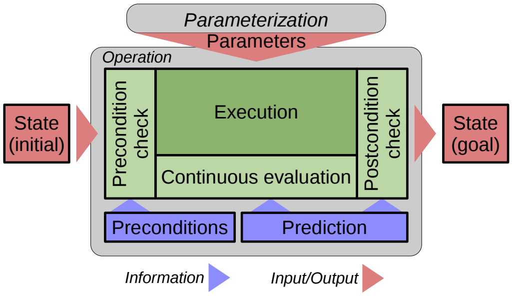


Fig. 4.2: The general model of a skill. The central part of the skill is the execution block which performs the desired manipulation of the world. The light green blocks make sure that the skill can only be executed in a safe and predictable manner. This is done by comparing the world and the world model against information from the blue blocks.

sufficient. Instead, a number of skills must be required. The operation part of the skill model also can intuitively be concatenated as shown in Figure 4.3.

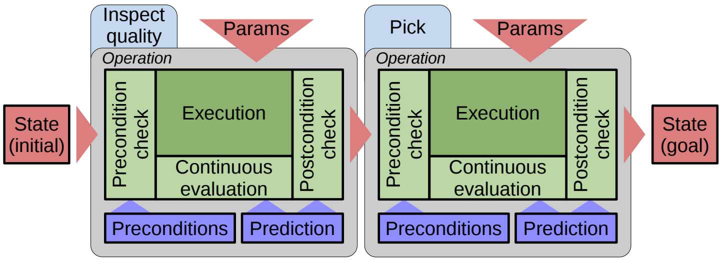


Fig. 4.3: The operation part of skills can directly be concatenated. The output state of one skill then becomes the initial state of the following skill.

4.1.3 Manual Parameterization

Different methods can be used for parameterization of skills including planner based approaches [Pedersen and Krüger, 2015, Rovida et al., 2014, Pedersen et al., 2015] and manual approaches [Schou et al., 2012, Schou et al., 2013]. The planner based approaches formalize world states and skill actions in PDDL (Planning Domain Definition Language) and employ STRIPS-like planners to combine skills into a program that can perform the requested state transformation. The manual approaches allow operators program on task level through an intuitive GUI and kinesthetic teaching. Planning based approaches have the advantage that no operator is required at all. They do, however, depend heavily on a comprehensive world model. Manual approaches instead keep humans-in-the-loop and take advantage of their knowledge of the world.

For the current thesis, a manual approach is used. The approach is illustrated in Figure 4.4. For each skill, a number of parameters need to be specified during task programming. Some of these can be specified offline in a GUI. This can for instance be robot velocity and the object type(s) to look for. Other parameters can intuitively be specified online through kinesthetic interaction with the robot. This can for instance be Cartesian positions and trajectories.

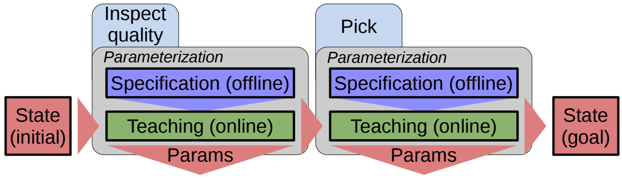


Fig. 4.4: Manual parameterization of skills. The skills are concatenated similar to the operation part of the skills in Figure 4.3. Teaching must be done in the correct order because teaching changes the world state in the same way as skill operation. Offline specification does on the other hand not affect the world and does therefore necessarily have to be done in order. Specification is only required to be completed before teaching of the same skill. Thus, all skills can (but are not required to) be parameterized before teaching begins.

During online teaching, the world gets transformed in the same way as during operation. Teaching of a pick skill, for instance, actually picks up an object and ends with the object being held by the robot. Therefore, teaching of skills must be done in the same order as during operation. Offline specification, on the other hand, does not transform the world and can therefore be carried out independently for each skill.

A combined model of skills can now be constructed with both an operation part and a manual parameterization part. This is illustrated in Figure 4.5. The model shows the parallelism between the two parts; teaching and execution, which transform from and to the same world states. They can therefore also be combined; e.g. for extending an existing task.

This ends conceptual introduction of robotic skills and task level programming. A skill model has been proposed which combines operation with manual parameterization including online teaching for intuitive robot programming. The following section presents an implementation of these concepts, SBS, which has been developed throughout this PhD project.

4.2 The Skill Based System

Two systems for skill based programming have been designed in cooperation with other PhD students during this project: The *Skill Based System* (SBS) [Schou et al., 2013] and the *Skill-based Robot Operating System* (SkiROS) [Rovida et al., 2014]. SkiROS is designed for automatic task planning and does not focus on human-robot interaction. SBS, on the other hand, implements

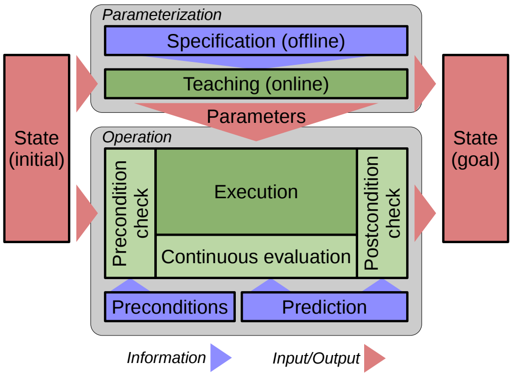


Fig. 4.5: Skill model including both operation and manual parameterization.

manual parameterization as introduced in the previous section, and is used for most of the scientific work in this thesis and it presented in the following. SBS is presented in the following.

4.2.1 Skill Based System Architecture

The conceptual architecture of SBS is illustrated in Figure 4.6, where the *layers* refer to the task/skill/device primitive layers in Figure 4.1. SBS is developed for the widely used *Robot Operating System* (ROS)[1], and all solid lines in the figure are ROS interfaces based on topics, services, and action servers. All overlapping circles are major classes in the same ROS node.

The central control keeps track of the current state and the world model during all stages of skill programming and operation. When skills are initially selected and specified offline, central control manages an internal, simulated world model which, based on preconditions and prediction of each skill, makes sure that only legal skills and parameters can be selected. During operation the same world model is being managed, but now compared to actual sensing data.

Each level (device primitive, skill, and task) has its own manager. The purpose of the managers is to decouple the functionality in each level from the other levels. In addition to these three managers, a UI-manager provides functionality to different user interfaces. The purpose is also for this manager to decouple the individual interfaces from specific functionality in the internal system. Currently, two user interfaces have been developed as part of the system and limited functionality is being provided to external interfaces using TCP/IP from rosbridge.

[1] Robot Operating System (ROS) [Quigley et al., 2009]. Refer to `http://www.ros.org/` for more information.

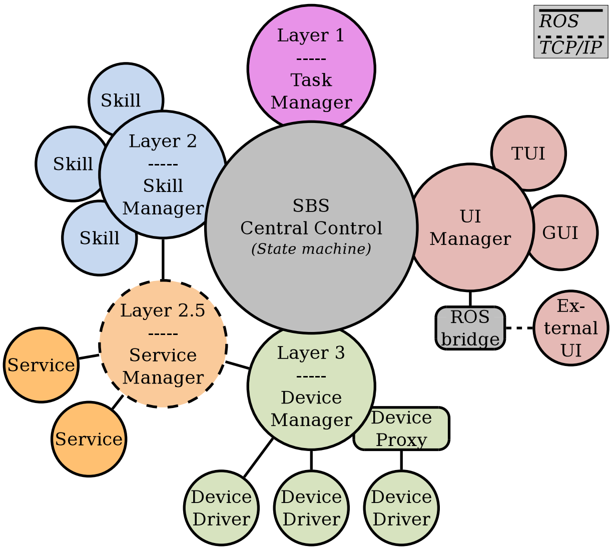


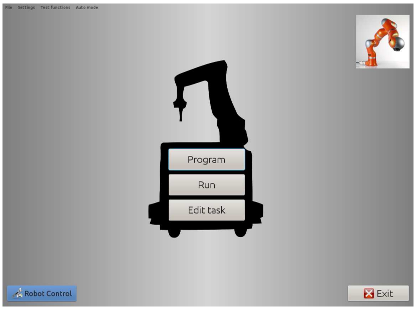
Fig. 4.6: The architecture of Skill Based System (SBS). Connected circles are different classes part of the same program and solid lines indicate communication via ROS.

Some functionality depends on multiple devices but does not perform an object-oriented an independent skill on its own. This can for instance be motion planning and in some cases object detection. Such functionality should be implemented as *services*. Services can, similar to device primitives, be used by different skills. Currently, the services are used directly by the relevant skills. As the as the system expands in the future, a dedicated service manager could keep track of existing services and standardize their interfaces seen from the remaining part of the system similar to the other managers.

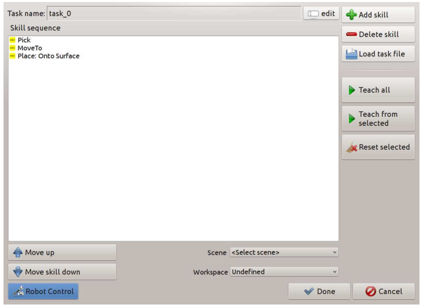
4.2.2 User Interface

Two interfaces are part of SBS; the *Textual User Interface* (TUI) and the *Graphical User Interface* (GUI). The TUI is intended for fast prototyping while the GUI is intended for non-expert end users. Different parts of the GUI is shown in Figure 4.7.

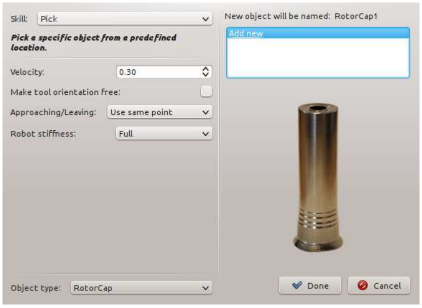
The typical program-and-execute process follows the images in 4.7(a)-(e). In (a), the user has access to different parts of the system. In (b), a new task is setup by selecting a sequence of skills. Is skill has to be parameterized, first offline and then online using manual, kinesthetic teaching. Figure (c) shows offline specification of a *pick* skill, and Figure (d) shows one step of guidance for online teaching. The top-left image in the figure provides direct instructions as to which inputs are required while the larger image gives an overview of the current teaching step. According to instructions in (d), the operator must



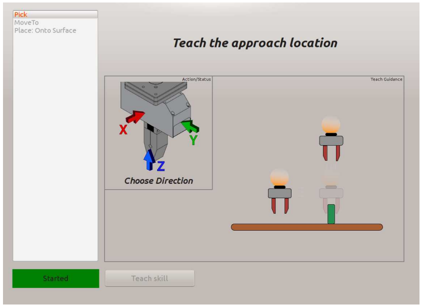
(a) Main window



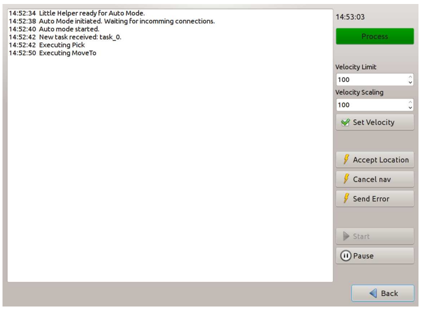
(b) Skill selection



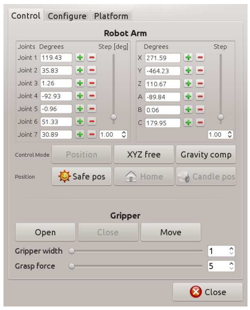
(c) Specification (offline)



(d) Teaching (online)



(e) Operation



(f) Robot control

Fig. 4.7: SBS GUI with offline selection of skills (b), offline specification of parameters (c), online teaching (d), and operation (e). The robot control in (f) provides direct control of the robot.

choose a direction to approach a particular location. The operator chooses by applying force in the desired direction. Figure (f) shows the robot control, which offers direct control over different parts of the robot.

A number of external graphical interfaces has been developed for specific purposes in addition to the integrated GUI. One of these is shown in Figure 4.8. This interface is developed as part of the CARLoS project [CARLoS, 2015]. It can be used to program and control stud welding operations both on Little Helper 3 using SBS and on a dedicated robot for the project which runs a different robot control system.

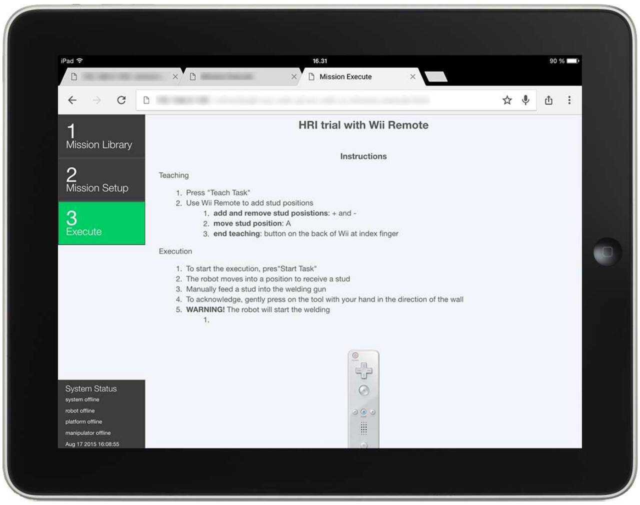


Fig. 4.8: External interface used with SBS on Little Helper 3 as well as with the CARLoS robot [CARLoS, 2015].

4.2.3 Vision Functionality in Skills

Vision functionality is relevant both for skill operation and for certain methods for online teaching. In Figure 4.9, various relevant vision functionality is listed against a number of developed skills as well as against different methods for online teaching. The figure should not be interpreted as giving indisputable answers as to what functionality should be included in which skills, but it does propose relevant functions for different stages of various skills as well as for teaching methods. Parentheses in the figure indicate that the function is relevant but not essential.

As is clear from the figure, many functions are relevant for several skills. The functions could therefore be integrated as device primitives. This would make the same function available for several skills, and it would support hardware independence seen from the skill layer. It is, however, a problem that many

Vision functionality	Recognize object	Pose estimate object (2D/3D)	Quality control / error detection	Detect environment (raw/markers)	Human detection and tracking	Detect occupied/free space
Teaching						
Projection mapping		*Teaching*		*Teaching*	*(Teaching)*	*Teaching*
Teaching-by-demonstr.					*Teaching*	
Skill (operation)						
Pick	*(Precondition)*	*Execution*		*(Execution)*		
Place			*Postcondition*			*(Execution)*
Place relative (into, onto)	*(Precondition)*	*(Precondition)*	*Postcondition*			*(Execution)*
Locate object	*Execution*	*(Execution)*				
Weld stud			*Postcondition*	*Precondition*		
Inspect quality		*Precondition*	*Execution*			
Calibrate to Workstation				*Execution*		
Drive to location						*Execution*

Fig. 4.9: Proposed vision and sensing functionality for different developed skills and methods for online teaching. For the operation parts of skills, it is listed in which part of the operation that the specific functionality is most relevant. Parentheses indicate that the function is relevant but not essential.

(if not all) of the functions could benefit from using multiple devices. For instance "Error detection" and "Pose estimate object" might have to move an eye-in-hand camera to get a better view of an object. A device primitive in the skill based framework should never depend on another device, and this would therefore not be possible.

An alternative is to implement all functionality as unique skills. Functions like "Recognize object" and "Pose estimate object" could then be selected during task programming similar to "pick". This, however, contradicts the definition of a skill, that it should "perform an object centered change to the world" as stated in Section 4.1.1. It would require more robot specific knowledge of operators to utilize such low-level skills.

Instead it is proposed to implement these functions as *services* as shown in Figure 4.6. Services can use device primitive, but are not available to operators directly. Instead, they provide functionality to skills similar to device primitives.

4.2.4 Integration with External Systems

The tight integration between SBS and ROS allows easy integration with systems developed by external partners. During this PhD project, a number of external systems have been integrated, including:

- Object tracking software [Choi et al., 2010] from the Georgia Institute of Technology where part of this PhD project was made.
- Pose estimation software from SDU[2] [Kiforenko et al., 2015] as part of the ACAT project [ACAT, 2015].

[2] University of Southern Denmark (SDU), `http://www.sdu.dk/`

- Motion planning software from the Austrian company CIT[3]as part of the TAPAS project [TAPAS, 2014].
- A remote tablet interface developed in HTML5 to work in SBS as well as for a partner's system as part of the CARLoS project [CARLoS, 2015]. A snapshot of interface is shown in Figure 4.8.
- A large number of publicly available ROS packages including drivers, navigation software, etc.

Some external systems have been integrated as services while others have been integrated directly into the relevant skill(s).

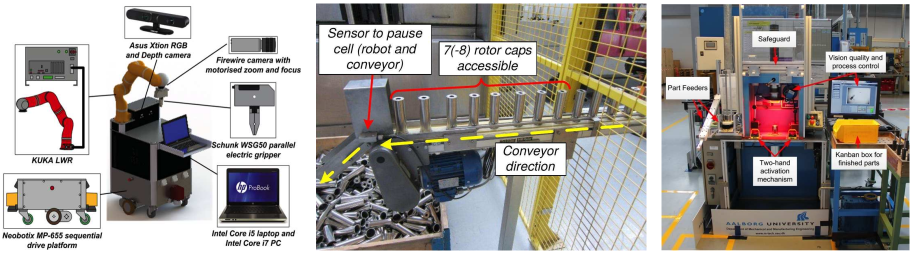
4.3 Paper A: Industrial Application of Skill Controlled Robots

To test and verify the functionality of the collaborative AIMM robots developed at Aalborg University as well as the skill based system SBS, a major real-world experiment was carried out spanning 10 days in an industrial manufacturing scenario at the Danish pump manufacturer Grundfos. Two Little Helper AIMM robots were used in the experiment, and the tasks of the robots cover the main application areas for AIMMs, ranging from logistic tasks to machine tending and assembly [Bøgh et al., 2012a]. The scenario can thus be considered a typical scenario for collaborative mobile robots, where fast and intuitive programming is essential. The contribution of the experiment is a proof-of-concept integration of skill based AIMM robots into real-world industrial manufacturing settings originally designed for human use. SBS is used for programming and execution, and the experiment thus partly answers research objective 1.1. The experiment is extensively described in the paper *Integration of Mobile Manipulators in an Industrial Production* [Madsen et al., 2015] and summarized here.

Figure 4.10 illustrates one of the robots in use (a), a machine tending station (b), and an assembly station (c) used during the experiment.

In the scenario, pump rotors are assembled from a rotor cap, a rotor core, a pressure ring, and eight magnets. Little Helper 3 (LH3) in Figure 4.10(a) first attempts to fetch rotor caps from the conveyor belt in Figure 4.10(b). This requires navigation, starting and stopping the conveyor belt, and picking rotor cores using vision. If no rotor caps are available, LH3 fetches them from a nearby warehouse station instead. LH3 then navigates to the assembly and press station in Figure 4.10(c), assembles the all the parts inside the press, and activates the press. The final part is picked from the press and placed in the kanban box left of the press. During assembly, LH3 communicates with an

[3]Convergent Information Technologies (CIT), `http://www.convergent-it.at/`



(a) Little Helper 3 (b) Rotor cap production (c) Assembly and press station

Fig. 4.10: One of the Little Helper robots uses in the experiment and two of stations. Both of these stations are build for manual use, but in this experiment slightly modified and used by two Little Helper robots.

external vision system and uses this to verify that critical assembly steps are completed satisfactorily. After assembly of a number of rotors, Little Helper 2 (LH2) picks up the kanban box and replaces it with an empty box from a warehouse station.

A total of 13 generic skills where used by LH3 at the stations during the experiments. The skills can be divided into 4 versions of pick, 4 of place, 2 for robot-workstation calibration, 1 for rotating objects, and one for performing quality control. One task, operation of the press, could not be carried out using generic skills. Instead, a specific *activate-press* skill had to be programmed from low-level device primitives.

The cycle time for production of one rotor was approximately 15 minutes divided into 5 minutes for fetching a rotor cap and 10 minutes for assembly. In comparison, the assembly operation can be carried our by a human worker in approximately 30 seconds. The task will therefore have to be sped up before it can be feasible from an economic perspective. However, it is an important contribution in itself to automate a task designed for human use, as well as using skill based programming program each task.

4.4 Conclusion

This chapter has presented robotic skills and extended the concept to include manual parameterization. An implementation of the concept, *SBS*, has been presented, and it has been analyzed and proposed how advanced vision functionality can be integrated in a modular manner that enables both reuse in different skills, hardware independence, and simplicity towards the end-user. Finally, an experiment has been presented where SBS is used for solving logistic tasks, machine tending, and assembly in a real-world setting. This answers research objective 1.1 and lays the ground for further development of vision-enabled skills as specified in objective 1.2. Vision skills are proposed and presented in the following chapter.

Chapter 5

Vision in a User Oriented Skill Framework

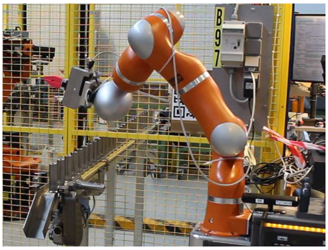
This chapter summarizes the part of the PhD project that concerns integration of vision functionality in the skill based framework. The chapter is based on three published conference papers, one presented extended abstract, and one technical report, which is synthesized from a peer-reviewed deliverable in the TAPAS project [TAPAS, 2014].

5.1 Paper B: Vision Skills

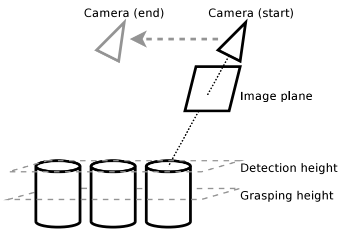
As part of the experiment at Grundfos presented in [Madsen et al., 2015], three skills were developed which directly use computer vision functionality. The experiment at Grundfos serves as a first proof-of-concept integration of vision functionality in a skill based system and thus a partial answer to research objective 1.2. The skills are described in detail in the paper *Human Assisted Computer Vision on Industrial Mobile Robots* [Andersen et al., 2013b] and summarized here:

Vision pick: The *vision pick skill* in SBS enables LH3 to detect objects with a camera mounted on the end-effector and to pick them based on taught parameters. The skill works for objects that are close to rotational invariant and located on a surface of fixed height. Although these requirements are quite limiting, they are in fact fulfilled by many items in a manufacturing plant. This is not least the case for pump parts, which by nature often are rotationally symmetric.

The setup and teaching parameters are shown in Figure 5.1. The locations of objects are detected using visual features such as edges. Which exact features to use are first specified manually on an external computer using a dedicated vision program based on Vision Builder in LabView. This program is currently in use throughout Grundfos' factories, and shop floor workers are able to configure it after very little training. In SBS, the name of the object class to pick is simply entered during offline parameterization of the vision pick skill. During operation, this name is used by the robot to request object detection over TCP/IP from the vision system.



(a) Setup



(b) Teaching parameters

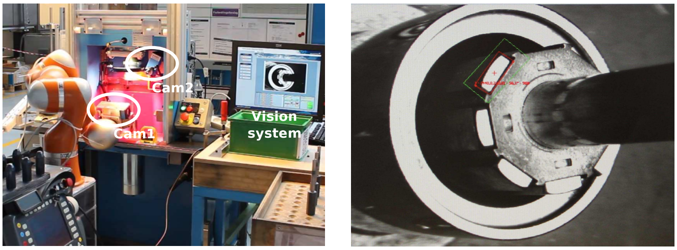
Fig. 5.1: Vision-pick-skill using external vision system.

Referring to Figure (b), the vision system detects objects in 2D in the image plane. During online teaching, the operator teaches how the object is grasped, primarily through three parameters: The height of the detected feature(s), the grasping height, and the grasping orientation. These features are taught using kinesthetic teaching. During operation, the 3D position of the detected feature can be found by extending the vector from the camera through the image plane to the detection height plane. The object is then picked using the taught grasping orientation in the taught grasping height. If no object is detected, the camera is moved along a taught line and it is attempted to detect objects every few centimeters.

During the experiment, the same skill with different parameters was used to pick objects from two locations; the conveyor belt in Figure (a) and a warehouse location.

Quality control: The vision system used for object detection is also used for binary success/failure quality control. In SBS, quality control is implemented as a skill with only one parameter being the name of the test in the remote vision system. From the skills concept, such quality control functionality should actually not constitute a skill on its own because its execution does not change the state of the world. Instead, a final system could implement this functionality as an optional post condition check for other skills.

During the assembly operation in the experiment, quality control was used for five different tests. In Figure 5.2(a) and (b) one example is shown where is is verified that a magnet has been placed correctly at the rotor core inside the press. The distance and angle between the magnet and the rotor core are measured and compared the pre-specified thresholds. In the example in the figure, the magnet passes the test.



(a) Setup

(b) Detection

Fig. 5.2: Quality control was used for five different tests. In this example, the position of the magnet in the green box in (b) is verified.

Fast calibration: Whenever an AIMM has driven to a new workstation it needs to determine it's exact position relative to the station. A function is developed for this based on an RGB-D camera mounted on the robot and QR codes fixed on the stations and implemented as a skill. During both teaching and operation, the pose of a fixed QR code is determined. Whenever the robot parks at a workstation, the pose of the fixed QR code is used to adjust for errors in the parking position and orientation.

No formal accuracy evaluation is presented in this paper, but preliminary tests indicate that the accuracy is at least ± 10 mm. The skill is used for calibration to the conveyor belt station which is shown in Figure 5.3.

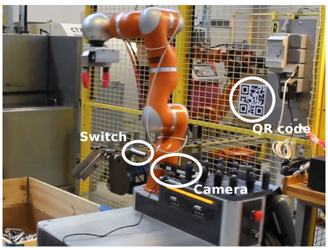


Fig. 5.3: QR calibration setup.

To sum up, three skills using vision were developed and applied successfully as part of a larger experiment in a real-world industrial setting. The skills were evaluated through the proof-of-concept experiment and no formal performance evaluation was carried out. The contribution is to describe how vision skills can be seamlessly integrated into a skill based framework, including how they can be parameterized by non-experts to solve new tasks.

5.2 Paper C: Fast Robot-Workstation Calibration

The QR calibration method presented in [Andersen et al., 2013b] is thoroughly described and evaluated in the paper *Fast Calibration of Industrial Mobile Robots to Workstations using QR Codes* [Andersen et al., 2013a]. The method supports camera-to-robot as well as robot-to-workstation calibration, and it is deeply integrated into the skill based framework. The contribution of the paper is a formal evaluation of the method and comparison with state-of-the-art. The main results are listed in Table 5.1.

Method	Duration	Precision	Source
Haptic	30-45 sec	± 1.0 mm	[Pedersen, 2011]
High speed	10 sec	± 1.0 mm	[Hvilshøj et al., 2010]
High precision	60 sec	± 0.1 mm	[Hvilshøj et al., 2010]
Proposed method	<1 sec	$< \pm 4.0$ mm	[Andersen et al., 2013a]

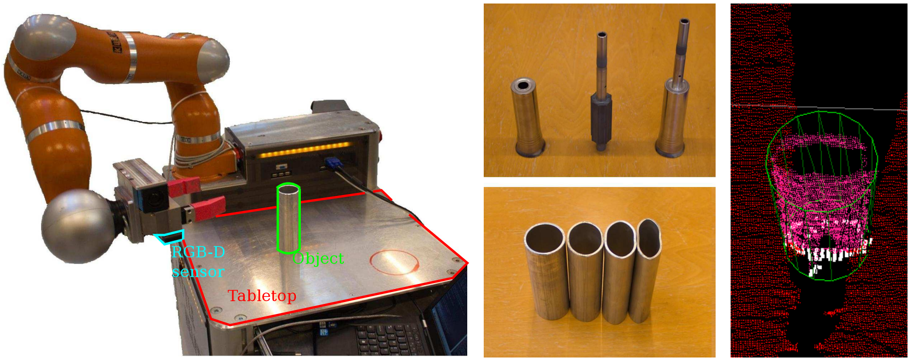
Table 5.1: Comparison of calibration methods. For the proposed method, the precision is estimated by repeated calibration and moving to the same measurable positions. Between each calibration, the platform is moved slightly (up to ± 15 cm and $\pm 10°$).

While the proposed method is not the most precise in the literature, and is significantly faster than comparable methods. This is especially advantageous for AIMMs that carry out logistic tasks and therefore need to move frequently between stations.

5.3 Paper D: Flexible Pick Skill using Depth Sensing

The paper *Using Robot Skills for Flexible Reprogramming of Pick Operations in Industrial Scenarios* [Andersen et al., 2014b] dives deeper into the skill based architecture and presents a more generic pick skill including full on-robot parameterization. Objects are detected using a point cloud from a depth camera as shown in Figure 5.4. Objects are segmented from the supporting plane, and containing cylinders are fitted around each valid object as shown in Figure 5.4(c). Objects are picked based on the position and size of this cylinder.

The contribution of the paper is to describe in full detail how the skill-based framework is used to develop a flexible vision-based pick skill, which fast and easily can be reprogrammed by non-experts. All necessary parameters are specified during parameterization, including the size of the object, where to look for objects, how to grasp the object, etc. Figure 5.5 shows the parameterization steps where parameters are specified and the operation steps where the same parameters are later used. Manual teaching is done solely using kinesthetic



(a) Setup (b) Test objects (c) Detection

Fig. 5.4: Objects are detected by segmenting a point could and thereafter picked using taught parameters. The skill is designed for close-to-cylindrical objects is tested successfully on the objects in (b). Figure (c) illustrated detection of the *most* skewed cylinder in (b).

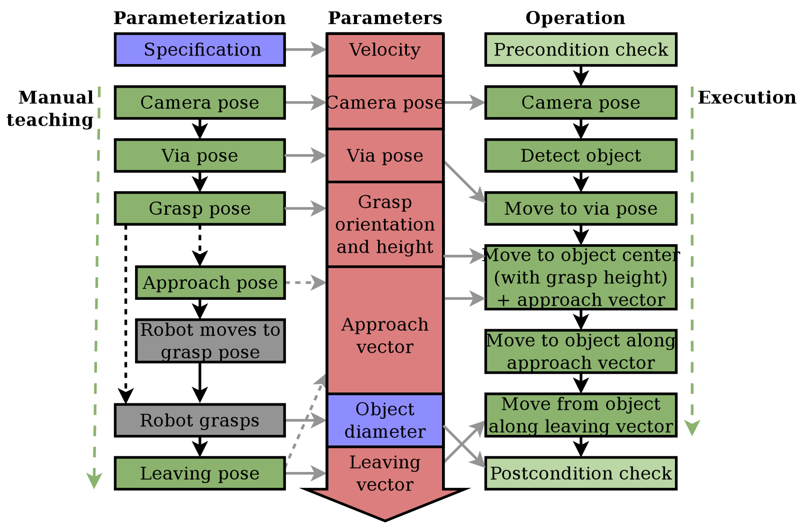


Fig. 5.5: Parameterization with manual teaching and operation of the pick skill. In the offline specification, the velocity of the robot is specified. It is also specified whether the same approach and leaving vector will be used. If that is the case, the "approach pose" is skipped during online teaching. The "object diameter" is blue because this is used for post-condition check. The remaining parameters are used for execution. *Manual teaching* is entirely performed by the operator using kinesthetic teaching.

teaching.

The figure also illustrates how parameterization and operation are parallel in the sense that the world is transformed in the same way. This makes it possible to continue teaching another skill, for instance a place skill, immediately after this skill has been taught. The skill is specifically designed for close-to-cylindrical objects, and tests show that it works with all objects shown in Figure 5.4(b).

The paper partially answers research objective 1.2 by presenting an advanced vision skill which is deeply integrated into the skill based system.

5.4 Report E: Adaptive Model Based Quality Inspection

Human workers naturally perform a visual inspection of all tasks that they carry out. This is also necessary for a flexible collaborative robot if more advanced tasks are to be carried out. In [Andersen et al., 2013b], we integrated quality control into the skill based system by allowing the robot to communicate with an external intuitive vision system. This approach allows a wide variety of tests to be used, but it does require a human to explicitly choose and set up tests in each scenario.

As part of the TAPAS project [TAPAS, 2014], a fully autonomous quality inspection skill was developed in cooperation with the partner company CIT[1]. It is described in detail in the attached report E, which is a synthesis of deliverable 3.8 in the TAPAS project. The skill enables a robot to detect errors using a depth camera mounted on the end-effector, only based on a CAD model and an approximate position of the object to inspect. It combines motion and next-best-view planning developed by CIT with error detection developed as part of this PhD project.

The purpose of the skill is to detect shape errors in the industrial objects handled in the TAPAS scenario. What characterizes these objects is that they are made of metal, their sizes range from 7 to 25 cm on the longest side, and they are reflective to a various degree. This is challenging characteristics to handle for most vision system. While large and expensive vision systems for object scanning do exist, typically based on laser scanning, the goal here is to fit a small system onto a collaborative robot. Small, off-the-shelf depth cameras, on the other hand, are not normally used for detecting reflective objects. Therefore, a number of small depth cameras were tested on one of the most reflective objects in the scenario. The evaluation criteria were:

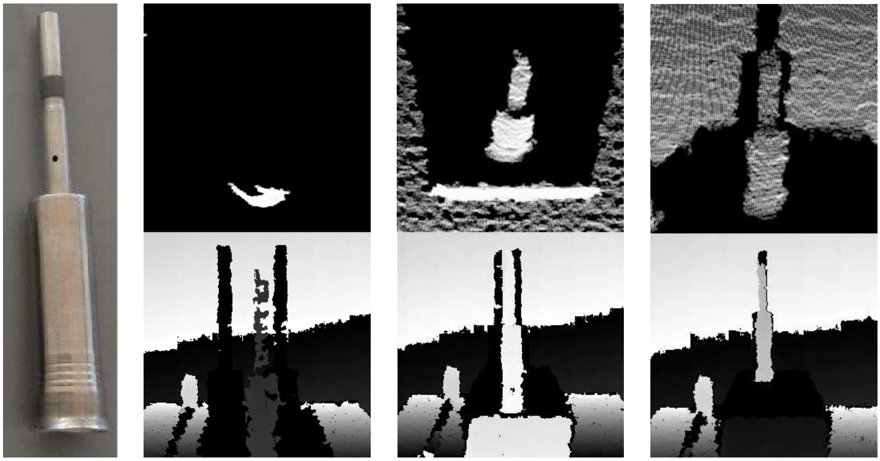
1. They should be able to detect as much of the surface as possible on distances up to 50 cm, which is reachable by typical collaborative robots,

[1] Convergent Information Technologies (CIT), `http://www.convergent-it.at/`

and

2. The shape of the detected surface should not have large errors.

The tested cameras were: PrimeSense Carmine 1.09, Microsoft Kinect 2, Asus Xtion, Mesa SwissRanger 4000, Intel RealSense R200, and Bumblebee XB3 using narrow-view and the Triclops stereo algorithm which comes with the camera. The Carmine 1.09 clearly outperformed the other cameras. The test object and results for the Carmine is shown in Figure 5.6.

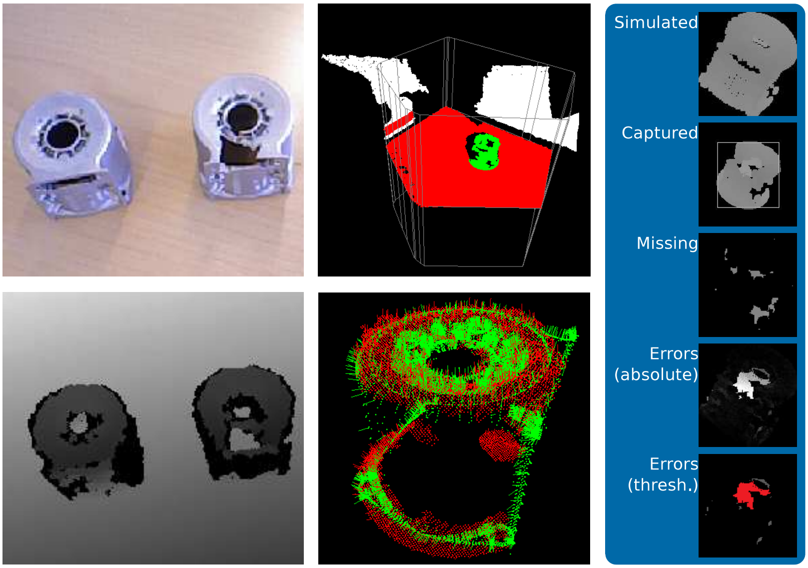


(a) Test object (b) 30 cm (c) 50 cm (d) 70 cm

Fig. 5.6: Tests of the PrimeSense Carmine 1.09 depth camera on a metallic, reflective test object. The bottom pictures are depth images of the test object on different distances. The top images are the same depth images seen as point clouds obliquely from above.

It is clear from the figure that the camera is able to detect the surface on 50 cm and above. However, the surface reconstruction is not perfect and some surface is not detected, especially near the borders. The quality control skill must able to handle these limitations. This is done by designing the system to be able to attempt re-detection of parts of the surface which could not be detected in the first iteration.

The error detection algorithm is illustrated in Figure 5.7. The objects in Figure 5.7(a) are used for testing, and the purpose is to detect the cavity error in the right-most object. The number of viewpoints which together makes it possible to detect the entire surface are first determined. The robot moves to all of them and captures depth images. The top image in Figure 5.7(b) shows the point cloud from one viewpoint. The object is segmented from its supporting surface. Point clouds of the object from multiple views are then merged into a single point cloud, and this is shown in the bottom image in red. The is matched to the *expected* point cloud which is shown in green.



(a) Test objects

(b) Segmentation (top) and matched point clouds (bottom)

(c) Error detection

Fig. 5.7: In (a) two test objects are show, one of which has an obvious cavity error. In (b) (top), one of the objects is detected and segmented from the supporting plane. In (b) (bottom), segmented point clouds from multiple views have been combined into the red point cloud. This is matched against the green point cloud, which is the points that the system *expected* to see. In (c), the error detection pipeline is shown for a single view. In this case, a large error is detected and highlighted red.

One way of detecting errors from the point clouds would be to take all points in the expected point cloud and determine the nearest point in the captured point cloud. If the distance is larger than a threshold, it should count as an error. A problem with such an approach is that is cannot easily distinguish between surfaces that the sensor failed to detect and actual errors. Therefore, the point clouds are instead projected to the camera views, as shown in the two top images in Figure 5.7(c). The third image in the figure shows surface that is *missing* in the captured point cloud. This cannot be used to conclude that there is an error in the object, but another iteration can instead try to get better data for these parts. The fourth and fifth image shown areas, where data is *present* but *different* in both point clouds. This cannot be due to sensor failures. If more surface than a predefined threshold is inconsistent, the quality check concludes that there is an error in the object. In the last image, the cavity error is detected and highlighted in red.

The contribution of the work is twofold: Firstly, the performance of different commercially available depth cameras is evaluated on short distances for detecting metallic, reflective objects. PrimeSense's Carmine 1.09 performs best, and this answers research objective 1.3. Secondly, it is shown how production errors can be detected fully autonomously and how feedback from the vision system can be used to compensate for imperfect sensing data. This functionality is integrated into a skill based architecture, and it the developed skill partly answers research objective 1.2.

5.5 Abstract F: Hand-Eye Calibration of Depth Cameras

For the adaptive quality control to provide good results, an accurate calibration between the robot's end-effector and the mounted depth camera is required. A typical way of calibrating RGB-D cameras such as the Carmine 1.09 used for the quality control is to calibrate the RGB camera and rely on the factory calibration between the RGB and depth cameras. This is a problem for two reasons. First, the VGA resolution of the RGB camera is not ideal for making an accurate calibration. Second, the factory calibration might not be completely accurate.

Instead, a dedicated method for hand-eye calibration of depth cameras is proposed in the extended abstract *Hand-Eye Calibration of Depth Cameras based on Planar Surfaces* [Andersen et al., 2014a]. This relates directly to objective 1.4. It is proposed to capture a series of point clouds of the same planar surface from different viewpoints. The situation is illustrated in Figure 5.8.

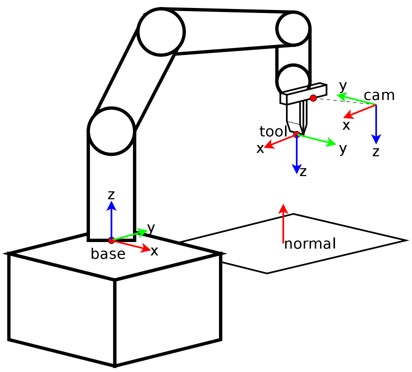


Fig. 5.8: The hand-eye calibration detects the same planar surface from multiple views. The figure illustrates all coordinate frames involved in the calibration.

For each view, an equation for the dominant plane is found using RANSAC

[Fischler and Bolles, 1981] followed by least squares fitting to all inliers. Two equations are now formulated, Equation (5.1) using the plane normal and Equation (5.2) using the point-to-plane distance:

$$^{\text{cam}}\overrightarrow{n}_{\text{plane}} = {}^{\text{cam}}_{\text{tool}}T \cdot {}^{\text{tool}}_{\text{base}}T \cdot {}^{\text{base}}\overrightarrow{n}_{\text{plane}} \tag{5.1}$$

$$D = \frac{^{\text{base}}\overrightarrow{n}_{\text{plane}} \cdot {}^{\text{base}}P_{\text{cam}} + d_{\text{plane}}}{|{}^{\text{base}}\overrightarrow{n}_{\text{plane}}|} \tag{5.2}$$

where the blue variables are unknowns, $^{\text{cam}}\overrightarrow{n}_{\text{plane}}$ is the estimated plane normal in each camera view, $^{\text{base}}\overrightarrow{n}_{\text{plane}}$ is the constant normal of the plane in the base frame, $^{\text{base}}P_{\text{cam}}$ is the position of the camera in each view, d_{plane} is the unknown but constant distance between the plane and the base, and D is the measured distance in each view. Note that Equation (5.1) is actually three equations, so in total four equations are used to estimate 12 unknowns. Therefore, as expected, more views are required.

The equations are combined into a single multivariate minimization function, $G(\overrightarrow{\theta})$. A cost function can then be constructed as $G^{\text{T}}(\overrightarrow{\theta})G(\overrightarrow{\theta})$, and this is minimized using gradient descent. To speed up the convergence rate, inertia is applied; meaning that a large fraction of the parameter update for each step is added to the update of the next. Also, the learning factor is increased gradually by a certain percentage. Both of these techniques risk causing serious overshoot, and if this happens, the update step is re-calculated with the inertia reset and the learning factor reduced by 50%.

The calibration algorithm is implemented in Matlab, and it has proven to work in simulated examples. For practical calibrations, surfaces which are horizontal relative to the base have been used. When the surfaces can be assumed to be horizontal, the minimization problem is reduced from 12 to 9 degrees of freedom. With this assumption, the algorithm converges in approximately 140 iterations for with real data.

5.6 Conclusion

In this chapter, a total of five vision-enabled skills have been presented. Two of these rely on an external system to perform image processing on monochrome images. The three remaining skills use depth sensing to detect and pick objects, perform quality control based on a CAD model, and to calibrate an AIMM to a workstation. Together these five skills answer research objective 1.2.

To support the skills that use depth-sensing, the performance of six commercially available depth cameras has been compared on short distances for detecting metallic, reflective objects. The best performing camera is the PrimeSense Carmine 1.09, and this answers research objective 1.3.

Finally, an algorithm has been developed for performing automatic hand-eye calibration of depth cameras to a robot end-effector. This answers research objective 1.4.

Chapter 6

Projection Based Task Space Interfaces

This chapter summarizes the part of the PhD project concerning projection based human-robot interaction. The chapter is based on two submitted conference papers and one previously unpublished technical report. Additionally one paper, [Andersen et al., 2015], has been published within the area. This is not included because it presents preliminary results which are included and expanded in [Andersen et al., 2016a].

6.1 Paper G: Intention Projection for Human-Robot Collaboration

Modern, collaborative industrial robots have the potential to function as co-workers to humans and collaborate on solving common tasks. For such collaboration to be fluent and seamless, it must be easy and intuitive to figure out the state and intentions of the robot. Compared to human co-workers, robots lacks abilities to signal intentions efficiently through body language, speech, and gestures. Instead, dedicated interfaces can be used. Traditional graphical interfaces do, however, require humans to focus attention on external monitors instead of the task at hand. The paper *Projecting Robot Intentions into Human Environments* [Andersen et al., 2016b] introduces the idea of enabling collaborative robots to project their intentions directly into the common workspace, the *task space*, using projection mapping. With information available to the human co-worker directly in task space, he can focus his attention solely towards this space, instead of being forced to continuously look at an external interface.

The main contribution of the paper is a human-robot interaction approach for collaborative industrial robots which uses projection mapping onto both tracked objects and static environments to indicate a robot's state and intentions.

A secondary contribution is a proposed pose estimation algorithm which is robust to different on-object projections. A robust pose estimation algorithm is necessary for continuous on-object projection. Figure 6.1 shows two use cases where the system has been tested. In (a), a car door is being tracked while a human moves it around. A wireframe projected onto the car door makes it visible to the human whether the door is being tracked correctly. In (b), warning signs are being projected onto an object that the robot intends to manipulate.

(a) Tracking visualized using a wireframe (b) On-object warning

Fig. 6.1: Interaction with tracked objects while information is being projected onto the objects in real-time. In (a), a wireframe is being projected onto a car door while it is being moved by a human. In (b), a sign warns a user that a robot intends to interact with the particular object.

The proposed pose estimation algorithm uses distance transformation to compare edges generated from an object model with edges detected in the camera frame. It is compared to an initialization algorithm based on SURF features by repeatedly estimating the pose of the car door in Figure 6.1(a) with a wireframe projected on top. The SURF based pose estimation (proposed by Choi and Christensen in [Choi et al., 2010]) has shown to perform very well when no graphics is being projected onto the object. However, the local SURF features are highly obscured by the graphics, even if different color channels are projected and captured, respectively. The tests documented in the paper show that the proposed method based on edges is significantly more robust to graphics projected onto the object, both with regards to accuracy and reliability.

The proposed interaction approach based on projection mapping is evaluated against traditional interaction approaches in a comparative usability study with 14 test persons. The test persons are asked to complete a task consisting of a series of subtasks in collaboration with a robot. Each subtask consists of either moving or rotating the object shown in Figure 6.1(b). Some subtasks are carried out by the robot and other must be carried out by the test person.

For each subtask, the test persons are either informed of what he has to do, or he is informed that the robot intends to manipulate the object. Three media are compared for providing the information: Text sheet, monitor, and projector. With the projector approach, information is projected onto the sides of the object as shown in Figure 6.1(b). With the monitor approach, the same information is provided on a monitor located next to the robot. With the text based approach, all subtasks are listed on numbered on a sheet of paper.

The usability is evaluated according to ISO 9241-11 (1998) [ISO, 1998], which defines usability as a combination of effectiveness, efficiency, and satisfaction. Effectiveness is in this study measured as the number of problems/questions. Because the task is relatively simple, only few problems and questions arose during the experiment. There is a tendency, though, that projection is most effective while text is least effective.

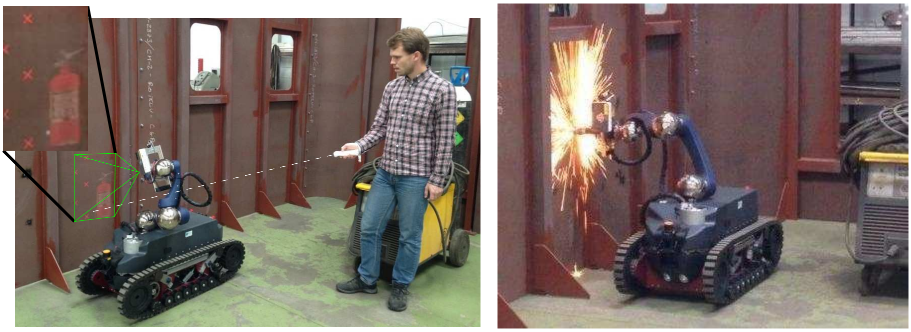
The efficiency is measured as the duration to complete the task, and this is almost identical for all methods. The satisfaction is evaluated through Lewis' After Scenario Questionnaire (ASQ) [Lewis, 1991]. The averaged and normalized scores place the projector interface best with a score of 14.3% (where 0% is optimal), monitor scores 16.7%, and text scores 21.8%.

To conclude, the paper and study answer research objective 2.1 by using projection mapping to communicate the state and intentions of a collaborative robot to human co-workers. It is showed that it is possible to combine real-time tracking of objects with projection onto the same objects, and an edge based method for pose estimation is proposed which is relatively robust to changing projections. The results of the user study show that the usability of projection based instructions for human-robot collaboration on average is higher than other methods on effectiveness and satisfaction and similar on efficiency. The results are, however, within the statistical uncertainty, and more research is required to draw definitive conclusions.

6.2 Paper H: Task Space HRI for Stud Welding Robots

The paper *Task Space HRI for Cooperative Mobile Robots in Fit-Out Operations Inside Ship Superstructures* [Andersen et al., 2016a] takes the idea of interacting with robots through projection based interfaces and applies it for a real task: Interacting with and programming of an autonomous stud welding robot. The welding operation is implemented as a robotic skill with online teaching, and projection mapping is used to program new and modify existing tasks. By implementing a task space interface for a weld skill, research objective 2.2 is answered.

Stud welding operations inside ship superstructures are today carried out manually. In the CARLoS project, it is proposed to automate part of the stud welding task with a mobile, autonomous robot [CARLoS, 2015]. The contribution of the work presented in [Andersen et al., 2016a] is an intuitive interaction approach for programming stud welding tasks. Projection mapping is used for providing information in task space, a Wii remote as IMU pointing device for modifying tasks, and a tablet for providing a high-level overview of the robot's state. The approach allows an operator to visually see and modify welding tasks of a robot while focusing his attention towards the task space. Figure 6.2 shows the system in use at the Valiña shipyard in La Coruña, Spain.



(a) Instruction (b) Welding

Fig. 6.2: Instruction of welding operations using an IMU pointing device and projection mapping. Welding positions are shown as red crosses and obstacles such as the fire extinguisher can also be shown. Stud positions can be added, deleted, and moved using a cursor controlled by the pointing device.

In Figure 6.2(a), the current task is being projected onto a wall segment. The task consists of welding positions and possibly obstacles; in the figure a fire extinguisher. The operator can move a projected cursor on the by wall the Wii remote. The cursor makes it possible to interact with the projected task information. Stud positions can be added, deleted, and moved, and obstacles can be deleted or moved as well if required. Additionally, the can be used to move the robot arm and thus enable the projector to project onto different parts of the wall. The initial task information can be loaded from a model file describing the compartment. If this is not available, a new task can be programmed online.

The interaction approach is evaluated in a laboratory usability study. A total of 17 test persons with diverse backgrounds participated in the study. The test persons were asked first to correct an erroneous stud distribution and second to instruct a new, specific stud welding task. In general, all test persons were able to carry out the tasks. The usability is evaluated according to ISO 9241-11 (1998) [ISO, 1998], which defines usability as a combination of effectiveness, efficiency, and satisfaction. The effectiveness and efficiency are here measured as the accuracy of the welding positions and the time for carrying out the tasks, respectively. The mean error is 14.7 mm and all errors are within 22 mm, which is acceptable for stud welding task. The time consumption is on average just above one minute for each task. It is difficult to compare these numbers to the manual process, though, since several independent steps are involved in this, including marking before welding can begin.

Satisfaction is evaluated through a custom questionnaire. Generally, all parts of the proposed system score very high, including projection of task information and the Wii remote interface. Some test persons suggested that additional step-by-step task information could be provided to assist novices.

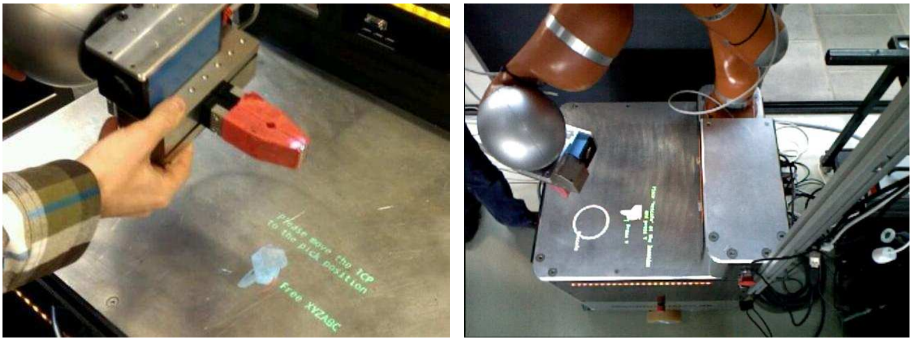
Parts of these results were published in the paper *Intuitive Task Programming of Stud Welding Robots for Ship Construction* in the *Proceedings of the 2015 International Conference on Industrial Technology (ICIT)* [Andersen et al., 2015]. This paper focuses on the skill implementation of the stud welding process. The newer paper, [Andersen et al., 2016a], instead includes more details on the system architecture, the usability study, and real-world tests from an actual shipyard.

6.3 Report I: Teaching Robotic Skills by Projecting into Task Space

The papers which are summarized in Chapter 4 and 5 present a skill based framework for manual programming of collaborative robots with various vision equipped skills. The papers presented in the current chapter propose to use projection mapping for interacting with robots; first for solving common tasks, and second for instructing welding tasks. In the latter case, projection mapping is used for instructing a particular robot skill; the weld skill. The technical report *Teaching Robotic Skills by Projecting into Task Space* merges the two research directions and presents a complete integration of a projection mapping in the skill based system as a replacement for the traditional monitor based GUI. This answers research objective 2.3.

The report proposes to project teaching instructions and additional required and helpful information into task space during manual, kinesthetic teaching robot skills. The hypothesis is that information in task space will increase the usability when compared to similar information being shown on a monitor. A secondary purpose of the report is to investigate the intuitiveness and usability of skill based teaching of relatively advanced vision skills such as object recognition and detection for pick. This is related to objective 1.2.

Figure 6.3 shows the proposed projection based interface on the Little Helper 3 robot. Teaching instructions are continuously projected onto modeled surfaces directly below the robot's end-effector, or as close as possible on an unoccupied surface. Also additional information is projected, including markings around detected objects, instructions on where to place objects, and warning areas which ask operators to stand clear. In Figure 6.3(b), the operator is for instance instructed to place a particular object in the circle so that the robot can learn it's appearance.



(a) Projected instructions (b) Setup

Fig. 6.3: The projection based interface displays information, which would otherwise be displayed on a remote monitor, near the robot's end-effector. The setup shown in (b) is used in a user study. The projector is placed near the camera used for taking the image, and it can project onto the two metallic surfaces below.

To evaluate the usability of the system, a comparative user study has been carried out where test participants after a short introduction to the system were asked to teach the same task online using 1) the projection based interface and 2) the monitor based interface. The task consisted of teaching three concatenated skills, which had been selected beforehand:

1. **Recognize:** The skill distinguishes between object classes using a discriminative model which is learned during teaching of the skill. The classification method is a bag-of-words approach based of SIFT features [Lowe, 1999]. During teaching, a number of monochrome images are captured of an object from each class, and SIFT features are extracted from the images. A vocabulary is constructed from all extracted features using K-means clustering with 20 bins. A classifier is then trained using a support vector machine (SVM).

   During execution, an object is classified by capturing and classifying three images. If the classifier does not predict the same class, more images are captured until at least 80% predict the same class. If this cannot be achieved, the skill fails. The skill accepts a specified class, and if another class is detected it is ignored.

2. **Pick-with-vision:** The skill uses a depth camera to detect objects on surfaces, fit a cylindric model around them, and pick them up. The skill is described in-depth in [Andersen et al., 2014b].

3. **Place-onto:** A previously grasped object is placed onto a surface using force sensing on a taught location. The skill has previously been presented in in [Schou et al., 2013].

A total of 20 persons participated in the study, and the usability is evaluated according to [ISO, 1998] as effectiveness, efficiency, and satisfaction.

Effectiveness: Measured as the average amount of assistance required for solving the tasks. This is reduced from 2.38 questions with the monitor system to 2.03 questions with the projection system on a scale weighted for seriousness in the required assistance. The difference is within the statistical uncertainty of the experiment.

Efficiency: Measured as the average time used to solve the tasks. This is reduced from 5:24 minutes with the monitor system to 5:20 questions with the projection system. This difference is also within the statistical uncertainty of the experiment.

Satisfaction: Measured subjectively through Lewis' ASQ questionnaire [Lewis, 1991] and specific questions concerning teaching as well as execution. The satisfaction measured as an average of all questions in the ASQ questionnaire increased from 5.60 with the monitor interface to 5.80 with the projection interface on a Likert scale from 1-7. This difference is within the statistical uncertainty. However, all specific questions which compare the two systems clearly favor the projection system to a statistically significant degree with a 95% significance level. This includes the preferred location of the information, the preferred method for indicating detected objects and instructed positions, and the projected warning area.

To conclude, the projection system scores on average slightly better on all usability factors, but both systems generally score very high. The differences between the systems are not statistically significant on most factors, and a larger study would be required to determine if there is an actual difference. Specifically on questions that compare elements of the interfaces directly, there is however a large and statistically significant advantage to the projection based interface. Most notably this includes the preferred position of the teaching information and instructions.

It is concluded that both interfaces have a high degree of usability for teaching a relatively advanced task including object recognition, detection, pick, and place. It can not be concluded from the user study that any interface is significantly better than the other in general. However, some elements of the projection based interface are clearly preferred by the test participants, including the position of teaching instructions in task space. As one test participant noted after the test, maybe a combination of the two interfaces would be better than any of the individual interfaces.

6.4 Conclusion

In this chapter, projection mapping has been proposed for improved interaction with industrial collaborative robots for collaborating on common tasks and for programming and teaching new tasks. First, a combined system of object detection, tracking, and projection mapping has been proposed for projecting intentions directly into a common workspace, the *task space*. Second, projection mapping has been proposed for instructing skills in task space; both a specific weld skill and a generic interface supporting all skills in the skill based system, SBS. Together with user studies of each suggested system this answers research objective 2.1, 2.2, and 2.3.

Three user studies have been carried out for evaluating the usability of projection based interfaces. In general, all studies showed that the tested interfaces had a high usability on all three usability factors; effectiveness, efficiency, and satisfaction. Comments from the test participants indicate that most participants found it easy to relate to information being projected into the real world. In two of the studies, projection based interaction is compared to traditional interfaces based on monitors or text, and on most factors, projection performed slightly better on average. The studies are not large enough to provide statistically significant conclusions on most factors, but on certain factors, projection is preferred to a significant degree. This includes for instance the preferred location of information during teaching of skills, where task space is clearly preferred compared to an external monitor.

A general conclusion is that the test participants appreciated information that provide an overview over the task contrary to only seeing the current step. A possibility can be to combine monitor and projector interfaces. Information that is required immediately can be projected while a monitor can be reserved for overview information which does not change frequently.

Chapter 7

Conclusions

This chapter sums up the main contributions of this thesis and suggests on this basis of the thesis relevant future research.

7.1 Contributions

The contributions of the thesis are listed below and numbered according to the research objectives stated in Section 3.1.

Vision in a user centered skill framework

1.1 *Task level programming with manual teaching of skills:* Chapter 4 introduces robotic skills and a skill based architecture which support manual parameterization. The *Skill Based System* is a modular implementation of a skill based architecture which has been developed as joint work between the current PhD, Casper Schou, and Jens S. Damgaard. The skill based architecture is presented in general and it is proposed specifically how advanced vision and sensing capabilities can be incorporated.

1.2a *Vision functionality as skills:* The papers A, B, C, and D together present a set of skills that includes vision functionality within the areas object picking, robot-workstation calibration, and quality control. All the skills can easily be parameterized using an intuitive GUI and online kinesthetic teaching. All skills have been tested in a laboratory and most of the skills have additionally been put into production in a factory.

1.2b *Autonomous quality inspection:* Paper B presents a quality control skill that relies on vision tests which are manually chosen in an external program. Contrary to this, Section 5.4 proposes to enable a robot to detect errors based autonomously solely on an object model. Next-best-view planning is combined with depth sensing to autonomously inspect object surfaces for inconsistencies between model and the physical object.

Depth cameras for industrial collaborative robots

1.3 *Evaluation of Depth Cameras:* To support objective 1.2, the performance of six commercially available compact depth cameras is evaluated for detecting reflective metallic objects on short ranges in Section E.2. It is concluded that the PrimeSense Carmine 1.09 performs best.

1.4 *Hand-eye calibration of depth cameras:* The extended abstract F proposes a method for performing hand-eye calibration of depth cameras based on planar surfaces. A preliminary Matlab implementation is presented, while a full integration in a robot control system is left as future work.

Projection Based Interfaces for Task Space Interaction

2.1 *Intention projection on environments and tracked objects:* Paper G proposes to make robots project their intentions onto both tracked objects and static environments in a workspace that is shared between human workers and robots. The idea to use a combination projection mapping and object tracking for collaborating with robots is novel. A user study indicates that the usability is higher for task space projection based interfaces than for traditional comparable interfaces.

2.2 *Instructing stud welding in task space:* The papers H and [Andersen et al., 2015] apply the idea of using projection mapping for human-robot interaction for instructing stud welding tasks inside ship superstructures. Stud welding is traditionally done manually, so automating the task is in itself novel. The contribution from the current PhD is to use projection mapping and an IMU pointing device to instruct and modify stud welding tasks. A user study shows that operators are able to quickly learn to use the system.

2.3 *Task space programming:* When an operator teaches skills he needs to understand the teaching steps. Report I proposes to use projection mapping to provide information directly in task space during teaching. A usability study shows that a projection based interface on average performs slightly better than a monitor interface on all usability measures (but within statistical uncertainties). On specific points, projection is strongly and significantly preferred. The users did for instance prefer to get teaching instructions near the robot's end-effector instead of on a remote monitor.

7.2 Concluding Remarks and Future Research

Since the beginning of this PhD project in 2012, collaborative robots have experienced nothing short of a boom in popularity, and it is impossible to predict exactly what the future holds. Robots such as Rethink Robotics' Baxter and and Sawyer, Universal Robots' newest variant UR3, and KUKA's iiwa help pave the way for new types of robots. Logistic robots such as Kiva Systems' warehouse solutions and MiR's ROS-based general purpose MiR100 transport robot are becoming more and more common, and combined industrial robots for both manipulation and logistics are beginning to move from research to industry with robots such as KUKA's KMRiiwa.

In this PhD project, various vision functionality has been developed and implemented in a user-friendly and hardware-independent skill based framework for collaborative robots. Rather than attempting to select every function and specify every parameter automatically, humans are kept in the loop during setup of new tasks. It has proven to be possible for robot experts as well as non-experts to use the developed vision skills, and some of the skills have additionally been put into experimental production at an industrial plant.

In a skill based framework, all skills should be on as high a level as possible and have a clear object and task oriented purpose for skills based programming to be easy. This PhD has suggested to implement vision and sensing functionality as services and make them available to the skill programmer and not to the end user as individual skills. This has the potential to facilitate both software and hardware modularization, and could make it feasible to develop even more advanced skills. For instance, it could be possible to develop a generic pick skill which incorporates detection, recognition, grasp planning, and picking, and which still is easy to setup using kinesthetic teaching and/or other intuitive interfaces. More research in this direction would be highly relevant.

Projection mapping interfaces have been implemented and tested in different scenarios in this PhD, and it is clear that projecting information into task space has advantages over traditional interfaces. Projection based interfaces can, however, also have disadvantages such as a lack of flat surfaces to project onto, objects and parts of the environment which have not been correctly modeled, etc. It would be relevant to investigate how an interface could be designed to take advantage of projection mapping *and* monitors or tablets most efficiently.

For the projection mapping interfaces themselves, there are several possibilities for improvements. One is to include a full model of the robot arm and end-effector. Thereby situations where the robot blocks projections could be avoided. A slightly more challenging improvement would be to integrate person tracking in the framework. This would enable better placement and orientation of projected information. Also, a single projector can only cover a relatively small working area. This area could be significantly increased if the projector was mounted on a calibrated pan-tilt unit.

The calibration issue leads to yet another path for future studies. There exist several solutions for fast calibration of one or more RGB cameras to each other and to a robot. In this PhD, easy and fast calibration routines have been investigated and developed for depth cameras in specific situations. Development of a general system for calibrating all elements in a setup with RGB cameras, depth cameras, projectors, and a robot would be highly relevant.

References

[ACAT, 2015] ACAT (2014-2015). Learning and execution of action categories acat project website. EU project funded under the European Community's Seventh Framework Programme grant no 600578. Web page: http://http://www.acat-project.eu/.

[Andersen et al., 2015] Andersen, R. S., Bøgh, S., Moeslund, T. B., and Madsen, O. (2015). Intuitive task programming of stud welding robots for ship construction. In *Industrial Technology (ICIT), 2015 IEEE International Conference on*, pages 3302–3307. IEEE.

[Andersen et al., 2016a] Andersen, R. S., Bøgh, S., Moeslund, T. B., and Madsen, O. (2016a). Task space HRI for cooperative mobile robots in fit-out operations inside ship superstructures. Submitted for publication at ICRA 2016.

[Andersen et al., 2013a] Andersen, R. S., Damgaard, J. S., Madsen, O., and Moeslund, T. B. (2013a). Fast calibration of industrial mobile robots to workstations using QR codes. In *Proceeedings of the 44'th International Symposium on Robotics (ISR)*, pages 257–262. Won the 'Best Paper' award.

[Andersen et al., 2014a] Andersen, R. S., Madsen, O., and Moeslund, T. B. (2014a). Hand-eye calibration of depth cameras based on planar surfaces. Presented at the 1'st International Workshop on Intelligent Robot Assistants (IRAS) at the 13'th International Conference on Intelligent Autonomous Systems (IAS).

[Andersen et al., 2016b] Andersen, R. S., Moeslund, T. B., Madsen, O., and Amor, H. B. (2016b). Projecting robot intentions into human environments. Submitted for publication at RO-MAN 2016.

[Andersen et al., 2014b] Andersen, R. S., Nalpantidis, L., Krüger, V., Madsen, O., and Moeslund, T. B. (2014b). Using robot skills for flexible reprogramming of pick operations in industrial scenarios. In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 678–685.

[Andersen et al., 2013b] Andersen, R. S., Schou, C., Damgaard, J. S., Madsen, O., and Moeslund, T. B. (2013b). Human assisted computer vision on industrial mobile robots. In *Proceedings of the 1'st AAU Workshop on Human-Centered Robotics*, pages 15–21. AAU.

[Archibald and Petriu, 1993] Archibald, C. and Petriu, E. (1993). Model for skills-oriented robot programming (skorp). In *Optical Engineering and Photonics in Aerospace Sensing*, pages 392–402. International Society for Optics and Photonics.

[Bandyopadhyay et al., 2001] Bandyopadhyay, D., Raskar, R., and Fuchs, H. (2001). Dynamic shader lamps: Painting on movable objects. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 207–216. IEEE.

[Bjorkelund et al., 2011] Bjorkelund, A., Edstrom, L., Haage, M., Malec, J., Nilsson, K., Nugues, P., Robertz, S. G., Storkle, D., Blomdell, A., Johansson, R., Linderoth, M., Nilsson, A., Robertsson, A., Stolt, A., and Bruyninckx, H. (2011). On the integration of skilled robot motions for productivity in manufacturing. In *Assembly and Manufacturing, IEEE International Symposium on*, pages 1–9.

[Bonasso, 1991] Bonasso, R. P. (1991). Integrating reaction plans and layered competences through synchronous control. In *IJCAI*, pages 1225–1233.

[Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23.

[Bøgh et al., 2012a] Bøgh, S., Hvilshøj, M., Kristiansen, M., and Madsen, O. (2012a). Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (aimm). *International Journal of Advanced Manufacturing Technology*, 61(5-8):713–726.

[Bøgh et al., 2012b] Bøgh, S., Nielsen, O., Pedersen, M., Krüger, V., and Madsen, O. (2012b). Does your robot have skills? In *Proceedings of the 43rd International Symposium on Robotics*.

[Cao and Balakrishnan, 2006] Cao, X. and Balakrishnan, R. (2006). Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 225–234. ACM.

[Cao et al., 2007] Cao, X., Forlines, C., and Balakrishnan, R. (2007). Multi-user interaction using handheld projectors. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 43–52. ACM.

[CARLoS, 2015] CARLoS (2014-2015). Cooperative robot for large spaces manufacturing. EU project funded under the European Community's Seventh Framework Programme grant no 606363. Web page: `http://http://carlosproject.eu/`.

[Chadalavada et al., 2015] Chadalavada, R. T., Andreasson, H., Krug, R., and Lilienthal, A. J. (2015). That's on my mind! robot to human intention communication through on-board projection on shared floor space. Presented at European Conference on Mobile Robots (ECMR).

[Choi et al., 2010] Choi, C., Christensen, H., et al. (2010). Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4048–4055. IEEE.

[Choi et al., 2013] Choi, S.-W., Kim, W.-J., and Lee, C. H. (2013). Interactive display robot: projector robot with natural user interface. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 109–110. IEEE Press.

[Connell, 1992] Connell, J. H. (1992). Sss: A hybrid architecture applied to robot navigation. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2719–2724. IEEE.

[Coovert et al., 2014] Coovert, M. D., Lee, T., Shindev, I., and Sun, Y. (2014). Spatial augmented reality as a method for a mobile robot to communicate intended movement. *Computers in Human Behavior*, 34:241–248.

[Ehnes et al., 2004] Ehnes, J., Hirota, K., and Hirose, M. (2004). Projected augmentation-augmented reality using rotatable video projectors. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 26–35. IEEE.

[EUROP, 2009] EUROP (2009). Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP.

[Fikes and Nilsson, 1972] Fikes, R. E. and Nilsson, N. J. (1972). Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Gat, 1998] Gat, E. (1998). On three-layer architectures. In Kortenkamp, D., Bonnasso, P. R., and Murphy, R., editors, *Artificial Intelligence and Mobile Robots*, pages 195–210.

[Ghiringhelli et al., 2014] Ghiringhelli, F., Guzzi, J., Di Caro, G., Caglioti, V., Gambardella, L. M., Giusti, A., et al. (2014). Interactive augmented reality for understanding and analyzing multi-robot systems. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1195–1201. IEEE.

[Goodrich et al., 2001] Goodrich, M. A., Olsen, D. R., Crandall, J. W., and Palmer, T. J. (2001). Experiments in adjustable autonomy. In *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, pages 1624–1629.

[Guerin et al., 2015] Guerin, K. R., Lea, C., Paxton, C., and Hager, G. D. (2015). A framework for end-user instruction of a robot assistant for manufacturing. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6167–6174. IEEE.

[Hartley and Pipitone, 1991] Hartley, R. and Pipitone, F. (1991). Experiments with the subsumption architecture. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1652–1658. IEEE.

[Huckaby and Christensen, 2014] Huckaby, J. and Christensen, H. (2014). Modeling robot assembly tasks in manufacturing using sysml. In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, pages 1–7. VDE.

[Huckaby et al., 2013] Huckaby, J., Vassos, S., and Christensen, H. I. (2013). Planning with a task modeling framework in manufacturing robotics. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5787–5794. IEEE.

[Hvilshøj and Bøgh, 2011] Hvilshøj, M. and Bøgh, S. (2011). " little helper"-an autonomous industrial mobile manipulator concept. *International Journal of Advanced Robotic Systems*, 8(2).

[Hvilshøj et al., 2010] Hvilshøj, M., Bøgh, S., Madsen, O., and Kristiansen, M. (2010). Calibration techniques for industrial mobile manipulators: Theoretical configurations and best practices. In *Robotics (ISR), 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–7. VDE.

[Hvilshøj et al., 2012] Hvilshøj, M., Bøgh, S., Skov Nielsen, O., and Madsen, O. (2012). Autonomous industrial mobile manipulation (aimm): past, present and future. *Industrial Robot: An International Journal*, 39(2):120–135.

[Hvilshøj and Bøgh, 2011] Hvilshøj, M. and Bøgh, S. (2011). "Little Helper" - an autonomous industrial mobile manipulator concept. *International Journal of Advanced Robotic Systems*, 8(2).

[Hvilshøj et al., 2009] Hvilshøj, M., Bøgh, S., Madsen, O., and Kristiansen, M. (2009). The mobile robot "Little Helper": Concepts, ideas and working principles. In *ETFA*, pages 1–4.

[Ishii et al., 2009] Ishii, K., Zhao, S., Inami, M., Igarashi, T., and Imai, M. (2009). Designing laser gesture interface for robot control. In *Human-Computer Interaction–INTERACT 2009*, pages 479–492. Springer.

[ISO, 1998] ISO, W. (1998). 9241-11. ergonomic requirements for office work with visual display terminals (vdts). *The international organization for standardization*.

[Jones et al., 2014] Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., and Shapira, L. (2014). Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 637–644. ACM.

[Kemp et al., 2008] Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., and Xu, Z. (2008). A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 241–248. IEEE.

[Kiforenko et al., 2015] Kiforenko, L., Buch, A. G., and Krüger, N. (2015). Object detection using a combination of multiple 3d feature descriptors. In *Computer Vision Systems*, pages 343–353. Springer.

[Leutert et al., 2013] Leutert, F., Herrmann, C., and Schilling, K. (2013). A spatial augmented reality system for intuitive display of robotic data. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 179–180. IEEE Press.

[Lewis, 1991] Lewis, J. R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the asq. *ACM SIGCHI Bulletin*, 23(1):78–81.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE.

[Madsen et al., 2015] Madsen, O., Bøgh, S., Schou, C., Andersen, R. S., Damgaard, J. S., Pedersen, M. R., and Krüger, V. (2015). Integration of mobile manipulators in an industrial production. *Industrial Robot: An International Journal*, 42(1):11–18.

[Mae et al., 2011] Mae, Y., Takahashi, H., Ohara, K., Takubo, T., and Arai, T. (2011). Component-based robot system design for grasping tasks. *Intelligent Service Robotics*, 4(1):91–98.

[Molyneaux and Gellersen, 2009] Molyneaux, D. and Gellersen, H. (2009). Projected interfaces: enabling serendipitous interaction with smart tangible objects. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 385–392. ACM.

[Molyneaux et al., 2012] Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A., and Gellersen, H. (2012). Interactive environment-aware handheld projectors for pervasive computing spaces. In *Pervasive Computing*, pages 197–215. Springer.

[Muszynski et al., 2012] Muszynski, S., Stückler, J., and Behnke, S. (2012). Adjustable autonomy for mobile teleoperation of personal service robots. In *RO-MAN, 2012 IEEE*, pages 933–940.

[Nilsson, 1993] Nilsson, N. J. (1993). *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers.

[Omidshafiei et al., 2015] Omidshafiei, S., Agha-Mohammadi, A.-A., Chen, Y. F., Ure, N. K., How, J. P., Vian, J., and Surati, R. (2015). Mar-cps: Measurable augmented reality for prototyping cyber-physical systems.

[Pedersen, 2011] Pedersen, M. R. (2011). Integration of the kuka light weight robot in a mobile manipulator. Master's thesis, Aalborg University.

[Pedersen and Krüger, 2015] Pedersen, M. R. and Krüger, V. (2015). Automated planning of industrial logistics on a skill-equipped robot. Presented at IROS 2015 workshop: Task Planning for Intelligent Robots in Service and Manufacturing.

[Pedersen et al., 2015] Pedersen, M. R., Nalpantidis, L., Andersen, R. S., Schou, C., Bøgh, S., Krüger, V., and Madsen, O. (2015). Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing.*

[Pinhanez, 2001] Pinhanez, C. (2001). The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *Ubicomp 2001: Ubiquitous Computing*, pages 315–331. Springer.

[Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5.

[ROBOTIQ, 2015] ROBOTIQ (2015). Collaborative robot ebook (sixth edition). Available at: `http://blog.robotiq.com/collaborative-robot-ebook`.

[Rovida et al., 2014] Rovida, F., Chrysostomou, D., Schou, C., Bøgh, S., Madsen, O., Krüger, V., Andersen, R. S., Pedersen, M. R., Grossmann, B., and Damgaard, J. S. (2014). SkiROS: A four tiered architecture for task-level programming of industrial mobile manipulators. Presented at the 1'st International Workshop on Intelligent Robot Assistants (IRAS) at the 13'th International Conference on Intelligent Autonomous Systems (IAS).

[Schmidt et al., 2012] Schmidt, D., Molyneaux, D., and Cao, X. (2012). Picontrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 379–388. ACM.

[Schou et al., 2012] Schou, C., Carøe, C., Hvilshøj, M., Damgaard, J. S., Bøgh, S., and Madsen, O. (2012). Human assisted instruction of autonomous industrial mobile manipulator and its qualitative assessment. In *Proceedings of the 1'st AAU Workshop on Human-Centered Robotics*, pages 22–28.

[Schou et al., 2013] Schou, C., Damgaard, J. S., Bogh, S., and Madsen, O. (2013). Human-robot interface for instructing industrial tasks using kinesthetic teaching. In *Proceeedings of the 44'th International Symposium on Robotics (ISR)*, pages 1–6. IEEE.

[Schuler, 1987] Schuler, J. (1987). *Integration von Förder-und Handhabungseinrichtungen*, volume 104. Springer-Verlag.

[SPARK, 2014] SPARK (2014). Strategic research agenda for robotics in europe 2014-2020 (SRA). Technical report, SPARK.

[Stricker et al., 2015] Stricker, R., Müller, S., and Gross, H.-M. (2015). Universal usage of a video projector on a mobile guide robot. In *Intelligent Robotics and Applications*, pages 25–36. Springer.

[TAPAS, 2014] TAPAS (2011-2014). Robotics-enabled logistics and assistive services for the transformable factory of the future. EU project funded under the European Community's Seventh Framework Programme grant no 260026. Web page: http://www.tapas-project.eu/.

[Tenorth et al., 2013] Tenorth, M., Perzylo, A. C., Lafrenz, R., and Beetz, M. (2013). Representation and exchange of knowledge about actions, objects, and environments in the roboearth framework. *Automation Science and Engineering, IEEE Transactions on*, 10(3):643–651.

[Waibel et al., 2011] Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., Haussermann, K., Janssen, R., Montiel, J., Perzylo, A., Schiessle, B., Tenorth, M., Zweigle, O., and van de Molengraft, R. (2011). Roboearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82.

[Wilson and Pham, 2003] Wilson, A. and Pham, H. (2003). Pointing in intelligent environments with the worldcursor. In *INTERACT*.

[Wilson and Shafer, 2003] Wilson, A. and Shafer, S. (2003). Xwand: Ui for intelligent spaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 545–552. ACM.

[Wilson and Benko, 2010] Wilson, A. D. and Benko, H. (2010). Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 273–282. ACM.

[Xiao et al., 2013] Xiao, R., Harrison, C., and Hudson, S. E. (2013). Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 879–888. ACM.

Part III

Papers and Technical Reports

Paper A

Integration of Mobile Manipulators in an Industrial Production

Ole Madsen, Simon Bøgh, Casper Schou, Rasmus Skovgaard Andersen, Jens Skov Damgaard, Mikkel Rath Pedersen and Volker Krüger

The paper has been published in
Industrial Robot: An International Journal, Vol. 42(1), pp. 11–18, 2015.

Integration of mobile manipulators in an industrial production

Ole Madsen, Simon Bøgh, Casper Schou, Rasmus Skovgaard Andersen, Jens Skov Damgaard, Mikkel Rath Pedersen and Volker Krüger
Department of Mechanical and Manufacturing Engineering, Aalborg University, Aalborg, Denmark

Abstract
Purpose – The purpose of this study has been to evaluate the technology of autonomous mobile manipulation in a real world industrial manufacturing environment. The objective has been to obtain experience in the integration with existing equipment and determine key challenges in maturing the technology to a level of readiness suitable for industry. Despite much research within the topic of industrial mobile manipulation, the technology has not yet found its way to the industry. To mature the technology to a level of readiness suitable for industry real-world experience is crucial. This paper reports from such a real-world industrial experiment with two mobile manipulators.
Design/methodology/approach – In the experiment, autonomous industrial mobile manipulators are integrated into the actual manufacturing environment of the pump manufacturer Grundfos. The two robots together solve the task of producing rotors; a task constituted by several sub-tasks ranging from logistics to complex assembly. With a total duration of 10 days, the experiment includes workspace adaptation, safety regulations, rapid robot instruction and running production.
Findings – With a setup time of less than one day, it was possible to program both robots to perform the production scenario in collaboration. Despite the success, the experiment clearly demonstrated several topics in need of further research before the technology can be made available to the industry: robustness and cycle time, safety investigations and possibly standardization, and robot and workstation re-configurability.
Originality/value – Despite the attention of research around the world, the topic of industrial mobile manipulation has only seen a limited number of real-world integrations. This work reports from a comprehensive integration into a real-world running production and thus reports on the key challenges identified from this integration.

Keywords Robotics, Automatic assembly, Man machine interface (MMI), Cooperative robots, Autonomous robots, Flexible manufacturing

Paper type Research paper

1. Introduction

Autonomous mobile manipulators that can be quickly moved and adapted to varying industrial needs to provide drastically new possibilities to manufacturing industries (Bøgh *et al.*, 2011). Contrary to the traditional stationary and pre-programmed production robots, mobile manipulators can provide assistance at multiple locations. They are able to provide highly flexible logistic possibilities and they can improve productivity by providing assistance in time-consuming, dangerous or straining situations. This constitutes a very different use case for mobile manipulators than the one known from traditional fixed robots. Thus, several new challenges arise indirectly from the increased flexibility and mobility; for instance, safety, navigation in human environments, programming complexity and time, and adaptability, to mention a few (Hepping *et al.*, 2007; Brogårdh, 2007; EUROP, 2009).

Research within the field of autonomous mobile manipulators goes almost 30 years back, and has the attention of many research groups around the world. In our previous work, we have studied the field of mobile manipulators for industrial purposes (Hvilshøj *et al.*, 2012b). We found that much research has gone into evolving the various aspects and capabilities of the mobile manipulator. Several researchers have conducted experiments with industrial components in industrial-like settings (Stopp *et al.*, 2002; Jamisola *et al.*, 2002; Früh *et al.*, 2007; Hamner *et al.*, 2010; Stolt *et al.*, 2011; Wang *et al.*, 2011). However, despite the large amount of research and the industrial needs and interest in the area, there are only few examples of implementation of mobile manipulators in real manufacturing environments (Datta *et al.*, 2008; Katz *et al.*, 2006; Helms *et al.*, 2002; Hentout *et al.*, 2010; Hvilshøj *et al.*, 2012a).

Hence, as part of the European Union-funded project TAPAS (FP7-260026), which aims at developing mobile manipulators, a number of experiments have been conducted with mobile manipulators in real-world industrial settings (TAPAS, 2010). The objective of these experiments is to analyze the gap between user needs and the developed

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
42/1 (2015) 11–18

[DOI 10.1108/IR-09-2014-0390]

The authors extend special thanks to Grundfos A/S for making this experiment possible and for continuous support and discussions regarding technology transfer to industry. This work was partially supported by the EU PF7 project TAPAS, FP7-2010-NMP-ICT-FoF, Grant number 260026.

Received 16 September 2014
Revised 7 November 2014
Accepted 10 November 2014

solutions in terms of performance, user-friendliness and cost-effectiveness.

This paper documents one such experiment where two Little Helper robots from Aalborg University have been implemented on the shop floor at the Danish pump manufacturer Grundfos.

2. The Little Helper robot

The Little Helper robot is a modular mobile manipulator intended for industrial use (Hvilshøj and Bøgh, 2011). It is designed as a re-configurable platform allowing for a variety of hardware to be mounted, as long as it complies with the general architecture. The Robot Operating System (ROS; ROS, 2014) is used as the software infrastructure to allow for distributed software nodes. Thus, ROS is used to connect the various drivers for each hardware component to the central software nodes handling the skill-based task programming and execution along with mission control and navigation.

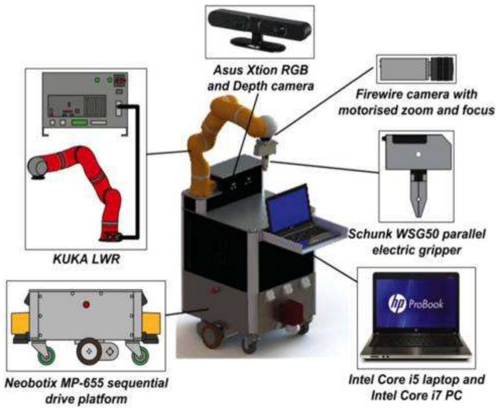
In this experiment, two Little Helper robots are used, both with a similar hardware setup. They both consist of a differential drive mobile platform (Neobotix MP-655) with onboard sensors (laser range, ultrasonic and motor encoders) used for motor control, navigation and safety. The manipulator used on both robots in this experiment is the seven-degree-of-freedom (DOF) KUKA Light Weight Robot (LWR). The KUKA LWR has, in our previous work, proven to be well suited for complex mechanical assembly tasks due to its force control, active compliance and seven-DOF. The extra DOF also provides improved manipulation flexibility in the logistic and machine tending tasks. The Little Helper robot is equipped with a tool-mounted FireWire camera with motorized zoom and focus. This camera is used for both object localization and quality control. Furthermore, an Asus Xtion RGB and depth camera is mounted on the robot platform and used to perform three-dimensional (3D) workstation calibration.

In this experiment, one Little Helper robot performs a complex assembly task, including quality control and part fetching logistic. The other robot performs logistic tasks supporting the assembly task. For further elaboration of the scenario, see Section 3. For clarification purposes, the two robots will be referred to as "LH1" (assembly) and "LH2" (logistics). LH1 is equipped with a Schunk WSG50 electric parallel gripper. This gripper provides full control of finger width and grasping force along with the ability to sense grasped objects. The jaws of the gripper have been designed for both rectangular-shaped objects and with a slot for providing a sufficient grip on cylindrical objects (when grasped from the side). This design has been found through empirical studies. LH2 has a passive tool specially designed for manipulating small-load carriers (SLC); hence, a type of plastic box. Figure 1 presents an outline of the LH1 architecture. Further information about the Little Helper concept can be found in (Hvilshøj and Bøgh, 2011).

2.1 Skill-based task programming

To enable an easy and operator-based setup of Little Helper for a particular production task a skill-based programming system has been designed. A skill is a (manipulating) action performed by the robot on an object using primitive functionalities of the different hardware components. It can be conceived as a building block with a predefined functionality that can be adapted to a specific task through a series of parameters. Thus, skills represent functionalities of the Little Helper as a whole. Skills can be instantiated and combined to solve complex robot tasks. A skill encapsulates robot knowledge, enabling the use of advanced robotics to non-robot experts. As a result, shop floor workers are able to instruct the robot and still solve complex assembly tasks (Bøgh *et al.*, 2012).

Figure 1 Outline of the Little Helper robot



The skill-based system implemented on the Little Helper consists of a graphical human–robot–interface (HRI), which is accessible through both a laptop on Little Helper and a tablet. This HRI allows the operator to setup the robot to a workstation, instruct new tasks and execute already instructed tasks. Programming a new task is done in two phases: a *specification phase* and a *teaching phase*. During the specification phase, a sequence of skills (e.g. Move <platform> to station <A>, Pick <Object>; Place <Object> into <workstation>) is chosen and partly parameterized. In the subsequent teaching phase, locational parameters are obtained through direct interaction with the manipulator.

During the specification phase, each skill in the skill sequence is chosen from a library of skills and several parameters (e.g. arm velocity) are provided through user input. While the specification phase is completely done through the Graphical User Interface (GUI), the teaching phase requires direct interaction with the manipulator of the Little Helper and is meant to provide the skills with location information. The teaching phase is done sequentially skill-by-skill so that the teaching sequence corresponds directly to the execution sequence, and thus creates a clear overview of the progress and outcome.

Taking a pick skill as an example, it has a predefined motion template. However, the motions are parametric and thus can be configured for the specific task through a set of parameters. For a pick skill, the motion template is to move to an approach location, move to the object location, close the gripper, enter compliance control mode and move to a depart location. The parameters are:

- Object (object type and location).
- Velocity.
- Compliance parameters.

- Approach direction and distance.
- Depart direction and distance.

The object parameters are either retrieved from the systems world model or provided by the operator through the teaching phase.

The production tasks in this experiment have been programmed using this skill-based programming framework. The programming time for setting up the experiment at the shop floor was less than 8 hours. Additionally, it was demonstrated that non-experts could perform the programming. More details about the skill-based programming framework including various user tests can be found in Schou *et al.* (2013).

3. The industrial setup

3.1 The production task

The task in the experiment was to produce rotor sub-assemblies for the Grundfos SQFlex pump. The rotor sub-assembly consists of a number of components (1 × rotor shaft, 1 × pressure ring, 8 × magnets and 1 × rotor cap), which are assembled in a fixture and pressed together to form the final product (see Figure 2).

3.2 The production scenario

The experiment takes place at the production facility of the SQFlex submersible pump; specifically in the area of rotor production. The rotor is the rotating part of the electric motor driving the pump. A sketch of the production layout with the respective workstation marked is shown on the left of Figure 3. The right side of this figure shows the corresponding navigation map obtained using the laser scanners on the platform.

The tasks of the two robots are as follows:

1 *Little Helper 1 (LH1)* – Assembly and quality control
   - LH1 *moves to Station 1*: Rotor cap production [or Station 1a (Rotor cap warehouse) if there are no rotor caps at Station 1].
   - *LH1* picks up N rotor caps at Station 1 (or Station 1a).
   - LH1 *moves to Station 2*: Assembly station.
   - *LH1* assembles N rotors at Station 2.

Figure 2 Overview of the components used in the assembly of the SQFle × rotor. 1 × rotor shaft, 1 × pressure ring, 8 × magnets and 1 × rotor cap are assembled into the SQFlex rotor

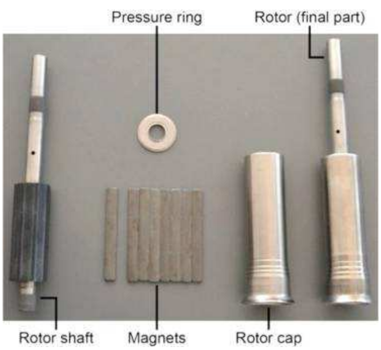
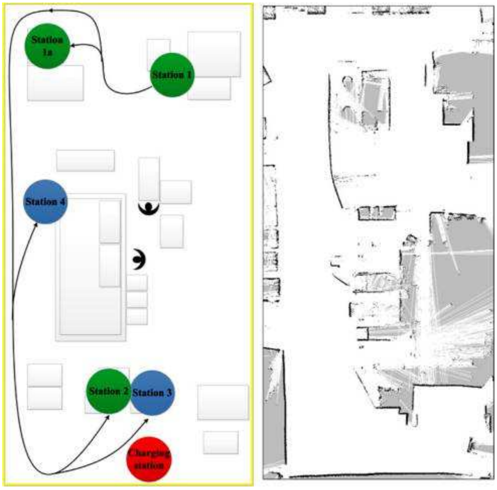


Figure 3 (Left) layout of SQFlex rotor factory with workstations and travel routes marked; green = LH1, blue = LH2, red = charging station. (Right) ROS SLAM navigation map obtained using the laser scanners on the Neobotix platform



   - *LH1* puts finished rotor assemblies in an SLC at Station 2.

2 *Little Helper 2 (LH2)* – Logistics and transportation
   - *LH2 picks up the full SLC with finished rotors at Station 3*: Finished SQFlex rotors.
   - *LH2 moves the full SLC to the warehouse system at Station 4*: Warehouse for SQFlex rotors.
   - *LH2* inserts the SLC in the shelf system at Station 4.
   - *LH2* picks up an empty SLC from the shelf system at Station 4.
   - *LH2* moves to Station 3.
   - *LH2* places an empty SLC at Station 3.

The digital map for navigation is learned during the initial task setup. The robot is jogged around the production area by a joystick, and the onboard laser range scanners acquire data about the surrounding area. The ROS GMapping library is used to create a two-dimensional (2D) map of the environment from the acquired laser data. The later autonomous navigation is performed by the Adaptive Monte Carlo Localization (AMCL) library based on the constructed map and live data from the laser scanners. Both the ROS GMapping and the AMCL have been adapted to the Neobotix mobile platform of Little Helper. This includes fusing the two point clouds from the laser scanners.

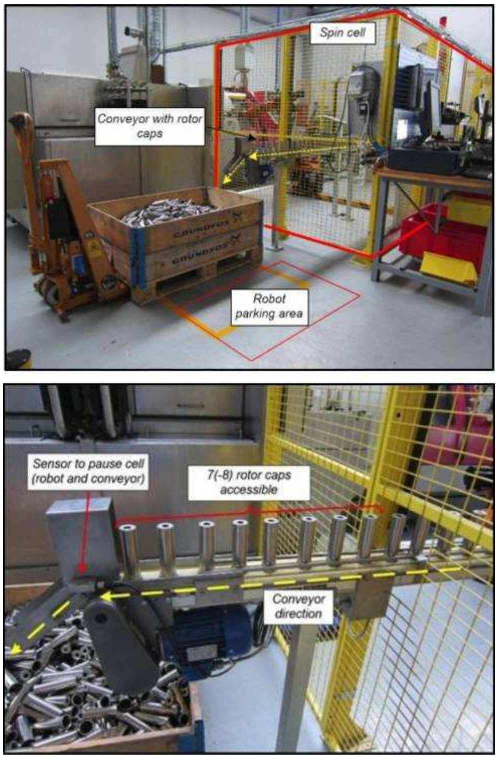
A central mission planner monitors the two robots and schedules them to the various tasks (Dang *et al.*, 2013). In this scheduling, the mission planner attempts to prohibit simultaneous navigation of both robots. This is done as the two robots must travel along intersecting routes and the interference of the laser scanners could cause undefined behavior.

3.2.1 Station 1: rotor cap production

At Station 1 LH1 will collect a number of rotor caps, a component used later in the assembly task at Station 2. These rotor caps are produced in an enclosed spin cell, exit on a conveyor belt and by the end they drop into a bin. The rotor caps are picked directly from the conveyor before they drop into the bin. A small mechanical switch is implemented to pause the conveyor and stop the rotor caps from moving (see Figure 4).

When arriving at the station, LH1 calibrates relatively to the workstation using a vision-based calibration skill. This skill uses the Asus Xtion camera mounted on the platform and a Quick Response (QR) marker on the workstation to perform a 3D vision calibration. This calibration skill including the 3D pose estimation of the QR marker is described in (Andersen *et al.*, 2013b). After calibrating, the robot activates the mechanical switch on the conveyor to pause the conveyor. Locating the rotor caps is done using a static 2D image captured with the tool camera. Hereby, the rotor caps are located in the horizontal plane by template matching (Andersen *et al.*, 2013a). The height of the rotor caps is instructed by the operator during the teaching phase. The location of the rotor cap is used by the robot to pick it up and place it in a fixture on the robot platform. After collecting the desired number of rotor caps, the robot re-activates the switch so that the operation of the conveyor, and inherently the spin cell, resumes. The task takes approximately 50 seconds to execute and is realized by four unique skills configured in a sequence of eight steps.

Figure 4 Station 1: rotor cap production. Rotor caps come from a spin cell on a conveyor. It is possible to grasp 7-8 rotor caps on the conveyor. The rotor caps are located using the onboard vision and lighting system. There is a small switch/sensor (lower) on the conveyor which the robot can turn on/off to pause the conveyor



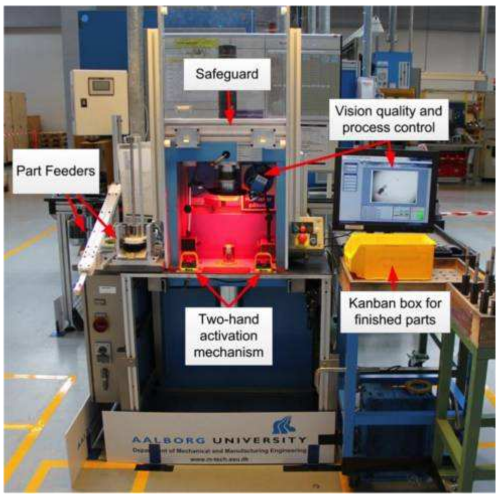
3.2.2 Station 1a: rotor cap warehouse

If there are not enough rotor caps available at Station 1, LH1 will move to Station 1a (rotor cap warehouse) and retrieve additional rotor caps. The object localization, pick-up and placing operations here are similar to the ones used at Station 1.

3.2.3 Station 2: assembly station

Figure 5 shows the assembly station at Grundfos. The components needed (see Figure 2) are all assembled in a fixture inside the housing of a hydraulic press. Upon assembly, the press is activated and the components are joined, thus becoming a finished rotor. The rotor shaft is picked from a trolley located next to the workstation. The trolley contains up to 96 rotor shafts, and in this experiment, it is brought to the workstation by a human operator. The rotor caps are collected by the robot itself at Stations 1 or 1a. Two feeders have been specially designed to hold the magnets and pressure rings. These feeders are designed for the robot to actuate (by pulling a lever/pushing a button) to extract a part. Using the robot to actuate the feeders has a negative effect on the cycle time, but it removes the need for electrical and communicational integration of the feeders. Decreasing the complexity of workstation adaptation complies with the scope of the TAPAS project. The finished rotors are put in an SLC located to the right of the press. When full, this SLC will be replaced by LH2.

Figure 5 Station 2 consists of the sub-assembly depicted in Figure 2. In the middle, the press for the assembly operation is located. To the right is Station 3 with a trolley for finished assembled parts



Upon arrival at the station, LH1 autonomously calibrates to the workstation using a haptic calibration skill. After calibration, the robot activates the pressure ring feeder to extract a ring, which is put into the fixture. Afterwards, the rotor shaft is picked and placed into the fixture. The magnet feeder is actuated to extract eight magnets, which afterwards are placed accurately into the fixture one by one. In between each magnet, the robot rotates the rotor shaft by 45 degrees. Finally, the rotor cap is picked from the robot platform and carefully placed in the fixture. After finishing the assembly, the robot closes a mechanical safety gate in front of the press to activate the hydraulic press. When the operation is finished, the robot opens the gate and moves the finished rotor to an SLC.

The safety gate has been designed for the experiment by Grundfos and was a strict requirement prior to the experiment for safety reasons. During human operation, the hydraulic press is operated by a two-hand activation mechanism.

An external 2D vision camera is mounted inside the press housing for acquiring images for quality control purposes. This camera is connected to a vision system developed by Grundfos. This vision system is designed for use by the operators on the shop floor, and thus complies well with the scopes of TAPAS. At certain steps in the assembly task, the Little Helper robot will request the vision system to perform a given quality check; for instance, check that the magnet has been properly inserted. Performing these quality checks has a negative effect on the cycle time, but is essential to ensure correct assembly, and thus the quality, of the final product. The tool-mounted camera is used for a single quality check outside the press housing. The Grundfos vision system interface is shown on the computer screen on Figure 6.

The assembly task at Station 2 takes approximately 10 minutes to complete with a skill sequence containing 118 steps (skills). The number of unique skills used in the task is 13 in total.

3.2.4 Station 3: finished SQFlex rotors

Little Helper 2 (LH2) moves to Station 3, calibrates using a QR marker, picks up the full SLC and places it on the mobile platform (see Figure 7). Afterwards, the SLC is transported to Station 4 (warehouse), where it is exchanged with an empty SLC. The robot returns to Station 3 with the empty SLC, which is placed next to the assembly station for LH1 to fill with finished rotors.

Figure 6 Vision system at Station 2 for process and quality control of the assembly process carried out by *LH1*

Figure 7 Station 3 is a pick-up area for the produced SQFlex rotors ready to be transported to the warehouse at Station 4

3.2.5 Station 4: warehouse for final SQFlex rotors

At Station 4 (see Figure 8), the finished rotors are stored in a shelf system for later use in the final pump assembly. At Station 4, LH2 inserts the full SLC retrieved from Station 3

Figure 8 The Station 4 warehouse shelf system for finished SQFlex rotors. The shelf system has capacity for both SLCs containing finished rotors and empty SLCs for Station 3. LH2 handles the logistical challenge of SLC handling between workstations

into the shelf system and extracts an empty SLC. The boxes and the shelf slots are located and identified via QR codes (see Figure 8).

3.3 Safety measures

The following safety measures were implemented at the SQFlex production facility in the duration of the experiment to protect people working in or near the testing area:

- Clearly marking the experiment area;
- Blocking the area with warning tape;
- Warning signs and instructions;
- Safety shoes; and
- Safety gate on the press.

Additionally, procedural, informational and personal protective measures including functional measures according to the industrial partners internal safety protocol were implemented. The Little Helpers were enhanced with the following safety measures (see also Figure 9):

- Industrial grade wireless emergency stop.
- *Signal indicators* – Yellow light indicators.
- Reduction of speed, torque and force.

The safety measures were implemented, but cannot be regarded as certified measures. This would require further analysis.

4. Experimental results

The experiment was carried out during a period of four production days, including hardware setup, workstation adaptation, robot programming, functionality tests, use case tests with shop-floor workers and integrated demonstrations in the live running production. The cycle time for production of one SQFlex rotor was approximately 15 minutes, including fetching one rotor cap at workstation 1, performing the assembly at Station 2 and the intermediate navigation between the workstations. A human worker performs the assembly in 30 seconds; however, this is excluding the part fetching. All tasks at the various workstations were programmed using the skill-based programming tool described in Section 2.1. In total, 13 different skills were used in the experiments (see Table I).

The main benchmark of the system was a four-hour consecutive operation to simulate half a workday of production. During this experiment, 26 errors occurred in total for both robots; resulting in a total downtime of 58 minutes combined for both robots. These errors were of various severities; however, three main errors were identified:

1. Errors in the navigation;
2. Errors in the assembly process; and
3. Errors in the communication with the central mission planner.

The errors in the navigation mainly occurred due to the navigation goals at the workstation being too close the production equipment. Errors in the assembly process were caused by too high tolerances in some of the feeding fixtures, resulting in improperly grasped objects. Other errors in the assembly process were also encountered. The communication with the mission planner experienced trouble because of two reasons; firstly, when one robot failed and was restarted, it could occasionally assume operation from the wrong step, thus confusing the mission planner. Secondly, much noise was experienced in the wireless communication between mission planner and robots occurring from manufacturing machines.

Figure 9 Yellow light indicators on the side of Little Helper flash during autonomous operation. Wireless emergency stop is implemented for the system

Table I Skills used in the experiment

Skill	Short description
Pick	Pick up an object
Place	Place an object
PlaceInto	Place an object into another object
PegInHole	Place an object into a hole (different approach than PlaceInto)
PickFromTrolley	Pick object from a trolley (several similar objects ordered in a pattern)
PickFromPlatform	Pick object from fixture on robot (several objects in pattern)
PlaceOnPlatform	Place object in fixture on robot
Rotate	Rotate an object
QCvision	Requests Grundfos vision system to perform a quality control
VisionPick	Use vision to locate and pick object
HapticCalibration	Calibrate to workstation using a haptic calibration
QRcalibration	Calibrate to workstation using 3D vision and a QR code
ActivatePress	Move gate on Workstation 2 to activate hydraulic press

5. Conclusions

In this paper, we have presented the results from implementing two mobile manipulators in a real-world manufacturing environment. The two robots have performed real-world manufacturing tasks using the actual equipment and workstations at the production facility. Smaller adaptations were made to the human-intended environment to enable the automation of these tasks; however, the workstation remained accessible and fully operational for human operators. Despite several errors during the four-hour scenario, the two robots successfully produced rotors and thus maintained the production area. Additionally, the experiment has shown that the task-level programming concept is applicable in real-world manufacturing tasks. The choice of tasks for this experiment has been done from a scientific perspective. Thus, the tasks are selected to provide a feasible yet challenging scenario for the robots. The assembly task has a cycle time more than 20 times longer than that of a human worker, which is considered too long; even though the robot does not need to be as fast as a human. As a result, the task is not feasible from an economical perspective. However, other aspects and benefits of automating the task might come into account. For instance, ergonomically issues in the manual task.

There are several reasons for the more than 20 times longer cycle time than that of a human worker. Firstly, the scenario and utilization of the robots could be improved, but the scenario was defined from a scientific desire. Secondly, the comparison does not take breaks, working hours, distractions, etc. into account. Thirdly, the human worker has two arms, very complex dexterous hands, complex and very fast compliant motions, and simultaneous performs quality control with the sight; the Little Helper robot only has a single robot arm, limited dexterity in grasping and slower compliant and force-controlled motions. Thus, the bottleneck in the robot task becomes the need to take the parts one by one and place them carefully and precisely in the fixture. We estimate that using a dual-arm setup for the assembly task could provide a significant increase in speed. Finally, the navigation speed, docking procedure and the subsequent calibration of the mobile manipulator all provide an increased cycle time compared to a human worker.

The experiment has shown that the technology of autonomous mobile manipulators can be implemented in real-world industrial manufacturing settings. Yet, the experiment has also clarified several challenges that still need to be solved to mature the technology to a level suitable for large-scale industrial implementation. Together with earlier real-world and laboratory experiments conducted in TAPAS (Hvilshøj *et al.*, 2012a), this experiment has indicated the following:

1. A number of relevant real-world industrial tasks can be solved with the existing technology, especially logistic tasks and assistive tasks, e.g.:
   - Continuous part feeding.
   - Simple assemblies/sub-assemblies.
   - Continuous quality and process control.
2. A skill-based programming methodology can be used for faster and more intuitive robot programming in industrial settings.
3. Integration into the surrounding manufacturing system is possible, but is a time-consuming task.
4. Adapting the robot to new tasks involves a number of hardware setup tasks; both intrinsic on the robot and extrinsic in the workstation (e.g. setup of feeders, grippers and access to machines). These tasks are quite time-consuming compared to the needed re-programming of the robots. In the experiments, the programming time was less than a day, whereas the hardware setup task was several weeks.
5. Many problems could be solved if the potential use of the mobile manipulator is built into the production system from the beginning, i.e. as part of the production system design.
6. Before a large commercial impact can be achieved, the following RTD areas have to be handled:
   - Robustness and processing speed must be increased.
   - Safety has to be solved, both through standardization and through new technologies.
   - System flexibility, re-configurability and usability must be improved/implemented.

References

Andersen, R.S., Damgaard, J.S., Madsen, O. and Moeslund, T.B. (2013b), "Fast calibration of industrial mobile robots to workstations using QR codes", *Proceeding of the 44th International Symposium on Robotics (ISR)*, IEEE, Seoul, pp. 1-6.

Andersen, R.S., Schou, C., Damgaard, J.S., Madsen, O. and Moeslund, T.B. (2013a), "Human assisted computer vision on industrial mobile robots", *Proceedings of 1st AAU Workshop on Human-Centered Robotics*, Vol. 1, Aalborg Universitetsforlag.

Bøgh, S., Hvilshøj, M., Kristiansen, M. and Madsen, O. (2011), "Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (AIMM)", *International Journal of Advanced Manufacturing Technology*, Vol. 61 Nos 5/8, pp. 713-726.

Bøgh, S., Nielsen, O.S., Pedersen, M.R., Krüger, V. and Madsen, O. (2012), "Does your robot have skills?", *Proceedings of the 43rd International Symposium on Robotics*, Verlag, pp. 29-31.

Brogårdh, T. (2007), "Present and future robot control development – an industrial perspective", *Annual Reviews in Control*, Vol. 31 No. 1, pp. 69-79.

Dang, V.Q., Nielsen, I.E., Bøgh, S. and Bocewicz, G. (2013), "Modelling and scheduling autonomous mobile robot for a real-world industrial application", *Proceedings of IFAC Conference on Manufacturing Modelling, Management and Control (IFAC)*, Vol. 7, *Saint Petersburg*, pp. 2098-2103.

Datta, S., Ray, R. and Banerji, D. (2008), "Development of autonomous mobile robot with manipulator for manufacturing environment", *International Journal of Advanced Manufacturing Technology*, Vol. 38 Nos 5/6, pp. 536-542.

European Robotics Technology Platform (EUROP) (2009), "The strategic research agenda for robotics in Europe – robotics vision for 2020 and beyond".

Früh, H., Keller, P. and Perucchi, T. (2007), "Intelligent mobile manipulators in industrial applications: experiences and challenges", *Lecture Notes in Computer Science*, Vol. 4850, pp. 370-385.

Hamner, B., Koterba, S., Shi, J., Simmons, R. and Singh, S. (2010), "An autonomous mobile manipulator for assembly tasks", *Autonomous Robots*, Vol. 28 No. 1, pp. 131-149.

Helms, E., Schraft, R.D. and Hägele, M. (2002), "rob@work: robot assistant in industrial environments", *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, IEEE, pp. 399-404.

Hentout, A., Bouzouia, B., Akli, I. and Toumi, R. (2010), "Mobile manipulation: a case study", in Lazinica, A. and Kawai, H. (Eds), *Robot Manipulators: New Achievements*, Intech, pp. 145-167.

Hepping, C., Fuhlbrigge, T.A. and Zhang, G. (2007), "Towards intelligent robotic assembly: trends in industry", *Assembly Automation*, Vol. 27 No. 3.

Hvilshøj, M. and Bøgh, S. (2011), "Little helper - an autonomous industrial mobile manipulator concept", *International Journal of Advanced Robotic Systems*, Vol. 8 No. 2.

Hvilshøj, M., Bøgh, S., Nielsen, O.S. and Madsen, O. (2012a), "Multiple part feeding real-world application for mobile manipulators", *Assembly Automation*, Vol. 32 No. 1, pp. 62-71.

Hvilshøj, M., Bøgh, S., Nielsen, O.S. and Madsen, O. (2012b), "Autonomous industrial mobile manipulation (AIMM): past, present and future", *Industrial Robot: An International Journal*, Vol. 39 No. 2, pp. 120-135.

Jamisola, R., Ang, V.M.H., Oetomo, D., Khatib, O., Lim, T.M. and Lim, S.Y. (2002), "The operational space formulation implementation to aircraft canopy polishing using a mobile manipulator", *Proceedings of IEEE International Conference on Robotics and Automation ICRA '02*, pp. 400-405.

Katz, D., Horrell, E., Yang, O., Burns, B., Buckley, T., Grishkan, A., Zhylkovskyy, V., Brock, O. and Learned-Miller, E.G. (2006), "The UMass mobile manipulator uMan: an experimental platform for autonomous mobile manipulation", *Workshop on Manipulation in Human Environments at Robotics: Science and Systems*.

ROS (2014), "Robot operating system", available at: www.ros.org (accessed 5 June 2014).

Schou, C., Damgaard, J.S., Bogh, S. and Madsen, O. (2013), "Human-robot interface for instructing industrial tasks using kinesthetic teaching", *Proceeding of the 44th International Symposium on Robotics (ISR)*, IEEE, Seoul, pp. 1-6.

Stolt, A., Linderoth, M., Robertsson, A. and Johansson, R. (2011), "Force controlled assembly of emergency stop button", *Proceedings of International Conference on Robotics and Automation (ICRA) 2011*, Shanghai, 9-13 May, pp. 3751-3756.

Stopp, A., Baldauf, T., Hantsche, R., Horstmann, S., Kristensen, S., Lohnert, F., Priem, C. and Ruscher, B. (2002), "The manufacturing assistant: safe, interactive teaching of operation sequences", *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 386-391.

TAPAS (2010), "Robotics-enabled logistics and assistive services for the transformable factory of the future (2010)", *Project under the European Community's Seventh Framework Programme*, available at: www.tapas-project.eu/

Wang, L., Tian, Y. and Sawaragi, T. (2011), "Case-based automatic programming in robotic assembly production", *Industrial Robot: An International Journal*, Vol. 38 No. 1 pp. 86-96.

Further reading

Mandow, A., Gòmez de Gabriel, J.M., Martìnez, J.L., Munoz, V.F., Ollero, A. and Garcìa-Cerezo, A. (1996), "The autonomous mobile robot AURORA for greenhouse operation", *IEEE Robotics & Automation Magazine*, pp. 18-28.

Corresponding author

Casper Schou can be contacted at: cs@m-tech.aau.dk

Paper B

Human Assisted Computer Vision on Industrial Mobile Robots

Rasmus S. Andersen, Casper Schou, Jens S. Damgaard, Ole Madsen, and Thomas B. Moeslund

The paper has been published in the
Proceedings of 1st AAU Workshop on Human-Centered Robotics, Vol. 1,
pp. 15–21, 2013.

Human Assisted Computer Vision on Industrial Mobile Robots

Rasmus S. Andersen * **Casper Schou** * **Jens S. Damgaard** *
Ole Madsen * **Thomas B. Moeslund** **

* *Dep. of Mechanical and Manufacturing Engineering, Aalborg University, Denmark*
** *Dep. of Architecture, Design and Media Technology, Aalborg University, Denmark*

ABSTRACT

Much research is directed at developing increasingly efficient and flexible production, and one important potential advancement is *Autonomous Industrial Mobile Manipulators* (AIMM's). The idea behind AIMM's is to have robots that have the ability to perform a wide variety of tasks, and which can easily and efficiently be reconfigured when the requirements changes. In this paper, the paradigm of *skill based* programming is investigated, and in particular how computer vision abilities can be integrated in this. Three applications of computer vision developed in a skill based framework are presented; namely vision pick, quality control, and fast calibration. All three are implemented on Aalborg University's AIMM, *Little Helper*, and tested in a real-life industrial environment at the Danish company Grundfos A/S.

1. INTRODUCTION

The globalization has for several decades moved manufacturing jobs from western countries to low-wage developing countries. This has put pressure on both wages and the productivity of production in the industrialized countries. One efficient way of increasing productivity is to automate production by using robots. A major limitation for the application of robots is, however, the scale of production. Construction of an automated production line is a major investment, and configuration of robots to perform the required operations is a time consuming task, that must be performed by highly specialized engineers. Thus, installation of a new, fully automated production line can only be justified if the quantity of identical items to be produced is very large. Robots have therefore proven to be particularly useful in industries such as in the car manufacturing industry, where a large quantity of identical products have to be produced.

For many kinds of production, the amount of identical items is, however, not large enough to justify investment in automated robotic production lines. Much research have therefore been directed towards developing more flexible types of automation. The organization *European Robotics Technology Platform* (EUROP) published in 2009 a *Strategic Research Agenda for European Robotics* (SRA), which outlined areas that European robotics research should focus on as well as metrics for each area EUROP (2009a,b). The core requirements for future robotics include:

- Reconfigurability: It must be possible to reconfigure both robots and other production hardware fast and easy, to prevent expensive idle time for long periods between production of (possibly small) production series.
- Human Robot Interaction: The communication between robot and human operators must be intuitive and to an increasing degree use languages and interfaces that are natural to humans.
- Autonomy (ability to function in less structured environments): On a car manufacturing plant, robots work in highly structured environments, virtually without any human presence at all. This approach is not sufficient for smaller production series. Thus, the robots must have a larger degree of autonomy, enabling them to perform tasks in dynamic environments, that cannot be precisely modeled before production begins.

One type of industrial robot that is well suited for such a production scenario, is the *Autonomous Industrial Mobile Manipulator* (AIMM). Although AIMM's are not yet in industrial use, they are already able to move autonomously around in changing environment and perform a wide variety of tasks. Since AIMM's must be designed to function in less than fully structured environments, they depend very much on their ability to sense both objects and the world around them. The focus of this paper is to investigate methods to do such sensing by using computer vision in a fast and easily reconfigurable way.

1.1 Related Research

Ordinary RGB cameras are used to give vision functionality to robots in various fields, including navigation, object manipulation, and interaction with humans, cf. Hvilshøj et al. (2009); Guizzo and Ackerman (2012); Nava et al. (2011). The human visual system includes additionally information about depth, and several approaches have been taken to provide this information to robots also, including stereo vision (Murray and Little, 2000), time-of-flight (TOF) depth cameras (Klank et al., 2009), and

⋆ This research was partially funded by the European Union project TAPAS under the Seventh Framework Programme.

depth cameras based on structured light (Siegwart and Nourbakhsh, 2004). While stereo vision is the closest analogue to the human visual system, it is far simpler to use active technologies such as TOF or structured light. With the launch of the Microsoft Kinect in 2010, which combines an ordinary RGB camera with a depth camera based on structured light, the price and accessibility of quality depth video imaging was all of a sudden reduced dramatically (Shotton et al., 2011; El-laithy et al., 2012). This dramatically increased the scientific interest in taking advantage of depth information in combination with RGB images for all areas, where RGB images was also previously used Tölgyessy and Hubinskỳ (2010); Benavidez and Jamshidi (2011); León et al. (2011). Recently, the smaller but equally powerful competitor *Asus Xtion Pro Live* was launched.

Since AIMM's must have the ability to move between workstation, calibration to new workstations is a particular useful aspect, which has also received some attention in the literature. In 2000, a general method for camera calibration was developed by Zhang (2000). This has later become extremely popular, due to implementations provided both for C/C++ in OpenCV and for Matlab in the Matlab camera calibration toolbox. This is designed specifically for estimating parameters, intrinsic as well as extrinsic, for cameras, and not directly applicable for calibration of robots. Another approach by Alici and Shirinzadeh (2005) calibrates industrial robots with very high precision, but this require a laser tracker to be located close to the calibration point. Thus, it is not suitable for AIMM's, which should be able to work in industrial environments without requiring extensive and/or expensive changes.

Two approaches from Hvilshøj et al. (2010) are specifically developed to AIMM's. A fast approach use in addition to a camera a laser for distance measurements, and a slower but very precise method makes only use of a camera on the tool. Both methods have, however, disadvantages: The fast approach requires that a laser is mounted on the tool of the robot. More equipment on the tool means less possible payload, and must therefore be avoided if possible. In the more precise approach, a large number of images are captured of a calibration board, and the execution time is about 60 seconds. If the robot is moving frequently between workstations, such non-productive time must be minimized.

A last approach, described in Pedersen (2011), applies haptic rather than vision based calibration. This approach is a able to calibrate very precisely in about 30 seconds by measuring locations on the workstation in three orthogonal directions. The disadvantage with this method is, in addition to the relatively long execution time, that the workstation must be have large surfaces in all three directions. Also, it is only applicable on robots with force feedback control.

1.2 Skill Based Computer Vision

A traditional and widely used way of programming robots is the *Sense-Plan-Act* (SPA) paradigm (Nilsson, 1993). Using this, the robot moves between the three states: Sense, plan and act. In the sensing state, information from sensors are used to update and maintain a world model. In the planning state, high level logic plans on basis of this world model what the robot has to do, and in the acting state the plan is carried out, typically using control theory. Two limitations of the SPA paradigm is that it does not well support reusability of code, and that the complexity of maintaining a complete world model can be very high.

The paradigm of robot skills attempts to counter both limitations of the SPA paradigm by introducing a layered architecture, where each layer executes its own SPA loop. The idea of using layers to provide better possibilities for reusing code was presented as early as in 1986 by Brooks (1986), but research to provide even more reusable and more generic solutions continue, cf. Gat (1998); Bjorkelund et al. (2011). In the skill paradigm, programming is divided in three layers. Different naming conventions exist, and here the layers are named *device primitives*, *skills*, and *tasks*. The purpose of the layered programming structure is to wrap the difficult and low level robotic knowledge in the lower levels, allowing non-expert users to focus on teaching tasks on a much higher level.

In the ongoing research project 'Little Helper' at Aalborg University, AIMM's have been developed on the basis of the skill paradigm since 2008. In close collaboration with both academic and industrial partners, it is attempted to make the technology ready for industrial use.

In this paper, the integration of computer vision in the skill based framework is presented. The vision algorithms are implemented using a commercial computer vision system based on *Labview* as well as the open source library *zbar*, and the focus here is on how to integrate and use this in the skill based robotic framework. First, the skill paradigm and the vision system applied are described in detail. Subsequently, three developed applications of computer vision are described: A generic pick skill using vision, quality control integration, and a fast calibration based on recognizing QR codes. Finally it is discussed how the results can be generalized, and where future research should be directed. The results presented in the paper are from a midway demonstration in the EU project TAPAS, performed in a factory owned by the Danish company Grundfos A/S.

2. METHODS

2.1 The Concept of Robot Skills

The architecture in the skill paradigm used here consists of three layers:

(1) **Device primitives:** Basic functions of one device, such as the robot, tool or a camera. Example: *Open gripper*.
(2) **Skills:** A predefined sequence of device primitives, that form a coherent action. In Björkelund et al. (2011), a skill is defined as "productive sensor-based robot motions". Example: *Pick up object* O_1.
(3) **Tasks:** Responsible for achieving the overall goals of the robot, while at the same time completely decoupled from the internals of the robot. The robot itself can thus in principle be replaced without replacing the task layer, as long as the new robot provides the same skills. Example: *Pick up 10 units of object* O_1 *at location* L_1*, and place them in a bin at location* L_2.

It does only make sense to execute a place skill, if the robot is holding an object. This means that a precondition for a place skill is, that an object is held. In general a skill has both pre- and post-conditions, and the skill only functions if these are met. This property is in general called that skills are *situated*.

In the skill paradigm, each skill has a *teach* and an *execute* phase. If the user wants the robot to pick up an object of type O_1 from location L_1, a pick is chosen and taught. After teaching is completed, the robot is able to execute the same skill, thus picking a new object of the same type from location L_1 on its own. Prerequisites include here include that the robot is already located at (or close to) location L_1, and that an object of type O_1 is present at the location.

2.2 Flexible Setup with External Computer Vision System

The computer vision system used here is based on the Vision Builder software in Labview. It is able to perform a large number of 2D vision tests, and it employs an intuitive interface, allowing non-experts to configure it with very little training. A screen image is shown in Figure 1.

This paper is, however, not concerned with the vision system itself, but with the integration and use of vision systems in general in the skill based robotic framework. The vision system is not installed on the robot itself. Instead, a protocol based on TCP/IP has been developed, which allows the robot to communicate with an external vision system. To be able to integrate the robot in different kinds of production lines, support for both local camera (mounted on the robot) as well as external cameras has been included in the protocol. For the experiments using the vision system that are described here, two cameras are used; one placed on the tool of the robot, and one fixed at an assembly cell. Both cameras are of the type DMK 31BF03-Z2, which have motorized zoom and a resolution of 1024×768. The first camera is used to detect and pose estimate rotor caps to be able to pick them up, while the fixed camera is used to perform quality control during assembly. Each of these cases are presented in the following subsections, along with the remaining experiment; providing fast calibration by pose estimation of QR codes.

2.3 Generic Vision Pick

The developed vision pick skill consists like all skills of a training and an execution phase. In the training phase, the following parameters are taught:

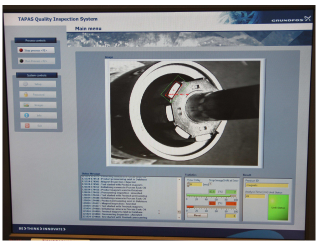


Fig. 1. Vision detection system in execution mode.

Start and end position of camera: These positions define both the route the camera will take when searching for an object to pick up, as well as an *acceptance region* for objects. During execution, the robot will move the camera from the start towards the end position. With short intervals, the robot will stop and grab an image in search of an object to pick up. Whenever an object has been found, it is calculated if the object is located in the acceptance region; between the two points. If this is the case, the robot cancels the movement towards the endpoint and picks up the object instead.

Detection height: Height of the feature, that the vision system is able to detect.

Grasping height: Appropriate height for grasping the object.

All the required parameters are taught by manually moving the robot arm around, and thus no programming skills are required. The parameters are illustrated in Figure 2. The principle for execution is to first capture an image, and then try to detect the location of a particular feature on the object to pick up in this image. This 2D position can be transformed into a 3D vector from the camera's focal point to the image plane, by applying the intrinsic parameters of the camera. By extending this vector, it will ultimately intersect the plane with the (taught) detection height, as illustrated in Figure 2. If this 3D intersection point is located in the acceptance region, the object can be picked. An additional height; the grasping height, is taught during training, and this enables the robot to grasp the object on a suitable position. The algorithm for execution of the vision pick skill is described in detail in Table 1.

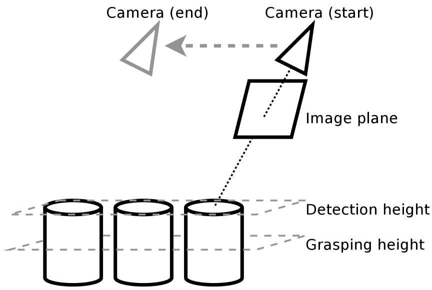


Fig. 2. Execution of the vision pick algorithm require the parameters $\boldsymbol{p}_{\text{cam,start}}$, $\boldsymbol{p}_{\text{cam,end}}$, h_{detect}, and h_{grasp} which are shown in the figure. All the parameters are specified during teaching.

The setup is shown in Figure 3 for two different locations. The Figures 3(a) and 3(c) show the robot searching for rotor caps, while Figure 3(b) and 3(d) show the robot actually picking up a robot cap. Note that the same skill is used at the different locations; only the parameters that are set during teaching differ.

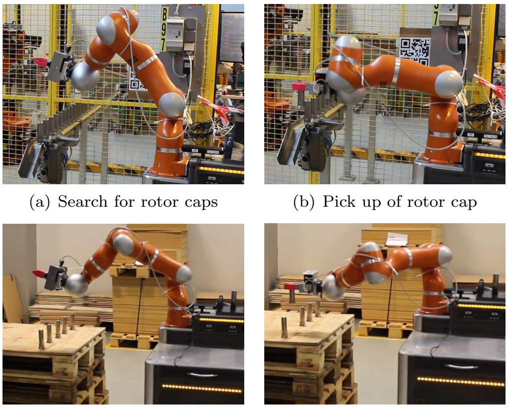
2.4 Quality Control

As with the vision pick skill, also the quality control is developed by utilizing the vision system described in Section 2.2. Thus, both cameras on the robot as well as external cameras can be used. Configuring quality control

(1) Move camera to (taught) start position, $\boldsymbol{p}_{\text{cam,start}}$.
(2) Capture image and send it to the vision system.
(3) Vision system detects object in the (2D) image, and returns this position to the AIMM.
(4) **IF** *the AIMM does not receive a valid position*
THEN
- Move camera one step along the line from $\boldsymbol{p}_{\text{cam,start}}$ to $\boldsymbol{p}_{\text{cam,end}}$.
- **IF** *the camera already was at* $\boldsymbol{p}_{cam,end}$
THEN the skill has failed
ELSE continue at 2.

(5) Calculate a 3D line from the camera through the (undistorted) image location, received from the vision system.
(6) Calculate the intersection point between the line and the horizontal plane with the (taught) height h_{detect}. This gives the 3D position of the object in camera space.
(7) Transform the object position from the camera space to the robot's base space.
(8) Replace the height of the position with the (taught) value, h_{grasp}, to make the robot grasp the object at a suitable location.
(9) Project the object position on the line between $\boldsymbol{p}_{\text{cam,start}}$ and $\boldsymbol{p}_{\text{cam,end}}$.
(10) **IF** *the projected object position is between* $\boldsymbol{p}_{cam,start}$ *and* $\boldsymbol{p}_{cam,end}$
THEN pick up the object at the calculated 3D position
ELSE the object position is not in the acceptance region, continue at 2.

Table 1. Algorithm for execution of vision based pick skill.



(a) Search for rotor caps (b) Pick up of rotor cap

(c) Search for rotor caps (d) Pick up of rotor cap

Fig. 3. Execution of vision pick skill. The robot captures images while moving in (a). The images are send to the quality control system shown in Figure 1, and whenever a rotor cap is detected, the robot picks it up, as shown in (b). Figures (c) and (d) show the same skill executed at a different location.

is mainly done at the vision system, and the robot itself only needs to know the name of the particular test to perform. In the skill framework, quality control can in general be viewed as a post-condition check, and if this fails, appropriate handling must be implemented. For the tests described here, this can either be to report an error, or to wait a short while and try again. The algorithm for execution of the developed quality control skill is shown in Table 2.

(1) The AIMM signals to vision system to perform (taught) quality control.
(2) Vision system performs control, and replies success/failure.
(3) **IF** *success*
THEN the AIMM continues
ELSE perform appropriate error handling (wait and try again, or call operator).

Table 2. Algorithm for execution of quality control skill.

2.5 Fast calibration

As mentioned in the introduction, the purpose of this skill is to provide calibration in three dimensions, faster than the existing calibration approaches developed by Hvilshøj et al. (2010), which have durations of 10 seconds and above. This is attempted by using the Kinect-like camera Asus Xtion Pro Live, that provides calibrated and undistorted images in both RGB and depth. In our approach, the calibration is implemented as a unique skill, thus having both a teaching and an execution phase. The phases are, however, almost identical. The purpose of both teaching and execution is to find the coordinate system of the (fixed) QR code; the QR frame. Subsequently all locations must be given relative to this frame.

When initiating the calibration skill, the RGB images from the Xtion camera are searched for QR codes. There are several libraries available that provide this functionality, and here *zbar* is chosen, because this directly provide the location of the corners in the images. When a QR code has been found, the depth at each corner of the QR code is averaged over a number of images, and the QR code's coordinate system can then be calculated as:

$$\boldsymbol{x} = \frac{\boldsymbol{c}_0 - \boldsymbol{c}_3}{|\boldsymbol{c}_0 - \boldsymbol{c}_3|} \tag{1}$$

$$\boldsymbol{y} = \frac{\boldsymbol{c}_2 - \boldsymbol{c}_3}{|\boldsymbol{c}_2 - \boldsymbol{c}_3|} \tag{2}$$

$$\boldsymbol{z} = \frac{\boldsymbol{x} \times \boldsymbol{y}}{|\boldsymbol{x} \times \boldsymbol{y}|} \tag{3}$$

where $\boldsymbol{c}_{\text{n}}$ is the location of the n'th corner of the QR code, and the corners are numbered clockwise.

To be able to work in this coordinate system it must be converted into a complete transformation matrix. This is done by calculating a *translation* and a *rotation*. The translation $\boldsymbol{t}_{\text{QR}}$ is defined by the center of the QR code, and is thus calculated as the mean of the corners:

$$\boldsymbol{t}_{\text{QR}} = \sum_{n=0}^{3} \frac{\boldsymbol{c}_{\text{n}}}{4} \tag{4}$$

B.3. Results

The rotation $\boldsymbol{r}_{\mathrm{QR}}$ is best defined as Euler angles, which can be calculated directly from the axes. The transformation from the QR code to the camera, ${}^{\mathrm{C}}_{\mathrm{QR}}T$, can be determined by combining the translation and rotation. The desired transformation is between the robot's base and the QR code, ${}^{\mathrm{B}}_{\mathrm{QR}}T$, and this is computed as:

$$ {}^{\mathrm{B}}_{\mathrm{QR}}T = {}^{\mathrm{B}}_{\mathrm{C}}T \cdot {}^{\mathrm{C}}_{\mathrm{QR}}T \tag{5} $$

where ${}^{\mathrm{B}}_{\mathrm{C}}T$ is the (fixed) transformation from the camera to the robot's base.

Finally, the coordinate system given by this transformation is applied to the robot. The exact sequence of execution of the calibration skill is given in Table 3.

---

(1) **WHILE** *correct QR code not found*
- Search for QR code in RGB images
- Read QR code
- **IF** *the text of the QR code matches taught string* **THEN** exit while loop

(2) Capture a number of depth images.

(3) **FOR** *each corner of the QR code*
- At the location of the corner, calculate the mean of the depth values (ignore 0-values).

(4) **IF** *one or more corners have no depth values* **THEN** the skill has failed. Exit.

(5) Define a coordinate system at the QR code as in Equations (1)-(3).

(6) Calculate the **translation** of the QR code $\boldsymbol{t}_{\mathrm{QR}}$ as the mean of the corners.

(7) Calculate the **rotation** of the QR code's coordinate system $\boldsymbol{r}_{\mathrm{QR}}$ in Euler angles.

(8) Combine translation and rotation into a transformation matrix, ${}^{\mathrm{C}}_{\mathrm{QR}}T$.

(9) Calculate the transformation from the QR code's coordinate system to the robots base coordinate system, ${}^{\mathrm{B}}_{\mathrm{QR}}T$, as in Equation (5).

(10) Set the robots frame base to ${}^{\mathrm{B}}_{\mathrm{QR}}T$.

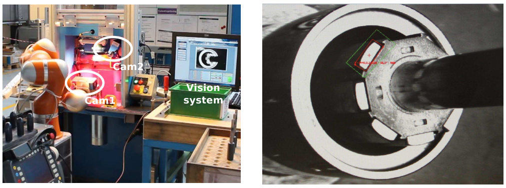
Table 3. Algorithm for execution of calibration skill based on QR codes. The Asus Xtion Pro Live was used for capturing RGB and depth images.

3. RESULTS

The three applications of computer vision have all been implemented on Aalborg University's AIMM *Little Helper*, and tested in a real-life industrial environment at a Grundfos factory. The vision pick skill was able to successfully pick an arbitrary number of rotor caps from two different locations, as shown in Figure 3. The precision was within ±5 mm, which was sufficient to correctly place the rotor caps at the desired locations afterwards.

The quality control was used for a variety of different tests. The application of this integration is only limited by the capabilities of the vision system itself, which is not described here. An example is shown in Figure 4, where it is detected that a magnet has been correctly placed beside the rotor core.

The setup for using the calibration skill is shown in Figure 5. The switch in the Figure is used to enable and disable the conveyor belt. The purpose of the calibration is here to make it possible for the robot to operate the switch,



(a) Assembly (b) Quality control

Fig. 4. Quality control setup. The robot is performing assembly tasks to the left in (a), while the quality control system is running externally, shown on the screen to the right. Figure (b) shows a close up of the result. The green box is the region of interest (ROI), and the red marking is the detected magnet.

and the position of the switch can thus be considered as a position of interest. The position of the QR code relative to the position of interest of course affects the calibration precision, and especially three factors affect the overall precision:

(1) The position estimate of the corners of the QR code the the cameras RGB image. These positions can be determined with sub-pixel accuracy, and at a distance of about 1 m as used in this setup, the precision of the corners is within ±1 mm.
(2) The relative error in the depth values at the corners for repeated measurements. The depth sensor in the Asus Xtion is the same as in the Kinect, and the absolute precision of the depth values provided by the Kinect has been shown to be within ±10 mm for distances between 0.8 m and 3.5 m when used indoor (El-laithy et al., 2012). No data are available on the relative repeatability error, but it has proven to be significantly smaller.
(3) The relative error in the depth values between the corners. No data are available on this precision, but this has also proven to be significantly less that the absolute error.

Especially the third factor; the relative error between the corners, is of interest, because this will cause the coordinate system at the QR code to have a slightly wrong orientation. A wrong orientation makes the error increase the longer the distance between the QR code and

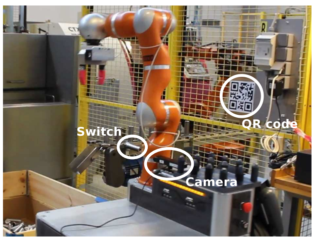


Fig. 5. Fast calibration using the Asus Xtion camera featuring calibrated RGB and depth images. The camera detects the pose of the fixed QR code, and subsequent movements with the robot are corrected accordingly.

the position of interest, and this is also what was found to be the case in the test scenario. Although no formal measurement of the precision has been carried out, visual inspection has shown that the precision is at least ±10 mm at any position. This precision proved to be sufficient to make it possible to operate the switch. In Table 4, the proposed calibration is compared to existing methods.

4. DISCUSSION

Integration of computer vision abilities into a skill based framework proved to be possible, and in this paper, three applications were successfully implemented. In particular the implemented quality control is very generic, and using the developed TCP/IP based protocol, the vision system could be changed without making any changes to the robot itself. This is also the case for the pick skill; however this has in the current implementation some limitations. It is currently assumed that the items to pick up are approximately placed in a line, as is for instance the case on a conveyor belt. Thus during teaching, the start and end location of the camera are taught. A further development should make it possible to define an arbitrary search region during teaching of the skill. For this, an optimal search pattern should automatically be calculated by the robot, taking into account that objects closest to the robot must be picked first. Positions and orientations of the camera during search should also be automatically determined.

The implemented calibration skill makes it possible to perform a very fast calibration compared to existing methods. This is especially important for industrial robots that are moving frequently between workstations. The precision was sufficient to perform the experiments described here, but for high-precision tasks it will be insufficient. There are two obvious ways of doing this:

- The Asus Xtion camera used, does in principle support RGB images with a 1280×960. A bug in the available open source drivers limited, however, the available resolution in our implementation to 640×480. Use of the full resolution images will definitely increase the precision of the QR code detection.
- From the depth image, only the four corner points were used. A better performance could be achieved by using the entire surface of the QR code, for instance by applying the RANSAC algorithm to filter out outliers.

It is impossible to say how much the precision can be improved. However, an experiment should be carried out to determine the precision exactly.

Method	Duration	Precision
Haptic[1]	30-45 sec	±1.0 mm
High speed[2]	10 sec	±1.0 mm
High precision[2]	60 sec	±0.1 mm
Proposed method	<1 sec	< ±10 mm

Table 4. Comparison of calibration methods. 1 are from Pedersen (2011); 2 are from Hvilshøj et al. (2010).

ACKNOWLEDGEMENTS

This research was partially funded by the European Union project TAPAS under the Seventh Framework Programme. The applications developed here was tested in an industrial environment at a factory owned by the Danish company Grundfos, and this was only possible due to a very close collaboration. Especially, thanks to J. Bigum and R. Larsen for assisting in developing the applied computer vision system.

REFERENCES

Alici, Gürsel and Shirinzadeh, Bijan. A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. *Mechanism and Machine Theory*, 40(8):879–906, 2005.

Benavidez, Patrick and Jamshidi, M. Mobile robot navigation and target tracking system. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*, pages 299–304. IEEE, 2011.

Bjorkelund, Anders; Edstrom, Lisett; Haage, Mathias; Malec, Jacek; Nilsson, Klas; Nugues, Pierre; Robertz, Sven Gestegard; Storkle, Denis; Blomdell, Anders; Johansson, Rolf; Linderoth, Magnus; Nilsson, Anders; Robertsson, Anders; Stolt, Andreas, and Bruyninckx, Herman. On the integration of skilled robot motions for productivity in manufacturing. In *Assembly and Manufacturing, IEEE International Symposium on*, pages 1–9, 2011.

Björkelund, Anders; Malec, Jacek; Nilsson, Klas, and Nugues, Pierre. Knowledge and skill representations for robotized production. In *Proc. 18th IFAC World Congress*, 2011.

Brooks, Rodney. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, March 1986.

El-laithy, RA; Huang, Jidong, and Yeh, Michael. Study on the use of microsoft kinect for robotics applications. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 1280–1288. IEEE, 2012.

EUROP. Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP, 2009a.

EUROP. Application requirements. Appendix to: Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP, 2009b.

Gat, Erann. On three-layer architectures. In Kortenkamp, David; Bonnasso, Peter R., and Murphy, Robin, editors, *Artificial Intelligence and Mobile Robots*, pages 195–210, 1998.

Guizzo, Erico and Ackerman, Evan. The rise of the robot worker. *Spectrum, IEEE*, 49(10):34 –41, october 2012. ISSN 0018-9235. doi: 10.1109/MSPEC.2012.6309254.

Hvilshøj, Mads; Bøgh, Simon; Madsen, Ole, and Kristiansen, Morten. Calibration techniques for industrial mobile manipulators: Theoretical configurations and best practices. In *Robotics (ISR), 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–7. VDE, 2010.

Hvilshøj, Mads; Bøgh, Simon; Madsen, Ole, and Kristiansen, Morten. The mobile robot "Little Helper":

Concepts, ideas and working principles. In *ETFA*, pages 1–4, 2009.

Klank, Ulrich; Pangercic, Dejan; Rusu, Radu Bogdan, and Beetz, Michael. Real-time CAD model matching for mobile manipulation and grasping. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 290–296. IEEE, 2009.

León, Adrián; Morales, Eduardo; Altamirano, Leopoldo, and Ruiz, Jaime. Teaching a robot to perform task through imitation and on-line feedback. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 549–556, 2011.

Murray, Don and Little, James J. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.

Nava, Federico; Sciuto, Donatella; Santambrogio, Marco Domenico; Herbrechtsmeier, Stefan; Porrmann, Mario; Witkowski, Ulf, and Rueckert, Ulrich. Applying dynamic reconfiguration in the mobile robotics domain: A case study on computer vision algorithms. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 4(3):29, 2011.

Nilsson, Nils J. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers, January 1993. ISBN 0934613109.

Pedersen, M. Rath. Integration of the kuka light weight robot in a mobile manipulator. Master's thesis, Aalborg University, 2011.

Shotton, Jamie; Fitzgibbon, Andrew; Cook, Mat; Sharp, Toby; Finocchio, Mark; Moore, Richard; Kipman, Alex, and Blake, Andrew. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. IEEE, 2011.

Siegwart, Roland and Nourbakhsh, Illah R. *Introduction to autonomous mobile robots*. MIT press, 2004.

Tölgyessy, Michal and Hubinskỳ, Peter. The kinect sensor in robotics education. In *Proceedings of 2nd International Conference on Robotics in Education*, 2010.

Zhang, Zhengyou. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

Paper C

Fast Calibration of Industrial Mobile Robots to Workstations using QR Codes

Rasmus S. Andersen, Jens S. Damgaard, Ole Madsen, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 44th International Symposium on Robotics (ISR), 2013.
Won the Best Paper Award

Paper D

Using Robot Skills for Flexible Reprogramming of Pick Operations in Industrial Scenarios

Rasmus S. Andersen, Lazaros Nalpantidis, Volker Krüger, Ole Madsen, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP), Vol. 3, pp. 678–685, 2014.

Excluded in the public version of the thesis due to copyright issues.

Paper E

Adaptive Model Based Quality Inspection

Rasmus S. Andersen, Ole Madsen, and Thomas B. Moeslund

This technical report is synthesized from deliverable 3.8 in the TAPAS project [TAPAS, 2014].

Adaptive Model Based Quality Inspection

E.1 Introduction

Human workers naturally perform a visual inspection of all tasks that they carry out. This is also necessary for a flexible collaborative robot if more advanced tasks are to be carried out. In [Andersen et al., 2013], quality control is integrated into the skill based system by allowing the robot to communicate with an external vision system which itself is designed with a simple, intuitive interface. This approach allows a wide variety of tests to be used, but it does require a human to explicitly choose which test to use in each scenario.

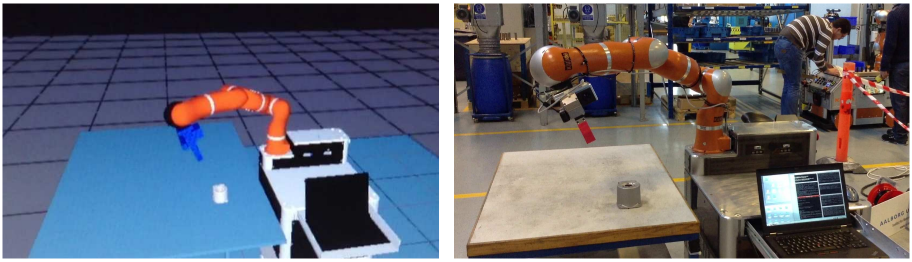
As part of the TAPAS project [TAPAS, 2014], a fully autonomous quality inspection skill was developed in cooperation with the partner company CIT[1]. The skill detects errors based on a CAD model and an approximate position of the object to inspect only. The system combines motion and next-best-view planning developed by CIT with error detection developed as part of this PhD. It is designed to inspect industrial objects in the TAPAS scenario presented in Paper A [Madsen et al., 2015]. The scenario is shown in Figure E.1. Specifically the requirements are:

- Inspection of texture-free shiny metal objects
- Flexibility to handle a large variety of objects
- Feedback-control to adjust for its own limitations

E.2 Evaluation of Depth Cameras

To compare an object to its model, the object shape must be acquired. Large and expensive systems for object scanning, typically based on laser scanning, do exist and are in use in the industry. However, these cannot easily be fitted onto a collaborative robot. Instead, the performance of cheaper off-the-shelf depth sensors is tested and compared on the metallic rotor in Figure E.2.

[1] Convergent Information Technologies (CIT), `http://www.convergent-it.at/`



(a) Simulation (b) Actual

Fig. E.1: Simulated and actual capturing of point clouds from different angles.

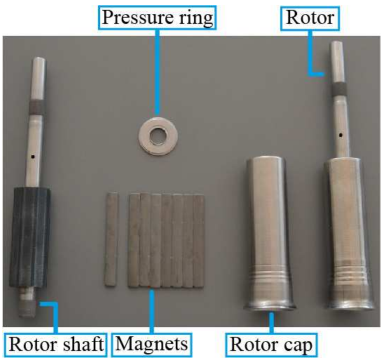


Fig. E.2: All the objects used for assembly in the TAPAS project at Grundfos A/S. In industrial scenarios such as this, many objects are metallic and shiny. The cameras used for inspection must therefore be able to detect such surfaces. The assembled rotor to the right in the image is used as test objects for comparing cameras.

For the inspection task, the camera must be able to detect the metallic surfaces. This must be possible on distances short enough to allow a robot arm holding the camera to capture views from multiple angles. The robot arm used for the experiment is the KUKA LWR 4 which has a reach of 1.178 m. Therefore, the surfaces should preferably be clearly detectable on distances of 0.5 m or shorter. The necessary accuracy depends on the size of production errors that should be detected, and no absolute number has been specified for the current project. However, the more accurate that the shape of the surface can be estimated the better.

Figure E.3 shows three test images from each tested camera and the performance of the cameras is assessed in Table E.1.

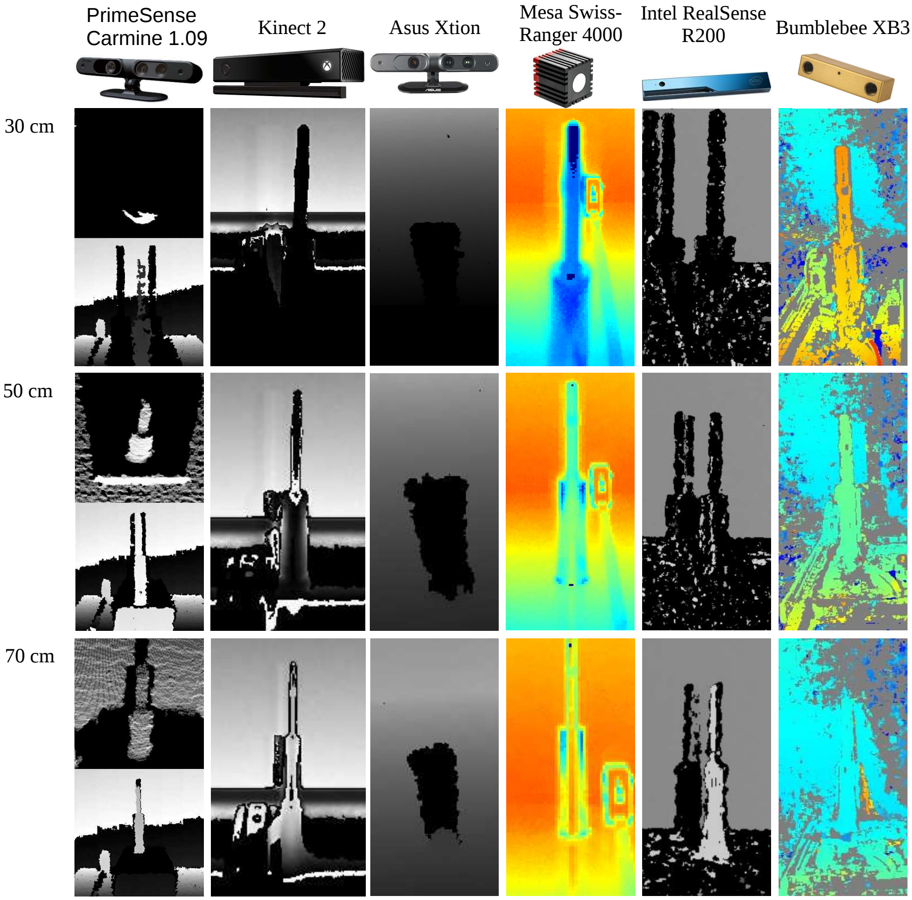


Fig. E.3: Depth maps from different cameras of a metallic and reflective object. As test object, an assembled rotor of the Grundfos SQFlex pump is used (see Figure 5.4(b) top-left). Depth maps are shown for each camera with the rotor placed in 30 cm, 50 cm, and 70 cm away. For the PrimeSense Carmine 1.09, point clouds are also shown seen inclined from above.

	Camera	Dist/m	Res	Assessment
Structured light	Asus Xtion	0.8-3.5	320x240 (640x480)	The sensor is actually able to get data from a closer distance than the specified 0.8 m as seen on the table surface. The metallic surface is not detected properly, though, at any distance.
	PrimeSense Carmine 1.09	0.35-1.4	320x240 (640x480)	The sensor provides valid object data on 50 and 70 cm and even some data on 30 cm. The rounded shape is correctly detected, except for the sides with high surface angles relative to the viewpoint of the camera.
	Intel RealSense R200	0.5-3	640x480	On 70 cm, data is acquired that is comparable to the Carmine 1.09. On closer distances, the object is, however, not detected.
Time-of-Flight	Mesa SwissRanger 4000	0.8-5 (0.1-5)	176x144	Data is acquired on all distances, but the depth values are highly dependent on the surface angle seen from the camera. This is especially pronounced in the center part of the rotor cap on 70 cm and on the top corners on both 50 and 70 cm.
	Kinect 2	0.5-8	512x424	The images are generated by taking the least significant bit in the depth values, and this causes increasing depths to be shown as repeated patterns of black-to-white. Data is acquired on 50 and 70 cm, but the same issue with the surface angle is present as with the SwissRanger. The border regions have wrong depth values while the center region on 70 cm is not registered.
Stereo	Bumblebee XB3 (narrow view)	—	1280x960	A baseline of 12 cm has been used with the Triclops stereo algorithm that comes with the Bumblebee camera. Data is acquired on all distances and although there are holes in the depth map, the density on the object surface is acceptable.

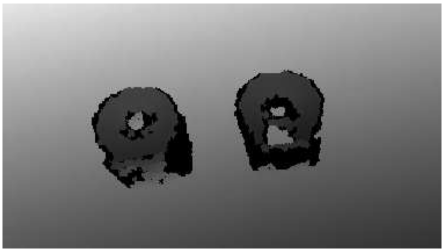
Table E.1: Comparison of cameras. Resolutions in parenthesis are upscaled and distances in parenthesis are specified as non-optimal.

The quality of the depth maps is best for the PrimeSense Carmine 1.09 and the Bumblebee XB3. The remaining cameras are either unable to detect the rotor on short distances, or show artifacts when the surface angle is either very large or very small seen from the camera. The resolution of the Bumblebee camera is higher than that of the Carmine, but the Carmine, on the other hand, produces depths almost without holes on the surface of the rotor. Also, the Carmine is significantly smaller and therefore fits better on a robot. The PrimeSense Carmine 1.09 is therefore chosen for the quality inspection system.

E.3 Point Cloud Acquisition and Matching

Figure E.4 shows images of the test objects used for the quality inspection taken with the PrimeSense camera. The goal for this object is to autonomously detect the cavity error present in the right-most object.

(a) RGB image



(b) Depth map

Fig. E.4: Test objects without/with a cavity production error. The images are captured with a PrimeSense Carmine 1.09.

As seen in Figure E.4(b), the error is clearly detectable by the depth camera. However, not all visible parts of the surface are detected. It cannot be predicted exactly which surfaces that will be detected, and several iterations of capturing images might therefore be necessary. Figure E.5 illustrates the vision-and-planning flow through the system. The inspection skill needs to be parameterized with the object type, surfaces to inspect, and approximate expected location of the object. The object type and surfaces to inspect can be chosen by an operator through a GUI. The inspection surfaces can either be all outer surfaces or a subset of these. Next, the operator teaches the robot where to look for the object and this concludes the parameterization.

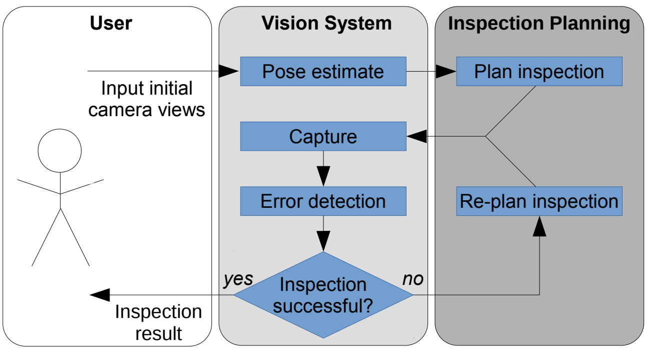
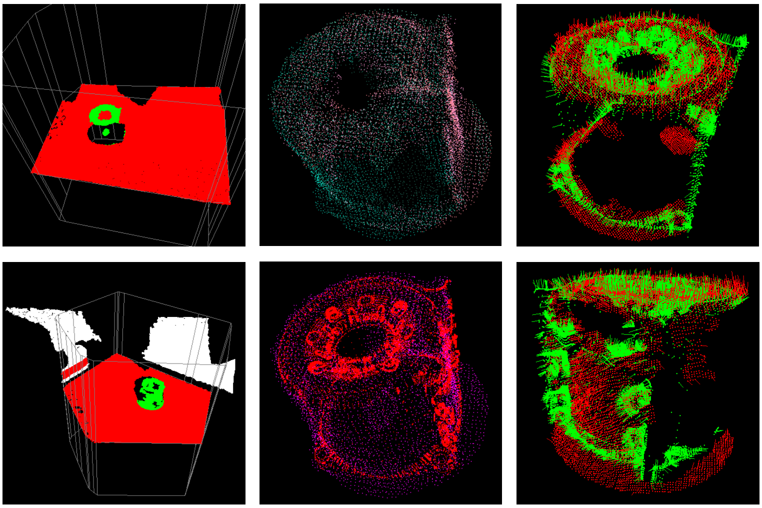


Fig. E.5: Adaptive planning for quality control.

When the skill is executed, the robot first estimates the exact pose of the object. The pose is then used to plan an inspection of the selected surfaces, including viewpoints, trajectories between viewpoints, and expected surfaces at each viewpoint. The vision system captures a point cloud from every viewpoint and uses these to estimate both whether there are errors and whether the entire specified surface has been successfully detected. If a large area could not be detected, a new inspection can be planned to cover the remaining parts of the surface. When the specified surface has been detected to a sufficient degree, the model is analyzed from each viewpoint to determine if significant errors are present.

The point cloud segmentation and matching is illustrated in Figure E.6. In (a), the object is detected and segmented from the dominant plane in the scene. The reader is referred to Paper D for a more thorough description of the segmentation algorithm.



(a) Segmentation.

(b) Registration (top) and matching (bottom).

(c) Expected (green) and detected (red) point clouds with normals.

Fig. E.6: Object segmentation, pose estimation, and matching against expected point clouds.

The segmented object point clouds are combined into a single denser point cloud in Figure E.6(b) (top). This point cloud registration uses the pose of the camera at each viewpoint for an initial estimate and translates the point clouds for a better accuracy using translational ICP (Iterative Closest Point) [Besl and McKay, 1992]. The merged detected point cloud is shown in the bottom image in Figure E.6(b) (in purple) matched against the model (in red). This matching step utilizes surface matching from the commercial computer vision library *HALCON*[2].

E.4 Error Detection

For each view, the inspection planner generates an expected point cloud of the object to inspect. The point cloud generated based on an error-free model. Figure E.6(c) shows the expected point cloud in green for two different views. The captured and segmented point clouds are shown in red.

The goal of the error detection algorithm is to determine significant differences between the simulated and the captured point clouds. There can be (at least) two types of differences:

1. Areas of the object that could not be detected. This can be caused by limitations of the sensor.

2. Areas of the object where the two point clouds differ, which can only be caused by actual differences between the object and the model, i.e. production errors.

For the first differences of type 1, a re-inspection can be necessary to better inspect the problematic areas. Differences of type 2, on the other hand, will count as detected errors if their area exceeds a predefined threshold.

If the point clouds were compared directly point-to-point in 3D space, there would not be any way to determine if missing detected points were caused by object or sensing errors. That is, it would not be possible to distinguish between the two types of differences. Therefore, both point clouds are instead projected to the image plane of the camera to form depth maps. This is illustrated in the first two columns of Figure E.7. No information is lost in this projection because the point clouds where both simulated and captured from this exact viewpoint. Type 1 differences can now be detected as areas of the captured depth maps with *no data*, while type 2 errors will show as areas with *different values* when compared to the simulated depth maps.

Type 2 differences are shown in the two last columns of the Figure. The first of these show absolute depth differences between points, which are present in both the simulated and captured view. In the last column, these values are thresholded. Large errors are detected through a connected component analysis. In the Figure, a significant error is detected for view 2 of object 1.

Type 1 differences are shown in the middle column. A significant number of these differences will make it necessary to perform a re-inspection. This is done by first constructing a point cloud from the pixels and feed this back to the inspection planner as a new surface to inspect. The planner then generates

[2]HALCON is a commercial machine vision library and IDE developed by MVTec Software GmbH.

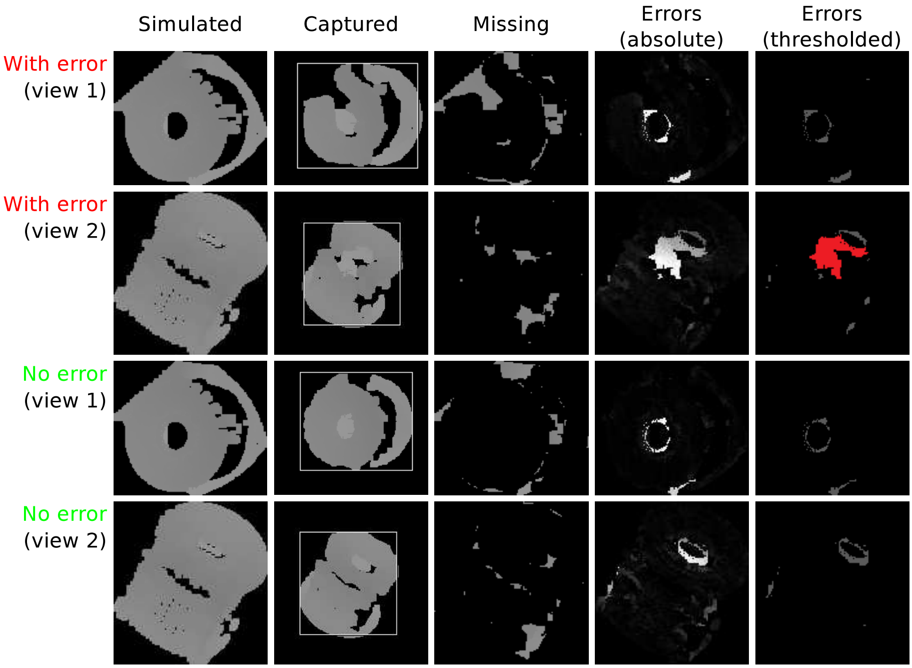


Fig. E.7: Results of adaptive quality inspection. The top two rows are with the erroneous object in Figure E.4 and the bottom two rows are with the error-free object. For both objects, two camera views were planned by the inspection planner. The first column shows the simulated point cloud projected to a depth image, and the second column shows the corresponding captured point cloud. The middle column shows the points that are missing in the captured point clouds but present in the simulation. The last two columns show the points that are present in both point clouds; with the depth differences shown as intensity or thresholded to a specific value, respectively. Connected regions larger that a certain threshold are considered as errors. This is the case for view 2 of object 1.

new camera views, and all the steps in the vision system are repeated with only these remaining points taken into account. This process can be repeated until either:

- An error is detected.
- No errors are detected and the entire surface could be detected (possibly excluding patches smaller than a predefined threshold). The product is concluded to be error-free.
- A large part of the surface could not be detected, even after re-inspections.

In the two first situations, the object is successfully inspected. In the last situation, the vision system has failed, and the product will have to be inspected by a human.

E.5 Conclusion

A fully autonomous quality inspection system has been developed by combining model-based error detection developed as part of the current PhD project with motion and next-best-view planning developed at the company CIT. The system is able to detect relatively large errors such as the cavity production error shown in Figure E.4. The contribution of the work is to show how production errors can be detected fully autonomously and how feedback from the vision system can be used to compensate for imperfect sensing data. Additionally, it is shown how this functionality can be integrated into a human-centered skill based architecture.

Errors significantly smaller than the one in Figure E.4 cannot currently be detected due to limitations of the used depth camera. However, the proposed methods can be used directly with any depth camera, and commercial depth cameras are at the moment improving fast. Significantly better cameras must be expected to be available within few years which will improve the capabilities of the system.

References

[Andersen et al., 2013] Andersen, R. S., Schou, C., Damgaard, J. S., Madsen, O., and Moeslund, T. B. (2013). Human assisted computer vision on industrial mobile robots. In *1st AAU Workshop on Human-Centered Robotics*, pages 15–21. AAU.

[Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics.

[Madsen et al., 2015] Madsen, O., Bøgh, S., Schou, C., Andersen, R. S., Damgaard, J. S., Pedersen, M. R., and Krüger, V. (2015). Integration of mobile manipulators in an industrial production. *Industrial Robot: An International Journal*, 42(1):11–18.

[TAPAS, 2014] TAPAS (2011-2014). Robotics-enabled logistics and assistive services for the transformable factory of the future. EU project funded under the European Community's Seventh Framework Programme grant no 260026. Web page: http://www.tapas-project.eu/.

Paper F

Hand-Eye Calibration of Depth Cameras based on Planar Surfaces

Rasmus S. Andersen, Ole Madsen, and Thomas B. Moeslund

This extended abstract has been presented excluding the *Method* section at the peer-reviewed workshop
1'st International Workshop on Intelligent Robot Assistants,
as part of the
13'th International Conference on Intelligent Autonomous Systems, 2014.

Hand-Eye Calibration of Depth Cameras based on Planar Surfaces

Rasmus S. Andersen[1], Ole Madsen[1] and Thomas B. Moeslund[2]
[1]*Department of Mechanical and Manufacturing Systems, Aalborg University, Denmark*
[2]*Department of Architecture, Design and Media Technology, Aalborg University, Denmark*
{rsa,om}@m-tech.aau.dk, tbm@create.aau.dk

Keywords: Hand-eye calibration, robot vision, depth cameras, depth calibration.

1 INTRODUCTION

For robots to be able to perform advanced tasks, it is a necessary to use various sensors. This is the case both for industrial tasks (pick-and-place operations, bin-picking), for home service robots (identification of humans, navigation), and for military robots (local/global navigation, obstacle identification). There are three typical ways to mount sensors relative to the robot:

1. Mounted on the robot in a fixed or movable position (pan/tilt).
2. Mounted in the environment in a fixed or movable position.
3. Mounted on the end-effector of the robot arm, that is supposed to interact with the environment.

The calibration between the sensor and the robot is essential for all of these mountings. In this work we focus on calibrating a depth camera to the end-effector; also known as *hand-eye* calibration. Hand-eye calibration is necessary for all sensors mounted on an end-effector. The most popular sensor type to mount on end-effectors is visible light cameras, and calibration of these have therefore been investigated thoroughly. Depth cameras is another popular choice, which have also been used on robots for several decades. Especially since the launch of Microsoft's Kinect in 2010, their popularity have increaed (El-laithy et al., 2012). The depth sensor in the Kinect works by projecting infrared structured light onto the scene. The depth is measured by capturing the known projected pattern, and based on this compute the depth. Other technologies for capturing depth images include Time-of-Flight (ToF) (Fuchs, 2012) and stereo vision.

1.1 Existing Methods

The problem is illustrated in Figure 1. The unknown transformation is the one between tool and camera, while the transformation between the base and tool is assumed to be known.

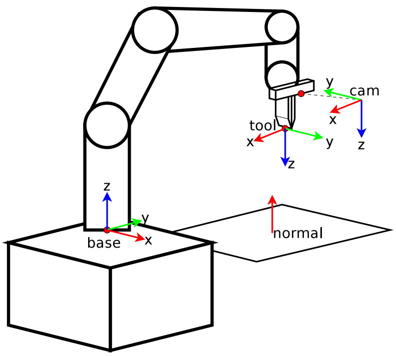


Figure 1: Coordinate systems involved in the hand-eye calibration.

Several approaches to depth camera hand-eye calibration exist. One popular approach is the TurtleBot calibration algorithm, which is available through ROS. This works specifically for Kinect-like cameras by first pose estimating a calibration board using the RGB/D sensors, and afterwards localizing it in the base frame by moving the robot tool to the four corners on th board. The problems with this approach include that is relies both on the imperfect internal RGB-D calibration of the Kinect and of the model of the tool.

Use of the tool can in some cases be avoided when using a calibration board (Tsai and Lenz, 1989; Hvilshøj et al., 2010). For the Kinect, a transformation between the depth camera and a visible light camera is known beforehand. Thus, the visible light camera can be calibrated first and used to indirectly calibrate the depth camera. This approach ofcourse relies on a transformation, which is not perfectly known. Another approach that can be used for a Kinect is to directly calibrate the internal infrared camera using an infrared light source and a calibration board. A problem with this is that the internal depth computations of the Kinect are circumvented. Also, it only works for depth cameras that is based on an infrared camera.

A few methods focus specifically on depth cameras. In (Pomerleau et al., 2011) the motion of the depth camera is continuously logged based on ICP. The hand-eye transformation can then be computed by comparing to the movement of the end-effector. A problem with this approach is the matches do eventually drift, causing the calibration to be less precise. In (Kahn et al., 2014) it is instead suggested to design a 3D shape to be optimal for 3D pose estimation from a point cloud, and use this to find the camera pose. The pose of the object relative to some world frame (such as the robot's base frame) must however be known beforehand. This is a severe limitation for general purpose hand-eye calibration.

1.2 Suggested Approach

The approach that we suggest here is to do calibration of depth cameras using only the point clouds as in (Kahn et al., 2014), but to estimate equations for a simple planar surface instead of carefully designed 3D shapes. The advantage is that planes can be found in point cloud very fast and reliably using standard techniques such as RANSAC. Also, a sufficiently planar surface is nearby in most locations. A plane is estimated for the same surface from multiple positions, and using these, the tool-camera transformation can be found by minimizing an overdetermined system of non-linear equations. A total of 10 parameters are estimated; 6 for the tool-camera transformation and 4 for the plane. The idea is illustrated in Figure 1.

2 METHOD

The system of equations is based on both the plane normal and the distance between the estimated plane and the base origin (refer to Figure 1). These are derived in the following subsections.

2.1 Plane Normal

The transformation between the estimates plane normal in each depth image and the plane normal in the base coordinate system is given by:

$$
\begin{aligned}
{}^{\text{cam}}\overrightarrow{n}_{\text{plane}} &= {}^{\text{cam}}_{\text{tool}}T \cdot {}^{\text{tool}}_{\text{base}}T \cdot {}^{\text{base}}\overrightarrow{n}_{\text{plane}} \Leftrightarrow \\
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} &= {}^{\text{cam}}_{\text{tool}}T \cdot {}^{\text{tool}}_{\text{base}}T \cdot {}^{\text{base}}\overrightarrow{n}_{\text{plane}} - {}^{\text{cam}}\overrightarrow{n}_{\text{plane}}
\end{aligned} \tag{1}
$$

The unknowns are the transformation ${}^{\text{cam}}_{\text{tool}}T$ and the plane normal ${}^{\text{base}}\overrightarrow{n}_{\text{plane}}$, which in total have 15 unknowns. This is not the case for the problem though; this has only 6 degrees of freedom for the transformation and 3 degrees of freedom for the plane normal. Here it is chosen to use ZYX Euler angles for the rotational degrees of freedom, and thus the unknown parameter vector to be optimized is $\overrightarrow{\theta} = [A,B,C,x,y,x,a,b,c]^{\mathrm{T}}$, where $[a,b,c]^{\mathrm{T}}$ is the normal vector of the unknown plane. When minimizing Equation (1), each value in ${}^{\text{cam}}_{\text{tool}}T$ is therefore replaced by the corresponding equation based on $\overrightarrow{\theta}$.

2.2 Plane Distance

The general equation for point-plane distances is given by:

$$D = \frac{ax_0 + by_0 + cz_0 + d}{\sqrt{a^2 + b^2 + c^2}}$$

where the plane equation is $ax + by + cz + d = 0$, $[x_0, y_0, z_0]^{\mathrm{T}}$ is an arbitrary point, and D it the shortest distance between the point and the plane. Inserting the plane parameters and the camera position in the base coordinate system (${}^{\text{base}}P_{\text{cam}}$) gives:

$$0 = \frac{{}^{\text{base}}\overrightarrow{n}_{\text{plane}} \cdot {}^{\text{base}}P_{\text{cam}} + d_{\text{plane}}}{|{}^{\text{base}}\overrightarrow{n}_{\text{plane}}|} - D \tag{2}$$

where d_{plane} is the actual distance between the plane and the base (and thus constant for all camera positions), and D is the measured distance. The norm of the normal vector below the fraction line is 1 for the normalized case, and can thus be removed.

2.3 Cost Function and Optimization

Combining Equation (1) and (2) we get the complete system of equations:

$$G(\overrightarrow{\theta}) = \begin{bmatrix} {}^{\text{cam}}_{\text{tool}}T \cdot {}^{\text{tool}}_{\text{base}}T \cdot {}^{\text{base}}\overrightarrow{n}_{\text{plane}} - {}^{\text{cam}}\overrightarrow{n}_{\text{plane}} \\ {}^{\text{base}}\overrightarrow{n}_{\text{plane}} \cdot {}^{\text{base}}P_{\text{cam}} + d_{\text{plane}} - D \end{bmatrix} \tag{3}$$

where the first line holds three independent equations and the second line one. Since the problem to be minimized has of 12 parameters, at least three measurement points is required, giving three sets of four equations.

The cost function is constructed by squaring and dividing by the number of equations:

$$F(\vec{\theta}) = \frac{1}{2n} \cdot G^{T}(\vec{\theta})G(\vec{\theta}) \tag{4}$$

where n is the number of measurement points. The optimization can then be done using gradient descent:

$$\vec{\theta}^{i+1} = \vec{\theta}^{i} - \alpha \vec{\nabla} F(\vec{\theta}^{i}) \tag{5}$$

To speed up the descend, a momentum approach is applied, where the learning factor α is gradually increased as long as the cost function $F(\vec{\theta})$ is decreasing.

3 PRELIMINARY RESULTS AND CONCLUSIONS

An initial implementation has been tested using 12 measurement points. In this it is assumed, that the plane normal is known to be vertical, and thus only 9 degrees of freedom is estimated. For each point, RANSAC has initially been used to estimate the plane, followed by least squares regression on inliers to increase precision. The learning process was stopped when $\frac{\Delta\theta}{\theta}$ got below a predefined threshold.

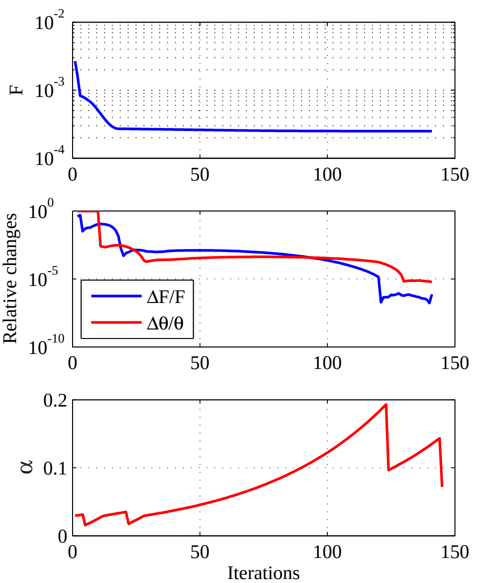


Figure 2: Coordinate systems involved in the hand-eye calibration.

In this extended abstract, a hand-eye calibration for depth cameras has been presented that is based on estimating a plane for the same surface from multiple positions. This gives an overdetermined non-linear system of equations with 12 parameters, that is minimized using standard least squares gradient descent. A preliminary implementation has proven the solution is feasible with a realistic guess for the parameters as a starting point for optimization. Our next step is to develop a full implementation and to evaluate its performance both with regards to precision and speed. We intend to publish the implementation as a publicly available ROS package.

ACKNOWLEDGEMENTS

This research was partially funded by the European Union under the Seventh Framework Programme project 260026 TAPAS - Robotics-enabled logistics and assistive services for the transformable factory of the future.

REFERENCES

El-laithy, R., Huang, J., and Yeh, M. (2012). Study on the use of microsoft kinect for robotics applications. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 1280–1288. IEEE.

Fuchs, S. (2012). *Calibration and Multipath Mitigation for Increased Accuracy of Time-of-Flight Camera Measurements in Robotic Applications*. PhD thesis, Universitätsbibliothek.

Hvilshøj, M., Bøgh, S., Madsen, O., and Kristiansen, M. (2010). Calibration techniques for industrial mobile manipulators: Theoretical configurations and best practices. In *Robotics (ISR), 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–7. VDE.

Kahn, S., Haumann, D., and Willert, V. (2014). Hand-eye calibration with a depth camera: 2d or 3d? In *International Conference on Computer Vision Theory and Applications (VISAPP)*. INSTICC.

Pomerleau, F., Magnenat, S., Colas, F., Liu, M., and Siegwart, R. (2011). Tracking a depth camera: Parameter exploration for fast icp. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3824–3829. IEEE Press.

Tsai, R. Y. and Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *Robotics and Automation, IEEE Transactions on*, 5(3):345–358.

Paper G

Projecting Robot Intentions into Human Environments

Rasmus S. Andersen, Ole Madsen, and Thomas B. Moeslund, and Heni Ben Amor

The paper has been submitted to the
25'th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2016)

In peer review

Excluded in the public version of the thesis due to copyright issues.

Paper H

Task Space HRI for Cooperative Mobile Robots in Fit-Out Operations Inside Ship Superstructures

Rasmus S. Andersen, Simon Bøgh, Thomas B. Moeslund, and Ole Madsen

The paper has been submitted to the
IEEE International Conference on Robotics and Automation (ICRA 2016)

In peer review

Excluded in the public version of the thesis due to copyright issues.

Paper I

Teaching Robotic Skills by Projecting into Task Space

Rasmus S. Andersen

This is a technical report which is intended for later publication.

Teaching Robotic Skills by Projecting into Task Space

I.1 Introduction

Skill based robot programming has the potential to allow users which are not experts in robotics to program a robot to solve new simple tasks fast and efficiently. This has been shown in several user studies, both for generic tasks [Schou et al., 2012, Schou et al., 2013, Pedersen et al., 2015] and (as part of this PhD) for the more specialized task stud welding [Andersen et al., 2015, Andersen et al., 2016]. For instance Schou et. al. presents a skill based programming system for the Little Helper 3 AIMM robot in [Schou et al., 2013], where a graphical user interface is combined with kinesthetic teaching. The system is tested in a user study where nine test persons with varying experience with robotics program the robot. All test persons managed to program the robot to perform two different pick-and-place tasks after having received only a short introduction and with minimal help. The programming time is approximately double for non-experts compared to experts familiar with the system. It is concluded that such a system is feasible; however it is noted that "The tests revealed that the interface still requires better instructions to support the operator during the teaching phase".

Manual kinesthetic teaching of skills makes it possible to program a task by *performing* the task. The operator directly interacts with the robot instead of programming an external interface on a stationary monitor or teach pendant. However, it is still necessary to pay attention to an external monitor to know the state of the robot and how to teach each step. The current report investigates if teaching can be improved by completely removing any external monitor and instead project the required information directly into task space. The idea is that projection mapped information allows the operator to focus all attention on the task space where the task is carried out instead of dividing it to a stationary monitor or teach pendant.

To evaluate the usability of the approach, a user study is carried out where users teach a relatively advanced series of skills involving object recognition, pose estimation, pick, and place. The applied pose estimation and pick is combined into a single vision-pick-skill which has previously been presented in [Andersen et al., 2014]. The study compares the usability of the approach directly to a slightly improved version of the graphical teaching interface presented by Schou et. al. in [Schou et al., 2013]. The main contribution of this report is a *projection based interface for manual kinesthetic teaching of skills and a comparative evaluation of its usability.* Secondary contributions are the presentation of an object recognition skill as well as a general usability evaluation of both the vision pick skill and the object recognition skill.

This report is structured as follows: The proposed projection based interaction system is first presented in Section I.2. In Section I.4, results from the comparative user study is presented. Finally, conclusions are drawn in Section I.5.

I.2 Projection Mapping Interface and Methods

Figure I.1 shows the prototype of the projection based teaching system. A projector can project guidance and graphics onto the surfaces below. The environment is modeled, and this makes it possible for the graphics to be prewarped to look correct on any surface. In the current setup, the projector is fixed. This limits the field which are covered by the projector. However, the field could easily be extended if needed by either mounting more projectors or by mounting the projector on a calibrated pan-tilt unit. The various elements in the system are described in the following.

I.2.1 Skill Teaching Instructions

The skill based programming method makes it possible to program a robot in the following steps:

1. Selection of a sequence of skills
2. Offline parameterization; for instance velocities and object types to handle
3. Online manual kinesthetic teaching

The first two steps have to be performed offline and a projection system is not relevant for those. The projection based teaching system instead replaces a monitor based system specifically in the online step. The online step contains, for instance for a pick-with-vision skill, the following instructed steps:

1. Press Y -> Move tool to camera position
   A position where the 3D camera in use can see the object and surroundings

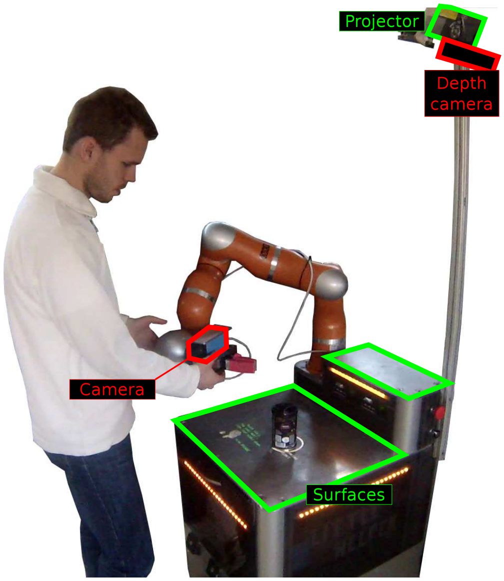
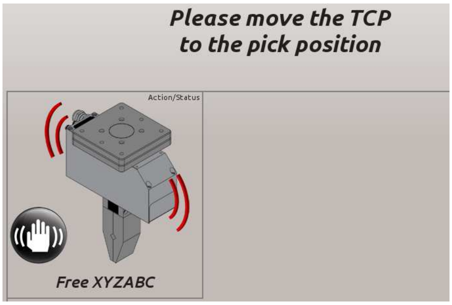


Fig. I.1: Setup of the projection based teaching system. A pole-mounted projector can project instructions and assisting graphics onto the surfaces below. The cameras are used for recognizing and pose estimating objects.

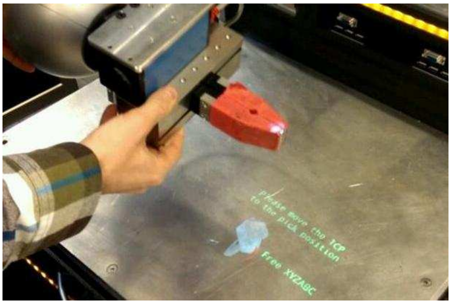
2. Press Y -> Move tool to via position, with the flat side down
   A position where the robot is "ready to grasp"
3. Press Y -> Move tool to grasp location
   A position where the object is in the middle of the open tool
4. Press X/Y/Z -> Select approach point
   Force input selects the approach direction and the desired distance is then moved manually

The action instructions ("Press Y") and the following contextual instructions are provided to the operator during teaching (the italic text is not part of the instructions). An example is shown in Figure I.2 for both the monitor and projection based systems. The tool image and the nearby text in the image informs the operator that the robot can now be moved in all degrees of freedom while the above text instructs the operator in what to do. When the tool is held stationary for a short duration, the position is stored. The tool image and nearby text will then change to indicate that the operator has to apply a force in the tools' Y-direction ("Press Y") to continue.

In the projection based system, the graphics is shown on the surface perpendicular beneath the end-effector. When the end-effector is moved, that graphics is also moved in real-time. If the end-effector is moved close to or



(a) Monitor



(b) Projector

Fig. I.2: Teaching instructions are displayed directly below the robot's end-effector. If this area is occupied or outside the field of projection, the instructions are displayed as close as possible.

beyond the surface edges, the graphics is shown on the surface as close to the end-effector as possible. Whenever the robot is aware of objects occupying parts of the surface, the graphics is moved away from these. Similarly, when the end-effector is very close to the surface, it itself is considered an obstacle, and the graphics is shown in front of the end-effector instead.

I.2.2 Projection-Supported Skills

The projection based system is fully integrated into the skill based system used for previous experiments at Aalborg University. It supports therefore a diverse range of skills. This includes the 13 skills which are presented and used for logistic tasks, machine tending, and assembly in [Madsen et al., 2015] and [Bogh et al., 2014]. The only requirement is that surfaces in the environment suitable for projection are modeled beforehand.

The projection system has been deeper integrated particularly in the skills used in the user study included in this report. The skills are:

Pick-with-vision: The skill uses a depth camera to detect objects on surfaces, fit a cylindric model around them, and pick them up. The teaching sequence was described in Section I.2.1. The skill has previously been presented in [Andersen et al., 2014], and a full description is outside the scope of this report.

Place-onto: The skill uses force sensing to place a previously grasped object onto a surface on a taught location. The skill has previously been presented in [Schou et al., 2013], and a full description is outside the scope of this report.

Recognize: The skill distinguishes between object classes using a discriminative model which is learned during teaching of the skill. The classification method is a bag-of-visual-words approach based of SIFT features [Lowe, 1999]. During teaching, a number of monochrome images are captured of an object from each class, and local features are extracted from the images. The features are extracted as SIFT features, but detected as STAR (or CenSurE) features [Agrawal et al., 2008]. This combination is chosen because it produces reliable classification results in practice. A vocabulary is constructed from the features extracted from all images by applying K-means clustering with 20 bins (or visual words). The classifier is then trained using a support vector machine (SVM).

During execution, an object is classified by first capturing three images and extract features in each, similar to during teaching. Each image is then classified based on the trained model. If the images are classified in the same class, this is chosen as the outcome. In case of disagreements, more images are captured until 80% predict the same class. If this cannot be achieved, the skill fails.

The skill can be set up to only accept a single class. Thus objects of a particular class can be picked, for instance, while other objects are ignored.

For these three skills, additional information is projected at specific times during both teaching and later during execution. The specific information is shown in Figure I.3. For I.3(a) and I.3(c), the projected information replaces similar information shown by the robot and on the monitor. For I.3(b) and I.3(d), the projected information cannot be indicated with the monitor based interface.

I.3 Evaluation Methods and Metrics

The projection and monitor based interaction systems are evaluated in a user study. The purpose of the study is to evaluate and compare the usability of each method.

I.3.1 Task

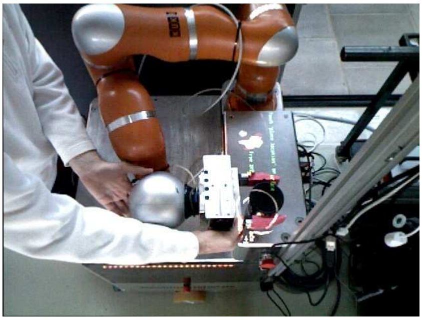
The test participants are asked to teach a sequence of three skills online with each system:

1. **Recognize** an object in a predefined area
2. **Pick-with-vision**: Detect the precise location of the object and pick it up

(a) Recognition: “Place ’Nescafé at the position” for capturing images. With the monitor interface, the position is indicated by pointing with the end-effector.

(b) Recognition: “Stay out of area” while the robot captures images. This is not available with the monitor interface.

(c) Pick-with-vision: “Detected object” before teaching how to grasp. With the monitor interface, this is shown in a pop-up window.



(d) Place-onto: “Place position” when object is placed. This is not available with the monitor interface.

Fig. I.3: Projected information during teaching of skills. The information in (c) and (d) are also shown during execution.

3. **Place-onto**: Place the object onto a surface at a new location (of the test participant’s choice)

The order in which the test participants use the two systems is changed for each participant to eliminate bias. Before the test persons are asked to teach the tasks themselves, they are introduced to the first system, they have to use. All of the required skills are demonstrated. After a test participant has tested the first system, the differences to the second system is verbally highlighted. They are not given a full introduction to the second system to reduce the time requirement. Execution of the taught tasks is demonstrated to the test participants immediately after teaching each system. If the task that the test participant has taught cannot be executed due to errors in the teaching, a pre-taught sequence is executed instead.

I.3.2 Usability Measures

The usability evaluation is based on ISO 9241-11 (1998) which defines usability as a combination of effectiveness, efficiency, and satisfaction [ISO, 1998]. The usability of each system is evaluated individually as:

Effectiveness "*The accuracy and completeness with which users achieve specified goals*"
Measured as the ability of the test participant to complete the teaching without assistance. Assistance requirements are divided into groups of increasing seriousness:

1. The test participant requests confirmation of his/her intended action. If a more thorough explanation than yes/no is required, the situation is counted as 2).
2. The test participant is in doubt on how to move forward and actively requests assistance.
3. The test participant makes an error during teaching which will make execution impossible or unreliable. If the test leader actively has to intervene to avoid such error, it is also counted.

When summing up the assistance requirements, they are weighted as $1/2$, 1, and $1 1/2$.

Efficiency "*The spent resources in relation to the accuracy and completeness with which users achieve specified goals*"
Measured objectively as the time spent to complete the task.

Satisfaction "*The freedom from discomfort and positive attitudes towards the use of the product*"
Measured subjectively through the Lewis' ASQ questionnaire [Lewis, 1991] and comparative questions on specific parts of the teaching. The Lewis' ASQ evaluates satisfaction with three Likert-scale questions (evaluated from 1-7):

1. Overall, I am satisfied with the ease of completing the tasks in this scenario
2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario
3. Overall, I am satisfied with the support information (projections, display on monitor) when completing the tasks

The systems are compared directly through the following specific questions:

1. The robot indicated the teaching steps by showing an image (of the tool) as well as text instructions. Where did you prefer to get this information?
   Refer to Figure I.2.
2. During teaching of OBJECT RECOGNITION, the robot indicated which object that should be placed at which location. With projector, projected circles were used. With monitor, the robot moved to the location. Which did you prefer?
   Refer to Figure I.3(a).
3. During teaching of OBJECT PICKING, the robot indicated which objects it had detected. With projection, using circles around the objects. With monitor, using a pop-up window. Which did you prefer?
   Refer to Figure I.3(c).
4. During teaching of OBJECT RECOGNITION, the robot took a number of pictures of each object. The area around the object was clearly marked with projection to help users staying clear of the robot and the images. Did you find this useful? (Where 1 is "Very useful" and 7 is "Not useful at all".)
   Refer to Figure I.3(b).

Question 1-3 is evaluated on a Likert scale from 1: "Clearly prefer projected" to 7: "Clearly preferred monitor".

Finally, the test participants are asked to also evaluate the interfaces during execution. There are, however, no interaction between to robot and the operator while execution is ongoing. Therefore, the test participants are instead asked to evaluate how safe they felt with each system and how well they understood what the robot was doing; also on a Likert scale from 1-7.

I.4 Results

The results for each of the systems in effectiveness, efficiency, and satisfaction respectively are presented in this section. The results are based on a total of $n = 20$ persons who participated in the experiment. The test participants had diverse backgrounds in robotics as well as in computers and IT in general. Their self-evaluated expertise in these fields are shown in Figure I.4.

I.4.1 Results on Effectiveness and Efficiency

The effectiveness results are shown in Figure I.5(a). The average number of errors was 1.65 for projection and 2.15 for monitor. When weighted for serious-

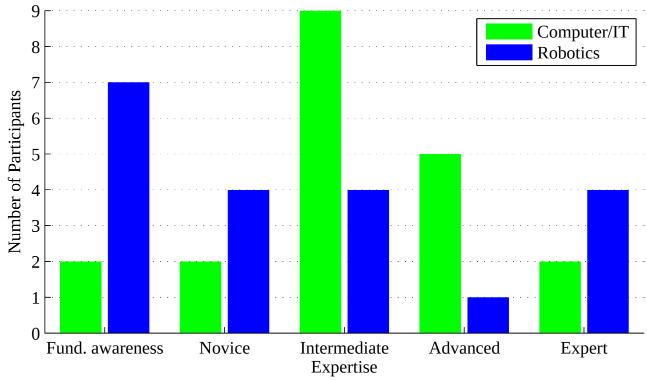
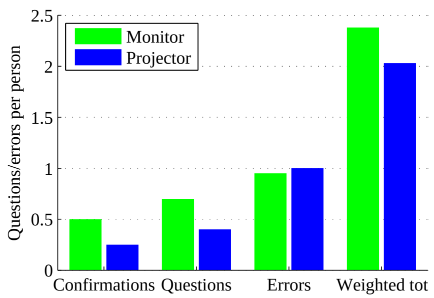
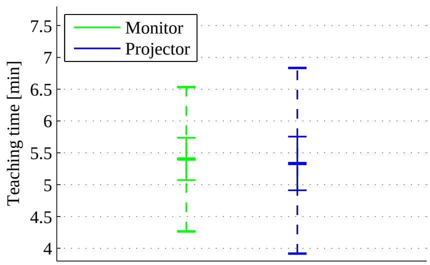


Fig. I.4: Test persons divided by self-estimated expertise in computers/IT and robotics.

ness, the numbers were 2.03 and 2.38, respectively. On average, the projection based interface thus caused the test participants to require less assistance.



(a) Effectiveness measured as the number of instances where assistance was required. The first three columns are instances of increasing seriousness and the last is a summation weighted after seriousness as defined in Section I.3.2.



(b) Efficiency measured as the time used for teaching. The middle horizontal lines are the means, the next lines are the 95% confidence intervals, and the top and bottom lines are the maximum and minimum values.

Fig. I.5: Effectiveness (required assistance) and efficiency (time consumption).

The results on effectiveness, the time required for teaching the task, are listed in Figure I.5(b). The average time was 5:20 minutes for projection and 5:24 for monitor. This covers over large differences in time consumption from person to person, as is evident from the Figure.

The results for both effectiveness and efficiency indicate that the projection based interface is better that the monitor interface. The differences are, however, too small to be statistically significant, and more research would be required to draw final conclusions.

I.4.2 Results on Satisfaction

The satisfaction of each interface is first evaluated though the three questions from Lewis' ASQ [Lewis, 1991] which are written in full in Section I.3.2. The test participants estimate their satisfaction with the ease of using the system, the time to complete the task, and the information offered by the system. The results are shown as T1-T3 in Figure I.6. There is a tendency that the projector interface is rated higher, except for the satisfaction with the provided information which is practically equal. None of the questions show statistically significant differences between the interfaces. It must be noted, however, that both systems receive very high satisfaction scores between 5.5 and 6.0 on the scale from 1-7.

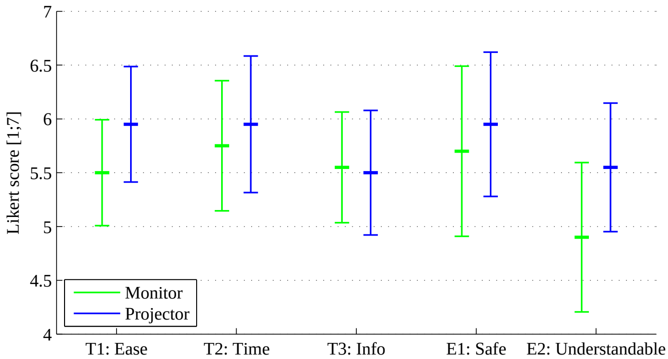


Fig. I.6: Satisfaction during teaching (T1-T3) and feeling of safety and understanding of the robot during execution (E1-E2). The teaching questions are from Lewis' ASQ and written in full in Section I.3.2. For all questions, the mean and the 95% confidence interval are shown. Also for all questions, higher scores indicate agreement and is considered better.

The evaluation of the test participants' feeling of safety and understanding of the robot's actions is included as question E1-E2 in Figure I.6. Similar to the teaching results, the projector interface is on average slightly better that the monitor interface. The differences are, however, not large enough to be statistically significant with a significance level of 95% in a 2-sided t-test.

The results of the direct comparison of different elements in the interfaces are shown in Figure I.7. The full questions are listed in Section I.3.2 as the second enumeration under *satisfaction*. Also, the "info position" refers to the placement of the graphics shown in Figure I.2 and the remaining questions to Figure I.3.

The test participants clearly preferred the projected interface for question 1-3 which compare the two interfaces directly. A score of 4 corresponds to the two interfaces being equally preferred. The null hypothesis that the actual means are 4 can be rejected in a 2-sided t-tests with a significance level of 95%. This is also the case for question 4 on the usefulness of the warning area. The

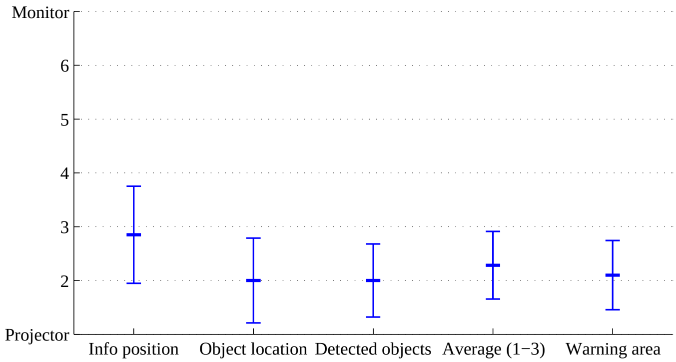


Fig. I.7: Teaching specifics.

p-values are below 0.02 for all questions. The projected interface can thus be concluded to be preferred in these specific areas to a statistically significant degree.

I.4.3 Discussions on the User Study

For both effectiveness and efficiency, the projection interface gave a slight improvement. The differences are, however, not large enough to be statistically significant. For satisfaction, projection also scored slightly better on all questions except for satisfaction with the information placement, where the monitor where marginally preferred. During teaching, the projection system on average increased the feeling of safety and the understanding of the robot's actions. None of these differences were, however, statistically significant. From the test participants' comments, it is clear that many felt it as a nuisance to continuously look away from the robot with the monitor based interface while they generally liked the projection based interface. A drawback with this was, however, that the robot arm occasionally blocked the projections, which could make it difficult or impossible to read the instructions. The projection system could therefore probably get a higher usability score by handling such situations. Or, as another test participant suggest, the two systems could be used simultaneously and benefit from each other's strengths.

The user test also compared different specific parts of the teaching systems, including the position of the information, the highlighted locations for instructed and detected objects, and a warning area requesting the operator to stand clear while the robot captured images. For all of these, projection was clearly preferred to a statistically significant degree with 95% significance levels.

It is particularly curious that the satisfaction with the provided support information ("T3: Info" in Figure I.6) is practically identical for the two systems, while the projected vs. monitor placement of the information ("Info position" in Figure I.7) shows a clear preference for the projector. One reason for this might be that many test persons are highly satisfied with both systems, and that no clear difference therefore emerges. When asked to compare the position directly, the projected position in task space is generally preferred.

I.5 Conclusion

This report proposes using projection mapping to improve human-robot interaction during programming of a task in a skill based system. Manual kinesthetic teaching enables an operator to program a robot to solve a task while actually solving the task. No complex programming system is required. The addition of projection mapping makes it possible for the operator to focus only on the area where the task is carried out and not on external interfaces.

The projection system projects information onto modeled surfaces directly below the end-effector of the robot or on the closest unoccupied area on the closest surface. The projection system has been integrated into an existing skill based programming system, enabling online projection based teaching of more than 13 skills. For three skills; recognize, pick-with-vision, and place-onto, the interface has been further improved by adding additional information which cannot be provided using a monitor.

The projection based interface has been compared to a monitor based interface in a user study with 20 participants. The study evaluates the usability as a combination of effectiveness, efficiency, and satisfaction, as it is defined in ISO 9241-11 (1998) [ISO, 1998]. It must be concluded that both approaches have a high usability after only a short introduction. Even tasks that require relatively complex computer vision such as recognizing and distinguishing between object classes and pose estimating for grasping could be taught by persons which were non-experts in robotics. The projection based interface performs slightly better than the monitor interface on all usability measures. The differences are too small to be statistically significant, though, and more research will be required to make a final conclusion. However, when the test persons compare the interfaces directly, the projection based interface is clearly preferred in every case to a statistically significant degree.

References

[Agrawal et al., 2008] Agrawal, M., Konolige, K., and Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer.

[Andersen et al., 2015] Andersen, R. S., Boegh, S., Moeslund, T. B., and Madsen, O. (2015). Intuitive task programming of stud welding robots for ship construction.

In *Industrial Technology (ICIT), 2015 IEEE International Conference on*, pages 3302–3307. IEEE.

[Andersen et al., 2016] Andersen, R. S., Boegh, S., Moeslund, T. B., and Madsen, O. (2016). Task space HRI for cooperative mobile robots in fit-out operations inside ship superstructures. Submitted for publication at RO-MAN 2016.

[Andersen et al., 2014] Andersen, R. S., Nalpantidis, L., Krüger, V., Madsen, O., and Moeslund, T. B. (2014). Using robot skills for flexible reprogramming of pick operations in industrial scenarios. In *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, volume 3, pages 678–685.

[Bogh et al., 2014] Bogh, S., Schou, C., Rühr, T., Kogan, Y., Dömel, A., Brucker, M., Eberst, C., Tornese, R., Sprunk, C., Tipaldi, G. D., et al. (2014). Integration and assessment of multiple mobile manipulators in a real-world industrial production facility. In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, pages 1–8. VDE.

[ISO, 1998] ISO, W. (1998). 9241-11. ergonomic requirements for office work with visual display terminals (vdts). *The international organization for standardization*.

[Lewis, 1991] Lewis, J. R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the asq. *ACM SIGCHI Bulletin*, 23(1):78–81.

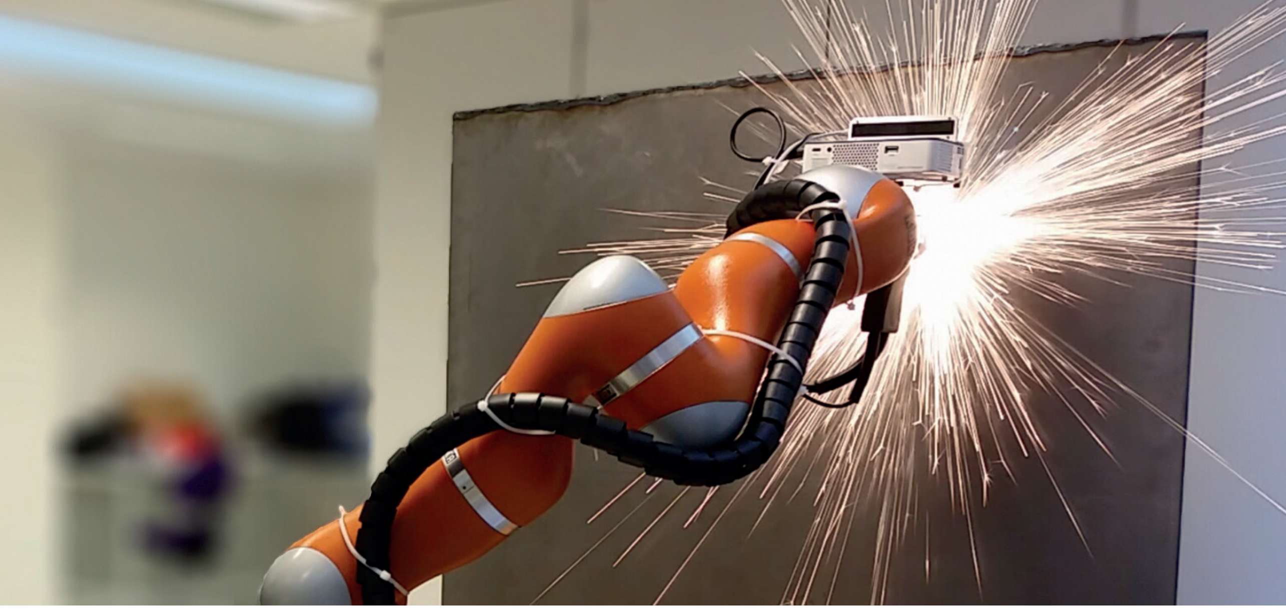
[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE.

[Madsen et al., 2015] Madsen, O., Bøgh, S., Schou, C., Andersen, R. S., Damgaard, J. S., Pedersen, M. R., and Krüger, V. (2015). Integration of mobile manipulators in an industrial production. *Industrial Robot: An International Journal*, 42(1):11–18.

[Pedersen et al., 2015] Pedersen, M. R., Nalpantidis, L., Andersen, R. S., Schou, C., Bøgh, S., Krüger, V., and Madsen, O. (2015). Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*.

[Schou et al., 2012] Schou, C., Carøe, C., Hvilshøj, M., Damgaard, J. S., Bøgh, S., and Madsen, O. (2012). Human assisted instruction of autonomous industrial mobile manipulator and its qualitative assessment. In *1st AAU Workshop on Human-Centered Robotics*, pages 22–28.

[Schou et al., 2013] Schou, C., Damgaard, J. S., Bogh, S., and Madsen, O. (2013). Human-robot interface for instructing industrial tasks using kinesthetic teaching. In *Robotics (ISR), 2013 44th International Symposium on*, pages 1–6. IEEE.



SUMMARY

The field of collaborative industrial robots is currently developing fast both in the industry and in the scientific community. Companies such as Rethink Robotics and Universal Robots are redefining the concept of an industrial robot and entire new markets and use cases are becoming relevant for robotic automation. Where industrial robots traditionally are placed behind security fences and programmed to perform simple, repetitive tasks, this next generation of robots will be able to work side-by-side with humans and collaborate on completing common tasks.

This thesis investigates methods for fast and intuitive programming and interaction with collaborative, industrial robots. The work is divided into two areas: Vision-enabled robotic skills and projection mapping interfaces. The purpose of robotic skills in general is to allow non-experts in robotics to program robots in an intuitive manner. It is investigated how a skill based architecture can incorporate advanced robot vision capabilities while keeping the robot programming fast and intuitive. Projection mapping, on the other hand, is the technique to project information onto the real world. It is investigated how projection mapping can be applied as part of human-robot interfaces to simplify and improve human-robot interaction in scenarios involving robot programming as well as human-robot cooperation.

ISSN (online): 2246-1248
ISBN (online): 978-87-7112-498-9

AALBORG **UNIVERSITY PRESS**