Seamless and safe human - centred robotic applications for novel collaborative workplaces – SHERLOCK

Grant Agreement No	:	**820689**
Project Acronym	:	**SHERLOCK**
Project Start Date	:	**1st October, 2018**
Consortium	:	**UNIVERSITY OF PATRAS (LMS)** **COMAU SPA (COMAU)** **VDL INDUSTRIAL MODULES (VDLIM)** **UNITED TECHNOLOGIES RESEARCH CENTRE IRELAND (LIMITED IRC)** **SOFITEC AERO SL (SOFITEC)** **FIDIA SPA (FIDIA)** **PILZ INDUSTRIEELEKTRONIK SL (PILZ)** **DEBBACHE-LAGIOS EE – GIZELIS ROBOTICS (GROBO)** **INTRASOFT INTERNATIONAL SA (INTRA)** **LIGHT & SHADOWS (LS)** **FUNDACION TECNALIA RESEARCH & INNOVATION (TECNALIA)** **FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V (FRAUNHOFER)** **CRANFIELD UNIVERSITY (CU)** **UNIVERSITY COLLEGE DUBLIN, NATIONAL UNIVERSITY OF IRELAND, DUBLIN (UCD)** **MCI BENELUX SA (MCI)**

SHERLOCK
UNLOCKING POTENTIALS

Title	:	Process perception modules – First prototype
Reference	:	D4.1
Availability	:	Public
Date	:	31/03/2020
Author/s	:	TECNALIA
Circulation	:	EU

Summary:

This document describes the outcome of T4.1 “Process perception for HRC applications” for the first period of the project. Specifically, it describes the design and implementation of the first prototype of the Process Perception Module (PPM) and shows its application at two use cases of SHERLOCK.

Table of Contents

1. LIST OF FIGURES

2. LIST OF TABLES

3. EXECUTIVE SUMMARY

This deliverable reports the development status of the initial prototype of SHERLOCK that will enhance the cognitive capabilities of robot, providing them perception of the process being executed as the outcome of T4.1 “Process perception for HRC applications”.

The outcome of this task is wrapped around the so call Process Perception Module (PPM).

The developed prototypes are applied to the OTIS and SOFITEC use cases of SHERLOCK. This document reports the design and the steps towards the implementation of the first prototypes.

The document is structured as follows: Section 4 provides an introduction to the aforementioned modules. Section 5 serves a brief description of the use case scenarios and the PPM application specifications for each one of the use cases. Section 6 describes the approach followed towards the development of PPM prototype for the OTIS and SOFITEC cases, while Section 7 provides details about the implementation of the approach. Finally, Section 8 concludes the work done and presents insights for the near future, towards the development of the final prototype of the module.

4. INTRODUCTION

Although the latest robotics introduced in the market during the last couple of years have been enhanced over the solutions existing before, they still lack advanced perception capabilities that will provide them cognition of the status of the shopfloor. The lack of perception capabilities is more intense in human robot collaborative scenarios, where the robotic solution, except from being able to know the position of the various resources and humans in the shopfloor, needs to know the tasks being executed by the operators and thus coordinate the production.

SHERLOCK aims to solve this issue by utilizing state of the art AI based tools to detect the actions of the operators inside the workplace and share them with the robots. These innovative tools will also be utilized to detect the parts that need to be manipulated by the robots and localize their position.

These functionalities will be merged under a module called “Process Perception Module” (PPM).

Computer vision enables the segmentation of the scene, identification and pose estimation of such objects on a dynamic environment. These perception pipelines are designed for 3D vision sensors. With this, a dataset of point clouds of the parts to be recognized is created and serves as the ground truth for detection.

In addition to recognition of parts to be manipulated by the robots as well as tasks being executed by humans, the technology developed can be utilized for easy online learning, so that operators can effortlessly and seamlessly introduce new parts into the system.

The aforementioned work is part of WP4 and specifically T4.1 “Process perception for HRC applications”.

5. DESCRIPTION OF THE PERCEPTION USE-CASES

This chapter provides a short description of the problem to be addressed and solved in each use-case. It is intended to give a better comprehension of the following chapters. A detailed description of all the use-cases, their current operations and the future objectives and contributions to be introduced by SHERLOCK to them can be seen in deliverable D1.1.

5.1. OTIS

The pilot work centre in OTIS is part of the mid-range door line Techna/D2200 and more specifically focuses on the workstation where the cab door panel hangers are pre-assembled prior to be assembled on the cab door lintel final assembly. The workstation dedicated for the assembly of the car hangers is adjustable to facilitate the assembly operations for all hanger types. Small components such as screws, nuts, rollers etc. are stored at the workstation in small boxes. Other components, dedicated for batched manufacturing orders, are prepared by the logistic responsible in bigger boxes before sent to the assembly lines. The hangers, before the assembly process, arrive to the workstation in standardized big boxes (containers), prepared by their supplier. Lastly, special carts are used after the completion of the assembly process in the discussed workstation, needed to transfer the assembled hangers to the next workstation for further processing. In its current form, a single assembly worker is working in assembly station.

Although they share common physical properties, each panel hanger has unique components to be assembled and specific instructions to follow in terms of location and orientation of components on the hangers. Right now, human operator does two stage control mechanism to ensure assembly is performed as stated in the work order : (1) *before* starting the assembly, *visually* checking and making sure that all order specific components are present in the station and delivered with the hanger (2) *after* completing the assembly, *visually* checking all components are assembled on the correct position and orientation. This visual control causes mental fatigue for operators and cause quality escapes as well. Therefore, one of the functionalities expected in SHERLOCK collaborative station is having process perception capability to support human operator for correctness of the assembly process. In that regard, United Technologies Research Center-Ireland (UTRC-I) is developing computer vision-based process perception module for both component verification and assembly validation purposes. In the first stage, *before* assembly starts, process perception module will detect the components present in the assembly station via a real-time depth camera and retrieve the list of required components to compare the ones with detected ones. In the second stage, *after* assembly is completed, the object detection will be enriched with detection of location of assembly parts and their orientation for comparing and validating the assembly regarding to CAD model. Detailed implementation plan is given in section 7.1.

5.2. VDL

In this use-case, the objective is to apply HRC to the assembly of panels used to cover microchip production machines. Panels have different sizes and consist of several components like outer and inner panels, hangers, handlers, screws, rings or rivets, with different sizes. The robot must be able to assist the operator in the assembly of the panels. To do so, it will perform operations such as riveting or marking the position where the operator should place a sealing strip with a laser beam.

Process Perception Module should support the seamless cooperation of the two actors, namely the human operator and the robot, by providing the following functionalities: i) detection of the small parts being installed on top of the panels, ii) verification of the correctness of assembly and iii) rivet holes detection for finetuning of the position of the robot prior proceeding with the riveting task.

5.3. SOFITEC

In the case of SOFITEC, an autonomous mobile dual-arm manipulation robot will be collaborating side by side with an operator. The scenario involves is co-transportation and co-manipulation of large-sized parts. For that purpose, the mobile robot needs to be able to detect the storage structures from which such objects will be retrieved, and the holding parts of the machines and workstations where they will be placed at. The process consists of the following sequence: First, the robot and the operator take the part from the storage unit in tandem and transport it to the ultrasonic inspection area. Once arrived, the robot should be able to recognize the part positioning tool in the ultrasonic machine, and collaboratively place one side of the part while the operator places the opposite. Once the inspection is finished, the operator and the robot remove the part form the machine and transport it to the dry and documentation area where they have to place it in specific stakeholders. After the part has been cleaned, they will transport to the final storage unit.

The important items to be recognized by the perception pipeline in this scenario are the storage shelves, the holding points of the inspection machines and the stakeholders in the dry and documentation area. While transporting the part, the robot will be guided by the force applied by the operator.

5.4. FIDIA

FIDIA produces milling machines. Through its three production plants, they perform the complete assembly of the machines until it is customer ready. The assembly of this kind of machinery is complex, where thousands of parts require collection, positioning and, testing. This process is extremely complex for introducing an HRC solution. Instead, a physical and cognitive assistance system for the operator will be adopted. This consists of a set of specific tools capable of reducing fatigue (both mental and physical) for the operator and increase the efficiency of the assembly process.

One of the tools will be the utilization of exoskeleton devices. These will support the operator in tasks which involve manipulation of heavy parts. The other novel tool will be a cognitive support system based on Augmented Reality technology, providing capabilities of detection and identification of the different components and parts, as well as provision of step by step instructions that will guide the operator through the assembly process.

6. APPROACH

From the analysis of the use case scenarios described at Section 5 and the identification of their requirements related to process perception, in was prominent that the PPM module should provide two functionalities:

- Identification of tasks being executed by human operators and validation of assembly
- Detection of parts to be manipulated by robots and localization of their position

Thus, the PPM internal architecture/structure can be summarized as shown in Figure 1:

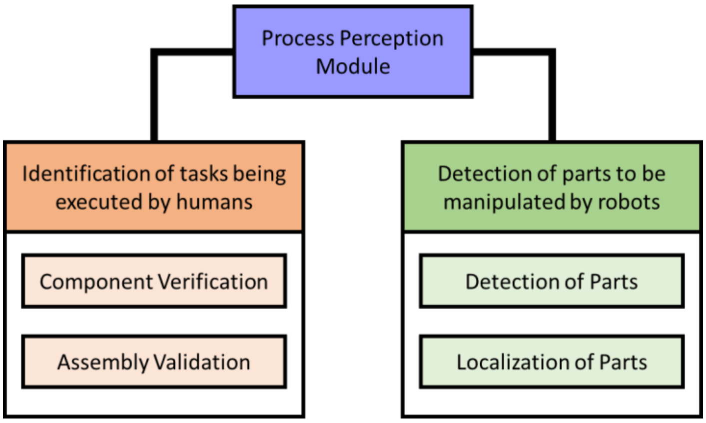

Figure 1: PPM architecture

Having the above in mind, a generic methodology was developed for each of those functionalities, applied as a reference example in two use cases of SHERLOCK – OTIS and SOFITEC cases-.

6.1. Identification of tasks being executed by humans - OTIS use case example

This section covers the hardware and software requirements of the equipment used to develop the feature of PPM related to identification of tasks being executed by humans and applied as an example to the OTIS use case.

A stationary RGB camera is installed above the matrix table and is used for verification of the number of components placed at the matrix by the human operator. As soon as all the relevant parts are placed, PPM informs the robot to move on and pick the matrix.

On the other hand, the Hololens device that is used to provide cognitive support to the operator (OSM module, D3.2), is used as a means of assembly validation. PPM will use data deriving from the built-in cameras of Hololens in order to track in real-time the parts that the operator assembles on the hanger and check if the parts are installed following the correct sequence.

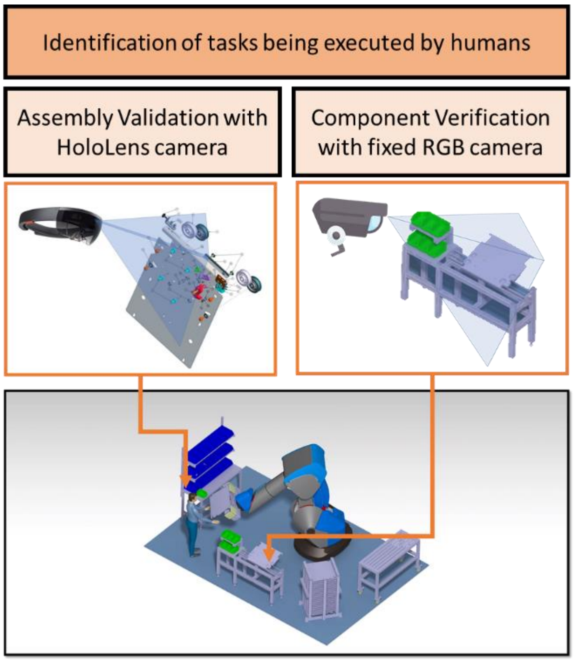

Figure 2: PPM applied at OTIS case

The hardware and software requirement for the Component verification and Assembly validation API are presented in this section.

A depth camera is utilized to capture video and corresponding depth information in the frames of the video and a gaming laptop which has a graphical processing unit (GPU) are the hardware requirements for the development of the API. The used hardware devices, and their features are shown in the following table (Table 1):

Table 1: Hardware requirements

Device	Features
ASUS ROG GL703G	Panel FHD 1920×1080 16:9. CPU Intel® Core™ i7-8750H Processor 2.2 GHz (9M Cache, up to 4.1 GHz). RAM DRAM DDR4 2666 16G. HDD1 PCIEG3x4 NVME 512G M.2 SSD. HDD2 SATA 1TB 5400RPM 2.5' Hybrid HDD (FireCuda). VGA NVIDIA GeForce GTX1070. WiFi 802.11ac (2*2) + Bluetooth. OS Windows 10 (64bit). Other 1 1A-Gunmatel Other 2 Gaming Backpack & Headset & Marketing Giveaway, Gaming mouse (Impact), Illuminated Chiclet Keyboard 4-Zone RGB
Intel RealSense D415	Dimensions: 99 mm x 20 mm x 23 mm Image Sensor Technology: Rolling Shutter, 1.4µm x 1.4µm pixel size RGB Resolution: 1920 X 1080 RGB Frame Rate: 30 fps Minimum Depth Distance (Min-Z): 0.16 m Depth Field of View (FOV): 65°±2° x 40°±1° x 72°±2° Depth accuracy: <= 2% Connectors: USB-C* 3.1 Gen 1* Maximum Range: Approx. 10 meters

Ubuntu 18.04.2 operating system will be used for the development of the API. The API will be implemented using in python language. Most of the scientific computing packages are available in the anaconda distribution. The Pillow and OpenCV with python wrapper are used for image processing

purposes. Tensorflow and Darknet provide open source neural network framework. Blender is an open source 3D creation suite used for image rendering purposes in this use-case. FreeCAD is an open-source parametric 3D modeler used to create mesh files of the individual components from the CAD files in OTIS case. The software tools, their versions and sources are given in the table below (Table 2):

Table 2: Software requirements

Software Product and Version	Source	Description
Ubuntu 18.04.2 Operating System	http://www.ubuntu.com/	Open source Operating System
Anaconda distribution with Python 3.5	https://www.anaconda.com/distribution/	Open-source distribution of the Python and R programming languages for scientific computing
Pillow==6.0.0	https://pypi.org/project/Pillow/	Open source package for image processing
OpenCV-python ==4.1.0.25	https://pypi.org/project/opencv-python/	Open source package for image processing with python wrapper.
Tensorflow-gpu==1.12.2	https://www.tensorflow.org/	Open-source software library for dataflow and differentiable programming across a range of tasks.
Darknet	http://pjreddie.com/darknet/	Open Source neural network framework
Blender	https://www.blender.org/	Open source 3D creation suite
FreeCAD	https://www.freecadweb.org/	Open-source parametric 3D modeler

6.2. Detection of parts to be manipulated by robots - SOFITEC use case example

This section covers the hardware and software requirements of the equipment used to develop the feature of PPM related to detection of parts to be manipulated by robots, developed by TECNALIA and applied as an example to the SOFITEC use case.

3D cameras installed on top of the mobile robot will be utilized to detect parts at three areas of interest, as shown in Figure 3.

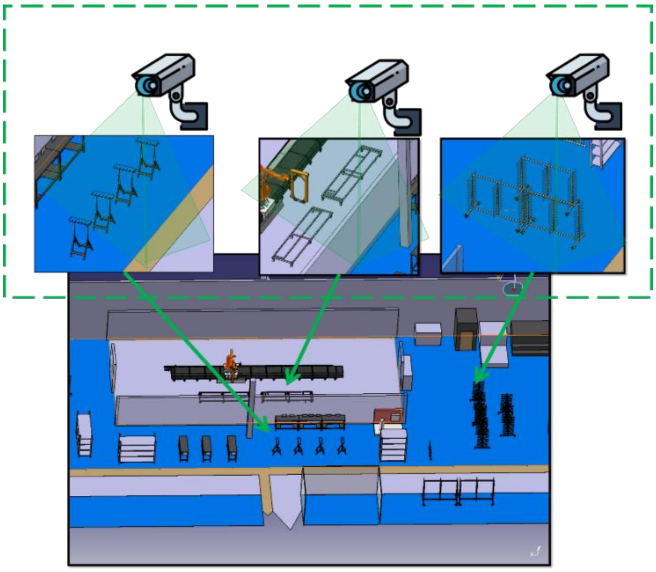

Figure 3: PPM applied at SOFITEC case

The used hardware devices, and their features are shown in the following table (Table 3):

Table 3: Hardware requirements

Device	Features
Dell Latitude 5590	15.6” 1920×1080. CPU Intel® Core™ i7-8650U Processor 1.90 GHz / 4.1 GHz. GPU Intel UHD Graphics 620 300 MHz. RAM 8GB DDR4, 2400 MHz. SSD DATA 512GB M.2
Zivid One+ Medium	Dimensions 226 mm x 86 mm x 165 mm, 2 kg. Structured light technology. RGB resolution 1920x1200. Range 0.6 – 2 m. Precision 0.06 mm > 1 mm . FOV 433x271 at 0.6m, 1330x871 at 2 m. Spatial resolution 0.23 at 0.6 m, 0.75 at 2 m.

The technology has been developed on Ubuntu Bionic (18.04), using open source software libraries, listed on the table below (Table 4).

Table 4: Software requirements

Software Product and Version	Source	Description
Ubuntu 18.04.2 Operating System	http://www.ubuntu.com/	Open source Operating System.
Point Cloud Library	http://pointclouds.org/	Open source package for point cloud processing.
OpenCV	https://opencv.org/	Open source package for image processing.
FLANN	https://github.com/mariusmuja/flann	Library for performing fast Approximate Nearest Neighbors searches in high dimensional spaces.

7. IMPLEMENTATION

This chapter covers explanations and schematics for the different solutions proposed in the Process Perception Modules.

7.1. Identification of tasks being executed by humans - OTIS use case example

The hardware and software specifications for the development of the Component verification functionality are mentioned in the section 6.1 of this document.

The API is implemented using the Darknet [1] deep learning framework and YoloV3 [2] model. The model is trained on a synthetic dataset comprising of components belonging to the OTIS case. The following section presents the creation of synthetic dataset of the components based on their CAD models and details regarding the hyper parameters for training the object detector for component detection. After detection of components in picture/frame, there are two different processes viz., component verification and assembled component validation are also presented in this section. Figure 4 shows the illustration of the synthetic dataset creation using CAD models of the Components. Figure 5 shows examples images from the Synthetic dataset of OTIS case parts.

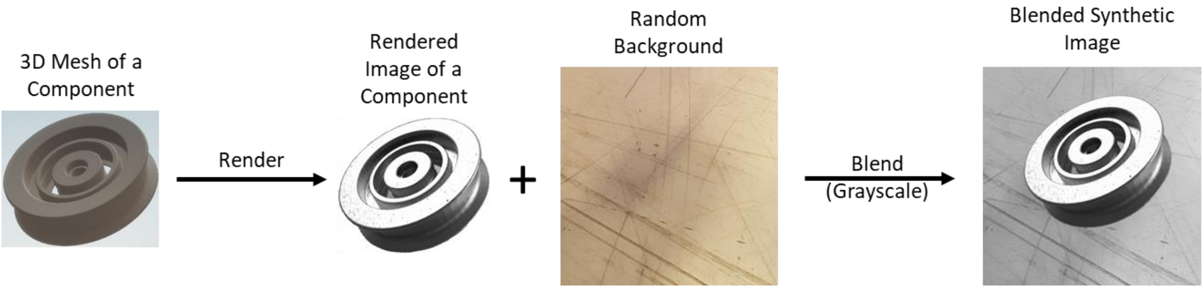

Figure 4: Illustration of synthetic dataset creation using CAD models of the Components

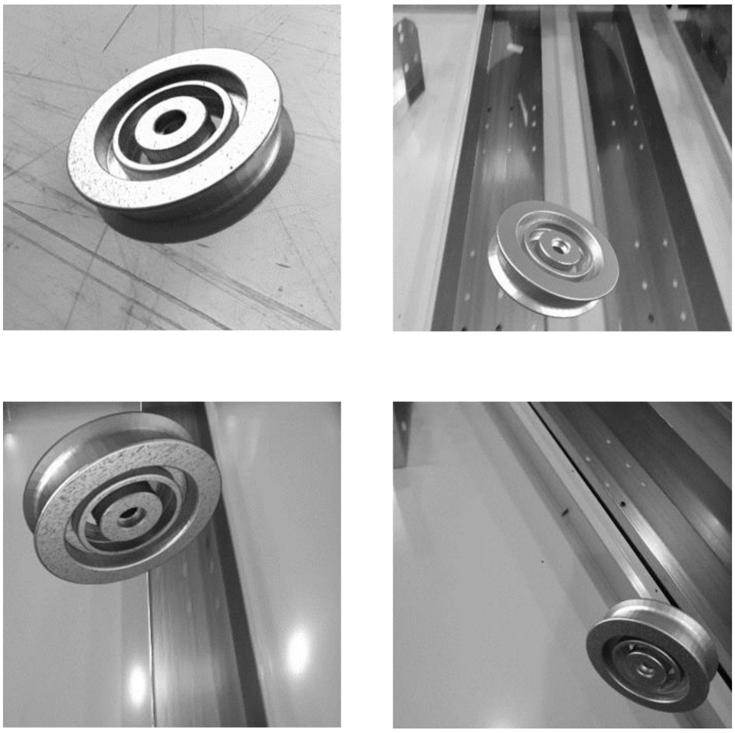

Figure 5: Example images from the Synthetic dataset of Components.

A large synthetic dataset comprising of components related to OTIS Case is created using the CAD models of the components. Initially, FreeCAD [4] is used to create/extract mesh files of the individual components from the CAD files. Blender [3] has been used to render the photorealistic textures on the CAD models with lighting using High-Dynamic-Range imaging. The rendering of the components is done for random components orientation and random camera viewpoints with respect to image. Once, a component is rendered, the rendered components are blended with random background images with some Gaussian noise and blur, random change in brightness and converted to greyscale images in order to make the detector color invariant. In this dataset, 500 hundred images are rendered for every component category for training the object detector.

Further for component detection, the network is trained using SGD with the initial learning rate of 10^{-4} for 120 epochs. We use the standard training strategies such as momentum of 0.9 and 0.0005 weight decay. The batch size is set to 32 in all our experiments. The size of the input image in our experiments is set to 416×416. After training the object detector, the trained model is tested on the using pictures of the components or a streaming video to detect various components involved in the OTIS case. Figure 6 shows the workflow of component verification based on the components detected and the list of the required components for the assembly in the OTIS Case.

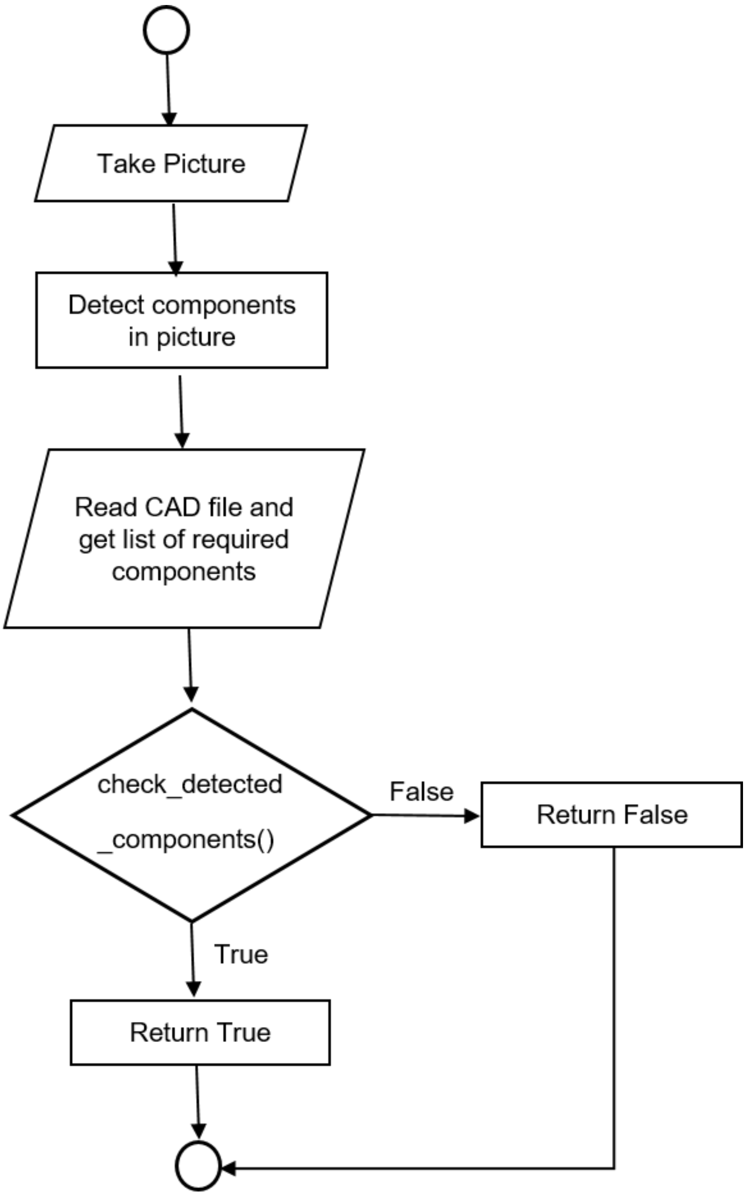

Figure 6: Diagram of the component verification workflow.

The pseudo code for the workflow is depicted in the Figure 7.

```
missing_components = []
for comp in required_compoents:
    if not comp in detected_compoents:
        missing_components.append(comp)

if len(missing_components)>0:
    return True, missing_components
else:
    return False, missing_components
```

Figure 7: Pseudo code of the component verification process.

Figure 8 summarizes the workflow of assembly validation based on the components detected to ensure correct assembly by verifying the sequence of operations based on the list of the assembly steps involving corresponding components in the OTIS Case.

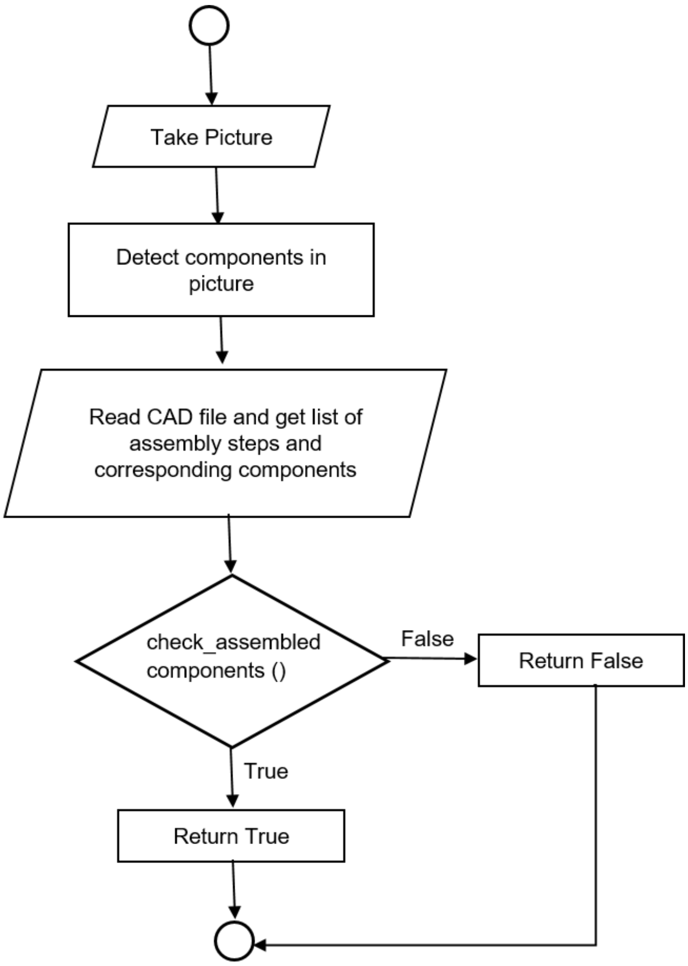

Figure 8: Diagram of the assembly validation workflow.

The pseudo code of the workflow is depicted in the Figure 9.

```
assembly_components = {}
for step in assembly_step:
    comp = step.component
    assembly_components[comp] = {}

    if comp in detected_components:
        assembly_components[comp]["detected"] = True

        location_error, orientation_error = comp.location-
detected_componet[comp].location,
                                            comp.orientation-
detected_componet[comp].orientation

        if location_error < location_thres and orientation_error <
orientation_thres:
             assembly_components[comp]["correct_placement"] = True
        else:
            assembly_components[comp]["correct_placement"] = False

    assembly_components[comp]["detected"] = False

# Only return True if all components are correctly placed
for comp in assembly_componets:
    if not comp["detected"] or not["correct_placement"]:
        return False

return True
```

Figure 9: Pseudo code of the assembly verification process.

7.2. Detection of parts to be manipulated by robots - SOFITEC use case example

This methodology developed for the SOFITEC use case builds an *expert system*, able to determine the best computer vision pipeline to detect and estimate the pose of industrial objects. A recognition pipeline is a three-step process:

1. Object detection: Finding the object of interest in an image.
2. Feature description: Finding a data structure to represent keypoints that belong to this object, so that it can be found in different images.
3. Feature matching: Finding correspondences between descriptors from different images.

Multiple feature descriptors and algorithms for detection and matching exist, as there is no universal recognition pipeline, since their performance is heavily affected by object properties such us colour, texture, shape, brightness, reflectance etc. The role of this expert system is to be able to recognize which pipeline suits better for a given object based on its characteristics Figure 10. To do so, the system must be trained on learning how to classify different objects into groups (or clusters) that share properties that makes it suitable for being recognized by a particular pipeline. These groups are called typologies.

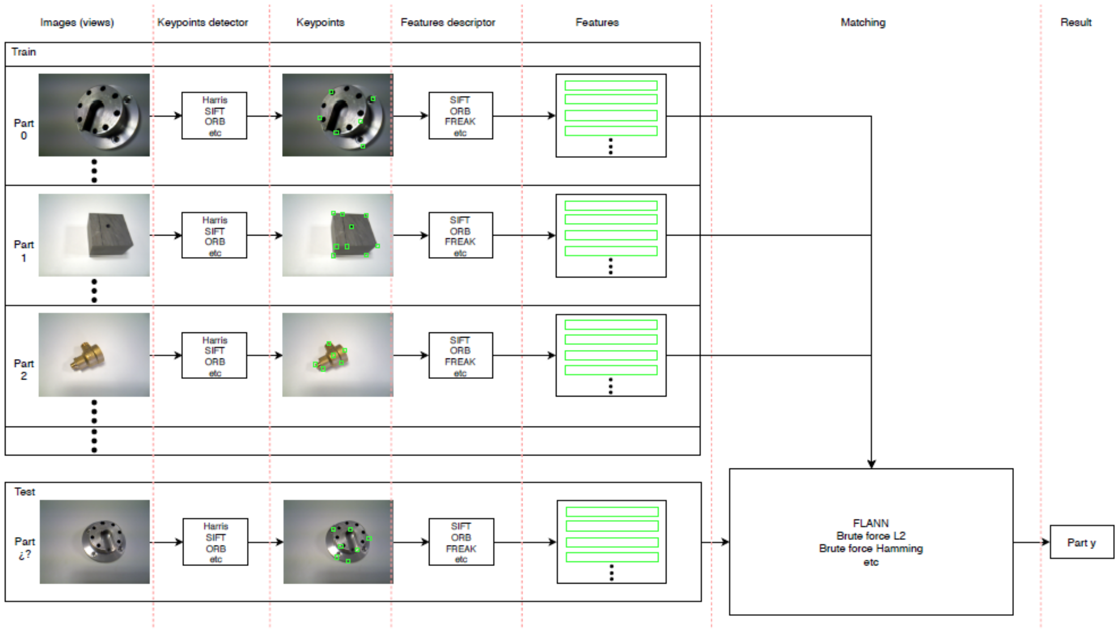

Figure 10: Schematics of the expert system for identifying best recognition pipelines.

The expert system is trained offline. To do so, we built a part-dataset, composed of 3D point clouds (called views) taken from each object to be recognized by our system. From each part we produce a total of 50 views, obtained by scanning each part with a Zivid One camera[1] (structured light). Alternatively, if the part is too big or it is not physically available, it is possible to generate the views from the CAD model and with a simulated 3D camera. Once the dataset is generated, each object undergoes the set of available pipelines used. For each industrial part we have a set X of views (as said, 50) and the object label, denoted as *Y*. The training method used Leave-One-Out Cross-Valiation. It is based on performing *X* iterations for each object, in which the dataset is divided in a training set, with $(X - x_i)$ and a test set composed by only the one x_i. Here, *i* takes values from 1 to 50. Each recognition pipeline is evaluated by finally computing, a score based on the metric F_1:

$$F_1 = 2 \cdot \frac{precision_y * recall_y}{precision_y + recall_y} \quad (1)$$

based on the precision -correct predictions divided by all predictions- and recall correct predictions divided by all views. The final score $\overline{F_1}$, which is the mean of all the F_1 scores for each object, is used to retrieve the best pipeline for such object.

The algorithms used by the expert system are classical computer vision detectors, including SIFT, SURF, FAST, BRISK or ORB. The feature matching methods considered include algorithms such as Nearest Neighbours, FLANN and Brute-Force Hamming.

The ultimate goal of the expert system is to build a hierarchical classification of industrial objects based on their similarity. This similarity is related to properties such as texture and shape, which make such objects suitable for a particular detection algorithm. To build the classification system, a matrix containing all the objects used (rows) and the F_1 scores for all the pipelines available (columns) is used as input. To this matrix, we apply K-means clustering algorithm to generate the typologies mentioned

[1] Zivid One camera, http://www.zivid.com/, 31/3/2020

above Figure 11. This training system method is systematic, making it easy the introduction of new parts to the process global dataset.

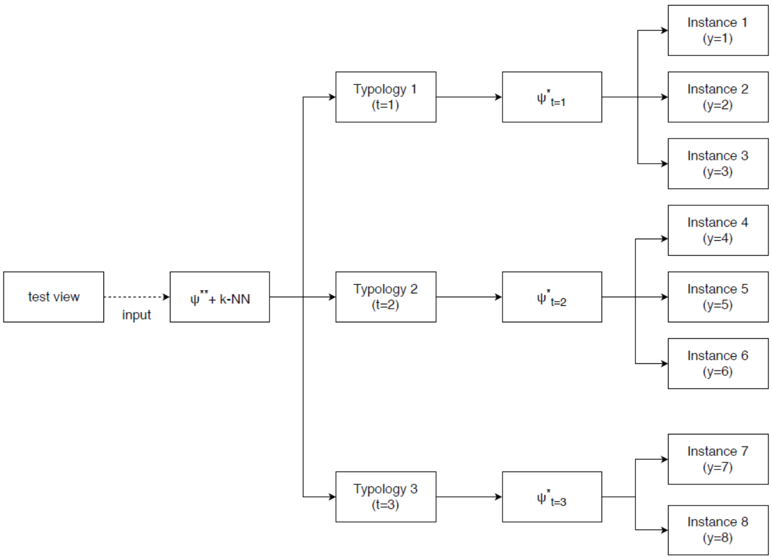

Figure 11: Hierarchical classification of industrial parts using the expert system.

With a fully trained expert system, the recognition process for the parts involved in the SOFITEC case (Figure 12) would consist on two major steps: Identification of the typology to which each part belongs (offline), and detection and pose estimation using the best scoring recognition pipeline for such typology (process perception module).

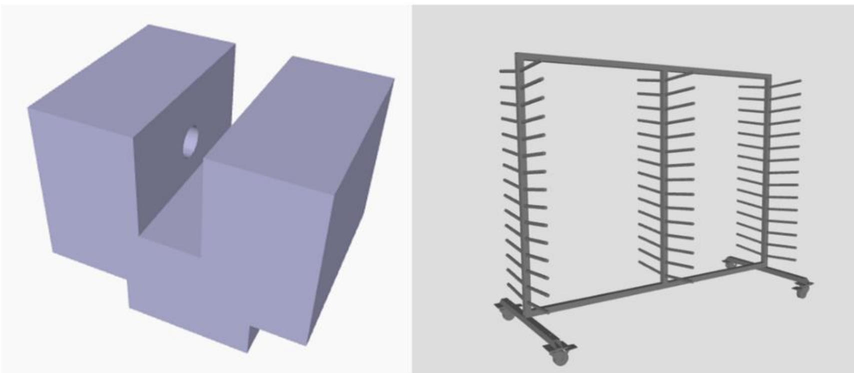

Figure 12: CAD models of the ultrasound fixing point and the storage unit of the SOFITEC plant.

The core API for recognition, is a collection of computer vision software libraries that can be added and interchanged in a plug-in fashion. This scheme allows for easily building recognition pipelines with different detection and matching algorithms on the go. The final perception module will be integrated in a ROS system to provide easy interfacing. The idea is to have the vision system as a ROS node providing a detection service whenever the robot requires usage of the vision system (look and move scheme), so that point cloud traffic in the network is reduced and only used when needed. This way, the process perception can be systematic:

1. Robot arrival to the workspace.

2. Service call to the vision node.
3. Image acquisition.
4. Detection and pose estimation of the part.
5. Object pose transformed to robot task frame and saved.

The solution for the process perception module regarding SOFITEC use case was published in [5].

8. CONCLUSIONS AND FUTURE WORK

As of the date of submitting this deliverable, UTRC-I has completed developing of the first prototype of the identification of tasks being executed by humans functionality of PPM demonstrated, at the OTIS use case, focusing of the development of the component detection and verification algorithm explained in Section 7.1.

Future steps defined for this functionality of PPM applied at the OTIS case, among others, involve:

i) Provision of an API that will facilitate the integration of PPM with the rest of the modules of SHERLOCK (M24)
ii) Integration of the first prototype of PPM with the rest of SHERLOCK modules (M24)
iii) Development of an automated pipeline for retrieving the relevant customer order data to compare with the detected components (M36)
iv) Enhancement of the component verification algorithm by adding location and pose detection features for assembly validation (M36)
v) Utilization of the Hololens headset used in Operator Support Module (OSM) (D3.2), as video input device for component detection and verification algorithm.

Regarding detection of parts to be manipulated by robots functionality of PPM, the first prototype of PPM was delivered, demonstrated in SOFITEC use case, trained to work on both physical parts and CAD representations of the parts involved in the SOFITEC case. Up until now, testing of the working prototype has been performed offline.

Future steps defined for this functionality of PPM applied at the SOFITEC case, among others, involve:

i) Integration of the first prototype of PPM with the rest of SHERLOCK modules (M24)
ii) Testing of the working prototype at the integrated SOFITEC scenario and enhancement of the module based on the results of the testing sessions (M36)

As far as the VDL and FIDIA cases are concerned, the two functionalities of PPM demonstrated at the OTIS and SOFITEC case will be applied to the VDL and FIDIA scenarios (M36).

9. GLOSSARY

AR	Augmented Reality
PPM	Process Perception Module
OSM	Operator Support Module
RHCM	Robot to Human Communication Module
HRI	Human Robot Interaction
HRC	Human Robot Collaboration/Collaborative
EED	Enhanced Exoskeleton Device
MDAM	Mobile Dual Arm Manipulator
HPCM	High Payload Collaborative Manipulator
LPCM	Low Payload Collaborative Manipulator
CAD	Computer Aided Design
API	Application Programming Interface
AI	Artificial Intelligence

10. REFERENCES

[1] Redmon, J. (2013). Darknet: Open source neural networks in c.

[2] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[3] Blender, https://www.blender.org/

[4] FreeCAD, https://www.freecadweb.org/

[5] Ibon Merino, Jon Azpiazu, Anthony Remazeilles, Basilio Sierra. 2D Features-based detector and descriptor selection system for hierarchical recognition of industrial parts. International Journal of Artificial Intelligence & Applications (IJAIA) Vol 10. No. 6 (2019).