Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 55 (2016) 1 – 5

www.elsevier.com/locate/procedia

5th CIRP Global Web Conference Research and Innovation for Future Production

High level robot programming using body and hand gestures

Panagiota Tsarouchi[a], Athanasios Athanasatos[a], Sotiris Makris[a], Xenofon Chatzigeorgiou[a], George Chryssolouris[a]*

[a]*University of Patras, Department of Mechanical Engineering and Aeronautics, Laboratory for Manufacturing Systems and Automation, 26500, Patras, Greece*

* Corresponding author. Tel.: +30-2610-997-264; fax: +30-2610-997-744. E-mail address: xrisol@lms.mech.upatras.gr

Abstract

Robot programming software tools are expected to be more intuitive and user friendly. This paper proposes a method for simplifying industrial robot programming using visual sensors detecting the human motions. A vocabulary of body and hand gestures is defined, allowing the movement of robot in different directions. An external controller application is used for the transformation between human and robot motions. On the robot side, a decoder application is developed translating the human messages into robot motions. The method is integrated within an open communication architecture based on Robot Operating System (ROS), enabling thus the easy extensibility with new functionalities. An automotive industry case study demonstrated the method, including the commanding of a dual arm robot for single and bi-manual motions.

Keywords: Gestures, Programming, Human Robot Interaction, Dual arm robot.

1. Introduction

Robot programming tools are expected to be more user friendly involving the latest trends of artificial intelligent, function blocks, universal programming languages and open source tools, anticipating the traditional robot programming methods [1]. Even non-expert robot programmers will be able to use those through shorter training processes. The intuitive robot programming includes demonstration techniques and instructive systems [2]. Different methods and sensors have been used for implementing more intuitive robot programming frameworks. Some examples include vision, voice, touch/force sensors [3], gloves, turn-rate and acceleration sensors [4], as well as artificial markers [5]. The main challenge faced in this area concerned the development of methods for the sensor knowledge data gathering and reproduction of these data from robots with robustness and accuracy [6].

An extended review on the use of human robot interaction (HRI) techniques was presented in [1]. The main direction of such systems includes both the design and development of multimodal communication frameworks for robot programming simplification. Among these research studies, a significant research attention has been paid on the use of gestures. For example, hand gestures are recognized by using data gloves in [7-8], where in [4] the sensor Wii is used. Examples using 3D cameras can be found in [9-12]. In [13] the human-robot dialogue is achieved with the help of a dynamic vocabulary. Kinect has been used as the recognition sensor in [14-18] enabling the online HRI. Kinect has also been used for ensuring safety in human robot coexistence [19]. Apart from HRI, the use of Kinect can also be found in studies for human computer interaction (HCI) [20-21]. Last but not least, the leap motion sensor was also selected for HRI [22-23].

Despite the wide research interest on intuitive interfaces for interacting and robot programming, there are still challenges related to the need for universal programming tools, advanced processing, transformation methods and the environment uncertainties. Related to the environment uncertainties, the natural language interactions, such as the gestures and voice commands, are not always applicable to industrial environments, due to their high level of noise, the lightning conditions and the dynamically changing environment.

doi:10.1016/j.procir.2016.09.020

Additionally, the development of gesture based vocabularies with complex and numerous gestures seems not to be user friendly for the human.

Taking into consideration the above challenges, this paper proposes a high level robot programming method using sensors detecting both body and hand gestures. The defined body gestures are static and the hand gestures are based on the dynamic movements of the hands. The user has two different options for interacting with the robot. There is no need for special robot programming training, even for commanding an industrial robot in this case. Safety can be ensured using the traditional emergency buttons, during the robot programming. The selected sensors for implementing body and hand gestures are the Microsoft Kinect and the leap motion respectively. Kinect sensor has become significantly popular in replacing the traditionally 3D cameras in applications where recognition accuracy is not needed. The introduction of leap motion in robot programming is relatively new allowing the detection of hand gestures for commanding the robot.

2. Method overview

The proposed high level robot programming method is based on the open architecture presented in Figure 1. In this architecture, the high level robot programming in achieved by the implementation of high level commands in the form of a gestures' vocabulary. This vocabulary includes both body and hand gestures, that are recognized using data from two different tracking devices. The first device is an RGB-d camera allowing the recognition of body gestures, while the hand gestures are recognized using the similar device called leap motion. This sensor includes two monochromatic IR cameras and three infrared LEDs.

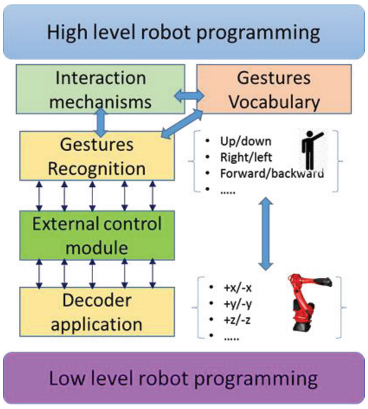

Fig. 1. System architecture.

The available interaction mechanisms (sensors) are represented within a software module (Fig. 1), allowing the detection and recognition of the gestures ('Gestures Vocabulary', 'Gestures Recognition'). The recognized gestures are published in "rostopic" as messages, where third applications can subscribe to listen these messages through ROS middleware. These messages represent the result of recognition in each topic e.g. /Left_gesture. These two modules do not communicate with each other but they are managed by the external control module. The control of the gestured based programming is managed by this module as well. This module is directly connected to the robot controller, by exchanging messages through TCP/IP protocol. Robot starts the movement that the human has commanded in order to allow the easy programming in tasks that high accuracy is not required.

A body and hand gestures vocabulary has been defined. The body gestures involves human body postures that are able to control the motion of a robot arm in different directions along a user frame. The defined gestures are static and allow to an operator to move the robot in the 6 different directions, namely +/-x, +/-y and +/-z around any selected frame (Fig. 2).

Similarly, to the body gestures, hand gestures are defined within this vocabulary (developed as functions using Python). These gestures are dynamic motions of the human hands, involving different number of fingers in each of them in order to secure the uniqueness of each motion. The use of these 6 different hand gestures enables the same motions as the body gestures. The hand gestures are more useful in the case that the human is closer to the robot for interaction of programming purposes.

The movement of the robot in the +x direction is achieved with the up gesture, where the operator extends the right arm upwards and holds the left hand down. The robot will move in the -x with down gesture and it can be done when the left hand is extended upwards and the right one held down. Using the hand gestures, the robot will go up if the operator uses one hand and swipes it upwards contrariwise it will go down if the finger is swiped downwards.

The movement along the +y direction is accomplished with the left gesture which includes the left hand extended at the height of the shoulder pointing to the left direction. On the contrary the robot will move along the -y with the left gesture involving the right hand extended with the same way but pointing to the right direction. Using the hand gestures, the gestures for right and left are recognized using two fingers swiping from left to right and the opposite.

The movement along the +z direction is achieved using the forward gesture, where the right hand is extended upwards and the left hand is at the height of the shoulder pointing to the left direction. The movement in the opposite direction is achieved by using the opposite hands, the left upwards and the right at the height of the shoulder pointing to the right (backward gesture).

The same movements are achieved using the relevant gestures, where three fingers swiping from left to right and the opposite. Finally, in order to stop the robot during a motion, two static body and hand gestures are available. In this way, the robot motor drivers are switched off and the robot will not remain active.

Following the recognition results of the gestures and how these are expected to be executed in the robot controller, communication between the external controller module and the decoder application is established. In this way, the robot receives the human commands, allowing the execution of them, translated to the described motions.

This method is offered for online interaction with an industrial robot, even in the case of multi-arm robotic systems, enabling the coordinated motions execution.

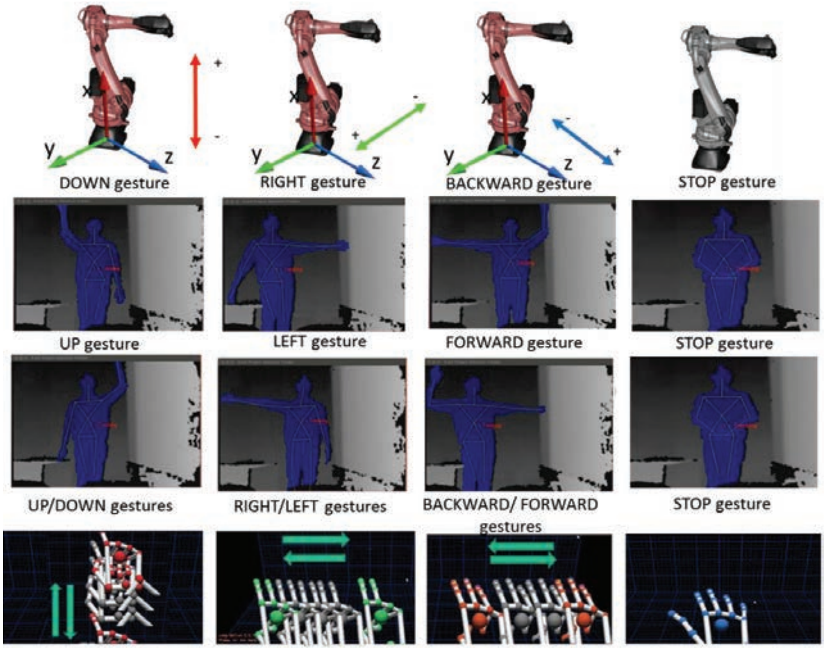

Fig. 2. High level commands for robot programming.

The definition of the vocabulary can be easily extended in order to involve more gestures. Additionally, in the proposed architecture other interaction mechanisms can be easily added such as data gloves, voice commands, augmented reality applications etc. Last but not least, time saving is achieved during a robot program, since direct and more natural interaction ways are used.

3. System implementation

The proposed method has been implemented as an independent software programming tool that can be applied in different robot platforms. ROS has been used as a middleware enabling the extensibility, maintainability and ability to easily transfer the tool in different platforms. The proposed system can be easily transferred in different robot platforms with minimum changes on the robot control side (decoder application). The hardware architecture is illustrated in Fig.3.

The two sensors are connected via USB to a Linux based PC where the programming software is running. This computer is connected to the robot controller through an Ethernet cable, being in the same network with the PC. The gestures vocabulary in the form of functions and their recognition modules are developed in Python, when they are using the available body and hand gestures tracker applications.

The Kinect tracker along with the proposed software allows the recognition of 18 human skeleton nodes (e.g. right or left hand node). Depending on the nodes values, the recognition module decides for the posture based on the functions that have been developed within the vocabulary.

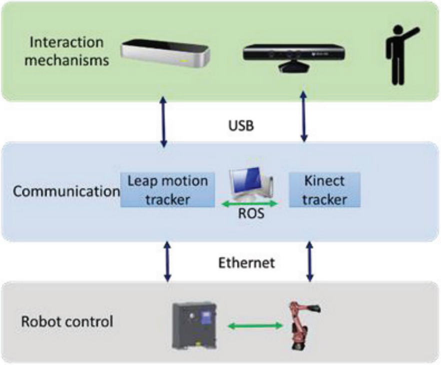

Fig. 3. Programming tool- hardware architecture.

In similar way, the hand gestures are recognized, using the data recorded from the leap motion sensor. The external_control topic, running on the same PC manages the communication between the human commands and the robot controller. The ROS graph of the proposed framework includes the developed rostopics (/Body_Tracker, /skeleton, /Hands_tracker, /Gestures_Recognition, /Hands_Gestures, /Body_Gestures) that are finally managed by the external_control topic (Fig. 4).

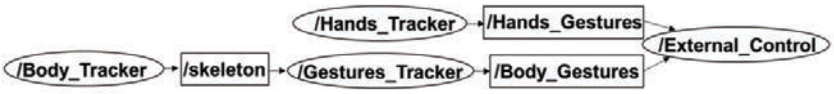

Fig. 4. The ROS graph of high-level robot programming framework.

The decoder application receives the data through a TCP/ IP socket in a string format. The robot motion starts since the commands are decoded. The robot speed is defined in 20% during programming. The response time for the two sensors is estimated around 43ms+-4ms, while the communication time with the robot controller is measured in 115+-6ms.

4. Case study

The proposed framework has been applied to an automotive industry case study for the programming of dual arm robot during the assembly of a vehicle dashboard vehicle. The robot program for this case, includes the motions for both single, synchronized in time and bi-manual assembly operations. A list of high level robot tasks are presented in Table 1. The sequence of these tasks has been described in [24].

The gestures that are used in each different robot operation are included in this table. Some examples of the high level operations and the gestures are presented in Fig.5. These example concern the programming of four different operations, starting from the 'pick up traverse' task to the 'place fuse box' task. The use of body and hand gestures are both demonstrated in this example.

Offline experiments with the developed recognition modules on body and hand gestures shown that the recognition rate was 93% and 96% respectively.

Table 1. High level robot programming- Dashboard assembly case.

High level robot task	High level robot operation	Gesture
Pick up traverse	ROTATE()	LEFT
	BI_APPROACH()	DOWN
	BI_INSERT()	BACKWARD
	BI_MOVE_AWAY()	UP
	ROTATE()	RIGHT
Place traverse	BI-APPROACH()	DOWN
	BI-EXTRACT()	FORWARD
	BI_MOVE_AWAY	UP
Pick up fuse box	ROTATE()	RIGHT
	BI_APPROACH()	DOWN
	CLOSE_GRIPPERS	-
	BI_MOVE_AWAY()	UP
	ROTATE()	LEFT
Place fuse box	BI_APPROACH()	DOWN
	OPEN_GRIPPERS	-
	BI_MOVE_AWAY()	UP

Experiments with 5 non-expert users shown that the use of body gestures was more convenient in the case that the accuracy in robot motions was not required, such as in the example of carrying an object.

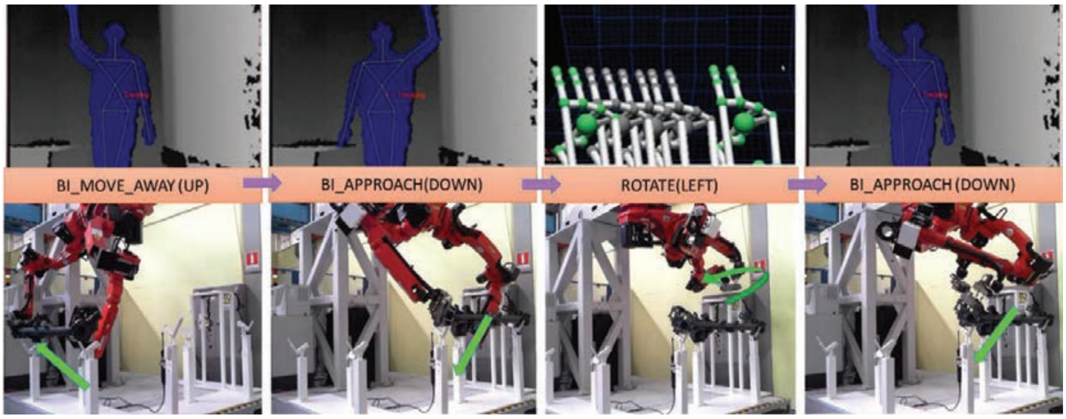

Fig. 5. High level programming through gestures in dual arm robot.

The hand gestures were more preferable to these users in the case that they should be close to the robot. In both cases, with the use of such gestures, the users mentioned that it is more convenient and less time consuming compared with the traditional ways (e.g. teach pendant).The users also described that it was easier to remember the static body gestures instead of the hand gestures, where the probability for making mistakes was increased.

5. Conclusions and future work

Robot programming requires time, cost and robot experts for an industry, making thus the need for new programming methods inevitable. The direction of new programming systems is oriented in the development of intuitive robot programming techniques. The proposed framework uses two low cost sensors exploiting many advantages for robot programming.

First of all, for a non-expert and familiar user, it is easier to understand how to move a robot using the natural ways of interaction. The definition of vocabularies can easily change and adapted to different industrial environments, depending on the user preferences. Extensibility is one more advantage of the proposed method. More methods can be easily implemented in the proposed open architecture , such as for example graphical interfaces for programming, voice commands for interaction, monitoring of human tasks, sensors for the process itself, external motion planners etc. In addition, this interaction framework can be also during the testing and execution of a robot program enabling the use of STOP gestures in case of unexpected robot performance. Last but not least, the proposed framework can be used in different robot platforms, making this method a universal approach for robot programming.

Investigation on more reliable devices in order to achieve better recognition results is one direction for future research. Additionally, investigation on the definition of more friendly and easy to remember gestures is one direction for improvements. Implementation of more interaction mechanisms and evaluation of them in terms of how easy they are to use and remember them could help also for designing more intuitive robot programming platforms.

Acknowledgments

This study was partially supported by the project X-act – Expert cooperative robots for highly skilled operations for the factory of the future and the ROBO-PARTNER – Seamless Human-Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future funded by the European Commission.

References

[1] Tsarouchi, P., Makris, S., Chryssolouris, G., 2016, Human – robot interaction review and challenges on task planning and programming, International Journal of Computer Integrated Manufacturing, pp. 1– 16, DOI:10.1080/0951192X.2015.1130251.

[2] Biggs G, Macdonald B (2003) A Survey of Robot Programming Systems. Proceedings of the Australasian Conference on Robotics and Automation, Brisbane.

[3] Aleotti, J., Skoglund, A., Duckett, T., 2004, Position teaching of a robot arm by demonstration with a wearable input device, International Conference on Intelligent Manipulation and Grasping, Genoa, July 1–2

[4] Neto, P., Norberto Pires, J., Paulo Moreira, A., 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, 37/2:137–147, DOI:10.1108/01439911011018911.

[5] Calinon, S., Billard, A., Stochastic gesture production and recognition model for a humanoid robot, in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), pp. 2769–2774.

[6] Makris, S., Tsarouchi, P., Surdilovic, D., Krüger, J., 2014, Intuitive dual arm robot programming for assembly operations, CIRP Annals - Manufacturing Technology, 63/1:13–16, DOI:10.1016/j.cirp.2014.03.017.

[7] Lee, C., Xu, Y., 1996, Online , Interactive Learning of Gestures for Human / Robot Interfaces, IEEE International Conference on robotics and automation, pp. 2982–2987, DOI:10.1109/ROBOT.1996.509165.

[8] Neto, P., Pereira, D., Pires, J. N., Moreira, A. P., 2013, Real-time and continuous hand gesture spotting: An approach based on artificial neural networks, in 2013 IEEE International Conference on Robotics and Automation, pp. 178–183.

[9] Stiefelhagen, R., Fogen, C., Gieselmann, P., Holzapfel, H., Nickel, K., Waibel, A., 2004, Natural human-robot interaction using speech, head pose and gestures, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), pp. 2422–2427.

[10] Nickel, K., Stiefelhagen, R., 2007, Visual recognition of pointing gestures for human–robot interaction, Image and Vision Computing, 25/12:1875–1884, DOI:10.1016/j.imavis.2005.12.020.

[11] Waldherr, S., Romero, R., Thrun, S., 2000, A Gesture Based Interface for Human-Robot Interaction, Autonomous Robots, 9/2:151–173, DOI:10.1023/A:1008918401478.

[12] Yang, H., Park, A., Lee, S., Member, S., 2007, Gesture Spotting and Recognition for Human – Robot Interaction, 23/2:256–270.

[13] Norberto Pires, J., 2005, Robot-by-voice: experiments on commanding an industrial robot using the human voice, Industrial Robot: An International Journal, 32/6:505–511, DOI:10.1108/01439910510629244.

[14] Qian, K., Niu, J., Yang, H., 2013, Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect, International Journal of Smart Home, 7/4:203–208.

[15] Suarez, J., Murphy, R. R., 2012, Hand gesture recognition with depth images: A review, in 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, pp. 411–417.

[16] Van den Bergh, M., Carton, D., De Nijs, R., Mitsou, N., Landsiedel, C., 2011, Real-time 3D hand gesture interaction with a robot for understanding directions from humans, in RO-MAN, pp. 357–362.

[17] Mead, R., Atrash, A., Matarić, M. J., 2013, Automated Proxemic Feature Extraction and Behavior Recognition: Applications in Human-Robot Interaction, International Journal of Social Robotics, 5/3:367–378, DOI:10.1007/s12369-013-0189-8.

[18] Morato, C., Kaipa, K. N., Zhao, B., Gupta, S. K., 2014, Toward Safe Human Robot Collaboration by Using Multiple Kinects Based Real-time Human Tracking, Journal of Computing and Information Science in Engineering, 14/1:011006, DOI:10.1115/1.4025810.

[19] Ren, Z., Meng, J., Yuan, J., 2011, Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction, IEEE International Conference on Information Communication and Signal Processing, /1:3–7, DOI:10.1109/ICICS.2011.6173545.

[20] Rautaray, S. S., Agrawal, A., 2015, Vision based hand gesture recognition for human computer interaction: a survey, Artificial Intelligence Review, 43/1:1–54, DOI:10.1007/s10462-012-9356-9.

[21] Venkata, T., Patel, S., 2015, Real-Time Robot Control Using Leap Motion, ASEE Northeast Section Conference.

[22] Narber, C., Lawson, W., Trafton, J. G., 2015, Anticipation of Touch Gestures to Improve Robot Reaction Time, Artificial Intelligence for Human-Robot Interaction ,pp. 100–102.

[23] Isleyici, Y., Aleny, G., 2015, Teaching Grasping Points Using Natural Movements, Frontiers in Artificial Intelligence and Applications, 277:275–278, DOI:10.3233/978-1-61499-578-4-275.

[24] Tsarouchi, P., Makris, S., Michalos, G., Stefos, M., Fourtakas, K., Kaltsoukalas, K., Kontovrakis, D., Chryssolouris, G., 2014, Robotized Assembly Process Using Dual Arm Robot, Procedia CIRP, 23:47–52, DOI:10.1016/j.procir.2014.10.078.