

# Spatial Programming for Industrial Robots Through Task Demonstration

Regular Paper

Jens Lambrecht<sup>1,\*</sup>, Martin Kleinsorge<sup>1</sup>, Martin Rosenstrauch<sup>1</sup> and Jörg Krüger<sup>1,2</sup>

<sup>1</sup> Department of Industrial Automation Technology, Institute for Machine Tools and Factory Management (IWF), Berlin Institute of Technology, Berlin, Germany

<sup>2</sup> Fraunhofer Institute for Production Systems and Design Technology (IPK), Berlin, Germany

\* Corresponding author E-mail: lambrecht@iwf.tu-berlin.de

Received 10 May 2012; Accepted 27 Dec 2012

DOI: 10.5772/55640

© 2013 Lambrecht et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract** We present an intuitive system for the programming of industrial robots using markerless gesture recognition and mobile augmented reality in terms of programming by demonstration. The approach covers gesture-based task definition and adaption by human demonstration, as well as task evaluation through augmented reality. A 3D motion tracking system and a handheld device establish the basis for the presented spatial programming system. In this publication, we present a prototype toward the programming of an assembly sequence consisting of several pick-and-place tasks. A scene reconstruction provides pose estimation of known objects with the help of the 2D camera of the handheld. Therefore, the programmer is able to define the program through natural bare-hand manipulation of these objects with the help of direct visual feedback in the augmented reality application. The program can be adapted by gestures and transmitted subsequently to an arbitrary industrial robot controller using a unified interface. Finally, we discuss an application of the presented spatial programming approach toward robot-based welding tasks.

**Keywords** Industrial Robot, Augmented Reality, Gestures, Programming by Demonstration, Object Recognition

## 1. Introduction

The demographic change and a general shortage of skilled workers are key challenges for manufacturing in western industrialized countries. In contrast to an ageing society is a new generation of "digital natives", young people who are familiar with using new technologies (e.g., computers and handhelds) intuitively and highly efficiently. One can meet the challenges of an ageing workforce both by taking preventive measures and through intelligent assistance systems, which support the worker in manual production processes toward ergonomics and efficiency. Hence, assistance systems could enable sustainable design of manual tasks. Regarding industrial robotics, manual online programming requires a high degree of expertise. Due to the time-consuming and complex programming process, small and medium-sized enterprises (SME) have reservations about investing in an industrial robot [1]. Accordingly, a recent report initiated by the European Union on robotic application [2] identifies human interaction as key technology for research with high likelihood of use in application. Furthermore, robot programming for everyone is presented as a main demand from SMEs for new robot advances. Thus, communication through human-like natural interfaces has increasingly

become the subject of scientific approaches for novel programming techniques in industrial robotics. Similar to human-to-human communication, most approaches use methods of communication, which are multimodal and more abstract than the robot language of classic programming approaches. High level programming aims at programming by teaching a task solution to the robot. The term programming by demonstration (PbD) refers to a task solution executed by a human operator, which is sensory observed by the robotic system. The demonstration is mapped into native robot commands and can be imitated by the robot [3]. Multimodal human robot interaction includes human-like communication, which mostly consists of speech (auditory) and gestures (visual). In terms of ergonomics, the main objectives of using natural communication channels are a shortening of both training times and operating cycles. Multimodality, i.e., a use of multiple communication channels simultaneously or successively, plays an important role in the design of intuitive control systems. Scientific publications in the field of industrial robotics, e.g., [4, 5], showed that multimodal control systems are more efficient and user friendly in comparison to conventional systems. However, an essential requirement is an adequate system design, which is related to the target application and the target user. Multimodal control systems typically use gestures, e.g., hand and finger gestures, haptics and speech for communication. Relevant publications in the field of gestural control of robots can be found in [6–9]. Another type of interaction between humans and machines is visualization of information through user interfaces. Unlike conventional forms of visualization, augmented reality (AR) enhances a camera image by adding spatially related information. In the field of industrial robotics, the user can be supported, e.g., by providing spatial information about robot programs. This means that poses, trajectories and information at the task level of the robot can be visualized in a real robot environment. Similar to simulation systems, a virtual robot can run the program with regard to accessibility and collision control. Thus, simulation-based features are carried out in a real environment [10]. Besides industrial robotics, some recent systems in research combine both modalities: AR and gestures. Huang and Alem created a wearable system for collaborative remote support in the mining industry [11]. Igarashi et al. [12] developed a control and programming system for mobile robots using touch gestures in an AR application. Regarding both approaches, empirical results toward usability appear to be promising. A more precise introduction of technically related systems and research projects using AR and gestures in the field of industrial robotic is given in section 2.

In this manuscript, we introduce a programming system for industrial robots using markerless gesture recognition and mobile AR in terms of programming by demonstration (PbD). The approach covers gesture-based

task definition and adaption by human demonstration. This is achieved through the manipulation of real or virtual objects. Additionally, the system covers program evaluation in a mobile AR applications on modern handheld devices. Because of 3D interaction through both gestures and AR, we call our approach 'spatial programming'. We developed an exemplary application of the spatial programming approach toward PbD of pick-and-place tasks. Furthermore, we discuss an application toward welding tasks which is part of ongoing work. A precise introduction of technically related systems and research projects using AR and gestures in the field of industrial robotic is given in section 2.

Beyond the state of the art, the paper is organized as follows: section 3 presents the objective of our spatial robot programming system. Section 4 introduces methods and experiments for an implementation towards a PbD of an assembly task. Finally, section 5 presents our conclusions and an outlook for spatial robot programming in additional fields of application.

## 2. Related Work

In this section, we introduce related programming systems and projects using gestures and AR in the field of industrial robotics. The main criteria for the delimitation of our approach are 3D bare-hand gestural interaction and mobile AR on modern handheld devices.

In 2003 the research project Morpha was completed [13]. The project was funded by the German Federal Ministry of Education and Research, and consisted of a broad consortium of robot manufacturers, research facilities and application companies. The main research focused on the development of interactive and innovative technologies for human-robot interaction through natural communication. The research results are still the basis for many research projects in the domain of multimodal control for service and industrial robots. The consortium introduced a rudimentary gesture-based motion control, task-oriented programming, as well as a static AR application for visualization of coordinate systems and axial moving directions. Within this project Dillmann et al. [8] evaluated different methods for interactive natural programming of robots and implemented a programming by demonstration system for dual-arm manipulation tasks using petri nets for task coordination [14]. The work on imitation learning of dual-arm manipulation continues in the Collaborative Research Center (SFB) 588 on humanoid robots [15].

The project SMErobot [1] within the 6th Framework Program of the EC aspired to develop new robot solutions tailored to the demands of SMEs in manufacturing. The main fields of research was automatic programming and understanding of human-like instructions, as well as safe

and productive human-robot cooperation. In addition to his previous and further work on robot programming [3, 6], within this project J. N. Pires of the University of Coimbra addressed natural high-level and multimodal programming by gestures and speech.

Meanwhile, many international research projects dealt with the application of AR to industrial environments. For a broad overview of industrial AR applications, technical requirements and implementations, refer to [16]. In the following we give an overview on recent scientific publications strongly related to our project.

W. Vogl [17] addressed the gesture-based definition of trajectories as well as AR-based evaluation of welding tasks. Moreover, a spatial interface was created to interact with virtual displays, which are projected onto material surfaces by means of a laser projector. This has been used to adjust parameters for welding tasks. From our point of view, the disadvantages of this system are: the definition of trajectories by an additional tool ("Magic Pen"); and the use of an expensive motion-tracking system tracking fiducial markers. Furthermore, spatial program adaption is limited to these 2D displays.

In 2011, Akan et al. [4] introduced an AR application with the objective of task-oriented programming of industrial robots. The camera is fixed in the workspace of the robot or mounted to the robot. Moving virtual objects in a graphical user interface enables the definition of assembly tasks. Further programming methods and gestures are not considered. The Augmented Reality and Assistive Technology Laboratory at the National University of Singapore studied different AR applications. They have dealt with AR-based programming of a virtual robot for arc welding by using a marker tool for spatial demonstration of trajectories [18]. In 2009, an approach of a marker cube-based teaching of trajectories and stereo vision-based methods for virtual object registration was used for programming virtual and real robots [19]. Recently, the research group developed a bare-hand gesture-based interaction with virtual 3D objects for AR applications [20]. The gesture recognition is based on 2D data from a single camera. A possible extension of 2D to 3D gesture interaction through a stereo camera system is mentioned, but not as yet implemented. Other publications of this research group consider a gesture-based definition of complex assembly tasks in an AR application [21, 22].

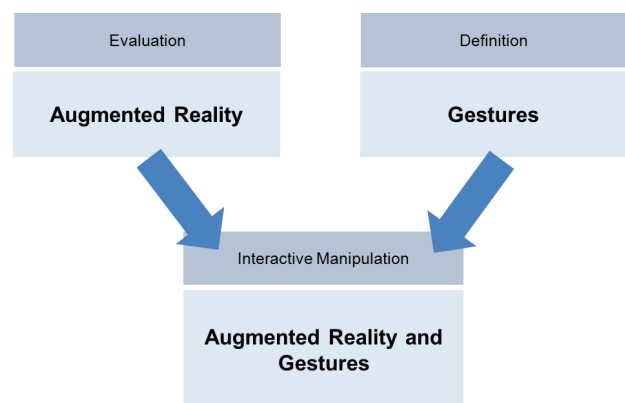
### 3. Interaction System

In this section we present the objective of our project, introducing the conditions and main challenges, as well as general solution statements. The first subsection is about the desired interaction system, while the last subsection describes the sensory and algorithmic

requirements toward perception. A specific implementation with a detailed description of the implemented methodologies can be found in the next section.

#### 3.1 Spatial Programming

The overall objective is to establish a modular system for intuitive industrial robot programming. The presented spatial programming system provides an efficient assistance system for online programming with the help of gestures and AR (see Fig. 1). In contrast to current state of the art technologies, our spatial programming system comprises the support for different phases of the programming process, as well as different levels of robot programming. Single programming modules are also applicable in combination with conventional online and offline programming methods. More distinguishing features are the use of 3D bare-hand gestural interaction and a mobile AR-based application on common handheld devices such as smartphones or tablet PCs.



**Figure 1.** Modalities for spatial online programming.

In this paper, we put an emphasis on gesture-based task definition and adaption of robot tasks. In order to achieve this goal, the user can interact naturally in 3D space with a real or virtual object through bare-hand gestures, i.e., the programmer can translate and rotate the virtual objects lying in front of him on a table in the robot workspace. Subsequently the robot program is adapted automatically according to the changes through spatial interaction. A further interaction on a lower programming level, e.g., gesture-based definition of poses and trajectories, is also part of the system, but will be considered in detail in another publication. To achieve flexibility in task definition, we aim for a multi-level programming approach to enable the skilled user to define new task representations by sequences of commands and parameters. The human demonstration of the task should be mapped into the high-level commands by a unified user interface. In terms of spatial evaluation of the robot program, we aim for a mobile AR solution on conventional handheld devices. Thus, the programmer is not confined to a fixed computer workstation but can

define poses, trajectories and tasks, on-the-move, as well as watching the spatial arrangement of the program components and manipulate single objects.

### 3.2 Perception

The perception of the system can be divided into two scopes: on the one hand, we have to recognize the procedure of task definition and adaption. Prerequisite for this work is a hand recognition and tracking, as well as a recognition of finger gestures. In addition to finger gestures, which can be recognized in 2D images, we intend to use 3D trajectory information of the hand to carry out precise spatial interactions.

On the other hand, we need a scene reconstruction to recognize objects for enabling the evaluation of the programming process in the AR application. For this reason, we aim to determine object class and pose. In terms of visual object recognition, the camera could be mounted to the end effector of the robot or could be positioned in the robot workspace. Another solution is to use the camera of the handheld device. Besides the advantage of no need for a additional camera, the use of the handheld camera has a high error susceptibility and requires more robustness of the image processing algorithms. Nevertheless, within our project we aim for a mobile solution for object recognition to be more flexible, regardless of location, robot position or additional sensors.

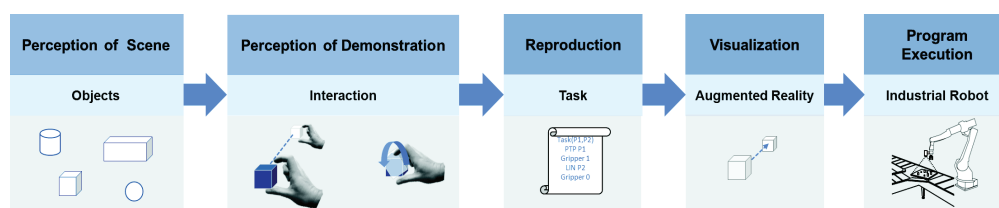
There is still no broad supply and demand for industrial machine vision on mobile devices. Additionally, currently in the field of robotics no software tools exist for high quality 3D scene reconstruction under free camera motion, which have an adequate usability for usage in

arbitrary scenarios. Common machine vision tools are optimized for specific environmental conditions and require high computing power to fit industrial demands.

In recent years, related work on the topic of 3D scene reconstruction has made great progress toward general applicability. The multi-view-stereo algorithm's demand for computation power is the reason for the relatively long history of publications until the first applications were capable of everyday use. The state of the art in structure from motion [24, 25] software is able to reconstruct the surfaces of many environments. The computation time of these algorithms is also high for current available consumer handheld devices. A possibility to overcome this limitation could be a cloud computing approach. Against this background, we decided to develop a simple 3D pose recognition algorithm for a limited set of easily distinguishable and predefined objects, which runs in real-time on a mobile device. This resulted in the object recognition algorithm presented in section 4.3.

## 4. Methodologies

In this section, we introduce methods to implement the spatial programming approach toward a robotic assembly task. Fig. 2 illustrates the basic steps of the programming process: object recognition, bare-hand interaction, i.e., translation and rotation, task mapping of interaction, visualization of the task in AR, and the execution of the task through the real robot. In the first subsections, we describe the basic hardware components, conceivable methods of spatial object manipulation, as well as the implementation of the gesture recognition and AR application. In the experiments section, we present results and a discussion of the specific assembly scenario.



**Figure 2.** Steps of spatial programming for an assembly task.

### 4.1 Setup

To implement the proposed spatial programming approach, our system consists of the following basic hardware components: industrial robot, handheld device and motion tracking sensor (Fig. 3). The application of a motion tracking system enables the tracking of human hands. We use TCP/IP sockets for communication between the components of the system. A unified control layer for vendor independent control and programming of industrial robots via arbitrary input device was

presented in [23]. This control layer is also used in this project for program representation, execution and adaption. Within our application, it is necessary to transform pose information to different coordinate systems, e.g., from motion tracking system to robot.

Fig. 4 illustrates the most important coordinate systems and some examples for related homogeneous transformations. A marker coordinate system is used to provide the pose information of the handheld device for the AR application and to determine object poses in the

robot coordinate system. In order to transform pose information between the different coordinate systems, one has to calibrate the marker, motion tracking and robot coordinate system.

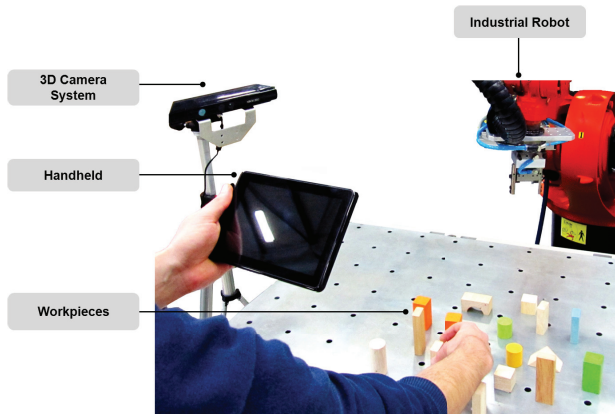


Figure 3. Basic components of the spatial interaction system.

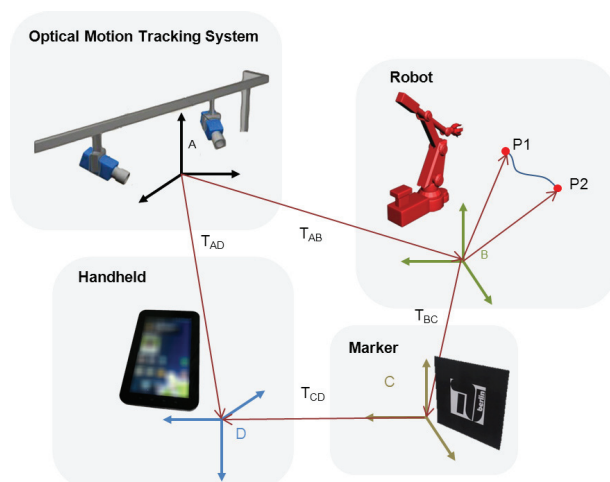


Figure 4. Relevant coordinate systems.

#### 4.2 Spatial Programming System

In terms of a holistic approach toward robot programming, we consider different programming levels for spatial interaction. Thus, the forms of interaction presented in this paper are applicable to different levels of programming: low-level and high-level programming. With regard to the task level, virtual objects cover task representations, e.g., workpieces for pick-and-place tasks. These virtual workpieces can be manipulated, e.g., by gestural snapping and moving in space. The robot program, which puts these gestural changes into practice, is created automatically. Besides pick-and-place tasks, an extension of graphical tasks' level representation and spatial interaction principles are possible due to a modular structure. The evaluation of the robot program is made possible through an AR application on the handheld device using a fiducial marker. Therefore, the programmer is capable of moving freely in the robot cell,

while the camera image is extended by spatial representations of the interpreted robot program. The spatial programming system can be divided into the AR application, gesture and hand trajectory recognition, a multi-level robot program representation and interfaces between these single modules. The module for object recognition is specific to task and application. In the following, it is considered separately.

##### 4.2.1 Mobile Augmented Reality

Prerequisite for a spatially adequate visualization of poses, coordinate systems, trajectories and task information in AR applications is the pose information of the camera. This means translational and rotational displacement between the camera coordinate system and a reference coordinate system must be known. There are various approaches using different sensor concepts for the determination of pose information for AR on handheld devices. Outdoor applications often use GPS in addition to internal rotation determination. Another approach uses fiducial markers with known dimensions. The markers are recognized by a pose estimation algorithm. Based on the marker position and orientation in the 2D camera image and its known dimensions, the algorithm estimates the pose information of the handheld in the marker coordinate system. Several frameworks and popular applications, e.g., ARToolKit, follow this approach. In this project we use a multi-marker approach using ARToolKit on an Android device as a general framework for marker tracking and visualization.

##### 4.2.2 Gesture-based Interaction

Gesture recognition for spatial interaction with virtual objects can be put into practice via 3D as well as 2D motion tracking. Enabling adequate 3D interaction based on 2D images from a single camera works only under fixed constraints. Otherwise, the algorithms are inaccurate because of the missing depth information. However, a rough determination of 3D movements for hands with known dimensions still is possible, e.g., see [21]. Due to the fact that finger gesture recognition based on 3D optical motion tracking data is complex (see application for MS Kinect [26]), we choose a novel approach to provide generic gestural interaction. For the manipulation of the virtual objects in AR, we combine 2D gestures recognized through the camera image of the handheld with 3D hand trajectories information, tracked by the external motion tracking system.

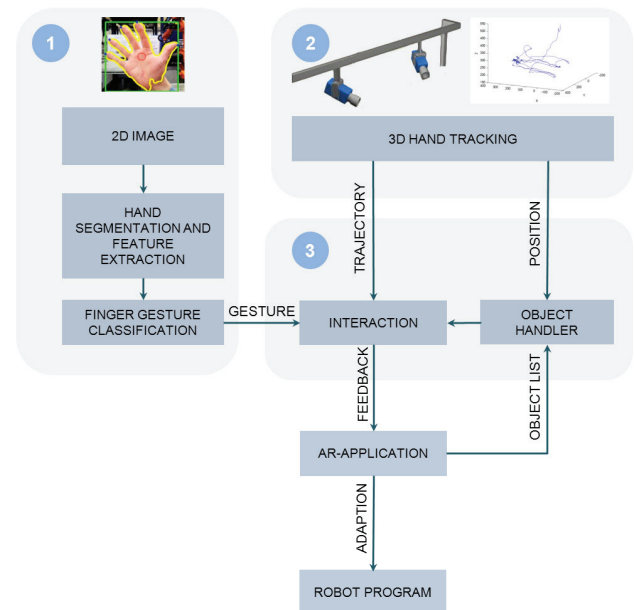
Fig. 5 illustrates the flow of information combining 2D command gesture recognition with 3D motion tracking. The reasoning and processing unit provides feedback about the gestural manipulation via AR and vibration on the handheld device. Finally, it adapts the robot task according to the gestural manipulation. In the following, we give a closer insight into finger gesture recognition.

The recognition of finger gestures contains the segmentation of skin colour region, extraction of fingertips as features and a shape-based pattern classification. Segmentation is done through skin colour tracking. The main challenge is to make the application robust to different skin colours and lighting variations. This is a difficult task due to limited computational effort and poor camera parameter handling on the handheld, e.g., it is not possible to completely turn off brightness and colour control. As a result, we follow a probability-based approach for robust and fast skin colour detection in [27]. First of all, we convert the image to HSV colour space ignoring the V channel. For a sample skin image, we determine a histogram, which is used as a calibration model for skin colour. For segmentation, we finally compute backprojection, i.e., the probability that a pixel has skin colour, threshold and smooth detected skin regions. For the purpose of features, we consider the fingertips, which are determined through convex hull according to the principle in [28]. For the classification of gestures, we implemented an algorithm for 2D shape analysis. For this purpose, we consider Procrustes analysis [29]. The algorithm compares a trajectory with a reference trajectory by translation, uniform scaling, rotation and finally shape comparison. Therefore, we are able to detect snap and release gestures based on the trajectories of the fingertips. In order to achieve a robust interaction for rotation and translation of virtual objects, we consider further constraints toward collision control of fingertips and object in the 2D camera image and spatial distance of hand and object.

#### 4.2.3 Task-oriented Programming

We implemented a basic robot programming system on a handheld device in [23]. This approach is enhanced by a task level. A task is represented by a subprogram containing a template for a motion structure of the robot, which can be parameterized by pose information. We enable the experienced user to define new motion sequence through drag and drop operations of conventional robot motion and action commands. Fig. 6 illustrates the structure of a simple pick-and-place task definition by pseudo-code including Point-to-Point (PTP) and linear (LIN) motion commands, as well as commands for the gripper control. The pose parameters can be set through a classic Teach-In approach, object definition or the spatial demonstration process. In order to map the demonstration of a pick-and-place task into this subprogram, we define the object pose as the start pose of the pick-and-place task. To determine the target pose, we incorporate pose displacements from 3D motion tracking. Depending on the specific robot task, the mapping of interaction poses has to be adapted. Demonstrating various tasks one after another creates a sequence of tasks, which is displayed on the handheld textually and spatially in AR. The sequence can also be adapted by

drag and drop operations with the help of touch gestures on the handheld device.



**Figure 5.** Interaction based on gestures and AR consisting of 1) 2D gesture recognition using the camera of the handheld, 2) tracking of 3D trajectories of the hand tracked by a motion tracking system, and 3) reasoning and processing unit.

#### 4.3 Object Recognition

The following subsection explains the particular steps of the successive image processing in more detail (see Fig. 7). The aim is to reconstruct the position and orientation of known geometrical objects out of 2D images on the handheld device. We assume the handheld pose to be known due to marker tracking of the AR application. Hence, we could determine object poses in the robot coordinate system. For the implemented algorithm, the following assumptions are considered: the objects and their geometrical dimensions have to be known. This means that only prespecified objects can be recognized and thus be located. In addition, the floor plate should be unicoloured, optimally matt white, and the ambient light conditions should be diffuse. The position and the orientation of the handheld integrated camera are given by marker tracking and the algorithm only handles acquisitions without partly hidden objects.

##### 4.3.1 Feature Detection

At the beginning of the algorithm, the acquired image (Fig. 8a) needs to be undistorted. Subsequently, an edge detection (Fig. 8b) is performed, followed by a closing to eliminate gaps (Fig. 8c). This enables the segmentation of all surfaces (Fig. 8d) that can be seen by the camera. Now it is possible to identify the single corners of each segment, here done by the calculation of the intersection points after detecting the segment borders with the Hough transform for line detection (Fig. 8e).

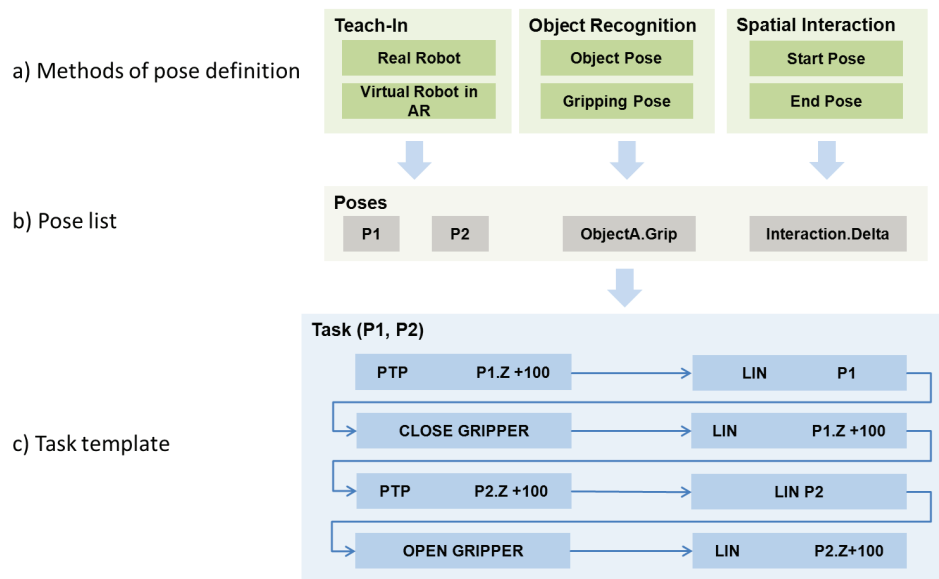
#### 4.3.2 3D Reconstruction

With the help of the resulting separate point list for each object, it is possible to solve the 3D position and orientation out of 2D data by using a perspective transformation. All in all, the computation is based on the following optical connection:

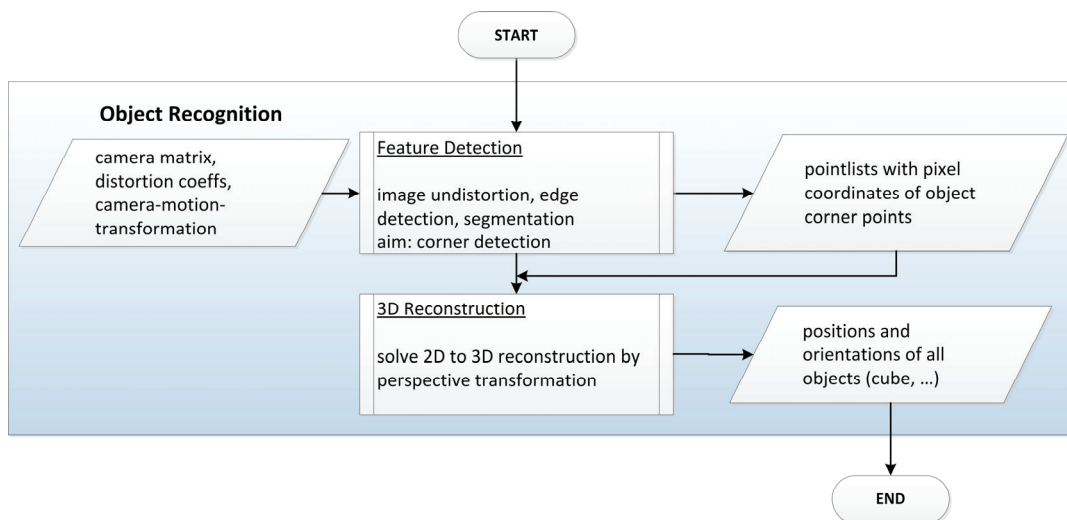
$$s \cdot m^T = A \cdot [R|t] \cdot M^T$$

$s$  is the scale factor,  $m$  a vector with the coordinates of the projection point in pixels,  $A$  the camera matrix,  $[R|t]$  the camera motion and  $M$  the coordinates of the 3D point in the world coordinate space. The colouring of the object surfaces ensures a correct association between the 2D and their corresponding 3D points in the algorithm.

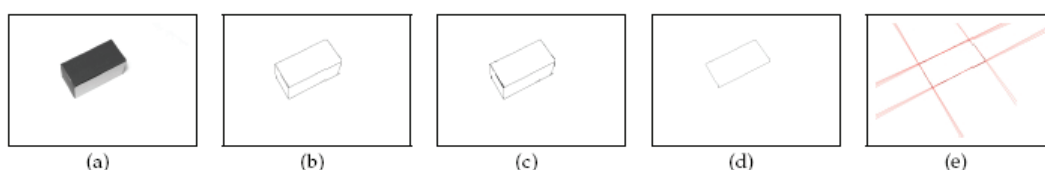
The additionally required information (focal lengths, principal point) are given with the camera matrix  $A$  and the used method is based on a paper by Gao et al. [30].



**Figure 6.** Structure of flexible task definition with the help of a robot program template and the mapping of specific poses from Teach-In, object recognition or spatial interaction. The sample template represents a simple pick-and-place task covering point-to-point (PTP) and linear (LIN) motion commands, as well as gripper commands.



**Figure 7.** Scheme of object recognition.

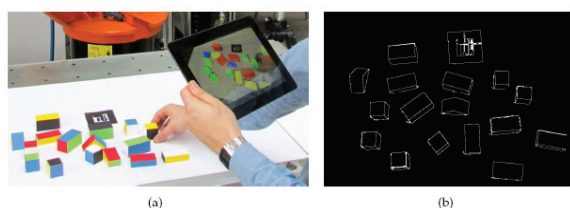


**Figure 8.** Workflow of image processing: a) acquired image, b) edge detection, c) closing d) segmentation of surfaces and e) line detection.

In order to maximize the precision, the distance between object and handheld should be of low values. Sporadic association errors in the 2D to 3D transform are compensated with over-determined point lists and outlier detection methods such as RANSAC.

#### 4.4 Experiments

In this subsection, we present an implementation of spatial programming toward a specific assembly task, show results and discuss further work. We developed an Android App for the ASUS Eee Pad Transformer Prime TF201, enabling an AR application based on the framework ARToolKit for the fiducial marker tracking and OpenGL ES for further visualization. For 2D gesture and object recognition, we use OpenCV 2.4.0 Java bindings for the Android 4.0.3 platform. 3D hand tracking is carried out by Kinect sensor and OpenNi Framework. To illustrate the intended use of our system, we consider a simple assembly task with initial situation (see Fig. 9a) and target situation (virtual objects in Fig. 11). All objects and dimensions are known and their CAD models are implemented in the AR application on the handheld. In order to support the robustness of the object recognition algorithm by additional colour segmentation, we assume the surface of a single object to be of a different colour. Fig. 9b illustrates the acquired scene image after edge detecting and closing algorithm. Since not all object edges are detected in a single image, one has to record several images for recognition of all objects. Based on the object pose estimation, virtual objects, which represent the real objects for interaction, are implemented into the AR application. Fig. 10a shows the skin colour segmentation and extracting of the fingertips for bare-hand interaction and Fig. 10b shows the gestural displacement of a virtual object for the demonstration of an assembly step. Fig. 11 shows the processing of the obtained robot program.

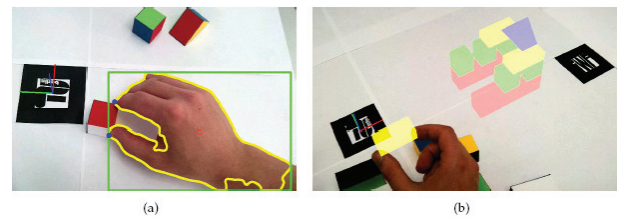


**Figure 9.** Initial situation (a) and image processing for object recognition (b).

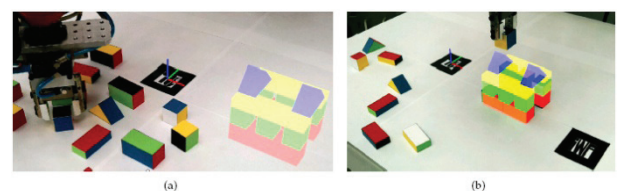
##### 4.4.1 Results and Discussion

Currently, we achieve low frame rates in the finger gesture recognition process for high resolution images. This results in a decrease of the intuitiveness within the task demonstrating and simultaneous evaluation in AR. Object recognition is unaffected by this effect, because it is carried out before interaction on single images. Nevertheless, it is possible to increase the frame rate

through code optimization and the use of C++ OpenCV. Initial project results show that the accuracies of object recognition are strongly dependent on the distances between the handheld device, marker and object. In most instances, this is still appropriate for the realization of robotic pick-and-place tasks, but there is potential for optimization regarding the sensor concept and object recognition algorithm. Insufficient accuracies may be compensated with the help of additional sensors, e.g., visual servoing using the eye-in-hand principle.



**Figure 10.** Skin colour segmentation and extraction of fingertips (a) for bare-hand interaction with virtual objects. Demonstration of assembly task (b).

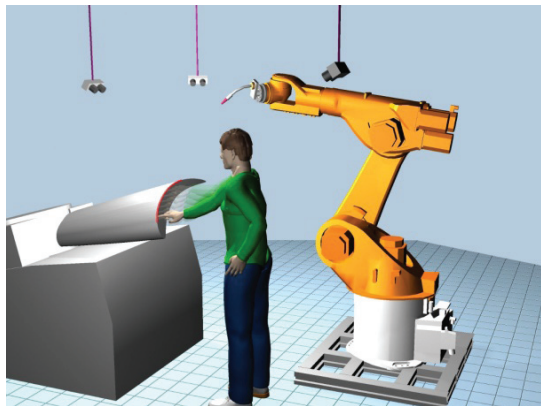


**Figure 11.** Assembly task representation through virtual objects (a) and semi-finished program execution (b).

Despite low absolute accuracies of the Kinect sensor, we consider the relative 3D translation to be adequate for robust and intuitive 3D interaction regarding pick-and-place tasks. Furthermore, vibration feedback about the recognition of command gestures and a continuous visual feedback about the object displacement during the demonstration process makes it possible to detect and avoid errors. Nevertheless, we aspire to replace the Kinect sensor with a more precise multi-camera-system. Hence, one can replace fiducial marker tracking for AR by a dynamic object tracking of the handheld device. Further work will also compare different object tracking principles and filtering methods, as well as resulting tracking accuracy and visualization errors in AR. Thus, we aim to identify an adequate sensor concept for broad applicability. Furthermore, novel gestures and interaction methods for spatial programming will be examined.

Further work is also on integrating our mobile programming system into software tools of the digital factory. CAD files of workpieces could be transmitted to the handheld device automatically for recognition or manipulation purposes. Subsequently, the extraction of the dimensions for object recognition could be automated. In accordance with the latest trend toward cyber-physical systems, data between handheld, robot and a central server or cloud architecture could be

exchanged, e.g., to provide fast information about robot programs and programming processes to enhance process planning and operating.



**Figure 12.** Robot-based path welding by demonstration.

The possible fields of application are numerous. A gesture-based definition of complex tasks could be particularly interesting regarding spray painting, arc welding and adhesive bonding tasks. Within an ongoing project, we are currently examining an arc welding scenario which consists of a gestural demonstration of infeed and welding path through finger movements. Subsequently, the welding path and poses are visualized in the AR application on the handheld, where the user can manipulate the virtual objects, e.g., to manipulate tool orientation of single poses. Fig. 12 illustrates the principle of robot-based path welding by demonstration. AR-based spatial evaluation of programs on ubiquitous handheld devices could potentially fit nearly any robotic application. Among the definition of pick-and-place tasks, a gesture-based definition of more complex tasks may demand additional sensors to achieve a required accuracy which currently cannot be reached by most markerless motion tracking systems and algorithms. Further limits of the applicability are caused by the complexity of special robot tasks, e.g., in-process path planning for dynamic environments.

## 5. Conclusion

In this paper, we presented an intuitive programming system for industrial robots on modern smartphones and tablet PCs. Programming by demonstration of the task was carried out by motion tracking of gestural bare-hand interaction. The evaluation through AR provides simultaneous information about the interaction and programming process. The overall potential of spatial programming lies in a reduction of programming times and of use. By means of spatial robot programming, the programmer is supported by the help of a highly efficient assistance system for online programming. The intuitive usage of spatial interaction represents a major simplification for industrial robot programming

compared to conventional methods. Our system enables non-specialists to define, evaluate and manipulate robot tasks without broad practice and experience.

## 6. References

- [1] Meyer Ch, Schraft R.D (2006) An Intuitive Teaching Method for Small and Medium Enterprises. *Intelligent Production Machines and Systems*. 2nd I\*PROMS Virtual International Conference. pp. 568-571.
- [2] Forge S, Blackman C (2010) A Helping Hand for Europe: The Competitive Outlook for the EU Robotics Industry. EUR - Scientific and Technical Research Reports. Publications Office of the European Union.
- [3] Pires J.N (2008) New challenges for industrial robotic cell programming. *Industrial Robot: An International Journal*. vol. 36 (1).
- [4] Akan, B, Ameri A, Curuklu B, Asplund L (2011) Intuitive industrial robot programming through incremental multimodal language and augmented. 2011 IEEE International Conference on Robotics and Automation (ICRA). pp. 3934-3939.
- [5] Marin R, Sanz P.J, Nebot P, Wirz R (2005) A multimodal interface to control a robot arm via the web: a case study on remote programming. *IEEE Transactions on Industrial Electronics*. vol. 52 (6): 1506-1520.
- [6] Neto P, Pires J.N (2010) High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition. *Industrial Robot: An International Journal*. vol. 37 (2).
- [7] Lambrecht J, Kleinsorge M, Krüger J (2011) Markerless gesture-based motion control and programming of industrial robots. 2011 IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA).
- [8] Dillmann R, Zoellner R, Ehrenmann M, Rogalla O (2002) Interactive Natural Programming of Robots: Introductory Overview. 2nd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments 2002.
- [9] Hasanuzzaman M, Zhang T, Ampornaramveth V, Ueno H (2006) Gesture-based human-robot interaction using a knowledge-based software platform. *Industrial Robot: An International Journal*. vol. 33 (1): 37-49.
- [10] Ameri A.E, Akan B, Curuklu B (2010) Augmented Reality Meets Industry: Interactive Robot Programming. Conference SIGRAD 2010 - Content Aggregation and Visualization.
- [11] Huang, W, Alem L (2011) Supporting Hand Gestures in Mobile Remote Collaboration: A Usability Evaluation. The 25th BCS Conference on Human Computer Interaction, Newcastle-upon-Tyne, UK.

- [12] Hashimoto A, Ishida A, Inami M, Igarashi T (2011) TouchMe: An Augmented Reality Based Remote Robot Manipulation. The 21st International Conference on Artificial Reality and Telexistence, Proceedings of ICAT2011.
- [13] Kazi A, Bunsendal J, Haag D, Baum R, Bischoff R (2005) Next Generation Teach Pendants for Industrial Robots. *Advances in Human-Robot Interaction*. Springer-Verlag Berlin Heidelberg 2005. pp. 47-66.
- [14] Zollner R, Asfour T, Dillmann R (2004) Programming by demonstration: dual-arm manipulation tasks for humanoid robots. *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*. vol. 1: 479 - 484.
- [15] Asfour T, Azad P, Gyarfas F, Dillmann R (2008) Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. *International Journal of Humanoid Robotics*. vol. 5 (2): 183-202.
- [16] Ong S.K, Yuan M.L, Nee A.Y.C (2008) Augmented reality applications in manufacturing: a survey. *International Journal of Production Research*. vol. 46 (10): 2707-2742.
- [17] Reinhart G, Munzert U, Vogl W (2008) A programming system for robot-based remote-laser-welding with conventional optics. *CIRP Annals - Manufacturing Technology*. vol. 57 (1): 37-40.
- [18] Chong J.W.S, Nee A.Y.C, Youcef-Toumi K, Ong S.K (2005) An Application of Augmented Reality (AR) in the Teaching of an Arc Welding Robot. *Singapore-MIT Alliance Symposium*.
- [19] Fang H.C, Ong S.K, Nee A.Y.C (2011) Interactive robot trajectory planning and simulation using Augmented Reality. *Robotics and Computer-Integrated Manufacturing*. vol. 28 (2): 227-237.
- [20] Ng L, Oon S, Ong S.K, Nee A.Y.C (2011) GARDE: a gesture-based augmented reality design evaluation system. In: *International Journal on Interactive Design and Manufacturing*. vol. 5: 85-94.
- [21] Ong S.K, Wang Z.B (2011) Augmented assembly technologies based on 3D bare-hands interaction. *CIRP Annals - Manufacturing Technology*. vol. 60 (1): 1-4.
- [22] Wang Z.B, Shen Y, Ong S.K, Nee A.Y.C (2009) Assembly Design and Evaluation Based on Bare-Hand Interaction in an Augmented Reality Environment. *CW '09. International Conference on CyberWorlds*. pp. 21-28.
- [23] Lambrecht J, Chemnitz M, Krüger J (2011) Control layer for multi-vendor industrial robot interaction providing integration of supervisory process control and multifunctional control units. *2011 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. pp. 115-120.
- [24] Furukawa Y, Ponce J (2010) Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. vol. 32 (8):1362-1376.
- [25] Esteban I, Dijk J, Groen F.C.A (2010) FIT3D toolbox: multiple view geometry and 3D reconstruction for matlab. *International Symposium on Security + Defence Europe (SPIE)*.
- [26] Oikonomidis I, Kyriazis N, Argyros A (2011) Efficient model-based 3D tracking of hand articulations using Kinect. *Proceedings of the British Machine Vision Conference 2011*, pp. 101.1-101.11.
- [27] Bradski G.R (1998) Real time face and object tracking as a component of a perceptual user interface. *Fourth IEEE Workshop on Applications of Computer Vision*. pp. 214-219.
- [28] Lee D, Lee S.G (2011) Vision-Based Finger Action Recognition by Angle Detection and Contour Analysis. *ETRI Journal*. vol. 33 (3): 415-422.
- [29] Wilhelm M, Roscher D, Blumendorf M, Albayrak S (2010) A Trajectory-Based Approach for Device Independent Gesture Recognition in Multimodal User Interfaces. *Haptic and Audio Interaction Design*, vol. 6306: 197-206.
- [30] Gao X.S, Hou XR, Tang J, Cheng H.F (2003) Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 25 (8): 930-943.