Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 44 (2016) 193 – 198
www.elsevier.com/locate/procedia

6th CIRP Conference on Assembly Technologies and Systems (CATS)

AR-Enhanced Human-Robot-Interaction - Methodologies, Algorithms, Tools

*Nils Andersson[a], Angelos Argyrou[b], Frank Nägele[c], Fernando Ubis [d], Urko Esnaola Campos[e], Maite Ortiz de Zarate [e], Robert Wilterdink [f]
[a]EON Development AB, Datavägen 10C S-436 32 Askim, Gotheburg,, Sweden
[b]Laboratory for Manufacturing Systems and Automation, Department of Mechanical Engineering and Aeronautic, University of Patras, 26500 Patras, Greece
[c]Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstasse 12, 70569 Stuttgart, Germany
[d]Visual Components, Vänrikinkuja 2, FI-02600 Espoo, Finland
[e]TECNALIA Research & Innovation, Parque Tecnológico de Alava, c/ Leonardo Da Vinci, 1,E-01510 Miñano (Alava), Spain
[f]Universal Robots, Energivej 25, 5260 Odense S, Denmark
* Corresponding author. Tel.: 46 (0) 31 748 43 34. *E-mail address:* nils@eonreality.com

---

Abstract

By using Augmented Reality in Human-Robot-Interaction scenarios we propose it is possible to improve training, programming, maintenance and process monitoring. AR Enhanced Human Robot Interaction means it is possible to conduct activities not only in a training facility with physical robot(s) but also in a complete virtual environment. By using virtual environments only a computer and possibly Head Mounting Display is required. This will reduce the bottlenecks for with overbooked physical training facilities. Physical environment for the activities with robot(s) will still be required, however using also virtual environments will increase flexibility and human operator can focus on training more complicated tasks.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
Keywords: Augmented Reality; Virtual Reality; Human-Robot-Interaction; Head-Mounted-Display

---

1. Introduction

The biggest handicap to the introduction of robots in assembly automation is that it is not "easy". Continuously changing product references make the assembly strategies continuously change and adapting the system to the changes is costly from the programming point of view. Our approach will utilize AR technology to combine the benefits of existing programming methodologies in order to develop a new novel and easy to use off-line programming by demonstration. AR-Enhanced Multimodal Robot-Programming Toolbox (AR-EMRPT) will be an integrated tool and it will encompass Programming-By- Demonstration, Programming-by-Instruction, Programming-by-Observation, and Contextual Programming techniques. The current practices in the field are presented in this study in order to analyze and select the most suitable enabling technologies for the development of AR-EMRPT. In the context of this study the AR-Enhanced Multimodal Robot-Programming methodology and prototypes are presented.

2. Current practices

2.1. Usage of AR/VR in robotics

Augmented and Virtual Reality (AR/VR) in Robotics as of today include how different hardware and software can collaborate with human and robot system to program, handle maintenance and errors. Most of the usage of AR and VR in Robotics today is limited to laboratory research projects and has not yet reached production. There are many very interesting projects at various locations throughout the world and the research results from the last years are very valuable [1], [2], [3].

The AR/VR in Robotics research presents a number of possibilities for the future like direct manipulation of robot skills and use of new low cost hardware.

Several research projects use some kind of hand tracking for moving robot arm. For the tracking part a number of different devices have been used: Kinect, Leap Motion, Optical tracking device [4]. All these projects show that it is possible to let a

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
doi:10.1016/j.procir.2016.03.022

robot arm follow a hand trajectory, however, the accuracy is not good for production with the possible exception of using an optical tracking system.

Modern mobile devices integrate gyro sensors very accurate. Using this capability it has been integrated solutions for controlling an industrial robot particularly the KUKA KR 6, robotic arm using an iPhone [5]. This approach is interesting since it combines rotation of a mobile device with touch input on the mobile display to control the robot arm.

Complete immersed Virtual Reality environment can be implemented with the Oculus VR device [2]. However the aforementioned solution lacks perception of real world and this is a serious barrier towards the utilization of such application in real assembly lines.

Recent research projects demonstrate what is possible to be performed and give good ideas for future advancements [4]. However, the industry does not yet really use AR/VR with robotics so far. Specifically, recent research projects have shown that it is possible to program a robot using AR and as well possible to move a robot arm using AR, however, none of the projects have really present a solution that works outside a well-controlled lab [6]. Our approach will focus on the development of application that will enable AR robot programming in non-structured environments.

2.2. Usage of simulation in robotics

Simulation has been recognized as an important tool in many areas. Robotics is no exception for this. In fact, simulation in robotics plays a very important role, perhaps more important than in many other fields [7]. Advanced robotics systems are complex and designing the right system is demanding and time consuming tasks, often even more complex than initially imagined [8]. Research effort has been also been carried out on simulation environments that would provide built-in algorithms for automatic optimization of workcell configuration [9]. Ability to visualize the cell configuration is essential not only for the designers themselves but also in communicating design issues among the design team and customers [15].

One major problem in program validation and cycle time analysis is that the algorithms defining the robot's motion behavior are proprietary and generally not publicly available. Having an insufficient model of the robot controller may significantly limit the usability of the simulation results. Aditionaly, it might lead to unexpected failures since the simulated behavior will not correspond to the actual behavior[8].

Towards this direction, many robot manufacturers provide emulators of their robot controllers. Virtual controller is a software that emulates the actual robot controller on a conventional personal computer. For the utilization of such tools a set of interfaces are requiered that will expose the information to simulation softwares. This is will be one of the main objectives of the current study.

2.3. Usage of tracker markers for augmented reality

Augmented Reality markers are useful object detection systems according to efficiency. Nevertheless, markers are not robust to rotations so other kind of detection systems should be carried out. AR markers are suitable solution for immediate future applications, but the system is not scalable as the number of detected objects increases.

Other kind of 2D detection approaches can be used in order to spot objects in the three dimensional space, such as intensity based approaches where texture information is required. For instance, AR markers are efficient but not scalable in terms of texture properties. In addition, detection of markers in high reflective materials is difficult task, while most of the industrial components involve metallic parts. Not to mention that 2D based approaches deal with serious occlusion issues.

To deal with rotation variations and occlusions, 3D based approaches are the most appropriate. 3D based approaches do not require texture information, although they can also make use of it in order to perform results. 3D based approaches are shape based detection techniques that use object's geometric properties in order to spot them in 3 dimensional space. The main 3D object localization approach is surface registration.

Surface registration method consists of obtaining the pose estimation of a particular 3D model by comparing it with a previously known located one. If a precise estimation is obtained, it means that the compared models have matched, so the object has been recognized.

Nevertheless, there is not an official categorization method, according to [10]. There are mainly two registration methods known as coarse and fine registration techniques. In general, both techniques may guarantee suitable pose estimation depending on the case, but sometimes none of them may reach a tight estimation. The main goal of the coarse registration is to compute an initial estimation between two clouds of 3D points using pair-correspondences of both. The goal of the fine registration is to come up with the most accurate solution minimizing distances among the closest correspondences.

Our approach aspires to reduce the extensive use of AR-Markers in industry. Figure 1 presents the steps the proposed methodology.

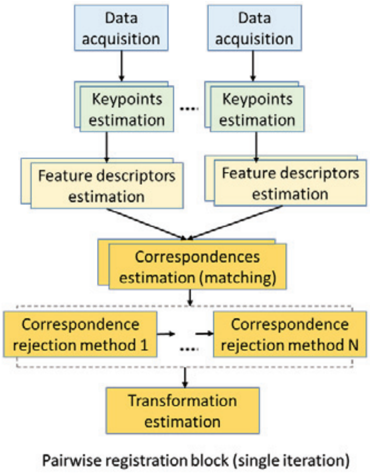


Fig. 1. An overview of pairwise registration.

2.4. CAD-based programming

Currently, industrial robots based automation represents the best solution for productivity and flexibility. Nevertheless, the programming of industrial robotic system for a specific application is still very difficult, time-consuming, and expensive. Today there are two main categories of robotic programming methods, which are, online programming (including lead-through and walk-through) and offline programming (OLP):

OLP method utilize 3D CAD data of a workpiece to generate and simulate robot programs, are widely used for automation system with large product volumes [12]. OLP is more complex than online programming, since the programming method not only needs to acquire the 3D robot targets but also needs to plan the trajectory of robot motion and optimize the sequence of the process [13]. The key steps of OLP are:

- Generation of 3D CAD model
- Tag creation
- Trajectory planning
- Process planning
- Post processing
- Simulation
- Calibration

The aim is to combine the features of both online and offline programming, extracting automatically the information of CAD to create tags (OLP) and create automatically trajectories using the skills of new robots(sensors), that help users to program a robot in an intuitive way, quickly, with a high-level of abstraction from the robot specific language [11].

Currently, there are some applications to extract information from CAD, but there are specific applications of determinate softwares and processes for example in Autodesk Inventor [13] or Kranendonk [14].

2.5. Programming by demonstration

The field of robotics, both in industrial and service contexts, has since its inception faced one challenge that yet remains to be solved: How can users program a robot to execute a task without the need to use a programming language? One of the dominant approaches that emerged during the 1980s to address this problem is Programming by Demonstration (PbD), also known as Imitation Learning (IL) or Learning from Demonstration (LfD) [16] [19]. This area of research aims at creating a framework that will allow robots to observe a human operator's actions by one of several means, abstract the motions and goals of the operation and then be able to reproduce them. At its core, this consists in finding the appropriate control policy to map states to actions [17].

With the robotics industry moving towards flexible, easily programmable robots, both in the industry and in the services, PbD might prove to be an important topic in coming years. This text will provide a brief survey of the area. Although the focus is on research applicable to industrial robotics, it is important to note that recent and current research centers on social and humanoid robotics. For a more comprehensive review, the reader is directed to [17] and [18], the first providing an overview of the historical development of PbD and recent techniques to augment or combine with it; the second, a comprehensive categorization of demonstration methods and ways to derive task policies from state-action examples.

Several demonstration methods for PbD have been widely investigated [22]. In [20], Tung and Kak demonstrate the use of a DataGlove to monitor the operators hand movements. The system is then able to infer the type of assembly task and then map the necessary actuator and grasp commands. Aleotti et al. propose combining a DataGlove with a virtual environment containing a model of the workspace in order to simplify teacher action tracking, reduce error and allow for simulation [21].Another approach is to use computer vision and object recognition algorithms to obtain a recording of the user demonstration.[21] and [23], both present early work using cameras for action recognition. A compilation of methods can be found in [18].

3. Approach

The approach chosen has been to divide the solution into four main different parts (Figure 2). Each part can be developed individually, the integration and communication between the parts is conducted by network using ROS (Robot Operating System).

ROS provides the framework [24] for interconnecting the different heterogeneous components described in this approach.

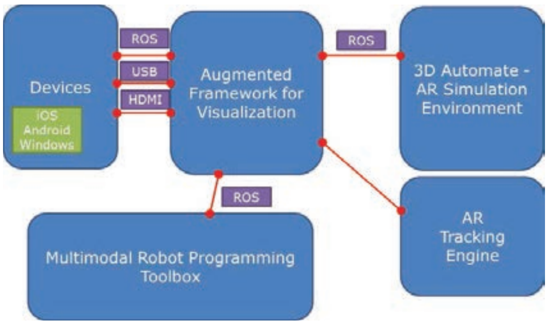


Fig. 2. Proposed solution overview.

3.1. Multimodal robot programming toolbox

The multimodal robot programming toolbox connects the various tools that may be used to program robots, like PbD or online/offline programming, to the data model that stores the program logic and application specific parameters.

To keep the whole architecture open for extension and flexible for change, the data model must not depend on the actual tools inside the toolbox. To decouple the data model from the tools, all changes to the data model are encapsulated within parameter requests. The requests are passed to the multimodal robot programming toolbox to be distributed to a tool without the data model knowing the actual tool providing the requested parameter. After a request is successfully executed by a tool, the returned values are stored in the data model.

This concept allows the user to select a desired tool from the various tools in the toolbox during runtime. For example, when a request for a robot target pose or trajectory is passed to the toolbox, a simple pop-up GUI asks the user whether to use PbD, a pose stored in the simulation tool or starting the free drive mode of the robot to jog it to the desired position.

ROS message types are used to define the parameter requests. The message types are hierarchical, which allows us to split up complex parameter sets into their (simpler) sub-components if no appropriate tool exists to define the parameter set in one single step. For example, a request for a robot pose can be split up into its sub-components position and orientation that can then be provided separately – or again be split up into their sub-components (e.g. x, y and z positions).

3.2. AR Simulation Environment

3D Automate is used for the AR Simulation Environment where the user can import CAD models, create 3D animations for workers and visualize the complete robotic cell.

It provides connectivity to ROS for reading out the joints' values of the simulated robot as well as reading in the joint's value of the virtual robot using rosbridge. In addition to the joints it is possible to access to additional properties of the simulated robot, the states and control the simulation.

The simulation tool will also provide with screenshots, graphics and videos to the AR-visualization framework for AR visualization used to visualize assembly task and instruct the operator.

The AR Simulation Environment will extend the functionalities of the Symbiotic Environment Design tool generating the simulation of the symbiotic workplace. This tool is used also for providing the visualization and animation of the simulated process.

3.3. Augmented reality framework for visualization

The AR Framework for Visualization will combine a number of algorithms, input from rosbridge for jobs and tasks and generate correct 3D Visualization images based on AR tracking and user position.

The main part of this framework is AR Visualization and instruction tool – visualizing state of robot cell and task to do. It is proposed to access to robot joint properties via the AR-Enhanced Simulation Tool. For programming this framework will use Robot Programming Tool Box and AR-Enhanced Contextual Programming. Communication with other parts in the system be realized through rosbridge Symbiotic Workplace Design Tool.

3.4. Devices

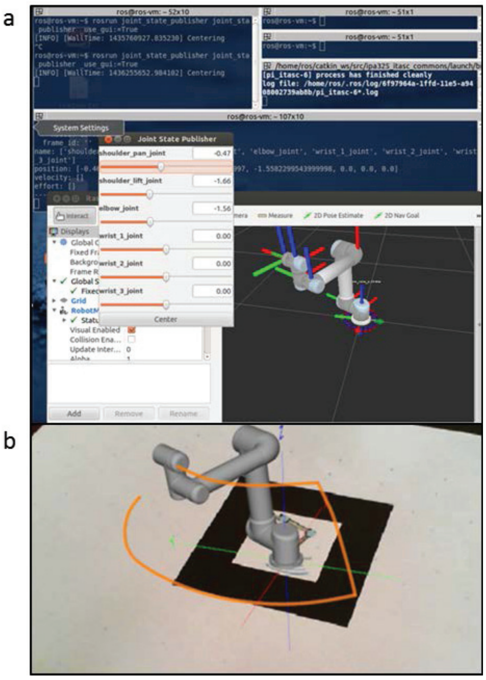
The devices subsystem will handle the supported display devices and also cameras. The proposed display devices to support consist of.

- Head Mounted Display(HMD) (PENNY)
- Tablet display (Android, iOS and Windows)
- Monitor (Windows)

HMDs and Monitors will be connected as standard display devices to a computer while tablets have the display built in to the device itself. Cameras will use rosbridge for communication with the other system parts like the AR Framework for Visualization.

4. Sample prototype

The feasibility of the idea is demonstrated through a demo scenario, using UR5 as a virtual model (Figure 3). The communication between RViz and EON Professional is presented. Specifically RViz publishes the state of the joints through RosBridge to EON and the latter visualizes the movement of the robot and vice versa.



```
ros@ros-vm:~$ rosrun joint_state_publisher joint_sta
 publisher _use_gui:=True
[INFO] [WallTime: 1435760927.835230] Centering
^C
ros@ros-vm:~$ rosrun joint_state_publisher joint_sta
 publisher _use_gui:=True
[INFO] [WallTime: 1436255652.984102] Centering
```

```
ros@ros-vm:~$
```

```
ros@ros-vm:~$
```

```
[pi_itasc-6] process has finished cleanly
log file: /home/ros/.ros/log/6f97964a-1ffd-11e5-a94
08002739ab8b/pi_itasc-6*.log
```

```
  frame_id: ''
name: ['shoulde                                  ', 'elbow_joint', 'wrist_1_joint', 'wrist_2_joint', 'wrist
_3_joint']
position: [-0.4                                  997, -1.5582299543999998, 0.0, 0.0, 0.0]
velocity: []
effort: []
```

shoulder_pan_joint -0.47
shoulder_lift_joint -1.66
elbow_joint -1.56
wrist_1_joint 0.00
wrist_2_joint 0.00
wrist_3_joint 0.00

Fig. 3. (a) RViz joint state publisher (b)EON Professional

5. Conclusions and future work

The current work constitute a roadmap for application of AR technologies in the industrial sector. The early implementation of the idea of such applications, constitutes a proof of concept for the proposed methodologies. Though the implementation of the first prototype the following points of interest have been identified:

- AR technology for enabling Human-Robot interaction in unstructured environment have been identified and categorized.
- It is easy to develop add-ons for communication into existing tools for our purpose by using a standard protocol, ROS.
- Algorithms for using various types of markers for AR is in global progress by various actors worldwide and we expect to see improved AR tracking in near future.

The proposed algorithms and methodologies will be enhanced and integrated together, in order to implement a generic and broad AR based Robot programming tool.

6. Acknowledgements

The work presented in the current paper is partially funded by the FP7 EU project LIAA (http://www.project-leanautomation.eu/).

References

[1] S. K. Ong, M. L. Yuan, and A. Y. C. Nee, “Augmented Reality Applications in Manufacturing: a Survey,” International Journal of Production Research, vol. 46, no. 10, pp. 2707–2742, May 2008.

[2] D. W. F. Van Krevelen and R. Poelman, “A Survey of Augmented Reality Technologies, Applications and Limitations,” International Journal of Virtual Reality, vol. 9, no. 2, p. 1, 2010.

[3] S. Iliano, V. Chimienti, and G. Dini, “Training by Augmented Reality in Industrial Environments: a Case Study.”

[4] Martínez Landa, “Problems Analysis and Solutions for the Establishment of Augmented Reality Technology in Maintenance and Education,” PhD Thesis, Tampereen teknillinen yliopisto. Julkaisu-Tampere University of Technology. Publication 1275, Tampere, Finland, 2015.

[5] iPhone Industrial Robot Control. [Online]. Available: https://www.youtube.com/watch?v=yFi7UL70zTo.

[6] J. Kosowatz, “Augmented Reality Not Quite Real for Engineers,” www.asme.org, 01-Sep-2009. [Online]. Available: http://www.asme.org/kb/news---articles/articles/design/augmented-reality-not-quite-real-for-engineers. [Accessed: 25-Sep-2012].

[7] Žlajpah L. Simulation of robotic manipulators. In 15th International Workshop on Robitics, Balatonfüred, Hungary; 2006.

[8] Nof SY. Handbook of industrial robotics. New York, NY, USA: John Wiley & Sons; 1999.

[9] Rossgoderer U, Woenckhaus C, Reinhart G ,Milberg J. A concept for automatical layout generation. In Robotics and Automation," IEEE International Conference on Robotics and Automation; 1995;8:800-805

[10] Salvi J, Matabosch C, Fofi D, Forest J. A review of recent range image registration methods with accuracy evaluation. Image and Vision Computing;2007;25:578-596.

[11] Pan Z, Polden J, Larkin N, Duin SV, Norrish J. Recent progress on programming methods for industrial robot. Robotics and Computer-Integrated Manufacturing; 2012;28:87-94.

[12] Polden J. Automated offline programming for low volume robotic manufacturing. Automated Offline Programming for Low Volume Robotc Manufacturing; 2014. p. 1-200.

[13] Neto P, Mendes N, Araújo R, Pires JN,. AP Moreira. High-level robot programming based on CAD: dealing with unpredictable environments. Industrial Robot: An International Journal; 2012:39;294-303.

[14] “Weld R. Hands-off robot programming. [Online]. Available: http://www.kranendonk.com/en/rinasweld.

[15] Lindskog E. Towards realistic visualisation of production systems (Thesis). Gothenburg, Sweden: Chalmers University of Technology; 2014.

[16] Billard A, Grollman D. Imitation learning in robots. In Encyclopedia of the Sciences of Learning, Springer; 2012. p. 1494-1496.

[17] Billard A, Calinon S, Dillmann R, Schaal S. Robot programming by demonstration. In Handbook of robotics, Berlin Heidelberg, Springer; 2008. p. 1371-1394.

[18] Argall BD, Chernova S, Veloso M, Browning B. A survey of robot learning from demonstration. Robotics and autonomous systems; 2009;57:469-483

[19] Friedrich H, Münch S, Dillmann R, Bocionek S, Sassin M. Robot programming by demonstration (RPD): Supporting the induction by human interaction. Machine Learning; 1996;23:163-189.

[20] Tung CP, Kak AC. Automatic learning of assembly tasks using a dataglove system. in IEEE/RSJ International Conference on Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots'; 1995.

[21] Aleotti J, Caselli S, Reggiani M. Leveraging on a virtual environment for robot programming by demonstration. Robotics and Autonomous Systems; 2004;47:163-161.

[22] Ikeuchi K, Suehiro T. Toward an assembly plan from observation. I. Task recognition with polyhedral bjects. IEEE Transactions on Robotics and Automation; 1994; 10:368-385.

[23] Kuniyoshi Y, Inaba M, Inoue H. Learning by watching: extracting reusable task knowledge from visual observation of human performance. IEEE Transactions on Robotics and Automation; 1994;10:799-822.

[24] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an Open-Source Robot Operating System,” in ICRA Workshop on Open Source Software, Kobe, Japan, 2009, vol. 3, p. 5.