INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING, 2018
VOL. 31, NO. 2, 129–137
https://doi.org/10.1080/0951192X.2017.1369165

ARTICLE

Automation of product packaging for industrial applications

C. Perez-Vidal[a], L. Gracia[b], J. M. de Paco[c], M. Wirkus[d], J. M. Azorin[a] and J. de Gea[d]

[a]Departamento de Ingeniería de Sistemas y Automática, Miguel Hernandez University, Elche, Spain; [b]Instituto IDF, Universitat Politècnica de Valencia, Valencia, Spain; [c]Technology Department, The European Organization for Nuclear Research, CERN, Geneva, Switzerland; [d]Robotics Innovation Center, German Research Centre for Artificial Intelligence, Bremen, Germany

ABSTRACT

This work presents a robotic-based solution devised to automate the product packaging in industrial environments. Although the proposed approach is illustrated for the case of the shoe industry, it applies to many other products requiring similar packaging processes. The main advantage obtained with the automated task is that productivity could be significantly increased. The key algorithms for the developed robot system are: object detection using a computer vision system; object grasping; trajectory planning with collision avoidance; and operator interaction using a force/torque sensor. All these algorithms have been experimentally tested in the laboratory to show the effectiveness and applicability of the proposed approach.

ARTICLE HISTORY
Received 24 September 2016
Accepted 17 July 2017

KEYWORDS
Collaborative robot; object grasping; force control; collision avoidance

1. Introduction

Not many examples of robotic applications can be found in shoe manufacturing industry (Pedrocchi et al., 2015, Hinojo-Perez et al., 2016, Dura-Gil et al., 2016, Jatta et al., 2004, Dulio and Boer, 2004). Next, some research projects that use robot systems to increase the productivity in shoe manufacturing are reviewed.

The EuroShoe project develops an innovative robotic cell (Nemec and Zlajpah 2008) that is able to perform finishing operations in shoe manufacturing, such as cleaning or polishing, using a force-controlled head. The purpose of project INTELISHOE (2013) was reducing the time-to-market in Small or Medium Enterprises of traditionally handcrafted goods such as footwear. The project SShoes (2013) implements an adaptive production processes for footwear and insoles and develops robotic demonstrators and 3D design tools. Other researchers present in Vilaca and Fonseca (2007) a software application for optimising shoe sole halogenation and lead roughing processes. The project CecMadeShoe (2008) developed advanced tools for the customisation process (magic mirror), whereas project FIT4U (2009) aims at responding to the growing demand for consumer-oriented product customisation, especially for sport footwear. The project RoboFoot (2012) developed manipulation for non-rigid objects, programming tools and sensor-based control approaches to overcome the complexity of automating the shoe manufacturing processes. Finally, the work described by Montiel (2007) uses object-oriented CAD systems for designing heels and insoles.

Motion planning is essential to use a robot system for practical applications. It consists of finding a path from the current configuration to the goal configuration satisfying constraints such as joint limits and collision avoidance with the obstacles in the environment (Mei and Lee 2016). An important issue when using robots in *unstructured environments* is that the robots system has to react to dynamic changes, i.e. the planned trajectories have to be modified online using data from sensors. For instance, this situation arises in human–robot interaction applications (Tsarouchi, Makris, and Chryssolouris 2016) (Tsarouchi et al. Forthcoming), where collisions between the robot and the human operator have to be avoided (Mohammed, Schmidt, and Wang Forthcoming). For this purpose, several types of *sensors* are used: vision systems, force/torque sensors, etc. For instance, a vision system can be used to visually detect the obstacles in the environment and to guide the robot to the goal configuration using, for example, visual servoing techniques (Vahrenkamp et al. 2008). In particular, in this work a depth camera (Microsof Kinect camera, KinectSpecs 2016) is used to generate a point cloud of the robot workspace in order to compute a trajectory that avoids the detected obstacles.

The structure of the paper is as follows. Next section introduces the industrial application to be solved using a robotic system, while Section 3 presents the hardware and software architecture proposed in this work to cope with it. Next, Section 4 develops the robot control system to properly perform the industrial task at hand. The proposed approach is applied in Section 5 to two actual robotic platforms to show its feasibility and effectiveness. Finally, some concluding remarks are given.

2. Industrial application

The approach proposed in this work for product packaging can be used for many types of industrial applications. However, the work is focused on a specific application, the shoe packaging, in order to illustrate the main characteristics of the method.

The footwear industry accounts for some of the shortest chain production runs to be found, e.g. eight pairs of shoes is the average order size in small- and medium-sized enterprises.

CONTACT C. Perez-Vidal carlos.perez@umh.es

Consequently, automation is nowadays more required to ensure competitiveness in this growing market. The introduction of intelligent robotic technologies can contribute to overcome the complexity in the automation of the associated production processes. The main difficulties to be faced are listed below:

(1) The high number of product versions due to the development of more than 200 different models for each season with different sizes, leather quality and colours per model.
(2) Complex manufacturing process: for each model it is necessary to develop and manufacture the last (the mould used to make a shoe); to produce the list of components (sole, heel, sock, strap, inner parts, etc.); to cut the inner and outside parts; and to stitch inner and outside parts to assemble them over the last.
(3) The assembly process is very laborious (around 25 different operations) and especially complex in fitting operations due to the non-uniformity and the different elasticity of the natural leather as well as the non-solid nature of the components, making even more complicated the use of a robotic manipulator as can be seen in Bonert, Shu, and Benhabib (2000).
(4) Extensive demand of specialised staff for quality control and packaging operations: each pair of shoes requires cleaning, final inspection, introduction in the shoe box and stacking up for shipment.

Figure 1 graphically represents the tasks involved on item 4. This phase is one of the operations with higher workforce impact. The 'Introduce shoes into box' block shown in Figure 1 is a part of the 'Packaging Process' that can be found in the 'Finishing phase'. The procedure is as follows. Workers take the shoes, visually inspect them and, if everything is fine and no defect is identified, proceeds with the packaging process, i.e. introducing each shoe into a plastic bag, and finally introduces the pair of shoes into the box. The process usually takes around 20–25 s to a human operator.

3. System architecture

3.1. Hardware

This section describes the hardware components used in the robot platform and its purpose. Figure 2 shows the main system components and the communication interfaces used between them.

3.1.1. Arm

Two robot platforms are used in this work to test the algorithms. Both are based on Schunk modules, have seven joint modules of four different sizes (PRL120, PRL100, PRL080 and PRL060), with peak out torques ranging from 10 Nm to 372 Nm. The robots have a Campus Area Network (CAN) bus line which links the modules with the central computer.

3.1.2. Robot hands

Four different grasping devices have been used for the tests in this project. The iCub hand from Italian Institute of Technology (used at the DFKI Lab), the IH2 Azzurra Hand from Prensilia (used at the UMH Lab), a Festo industrial gripper (HGP-25-A-B Festo parallel gripper) and a Schunk industrial gripper (servo electric 2-finger-parallel gripper type PG 70).

3.1.3. 3D sensor

A Kinect camera (KinectSpecs 2016) is used due to its high performance and low cost. This device comprises a RGB camera with 1280 × 960 resolution, an infrared (IR) emitter and an IR depth sensor. The minimum measuring distance for this 3D camera is around one metre.

3.1.4. Control PC

The computer used for the tests has the following characteristics: Intel Core i7 2600k, 4 cores, 3.40Ghz, 6Mb cache, Turbo Boost 2.0, RAM 16Gb, NVIDIA GeForce210 8Gb, HDD ST1000DL002-9TT153 ATA of 2Tb.

3.2. Software

A Robot Operating System (ROS) platform over Ubuntu has been used for the experiments developed in this work. ROS provides libraries and tools to help software developers to create robot applications as well as the services of an operating system: hardware abstraction, low-level device control, etc. In order to communicate the CAN modules (Schunk modules and robot hands) via a PCAN-USB interface, the ROS package *libpcan*, LibPCan (2016), has been used. Other ROS packages that have been used are *schunk_powercube_chain*, RosPCANdrivers (2016) and *schunk_hardware_config*. The robot description is set up using URDF (Unified Robot

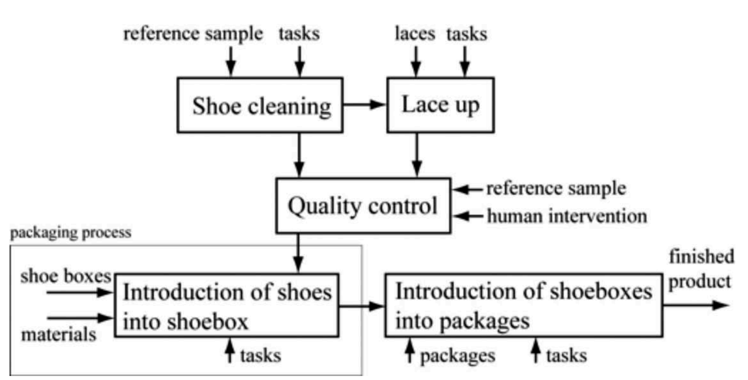

Figure 1. Description of the finishing phase of a shoe manufacturing process.

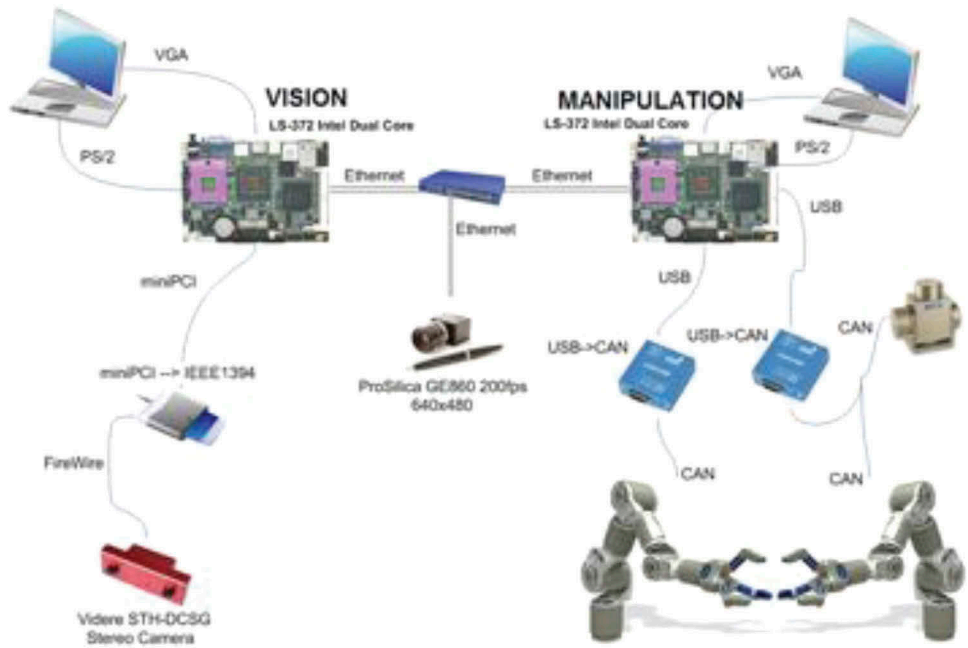

Figure 2. Hardware architecture.

Description Format, URDF 2016), which is an XML format for representing a robot model.

4. Robot control

4.1. Motion planning

As mentioned in the Introduction section, motion planning consists of finding a path from the current pose (position and orientation) to a goal pose satisfying constraints such as joint limits and collision avoidance with the obstacles in the environment. For this purpose, sensing is needed to avoid collision when operating in unstructured environments. For instance, using the Kinect camera mentioned in previous section, a point cloud is obtained, which is a set of 3D points representing the surface of the detected objects. Subsequently, the point cloud is processed (segmentation, filtering, model fitting, surface reconstruction, etc.) using the Point Cloud Library (PCL 2016). Moreover, the ROS *arm_navigation* stack (ArmNav 2016) allows to generate robot manipulation applications using a set of stacks. For instance, the *collision_environment* stack contains tools to create representations of the environment for collision checking (Hornung et al. 2013, see Figure 3).

Furthermore, the constraint-aware Inverse Kinematics solver (*arm_navigation* stack, Arm Navigation ROS stack, Website 2016) provides the constraint-aware kinematics for any serial robot combining the ROS collision tool with Orocos KDL (2012) forward and inverse kinematics solvers.

A repository of motion planning algorithms is available in the OMPL (Open Motion Planning Library) (Sucan, Moll, and Kavraki 2012): PRM, RRT, ESTS, SBL, KPIECE, BKPIECE, LBKPIECE, LazyRRT, RRTConnect, etc. Hence, it can be selected as the most suitable planner for the task at hand and the values for its parameters can be chosen. However, these planners may generate non-smooth trajectories and, hence, smoothing is needed before sending it to the controller.

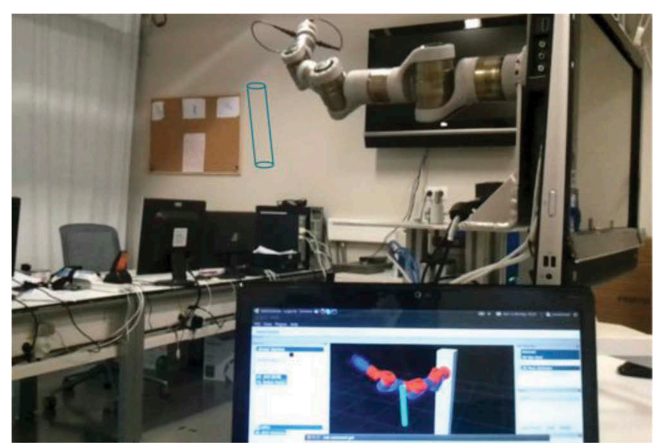

Figure 3. Avoidance of real or virtual objects.

4.2. Force control

Either for objects in the environment moving very fast or objects occluded from the vision system field of view, collisions between them and the robot cannot always be avoided. Therefore, for security reasons, a collision detection algorithm is required to protect the human operator, the robot, etc. For instance, when using the robot system described in Section 3.1, this can be achieved using the information provided by the force/torque sensor located at the robot's end-effector. Thus, if the collision algorithm detects a collision, the robot stops its current task and switches to a force-control mode. At this point, the human operator can move the robot by pulling at the end-effector, so that the operator can relieve the system from the collision situation or to inspect the carried item.

4.3. Object detection

The stack RoboEarth (Waibel et al. 2011) of ROS is used for object detection. This tool allows to build up an object model using the data from the Kinect camera. Furthermore, it allows to find objects in the scene which model has been previously stored. In the modelling phase, the object is placed onto two sheets of paper, where Augmented Reality Markers are printed on. The Markers are tracked to reconstruct the camera pose relative to the markers. When the camera is moved around the object, or vice versa, the camera pose is computed and the Speeded-Up Robust Features (SURF) (Bay, Tuytelaars, and Van Gool 2006) features are extracted using OpenCV implementation. After that, the 3D position is determined using the depth information from the Kinect sensor. The local point cloud of the object and the SURF features are stored for several frames. These are the reference poses that are used for the object detection. Thus, the SURF features are extracted from the camera image and matched sequentially against the stored feature descriptor for each reference pose of the learned objects. The computational cost of this process can be rather large depending on the number of objects and reference poses. In general, 3D object detection requires more reference poses to obtain a robust detection.

Furthermore, geometric model-driven approaches are used in this work to fit a 3D shape model of the object (shoe, box, etc.) into the scene and to track it (Teuliere, Marchand, and Eck 2010). These methods reliably compute the object pose information assuming that there is *a priori* knowledge of the location where the model fitting has started.

4.4. Grasping

This subsection outlines an algorithm to plan grasps using the object representation. The algorithm is based on the grasp hypothesis generation implemented in the Simox toolbox (Simox 2016). The algorithm presented there was extended to cope with restricted touch regions on objects. The inputs for the algorithm are a kinematic model of the robotic hand, a 3D mesh of the object to grasp and a grasp definition. The grasp definition contains a pre-shape of the hand and an approach vector. On the object, a random approaching point x is sampled on which the vector n_x normal to the surface is approximated. Then, the pre-shape is aligned to this normal vector using the pre-shape approach vector. The position in the space is selected in a way that with the hand opened (in pre-shape) there is no initial collision with the object. To do so, collision checking is performed starting with a position close to the hand and moving the hand along the approach vector in negative direction. When there is no initial collision for a position, the hand is closed with a constant speed until all fingers are in contact. If there are more than two contacts, it is checked whether all contacts are within the valid grasp regions. If so, grasp quality is calculated based on the number of contact points and it is checked whether the force closure property of the grasp is fulfilled.

Figure 4 shows a picture of the grasp planning with shaded regions. The red areas on the object indicate regions in which the robot is not allowed to touch the object (process manually

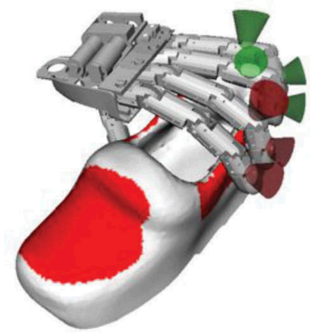

Figure 4. Grasp planning with restricted touch regions.

performed). The cones in the fingers indicate contact points that are valid or invalid contacts. In the case of the displayed grasp, the grasp planning algorithm would reject it.

5. Experimental results

This section presents two experiments on two different platforms to show the effectiveness of the trajectory planning algorithm to avoid the obstacles and the motion algorithm using force control.

5.1. Trajectory planning and collision avoidance

The trajectory generated by the motion planning algorithm (see Section 4.1) should avoid the collisions between the robot (or the object that is carrying) and the objects in the robot environment. In this experiment, the objects have been detected with a Kinect camera and the corresponding point cloud has been obtained. Figure 5 shows a sequence in which the robot is carrying an aluminium stick from one side to the other of a rigid structure, which is marked with black and yellow bands. The motion planning takes into account the starting and goal pose for the carried object, its shape and the environment in order to properly avoid collisions.

5.2. Robot motion using force control

Figure 6 shows an experiment, which has been developed to show the functionality of the collision detection and force control described in Section 3.2. The manipulator is moving the grasped shoe from one box to another located on a different table. At some point, the operator pushes the robot end-effector, the robot system detects the 'collision' using the information provided by the force sensor and the robot control switches to force control mode. At this point, the human operator can freely move the robot end-effector. Once no more external forces are applied, the robot goes back to follow the previously planned trajectory. However, if large deviations from this trajectory have occurred, the task could be considered as failed and a re-planning could be performed, e.g. using the vision system, in order to obtain a new trajectory.

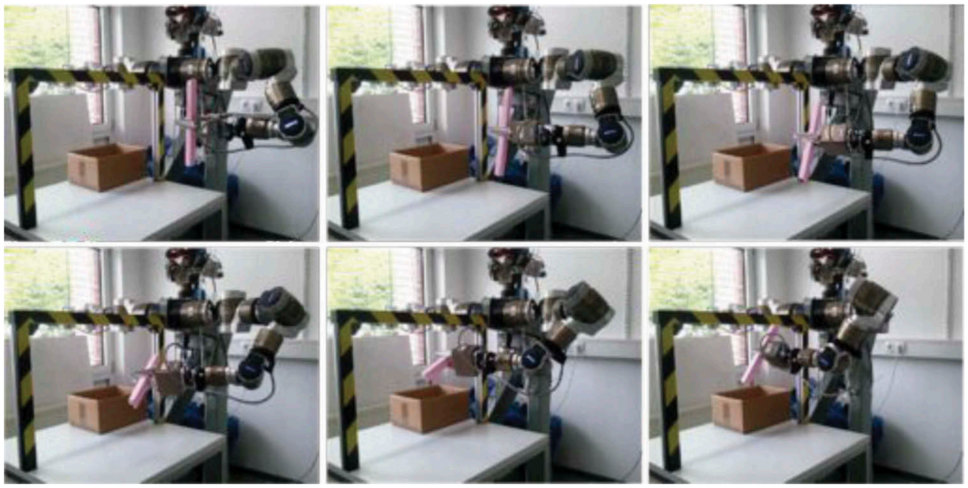

Figure 5. Automatic trajectory re-planning to avoid obstacle collision.

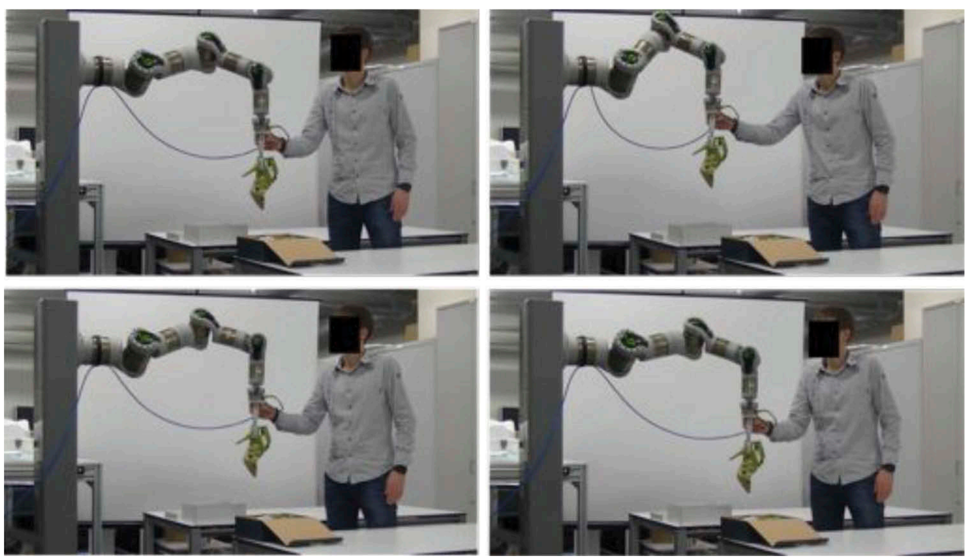

Figure 6. Experiment on force detection and force control: the operator is moving the robot.

5.3. Object recognition and grasping

The steps for grasping an object are as follows: first, the object has to be successfully detected; second, a suitable grasping position has to be computed; and, finally, the object is grasped. According to Moulianitis, Dentsoras, and Aspragathos (1999), there are three main techniques for gripping non-rigid objects:

- Mechanical surface, in which the material is clamped or pinched between gripper fingers to give high frictional holding.
- Intrusive, in which pins are fed into the surface or body of the material.
- Surface attraction, which includes the use of adhesives or vacuum.

The surface where the object is over is considered an obstacle for the planning algorithm. Given a certain desired grasp area for the shoe, the motion planner is requested to find an optimal trajectory which brings the robotic arm from its current position to a fixed distance relative to the shoe and with the approaching orientation vector according to the detected pose of the shoe. The final shoe grasping from here will be performed 'blindly' and the robot will move along the approaching vector towards the shoe and close its fingers. Tactile information on the tips and palms of the hand will provide the necessary contact information to complete the grasp motion (in case that the robotic hand has this capability). In this case, compared to parallel grippers, the use of a multi-fingered hand provides a higher level of robustness on the closing grasp, as the robotic hand will enclose the shoe within its fingers and the grasp point location.

Figure 7 shows a scene where a shoe and a plastic glass can be seen (left) and a milk package over the two sheets of paper (right) described in Section 4.3.

This object detection based on RoboEarth is used to locate the object that is going to be manipulated. Figure 8 shows an experiment where a milk package is detected and is going to be picked up. In this figure, real image and simulation image are blend to check that both of them matched during the task.

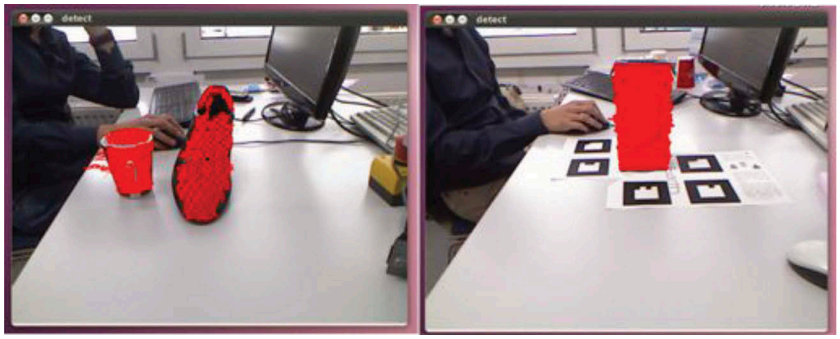

Figure 7. Detection of a registered object. At left, a shoe and a plastic glass have been located and at right, a milk package can be easily recognised.

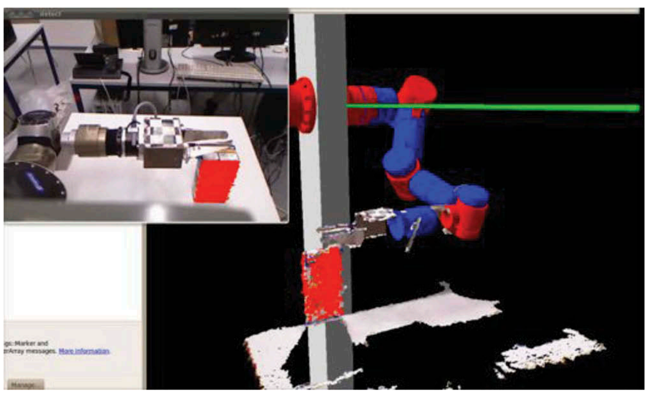

Figure 8. Combination of simulated and real experiment where a milk package is detected and manipulated.

To check the RoboEarth performance, an experiment inspired on Rigual et al. (2012) has been implemented. A group of four shoes are introduced in the system database and then they are recognised in the scene shown in Figure 7. To perform this test, different shoe models with different features have been chosen. Results are shown in Table 1. For shoe No.1, detection at 60 cm is around 90% but it decreases for larger distances. In case of shoe No. 2, detection rates drop due to the decorative holes of the object (toe hole and lateral holes).

RoboEarth detection does not work properly with glossy surfaces. Shoe No. 3 is a patent leather model and light reflections hamper recognition, but the worst case is for shoe No. 4. This is more a wire structure than an object and, for this reason, it could not be recognised using RoboEarth, not to mention about robotic grasping.

The shoe box (like a milk package) can be easily located due to its regular shape. In this sense, accuracy is not relevant to introduce a pair of shoes inside a shoe box because this task has a tolerance of several centimetres. In addition, accuracy depends on robot speed as well. Robot speed generates undesired inertia effect on carried object and in many cases the robotic hand has to be closed tighter to avoid drops or slippage. To generate a larger force with a robotic hand is not always possible due to object limitations or limitations of the robotic hand itself. In this work, slow movements have been performed and, by the moment, no productivity information has been obtained. Anyway, speeding up the robot movements remain as further work.

About grasping reliability, last row of Table 1 shows the number of times that the system has successfully grasped the shoe (out of ten). For the cases No. 1 and No. 3, around 90% of grasps were successful. For shoe No. 2 (more flexible upper), this amount dropped to 80%, but the worst case, obtaining poor results, was for shoe No. 4. In this case, it was almost impossible to pick the shoe up. Another important issue is weight. It establishes a speed limit for the robot movements. As heavier the shoe is as slowly the robot should move.

Regarding grasping other objects, Table 2 shows the number of times that the system has successfully grasped a milk package (out of ten). For all cases, between 9 and 10 times (out of ten) the package has been successfully grasped. Object detection is much more stable than in the case of shoes as well. For widen distances, right detections are between 90% and 100%.

Figure 9 shows the process of locating the shoe, performing motion planning, reaching and, finally, grasping it. Using the previously defined detection methods, the Kinect camera is used to detect the shoe and identify its pose.

It is interesting to remark that, in this work, initial grasping tests were performed using industrial parallel grippers at both research laboratories (UHM and DFKI) (see Figures 5 and 6). However, since certain shoe areas have to be avoided (e.g. not

Table 1. Detection results for shoes: number of correct detections for each distance (out of ten).

Shoe type	No. 1 			No. 2 			No. 3 			No. 4 		
Distance	0.6 m	1 m	2 m	0.6 m	1 m	2 m	0.6 m	1 m	2 m	0.6 m	1 m	2 m
Detections (out of ten)	9	8	6	7	6	4	5	3	1	1	0	0
Successfully grasped (out of ten)		9			8			9			0	

Table 2. Detection results for milk packages: number of correct detections for each distance (out of ten).

	No. 1 			No. 2 			No. 3 		
Object type									
Distance	0.6 m	1 m	2 m	0.6 m	1 m	2 m	0.6 m	1 m	2 m
Detections (out of ten)	10	10	9	10	9	9	10	9	9
Successfully grasped (out of 10)		10			9			10	

touched, see Figure 4), the use of a multi-finger hand provides a higher search space on which to find a suitable hand configuration that robustly grasps the shoe and, at the same time, avoids touching certain shoe areas. For this reason, the iCub hand (Davis, Tsagarakis, and Caldwell 2008) was used for the advanced grasping tests at one research laboratory and the IH2 Azzurra Hand (IH2Azzurra 2012) at the other.

6. Workspace layout for robot application

The robot system developed in this work for product packing has been tested in previous section in laboratory conditions. To transfer this system to an industrial plant, all the elements in the robot workspace have to be properly designed and located. All together form a Flexible Manufacturing Cell (FMC). Figure 10 shows an example of FMC to integrate a robot in a production line. Note that, an industrial robot has been depicted (instead of the self-developed robot used in the laboratory tests) together with safety barriers in order to meet safety regulations.

FMCs have been widely implemented in modern factories (Tubaileh, Hammad, and Kafafi 2007). For an efficient use of handled material, the FMC must take into account the special features of the process. When a FMC is introduced in the factory layout, its location is determined following a reach-ability and mobility criteria. The optimal FMC layout is obtained by minimising the cycle time of the robot joints required to perform a sequence of travel, as it can be seen by Fenton, Poon, and Davies (1992) and Tubaileh, Hammad, and Kafafi (2007).

Minimising the cycle time of the FMC increases production rate. Many works have been focused on optimising travel time. Fenton, Poon, and Davies (1992), Dissanayake and Gal (1994) and Mata and Tubaileh (1998) presented research works to get minimum travel time and minimum joint displacement between two positions. Once the FMC position is chosen at the end of the production line, FMC integration is required. In this sense, the following items need to be considered:

- **Installation**: The robot and cameras have to be affixed and its relative coordinates respect to conveyor belts have to be measured. After this task, the robot is properly located on its environment. In this step, emergency stops, speed limits, over-torque detectors, safety buttons (including dead

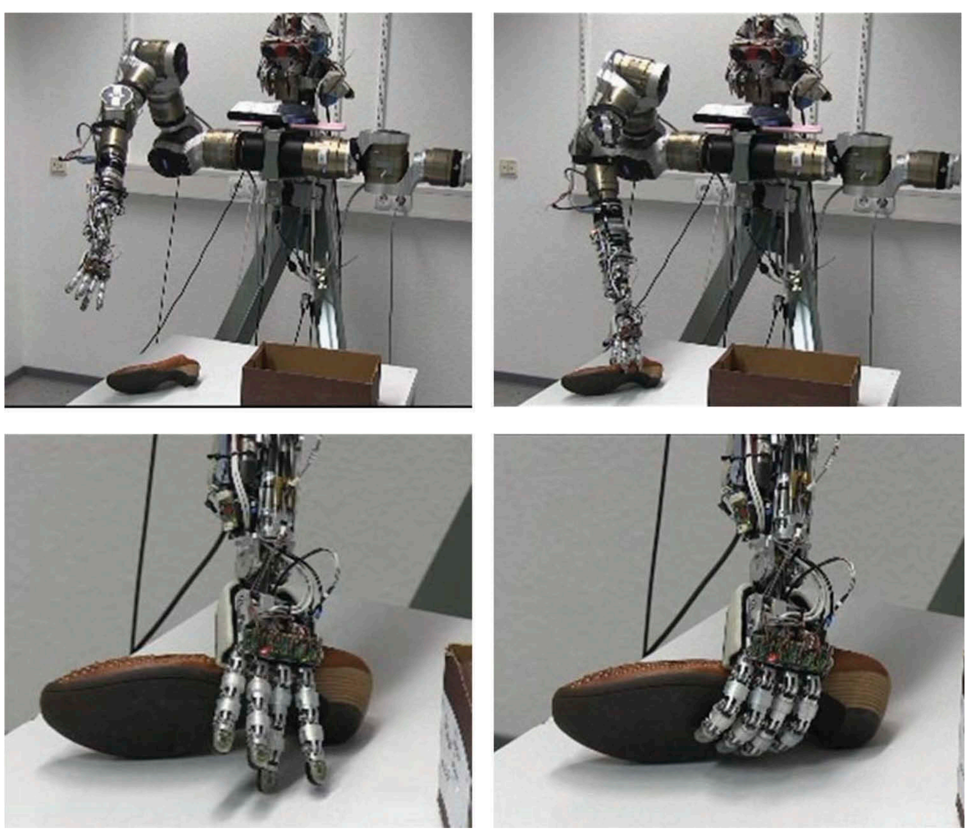

Figure 9. Complete shoe grasping process including shoe detection, motion planning, reaching and grasping.

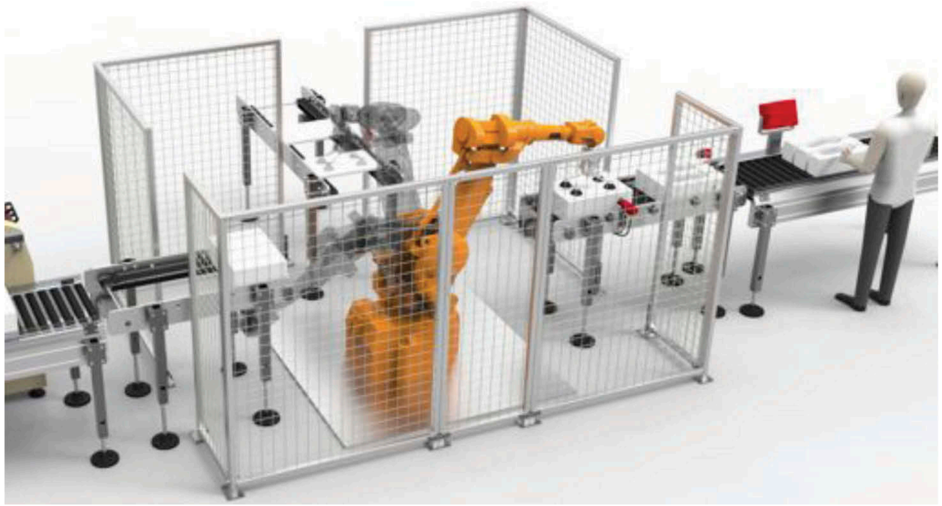

Figure 10. Example of robot integration in a production line.

man's switch if needed), passwords, fencing barriers and auto-diagnosis have to be included in the FMC.

- **Synchronisation**: Each event is triggered by a synchronisation signal. Synchronisation signals in this FMC come mainly from two sensors: the Kinect camera and the force/torque sensor. In addition, factories with one or more robotic devices are usually provided of an industrial network in which FMCs have to be connected. It allows to send and receive detailed parameters of the production.
- **User interface**: The user interface is the device that allows a factory operator to interact with the FMC. This interface must be adapted to the industrial sector, the specific application and even to the worker knowledge and terminology. This task includes operator training.
- **Safety regulations**: Although safety measures have been taken into account during the installation process, a special care about regulations must be considered. In this work, ISO EN 953 (2009), ISO 10218-1 (2011) and ISO 15066 (2016) have to be accomplished.
- **Optimisation**: When the device is installed in the factory and the above items are done, the robot still moves slowly. The most difficult task of the robot installation is to speed it up, making the FMC as productive as possible. It involves many times the substitution of slower hardware (commonly called bottlenecks) by a faster one. This is an iterative task that takes a long time and it can be included as a continuous improvement task. In this work, this is the toughest task of them all.

7. Conclusion

In this work, a robotic system to automate the product packaging in industrial applications has been developed. In particular, the proposed system has been specially conceived for the shoe manufacturing industry. However, other products with similar packaging processes may benefit with the obtained solution. An important part of this work has been selecting and testing the main algorithms for the robotic system: object detection, object grasping, trajectory planning, etc. In this sense, the experimental results obtained for all these algorithms have shown a good success and allow us to believe that the proposed approach could be used in industrial applications in the near future.

Acknowledgements

This work has been partly supported by Ministerio de Economia y Competitividad of the Spanish Government (Grant No. RTC201654086 and PRI-AIBDE-2011-1219), by the Deutscher Akademischer Austauschdienst (DAAD) of the German Government (Projekt-ID 54368155) and by ROBOFOOT project (Grant No. 260159) of the European Commission.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work has been partly supported by Ministerio de Economia y Competitividad of the Spanish Government [Grant No. RTC201654086 and PRI-AIBDE-2011-1219], by the Deutscher Akademischer Austauschdienst (DAAD) of the German Government (Projekt-ID 54368155) and by ROBOFOOT project [Grant No. 260159] of the European Commission.

References

Arm Navigation ROS stack, Website. 2016. Accessed September 16 2016. http://wiki.ros.org/arm_navigation

Bay, H., T. Tuytelaars, and L. Van Gool. 2006. "SURF: Speeded Up Robust Features." In *Computer Vision ECCV 2006. ECCV 2006. Lecture Notes in Computer Science*, edited by A. Leonardis, H. Bischof, and A. Pinz. 3951 vols. Berlin, Heidelberg: Springer.

Bonert, M., L. H. Shu, and B. Benhabib. 2000. "Motion Planning for Multi-Robot Assembly Systems." *International Journal of Computer Integrated Manufacturing* 13 (4): 301–310. doi:10.1080/095119200407660.

Custom, Environment and Comfort Made Shoe. 2008. *Project Funded by 6th FP [2004-2008]*.

Davis, S., N. G. Tsagarakis, and D. G. Caldwell. 2008. "The Initial Design and Manufacturing Process of a Low Cost Hand for the Robot iCub." In *8th IEEE-RAS International Conference on Humanoid Robots*, 40–45. Daejeon, February 20 2009.

Dissanayake, M., and J. A. Gal. 1994. "Workstation Planning for Redundant Manipulators." *International Journal of Production Research* 32 (5): 1105–1118. doi:10.1080/00207549408956990.

Dulio, S., and C. Boer. 2004. "Integrated Production Plant (IPP): An Innovative Laboratory for Research Projects in the Footwear Field." *International Journal of Computer Integrated Manufacturing* 17 (7): 601–611. doi:10.1080/0951192042000273168.

Dura-Gil, J. V., A. Ballester-Fernández, M. Cavallaro, A. Chiodi, A. Ballarino, C. Brondi, V. von Arnim, and D. Stellmach. 2016. "New Technologies for Customizing Products for People with Special Necessities: Project FASHION-ABLE." *International Journal of Computer Integrated Manufacturing*. doi:10.1080/0951192X.2016.1145803.

EN 953:2009, Safety of machinery - Guards - General requirements for the design and construction of fixed and movable guards. 2009. *German Version EN 953:1997+A1:2009 [ISO-2011] DIN EN*.

Fenton, R. G., D. Poon, and S. P. Davies. 1992. "Robotic Workcell Cycle Time Optimization Using Computer Graphics." *Annals of the CIRP* 41: 509–512. doi:10.1016/S0007-8506(07)61256-6.

Framework of Integrated Technologies for User Centred Products. 2009. *Grant Agreement No.: CP-TP 229336-2. NMP2-SE-2009-229336 FIT4U –7th FP*.

Hinojo-Perez, J. J., M. Davia-Aracil, A. Jimeno-Morenilla, L. Sanchez-Romero, and F. Salas. 2016. "Automation of the Shoe Last Grading Process according to International Sizing Systems." *The International Journal of Advanced Manufacturing Technology* 85 (July): 455–467. doi:10.1007/s00170-015-7947-8.

Hornung, A., K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. 2013. "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees." *Autonomous Robots* 34: 189–206. doi:10.1007/s10514-012-9321-0.

IH2 Azzurra Hand. 2012. Accessed September 16 2016. http://www.prensilia.com/index.php?q=en/node/40

INTELISHOE - Integration and Linking of Shoe and Auxiliary Industries. 2013. *Project Funded by 5th FP*.

ISO 10218-1:2011, Robots and Robotic Devices - Safety Requirements for Industrial Robots - Part 1: Robots (ISO 10218-1:2011). 2011. *German Version EN ISO 10218-1:2011*.

ISO/TS 15066:2016 - Robots and Robotic Devices – Collaborative robots. 2016.

Jatta, F., L. Zanoni, I. Fassi, and S. Negri. 2004. "A Roughing/Cementing Robotic Cell for Custom Made Shoe Manufacture." *International Journal of Computer Integrated Manufacturing* 17: 645–652. doi:10.1080/095112042000273212.

Kinect for Windows Sensor Components and Specifications, Website. 2016. Accessed September 16 2016. http://msdn.microsoft.com/en-us/library/jj131033.aspx

Mata, V., and A. Tubaileh. 1998. "The Machine Layout Problem in Robot Cells." *International Journal of Production Research* 36 (5): 1273–1292. doi:10.1080/002075498193318.

Mei, K. J., and R. S. Lee. 2016. "Collision Detection for Virtual Machine Tools and Virtual Robot Arms Using the Shared Triangles Extended Octrees Method." *International Journal of Computer Integrated Manufacturing* 29 (4): 355–373. doi:10.1080/0951192X.2015.1033755.

Mohammed, A., B. Schmidt, and L. Wang. Forthcoming. "Active Collision Avoidance for Human? Robot Collaboration Driven by Vision Sensors." *International Journal of Computer Integrated Manufacturing* 1–11. doi:10.1080/0951192X.2016.1268269.

Montiel, E. 2007. "Customization in the Footwear Industry." In *Proceedings of the MIT Congress on Mass Customization*.

Moulianitis, V. C., A. J. Dentsoras, and N. A. Aspragathos. 1999. A Knowledge-Based System for the Conceptual Design of Grippers for Handling Fabrics. *Artificial Intelligent Engineering Design Analysis Manufacturing* 13. January, ISSN: 0890-0604. doi:10.1017/S0890060499131020.

Nemec, B., and L. Zlajpah. 2008. "Robotic Cell for Custom Finishing Operations." *International Journal Computation Integrative Manufacturing* 21: 33–42. doi:10.1080/09511920600667341. January.

Orocos KDL Documentation, Website. 2012. Accessed September 16 2016. http://www.orocos.org/kdl

Package libpcan documentation, Website. 2016. Accessed September 16 2016. http://www.ros.org/wiki/libpcan

PCAN Driver for Linux, User Manual. 2016. Accessed September 16 2016. http://wiki.ros.org/schunk_powercube_chain

Pedrocchi, N., E. Villagrossi, C. Cenati, and L. M. Tosatti. 2015. "Design of Fuzzy Logic Controller of Industrial Robot for Roughing the Uppers of Fashion Shoes." *The International Journal of Advanced Manufacturing Technology* 77 (March): 939–953. doi:10.1007/s00170-014-6501-4.

Point Cloud Library (PCL), Website. 2016. Accessed September 16 2016. http://www.pointclouds.org/

Rigual, F., A. Ramisa, G. Aleny, and C. Torras. 2012. "Object Detection Methods for Robot Grasping: Experimental Assessment and Tuning." In *15th Catalan Conference on Artificial Intelligence, 2012, Alacant, in Artificial Intelligence Research and Development, Vol 248 of Frontiers in Artificial Intelligence and Applications*, 123–132. IOS Press.

RoboFoot Project Website. 2012. Accessed September 16 2016. http://www.robofoot.eu/

ROS Robot Model (URDF) documentation Website. 2016. Accessed September 16 2016. http://www.ros.org/wiki/urdf

Simox toolbox. 2016. Accessed September 16 2016. http://simox.sourceforge.net/

Special shoes movement. 2013. Project funded by 7th FP, NMP-2008-SME-2-R.229261. http://www.sshoes.eu

Sucan, I. A., M. Moll, and L. E. Kavraki. 2012. "The Open Motion Planning Library." *IEEE Robotics & Automation Magazine* 19 (December): 72–82. doi:10.1109/MRA.2012.2205651.

Teuliere, C., E. Marchand, and L. Eck. 2010. Using Multiple Hypothesis in Model-Based Tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 4559–4565. Anchorage, AK, May 3–7. : 1050-4729

Tsarouchi, P., S. Makris, and G. Chryssolouris. 2016. "Human? Robot Interaction Review and Challenges on Task Planning and Programming." *International Journal of Computer Integrated Manufacturing* 29 (8): 916–931. doi:10.1080/0951192X.2015.1130251.

Tsarouchi, P., A. S. Matthaiakis, S. Makris, and G. Chryssolouris. Forthcoming. "On a Human-Robot Collaboration in an Assembly Cell." *International Journal of Computer Integrated Manufacturing* 1–10. doi:10.1080/0951192X.2016.1187297.

Tubaileh, A., I. Hammad, and L. A. Kafafi. 2007. "Robot Cell Planning." *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*. 1,(2): 93–96.

Vahrenkamp, N., S. Wieland, P. Azad, D. Gonzalez, T. Asfour, and R. Dillmann. 2008. "Visual Servoing for Humanoid Grasping and Manipulation Tasks." In *8th IEEE-RAS International Conference on Humanoid Robots*, 406–412. Daejeon, December 1–3.

Vilaca, J. L., and J. Fonseca. 2007. "A New Software Application for Footwear Industry." In *IEEE International Symposium onIntelligent Signal Processing, 2007. WISP 2007*. Alcala de Henares, October 3–5.

Waibel, M., M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. M. M. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft. 2011. "Distributed Databases, Information Exchange, Knowledge Transfer, Robot Kinematics, Robot Sensing Systems, Semantics, Service Robots." *IEEE Robotics Automation Magazine* 18: 69–82. doi:10.1109/MRA.2011.941632.