Interactive Laser-Projection for Programming Industrial Robots

M. F. Zaeh and W. Vogl
Technical University Munich
Institute for Machine Tools
Boltzmannstrasse 15
D-85748 Garching, Germany

ABSTRACT

A method for intuitive and efficient programming of industrial robots based on Augmented Reality (AR) is presented, in which tool trajectories and target coordinates are interactively visualized and manipulated in the robot's environment by means of laser projection. The virtual information relevant for programming, such as trajectories and target coordinates, is projected into the robot's environment and can be manipulated interactively. For an intuitive and efficient user input to the system, spatial interaction techniques have been developed, which enable the user to virtually draw the desired motion paths for processing a work piece surface, directly onto the respective object. The discussed method has been implemented in an integrated AR-user interface and has been initially evaluated in an experimental programming scenario. The obtained results indicate that it enables significantly faster and easier programming of processing tasks compared to currently available shop-floor programming methods.

CR Categories: H5.2: [Information Systems]: User Interfaces, J.2 [Computer Applications]: Sciences and Engineering

Keywords: Human-Robot Interaction, Spatial AR, Projection

1 INTRODUCTION

Industrial robots are flexible production resources that are widely used within automated mass production today. While their performance has been steadily increasing over the last decades, their programming is still a time-consuming process that can only be conducted through trained experts. The major methods for programming industrial robots can be discerned into online, conducted at the real setup based on the robot controller and offline, conducted afar from the real setup based on a 3D-simulation of the robots environment and its controller. Offline programming is an efficient programming method, which however requires detailed 3D-models of the production cell and dedicated simulation experts. Furthermore, discrepancies between models and the real environment make later adaptations on the shop floor almost unavoidable.

For generation of new programs or adaptation of existing ones on the shop floor, online-programming, is the most widely used method. Commands and target coordinates are either entered alphanumerically to the robot's control panel or the robot is directed by its control panel and the passed target points are stored (teach-in), such that a motion program is successively created. This process is time consuming as the single steps of a program have to be defined mainly by moving the robot along the desired path while – apart from the current pose of the robot – no visual feedback of the programming process is provided to the user at all. Therefore this method is not very efficient and requires a high level of skills and caution from its user.

Augmented reality, i. e. interactively overlaying the real environment with virtual spatial information, can be used to make the advantages of graphically-interactive simulation directly available in the real production environment and to provide an efficient and intuitive communication channel for spatial information. The information overlay can be achieved by video mixing (video see-through, VST), optical combining (optical see-through, OST) or projection of the virtual data into the environment. AR offers the possibility to visualize the motions and trajectories of a robot overlaid with the real environment and enables the user to intuitively interact with the spatial information. This way, user interfaces can be realized, which make robot programming more efficient and far more intuitive.

2 STATE OF THE ART

The first applications of AR in robotics were in the area of telepresence, where virtual 3D orientation cues were rendered into a video stream of the remote environment to improve spatial perception of the human operator [5, 7]. Dillman et al. used AR to display and to program the paths of a mobile robot [4]. Also in the area of industrial robots, a training application has been presented [2], where coordinate systems and motion data are visualized to novel robot users in an intuitive way through AR. Pettersen et al. presented a prototype, where paint spraying results were visualized beforehand by means of VST and the motions of the robot could be defined using a tracked input device [6].

The above approaches are based on VST-technology and rely on head-worn or screen-based displays. However, using VST-HMDs or handheld tablets, exact fine manipulation, as it is required for defining target coordinates for the robot with respect to real object surfaces is cumbersome for the user [4]. Optical see-through displays can partially compensate this aspect, but also suffer from accommodation problems in close-range interaction. Furthermore, they are more complicated to handle due to the user-specific calibration. Both, OST and VST require display devices which impair the user and strongly restrict the field-of-view.

Projection-based or spatial AR, offers a chance to overcome these issues and to make the potential of AR-technology available in industrial robot applications without being dependent on display devices. Raskar, Bimber et al. have shown projection with compensation of both surface texture and shape, mosaicing of several video projectors and also stereoscopic projection-based augmentation [1]. So far, these approaches have been mainly focused on multimedia applications. For applications in robot programming, the resolution and visibility of the projection with lens-based beamers is rather limited. Even though, the focus of the projection can be shifted or extended through multiple projectors, the focus area is still mainly planar and focussed projection on complex 3D-geometries for e.g. visualizing processing trajectories on a work piece with strongly varying height profiles is not possible.

This article describes an approach of applying projection-based AR to the domain of industrial robot programming and presents a method, in which interactive laser projection with geometric feedback is used to realize precise spatial interaction between the programmer and the robot (section 3). A prototypical system has been implemented and integrated with classic VST-approaches in an AR-based programming interface for industrial robots (4). An experimental evaluation of the developed user interface is described in section (5) and a conclusion is given in section (6).

3 Interactive Laser-projection

Scanning laser projectors deflect a laser beam through a mirror attached to a galvanometer. They can project coloured polygon lines onto 3D-surfaces precisely and are widely used in the assembly domain to give static guidance for assembly workers e. g. for placing layers of composite materials. With these projectors, path information of the robot and processing trajectories, such as welding seams or adhesive dispensation, can be displayed to the user in exact reference to workpiece surfaces.

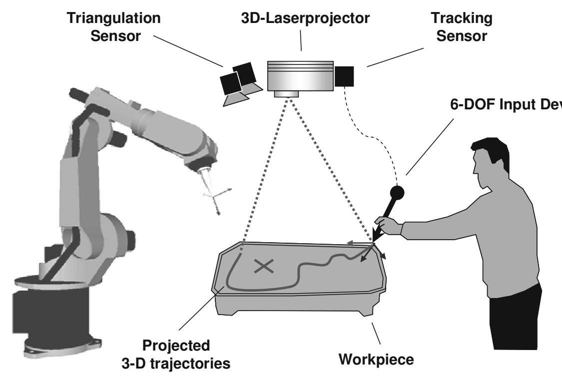

Figure 1: Interactive laser-projection for programming a processing task on a workpiece surface

3.1. Principle

An industrial laser projector is used to display the trajectories on the real workpiece. The device is capable of displaying 3D-polygon lines in 3 different colours (red, green, yellow) with high precision (0.5 mm RMS at 4 m distance) over a large working range (80° x 80° x 4 m). The projection contents are interactively generated according to the current program data of the robot and the user input. A handheld pointing device, whose position and orientation is being tracked, serves as central input device for the definition of target coordinates for the robot.

Combining the interactive projection and the handheld input device, a user interface is realized that enables the programmer to edit the processing coordinates for the robot directly on workpiece surfaces and which gives interactive visual feedback about the resulting motions and trajectories (compare fig. 1). In contrast to VST or OST approaches, the user is not restricted by an additional display device and precise, immediate interaction with the projected virtual information is possible. Furthermore, the projected laser light can be captured by calibrated cameras in order to measure surface geometry based on the principle of laser triangulation [3] and to realize projection with geometric feedback.

3.2. Pen-based input and interaction

A handheld pointer, which is tracked in real-time, serves as the main input device with which the user can define target coordinates for the frame of the robot's tool-centre-point-frame (TCP) in six degrees of freedom (6-DOF) and interact with the projected contents. Attached to the tool tip of the stylus, a virtual coordinate system, representing the robot's TCP-frame is projected, which indicates the target position and orientation of the robot. By updating the projected contents based on the tracking data at an interactive rate (here: ~12Hz), the impression of sketching the target coordinates onto the work piece surface can be given the user as if he was using a real pen (fig. 2). The stylus is equipped with a wireless infrared sender providing two input buttons, with which the user can give additional commands for e.g. selection or confirmation.

In this case, the input device is being tracked by an optical outside-in tracking system. Two infrared cameras with a baseline of 1 m track several retro-reflective markers attached to the pointing device. The tracking requirements of the application concerning precision, mobility and independence from environmental influence make this system preferable to other tracking principles, such as magnetic or acoustic tracking. The reference frame of the pointer is calibrated to the tip of the device, which can be tracked with an RMS of 0.4 mm concerning position and an angular deviation of 0.15° over the whole measurement volume (compare fig. 3). Using this device, the target frames for the robot can be edited with sufficient precision through both tactile measurement on surfaces and through positioning of the pointer in space.

3.3. Spatial interaction and editing mechanisms

For efficient and intuitive editing of processing trajectories for the robot in a graphically-interactive way, various spatial input mechanisms are provided to the user. These are based on interactive laser projection for visualization and the tracked input device, to which a virtual cursor is attached.

- Direct spatial input: Target frames for the robot are defined by pointing to the desired target position and orienting the cursor until the input is confirmed through pressing an input button.
- Selection by pointing: Points and path segments can be selected by approaching them with the pointer's tip closer than a predefined distance (here: 3 mm). The approached element is highlighted and the user can confirm the selection through the input button.
- Drag-and-drop: Path elements or single points are picked up by selecting them as described and are then moved around as long as the input button is pressed while the projection of the resulting path is permanently updated.
- Spline modeling: A B-Spline representation of the robots trace is fitted on the previously defined waypoints. By altering the spline's base points, a smooth path can be modeled and interactively edited by the user.
- Editing geometric features: Circles, arcs and squares are defined by characteristic points and then visually adapted by modifying these points.

With the described spatial interaction mechanisms, the user can efficiently edit the spatial coordinates of a robot program directly

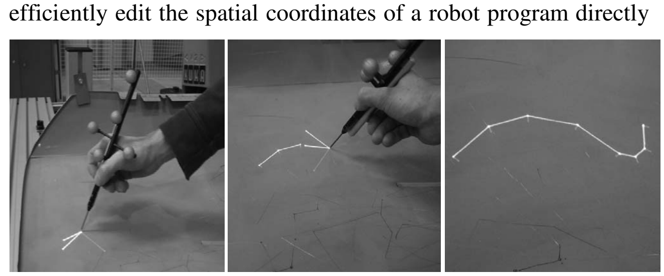
Figure 2: Interactive editing of the projected trajectories

on a workpiece's surface in a manner very similar to CAD-based programming environments with screen and mouse interfaces.

3.4. Combining projection and VST-Visualization

The projection can precisely display target coordinates on complex object surfaces. However, for the programming process, not only the target coordinates are relevant but also the motion behaviour of the robot. As there are always multiple solutions to the inverse kinematics of a robotic manipulator arm and the interpolation behaviour cannot intuitively be foreseen, it is necessary to give the programmer additional visual feedback about the motions of the robot. A VST-visualization is well suited to complement the projection interface for this purpose. An animated 3D-kinematic model of the robot is superimposed with the real robot and is shown to the user on a monitor. This visualization gives an immediate feedback of the robot's motions during the programming process on the one hand as a virtual model follows the movements of the input device in real-time (fig. 3, lower right). On the other hand, the AR-Visualization provides a safe preview of the program execution through an animation of the virtual model (fig. 6) for collision avoidance.

4 IMPLEMENTED SYSTEM

The discussed interaction method has been realized in a prototypical user interface. Figure 3 gives an overview of the implemented system: The central medium for visualization and editing of trajectories is laser projection (1), which shows target coordinates in exact reference to the real workpiece. For the definition of target coordinates and for direct manipulation, a handheld stylus (4), equipped with two input buttons is tracked in 6-DOF by an optical tracking system (2). For controlling the overall application and textual editing of the robot program, a 2D-GUI is provided, which is displayed on a tablet-PC (5). On this tablet-PC, an additional VST-visualization based on ARToolkit and OpenVRML indicates the poses and motions of the robot to the user in real-time. These are displayed overlaid on images of the robots working range captured by fixed video cameras (3).

Figure 4 shows the components and their interlinkage in a schematic way. The central element is the interaction kernel, which directs the control flow of the application and updates the model data. In each update loop, the user input, from the 6-DOF input device and from the 2D-GUI are processed and the model data, i. e. motion commands, trajectories and robot poses are updated accordingly and visualized consistently through the laser projection, the VST-visualization and the 2D-GUI. The system has been implemented in C++ and is running on a windows dual processor machine (2 x 2.4 GHz). The interaction control thread and the GUI are running at a rate of 25 Hz, the laser projection is updated at approx. 12Hz and the video rate is currently 25 Hz.

For the calibration of the overall system, a semi-automated calibration mechanism of the projector is used, which is based on four small reflective mirrors mounted in the working space. The positions of these mirrors are initially measured using the robot and a measurement tip, such that the relative transform between the projector and the robot's coordinate system can be computed. The IR-Tracking system is then calibrated with respect to the mirrors by probing them successively with the handheld measurement tool. Finally, the pose of the triangulation cameras is computed using the 2D-positions of the mirrors in the camera images and the known world positions of the mirrors.

Currently, the system is being transformed into a mobile setup, in which the relative position of the projector, the IR-tracker and the triangulation cameras is fixed. This way, the setup can be positioned freely on a tripod and quickly recalibrated with respect to the workspace by means of the projector's automated calibration, a process which takes less than 2 minutes.

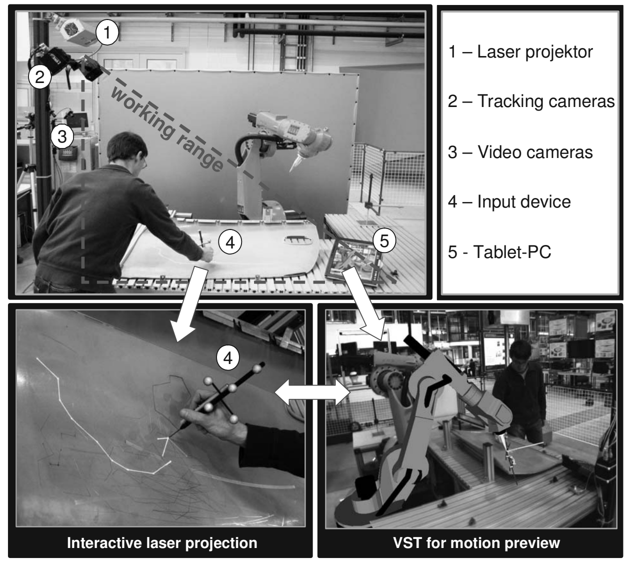

Figure 3: Implemented AR-Interface for robot programming

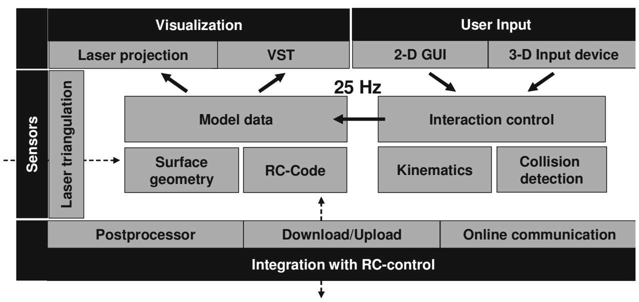

Figure 4: System structure

5 EXPERIMENTAL EVALUATION

The implemented system has been tested and evaluated in an experiment, in which the test persons had to accomplish a small programming task with a conventional shop-floor programming method (teach-in) on the one hand and with the developed AR-interface on the other hand. A trajectory connecting twelve target points, comprising linear motions, point-to-point motions and a circular arc, were engraved in the reference workpieces and had to be programmed by the test persons (compare figure 5). The scenario has been chosen, such that it resembles the geometric requirements as they occur in programming typical surface-related applications such as welding or dispensing of adhesives. Nine male test persons with equivalent background in production engineering and a medium level of experience in programming industrial robots had to accomplish the given task with both the classic teach-in method, in which the robot's tool (in this case symbolised by a metal cone) is moved along the desired trajectory and points are stored successively and the AR-interface. For both methods, the performance in terms of programming time and the accuracy of the results have been measured. The times that the users required for programming the reference tasks are depicted in figure 7. The duration with classic teach-in was on average 480 seconds, while the average time with the AR-based interface was only 90 seconds.

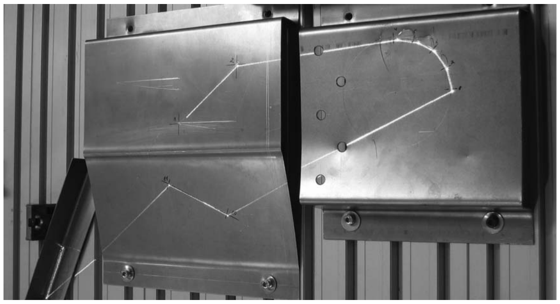

Figure 5: Programming task for experimental evaluation (with resulting processing trajectory)

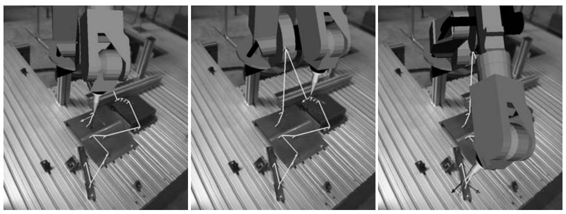

Figure 6: Motion preview with virtual robot model

On average, the definition of spatial target coordinates and trajectories was accomplished in less than one fifth of the time that a programmer requires with current conventional methods. The user interface and its central metaphor of drawing the desired trajectories were classified as very intuitive by all test persons. Even though the AR-interface was completely new to the test persons, the tasks were accomplished in significantly less time and with a comparable and partially even better precision. The positioning quality, in terms of maximum deviation from the demanded target points reached with AR was equal as the precision of the teach-in results for deviations lateral to the workpiece surface. However for deviations normal to the surface, the AR-based interface yielded even better results. This is mainly due to the difficulty inherent in positioning the robot's tool tip directly on surfaces. The experiments also showed that fine positioning of target points with the handheld pointer is limited to a precision of ~0.5 mm when the tip is in surface contact. It strongly declines, when the tip has to be positioned above a surface. With the current setup, occlusions of both the projected trajectories and the optical tracking system occasionally occurred during the experiments. This can be addressed by combining several projectors in one display and increasing the number of tracking cameras. Another option is the realisation of an integrated programming device, which can be attached to the robot manipulator and can thus be positioned in a suitable way that avoids occlusions.

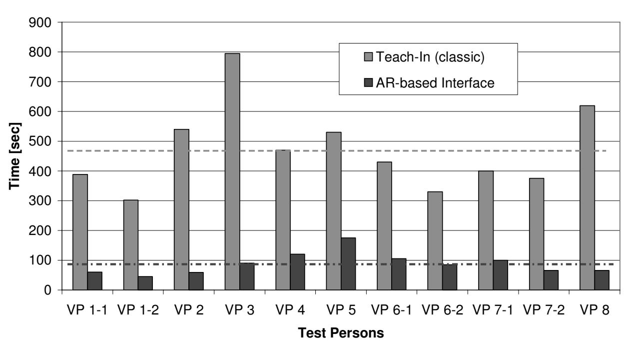

Figure 7: Time required for programming the reference task with classic interface vs. the AR-based user interface

6 Fields of Application

Applications for the system are mainly seen in small lot size production, especially in SMEs or handcraft companies, in which robots need to be reprogrammed frequently and in a flexible manner. The discussed system can thereby be used to program surface-related processing tasks such as joining and cutting of free-form parts. One example are robotic laser processes, in which a robot moves a laser optic along the work piece surface, such as laser welding, laser cutting or curing. All of these tasks require defined offsets between the surface and the laser and varying inclination angles relative to the surface. Programming these processes online through teach-in or by leading the robot through force-torque sensors is highly cumbersome. CAD-based offline programming would require concise CAD-models of the parts and dedicated simulation experts. Both are often not available in small lot size production and in SMEs respectively. Using the AR-interface, robot motions and processing trajectories for such applications can be edited quickly and intuitively, even by production workers without dedicated programming skills.

7 Summary

A method for programming industrial robots by means of interactive laser projection was described in this article and an integrated AR-based user interface was developed. The central elements of the interface are a tracked input device for definition of target coordinates and 3D-laser projection for interactive visualisation of the programmed trajectories, without requiring body-attached or handheld display devices. A prototype was realized and experimentally evaluated in comparison to conventional programming methods. These tests confirm that the method allows for a reduction of programming time of up to 80%. The tests also showed that the interface can be intuitively understood and its usage can be easily learned by novel users.

Acknowledgements

This project has been partially founded by the German Research Foundation (DFG) within the project "Augmented Reality in Assembly Planning". The authors also thank LAP Laser GmbH for the generous support of the project.

References

[1] Bimber, O.; Emmerling, A.; Klemmerer, T.: Embedded Entertainment with Smart Projectors. IEEE Computer 38 (2005) 1, pp. 48-55.
[2] Bischoff, R.; Kazi, A.: Perspectives on Augmented Reality Based Human-Robot Interaction with Industrial Robots. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent robotics And Systems (IROS); 2004, pp. 3226-3231.
[3] de la Hamette, P.; von Waldrich, M.; Tröster, G.: Laser Triangulation as a Means of robust Input for Wearable Computers. In: Proc. of 8th IEEE Int. Symposium of Wearable Computers; 2004
[4] Giesler, B.; Salb, T.; Steinhaus, P.; Dillmann, R.: Using Augmented Reality to Interact with an Autonomous Mobile Plattform. Proc. of the 2004 IEEE Conference on Robotics and Automation (ICRA), 2004, pp. 1009-1014.
[5] Milgram, P.; Drasic, D.; Zhai, S.: Applications of Augmented Reality in Human-Robot Communication. Proc. of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1993); 1993, pp. 1244-1249.
[6] Pettersen, T.; Skourup, C.; Engedal, T.: Users regain control of industrial robots. Proc. of the 2004 IEEE International Conference Mechatronics and Robotics (MECHROB). Aachen 2004.
[7] Pretlove, J.: Augmenting Reality for Telerobotics: Unifying Real and Virtual Worlds. Industrial Robot 25 (1998) 6, pp. 401-407.