Mobile dual arm robotic workers with embedded cognition for hybrid and dynamically reconfigurable manufacturing systems

Grant Agreement No	:	**723616**
Project Acronym	:	**THOMAS**
Project Start Date	:	**1st October, 2016**
Consortium	:	**UNIVERSITY OF PATRAS (LMS)** **PEUGEOT CITROEN AUTOMOBILES S.A. (PSA)** **SICK AG (SICK)** **FUNDATION TECHNALIA RESEARCH & INNOVATION (TECNALIA)** **ROBOCEPTION GMBH (ROBOCEPTION)** **DGH ROBOTICA, AUTOMATIZACION Y MANTENIMIENTO INDUSTRIAL SA (DGH)** **AERNNOVA AEROSPACE S.A.U. (AERNNOVA)** **INTRASOFT INTERNATIONAL SA (INTRASOFT)**

Title	:	Human detection and tracking – Initial Prototype
Reference	:	D2.3
Availability	:	Public
Date	:	31/03/2018
Author/s	:	TECNALIA, SICK, LMS
Circulation	:	EU, consortium

Summary:

This document reports the earliest implementations and tests of the components of the human detection and tracking system.

Table of Contents

LIST OF FIGURES 3
LIST OF TABLES 5
1. EXECUTIVE SUMMARY 6
2. INTRODUCTION 7
3. 2D LASER BASED HUMAN DETECTION 8
3.1. Motivation 8
3.2. Implementation 8
3.2.1. System overview 8
3.2.2. Ego Motion Estimation 10
3.2.3. Motion Filter / Background Filter 10
3.2.4. Segmentation 12
3.2.5. Segment Filter 12
3.2.6. Pre-Classification 12
3.2.7. Tracking 12
3.2.8. Post-Classification 13
3.3. Testing 14
3.3.1. Setup 14
3.3.2. Stationary MRP 15
3.3.3. Dynamic MRP 16
3.4. Conclusion 16
4. 3D BASED HUMAN DETECTION 17
4.1. Implementation 19
4.2. Testing 20
5. HUMAN-ROBOT INTERACTION 24
5.1. Wearable device application 24
5.1.1. Smartwatch application 24
5.1.2. Integration in the THOMAS overall system architecture 26
5.1.3. Prototype implementation 28
5.2. Augmented Reality 30
5.2.1. AR Operator Support application 30
5.2.2. Integration in the THOMAS overall system architecture 31
5.2.3. Prototype implementation 32
5.3. Human Gestures / Posture Recognition for direct MRP guiding 34
5.3.1. Gesture recognition application 35
5.3.2. Integration in THOMAS overall system 38
6. CONCLUSIONS 39
7. GLOSSARY 40
8. REFERENCES 41

LIST OF FIGURES

Figure 1: SICK microScan3 ... 8

Figure 2: System overview of the 2D laser scanner data based human detection and tracking ... 9

Figure 3: Basic structure of a tracking module ... 10

Figure 4: Exemplary excerpt of a laser scan containing one human around 4 m away from the sensor; both legs are highlighted in green ... 11

Figure 5: Velocity plot over time for a single leg of one of the test persons; red: center point of the detected leg; green and blue: two of the evaluated fits ... 13

Figure 6: left: mathematical plot of the angle between the two human legs, assuming the green motion model from Figure 5; right: estimation of the angle from the scan points ... 14

Figure 7: Object classified as human based on its motion; blue: plot of the angle between the two legs over time; green: Fourier transform of the curve ... 14

Figure 8: left: mockup of the MRP that was used to test the 2D tracking system; right: SICK microScan3 ... 15

Figure 9: Detection of the human ("Pedestrian") walking around the MRP ... 15

Figure 10: Example for false detections due to ego motion ... 16

Figure 11: Camera configuration on Tecnalia laboratory ... 17

Figure 12: Point Clouds received from cameras, non-fused vs fused ... 18

Figure 13: Detection of human by one of the cameras ... 18

Figure 14: Workspace safety areas ... 19

Figure 15: Quality of the camera comparison (SD, qHD, HD) ... 20

Figure 16: One camera configuration ... 21

Figure 17: Two camera configuration ... 22

Figure 18: Three camera configuration ... 23

Figure 19: Smartwatch interfaces ... 25

Figure 20: Smartwatch application – Message exchange sequence diagram ... 26

Figure 21: Station Controller with the available topics ... 27

Figure 22: HRI integration architecture ... 28

Figure 23: Smartwatch application integrated to station controller – Testing task takeover button ... 29

Figure 24: Smartwatch application integrated to station controller – Testing task completed button ... 29

Figure 25: AR application – Message exchange sequence diagram ... 31

Figure 26: AR application integration architecture ... 32

Figure 27: Assembly operation steps – Instructions and 3D model ... 33

Figure 28: Assembly operation– Instructions only ... 34

Figure 29: Alert message ... 34

Figure 30: Stance for Body classification ... 35

Figure 31: Medium Size Operator Gesture ... 37

Figure 32: Large Size Operator Gesture ... 37

Figure 33 Ros Message ... 38

Figure 34: Gesture application integration architecture..38

LIST OF TABLES

Table 1: Main services used by smartwatch application..27
Table 2: Description of status IDs...27
Table 3: Main service used by smartwatch application 31
Table 4: Main topics used by smartwatch application..31
Table 5: Description of status IDs...32
Table 6: Gesture - Robot arm movement assignment..36
Table 7: Main topics used by gesture recognition application..38

1. EXECUTIVE SUMMARY

This deliverable D2.3 “Human detection and tracking – Initial Prototype” reports the development status of the initial prototypes of the different components that compose the system that in THOMAS will be used for perceiving humans around the workplace.

The deliverable will be focused on reporting experimental setups and prototype implementations, without going into detail in the actual techniques, methods and algorithms used.

The document is organised in five main sections:

Section 2 will provide and introduction with some background and motivation for the work done.

The three following sections are the main technical sections. They contain the description of the prototypes for 2D laser based human detection under development by SICK (Section 3), the 3D based human presence detection under development by TECNALIA (Section 4) and the wearable based detection and Human-Robot interaction under development by LMS (Section 5).

The deliverable concludes with some summary and conclusions in Section 6.

2. INTRODUCTION

As has been stated repeatedly, safety and Human-Robot collaboration are two key objectives of the THOMAS project. These two objectives are strongly inter-related. While safety is always a very important element of any industrial system, the introduction of the Human-Robot interaction capabilities make safety a totally crucial element of the system. Since collaborating with robots imply close proximity with them, humans will put themselves constantly at risky situations that need to be properly managed. Moreover, current standard safety methods rely normally in purely proximity based warnings that are triggered by any object that comes inside a pre-defined safety area. In Human-Robot collaboration, humans have to actually come close to robots, so these standard safety methods need to be improved for effective collaboration.

Being so, the key element of safety in Human-Robot collaboration is the detection and tracking of the humans, and the differentiation of them from other objects or elements that can pose a risk. The single most important information required is "where" the humans are at any moment. Thus, fast and reliable human detection and tracking systems are required.

Human detection methods are a big trend of research, mainly in 2D vision based approaches. In THOMAS project, the intention is to develop a robust detection and tracking integrated system using both custom developments and off-the-shelf components, and mainly using sensors that will be already deployed in the robot, the workshop and the workstations.

The human detection and HRI system will be then composed of three elements:

- A laser based detection system developed by SICK. Safety laser scanner are ubiquitous in workshops. They are present in the MRP, the MPP and the workstations. They are also fast and accurate, so they are a very good source of information for human detection.
- A 3D based detection system developed by TECNALIA. RGBD cameras and other 3D cameras are very common in robotics and other applications. The MRP will be equipped with several and the deployment of them in the workstations is under study. Using of-the-shelf components, the development will be focused on improving detection robustness, increasing coverage and fusion of different sources of 3D information.
- A wearable based HRI system developed by LMS. Wearables are going to be used for interaction with the robots (send commands, validate actions, etc.). Wearables also include sensors that potentially can be used to detect and track the presence of the human wearer.

The development status of each system will be described in the following sections.

3. 2D LASER BASED HUMAN DETECTION

The primary objective of this research is the development of a module for the detection and tracking of humans that are located in close proximity of the MRP. This information can then be forwarded to and used by higher level modules such as the 3D-based human detection and integrated into the world model. The following sections will provide an insight into the current implementation and testing and will also address the challenges that still need further development.

3.1. Motivation

As described in D2.4, the safety design of the MRP will include two safety laser scanners, two SICK microScan3 (Figure 1). Their primary purpose is to mitigate risks that are imposed by the movement of the MRP and its robot arms by monitoring pre-defined safety fields.

Figure 1: SICK microScan3

In addition to their usage as safety-rated sensors, it is possible to retrieve the measurement data which consists of a 2D laser scan that provides a discrete representation of the environment in a 2D plane. The data can be post-processed to generate further information and usage, such as the detection and tracking of surrounding objects and reflector or contour-based localization.

As one important part of this project investigates human-robot interaction, presence detection of humans in the vicinity of the MRP is of particular interest. Use cases based on 2D perception data mainly include preventive measures, such as obstacle avoidance during the navigation of the MRP's platform and also the robot arms or a preventive adaption of their speed. While the 2D laser scanners are limited in scanning only a 2D plane, they however do cover a wide range that usually cannot be monitored with 3D perception sensors. In combination with 3D perception data, the previous cases can be extended to more complex use cases such as gesture and intention recognition.

3.2. Implementation

The description of the implementation starts by providing an overview of the sub-modules that the system for human detection and tracking based on 2D laser scanner data is comprised of.

Following up, each sub-module will be described in more detail.

3.2.1. System overview

When developing a tracking system, common challenges need to be dealt with.

The most important ones include:

- Ego motion compensation (i.e. the motion of the mobile platform hosting the sensors)
- Measurement noise
- Foreground / Background detection
- Initialization of new targets
- Modelling the target movement
- Data association between detections and known targets

- Object classification

The developed system consists of several sub-modules that address these challenges. An abstract overview of the sub-modules is given in Figure 2.

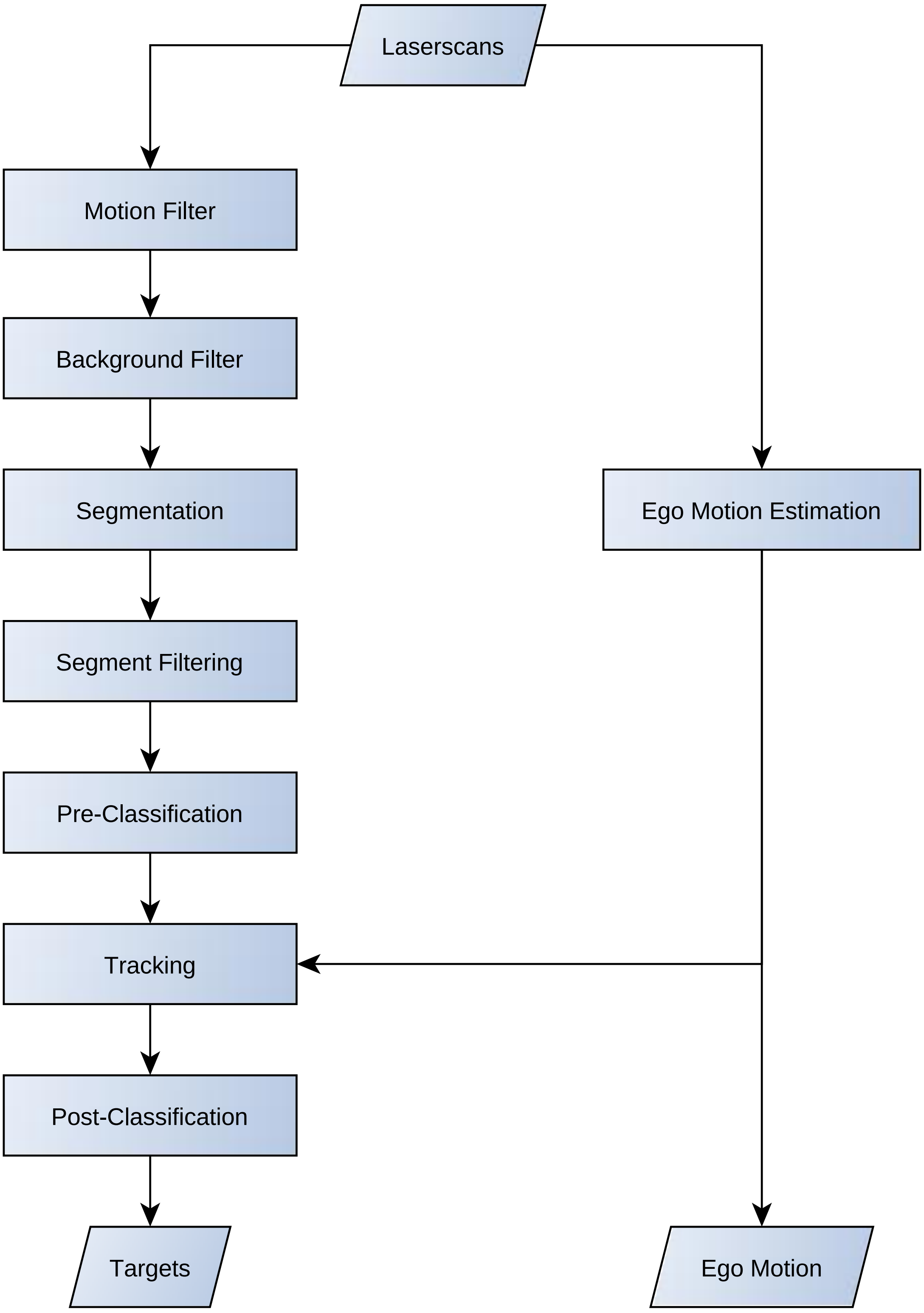

Figure 2: System overview of the 2D laser scanner data based human detection and tracking

3.2.2. Ego Motion Estimation

The typical tracking application follows a structure similar to the one depicted in Figure 3.

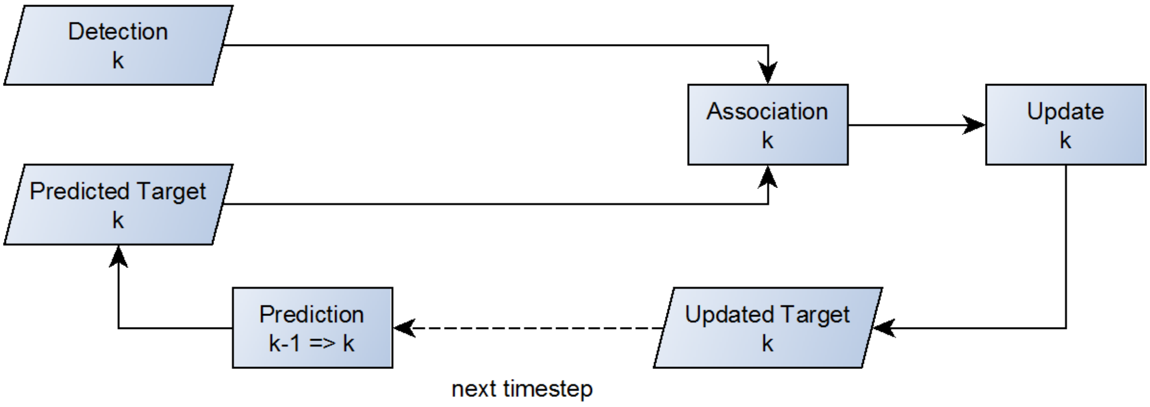

Figure 3: Basic structure of a tracking module

Sensor data is only available at a specific frequency and is thus discrete in time. As such, an association between consecutive time steps needs to be accomplished. At each time step, detections are computed from the raw sensor data which are then associated with known targets that were already tracked in previous time steps. For each detection that is not associated with a known target, a new target is initiated. For the association process, the motion of the target needs to be compensated. Thus, commonly a motion prediction of all known targets is performed. For this prediction, a motion model needs to be assumed. A well-proven approach – which is also used in this work – is the constant velocity assumption in combination with a white noise acceleration.

In addition to the motion of the targets between consecutive time steps, a possible movement of the sensor itself needs to be taken into account as well. This is also called ego motion compensation. There exist different approaches on how to determine the ego motion, for instance odometry measurements calculated from the wheel encoders.

Wheel encoder odometry measurements are exposed to the drawback that an integration of the encoders and the calculation of the odometry needs extra effort and is in particular prone to errors when having to deal with four independent wheels, which is the case for the MRP in this project. Thus, for this tracking system, the ego motion is determined from the laser scanner data itself. This is also known as visual odometry or scan matching. By finding the best transform between the current and the preceding scans, the motion of the sensors can be estimated.

3.2.3. Motion Filter / Background Filter

In a tracking application, there is only a small fraction of the measurement data that is of interest. In this case, the goal is to track humans that are located in close proximity of the MRP, but all other objects are of rather minor interest. As was explained in the previous section, each detection that is not matched to a known target (i.e. tracked object) will create a new target. A reliable tracking system thus needs to have reliable detection to keep the number of tracked targets low and to reduce the complexity in the association between detections and known targets.

For the THOMAS use cases, the humans around the MRP are of interest. Therefore, a perfect detection in our case would detect all humans in the detection area of the laser scanners. This is of course very challenging in the case of a 2D sensor, as there is not much data representing the object. An exemplary excerpt of such a 2D laser scan is presented in Figure 4. This excerpt only contains one human whose legs are highlighted in green colour. The legs are situated at around 4 m distance from the sensor and each leg consists of only 6 scan points. With increasing distance from the sensor, the number of scan points per leg will decrease further.

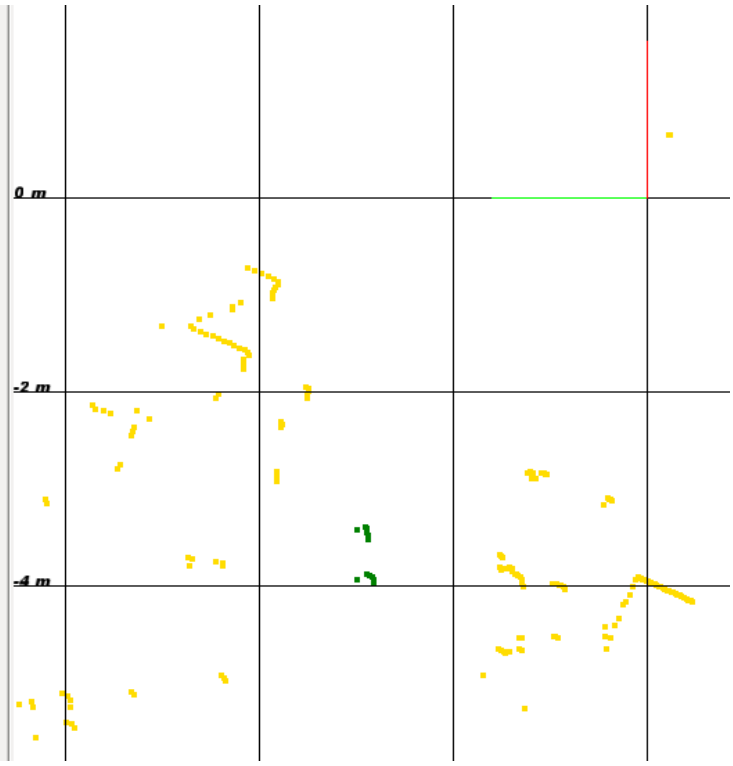

Figure 4: Exemplary excerpt of a laser scan containing one human around 4 m away from the sensor; both legs are highlighted in green

To facilitate a reliable detection of humans and human legs in the 2D laser scan, there exist several strategies that impose different restrictions.

- *Static maps:*
  When using the laser scanner measurement data for navigation purposes, a map of the environment is generated and then used to localize the scanner and the vehicle in the map. If the pose of the laser scanner inside the map is known, all static parts of the map can be masked, as they are not of interest for the tracking. However, this idea introduces a lot of dependencies to other modules and restricts the general usage of the tracking system, as it imposes the need for a pre-recorded map of the environment and a reliable localization algorithm.
  For this tracking system, this strategy is thus discarded at this moment with regard to the mentioned drawbacks to keep the system more encapsulated and to make it applicable to a wider field of use cases.

- *Temporal occupancy monitoring:*
  There exist several approaches that share the idea of detecting the static part of the environment as background and only focus on the temporal changes in occupancy, that are most likely due to motion. Since normal occupancy grids (Elfes, 1989) give a discrete grid representation of the environment with each grid cell holding a specific occupancy probability that is constructed over the entire history of the measurement data, temporal occupancy grids (Arbuckle, Howard, & Maja, 2002) were introduced to limit the influence of old data and only focus on a specific time span. However the discretization of the environment into a grid with grid cell borders enforces the influence of noise, for instance if measurements toggle between grid cells.
  Therefore, we extended the idea of temporal occupancy grids to a more general approach avoiding the discretization into grids. For each scan point, given a history of previous laser scans, the algorithm calculates the prior probability $p_{occ}^{k-n:k-1}$ that the environment is occupied at this specific scan point. The index k denotes the current time step; n specifies the number of time steps that are part of the laser scan history. If $p_{occ}^{k-n:k-1}$ is below a predefined threshold, the environment is considered to have been free at this point. Given the assumption, that a

detection at this spatial point needs to have been caused by a movement – as it had been unoccupied prior to the detection – this provides a filter for motion in the laser scan.

- *Polygon background filtering:*
  Typically, the area of interest is a lot smaller than the area, that the laser scanner is able to cover. For instance, most laser scanners have a maximum range of more than 30 m. For the application in the THOMAS use cases, humans in close proximity of the MRP are of interest. Thus, in our system we use a polygonal line that is used to mask the area of interest around the MRP.

3.2.4. Segmentation

While the previous sub-modules do not require any contextual knowledge about the scan points, the tracking procedure as sketched in Figure 3 needs detected objects, i.e. groups of scan points or segments. For our system, we use a Euclidian approach to group the scan points into segments by specifying a distance threshold for scan points that are assumed to be associated to each other.

3.2.5. Segment Filter

The result of the segmentation described in Section 3.2.4 is a list of segments. Prior to more advanced methods, a first plausibility check is done by simple segment filtering.

These filters address the following characteristics of the segments:

- Minimum number of scan points
- Minimum/maximum size
- Minimum/maximum height
- Minimum/maximum width

As these filters are prone to hard choices of the parameters, they are parameterized conservatively and only seen as a plausibility check for the following filters.

3.2.6. Pre-Classification

In our system, we differentiate a pre- and a post-classification. The pre-classification operates without any knowledge of the targets, i.e. only uses the detections; the post-classification uses characteristics of the targets for the classification.

For the pre-classification, characteristics of the detection are used as input, i.e. the shape of the detected objects. Human legs have a characteristic shape in the laser scan which can be used to generate a rough estimate if an object might represent a leg or not. As described in Section 3.2.3, a leg only consists of few scan points. Additionally, non-human objects in a typically unstructured industrial environment often lead to a similar shape in the laser scan, as illustrated in Figure 4. Therefore, the identification of legs in the laser scan by their shape will not be possible with a high confidence and is prone to a high false-positive rate.

Although the classification of legs by their shape is prone to a high false-positive rate, it can be used in combination with other filters to eliminate this drawback. For our system, we adapted the classifier developed by (Leigh, Pineau, Olmedo, & Zhang, 2015) for a usage with our sensor. It uses a multitude of features such as the width, length, standard deviation and circularity of the points to train a random forest classifier, that can then be used to give an estimate for the likelihood that the given detection is a leg.

In our system, we use this classifier as a filter to only consider segments that fulfil a given classification threshold.

3.2.7. Tracking

The structure of our tracking system follows the structure that is depicted in Figure 3. The filtered segments are taken as input detections for the data association with known targets. The prediction and update cycle is realized by using the prediction and update steps of a Kalman Filter in combination with a constant velocity (CV) motion model, modelling the acceleration as white noise (Kohler, 1997).

Although the CV motion model is not able to realistically model a human's motion, it has proven to be adequate for human tracking.

Each predicted target is then associated with the input detections using an iterative Nearest Neighbour approach. If a detection is not associated to any known target, a new target is initialized. If there is no detection to be associated with a known target, this target will not be updated in this time step but instead the prediction by the motion model will be used. If a target has not been observed for a given number of time steps, it will be removed from the list of tracked targets.

3.2.8. Post-Classification

In contrast to the pre-classification described in Section 3.2.6, that uses the detected segments, the post-classification uses the tracked targets for classification. As mentioned before, a classification of the shape may lead to a high false-positive rate when using 2D laser scanner measurement data. To enforce a more robust classification of objects as humans, the motion of the targets is taken as input for the post-classification.

To get an impression of the motion characteristics of humans in the 2D laser scanner data, several motion models were investigated. In a test setup, the motion of test persons was recorded and then analysed. One exemplary plot with two of the investigated motion models is shown in Figure 5.

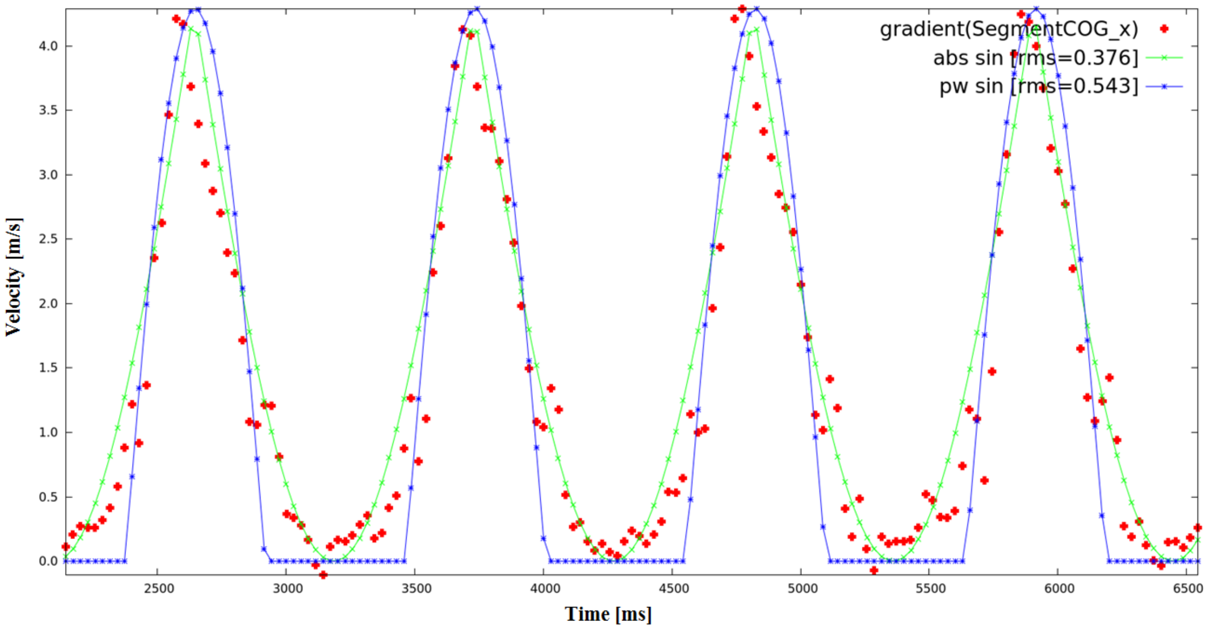

Figure 5: Velocity plot over time for a single leg of one of the test persons; red: center point of the detected leg; green and blue: two of the evaluated fits

The fit, that best described the motion of one leg (green line in Figure 5), is described by the following function:

$$v_{leg} = v_0 * (1 - |sin(\omega t + \phi)|)$$

Based upon this motion model, it can be shown mathematically, that the angle between the two legs follows a function depicted in Figure 6. The Fourier transform of this function has characteristic peaks due to its sinusoidal nature. By comparing the Fourier transform of the curve of the angle between the two legs to the ideal one, humans can be identified by their motion in the 2D laser scan (Fuerstenberg, 2009).

Figure 7 shows an example for the classification of an object as human and the corresponding Fourier transform.

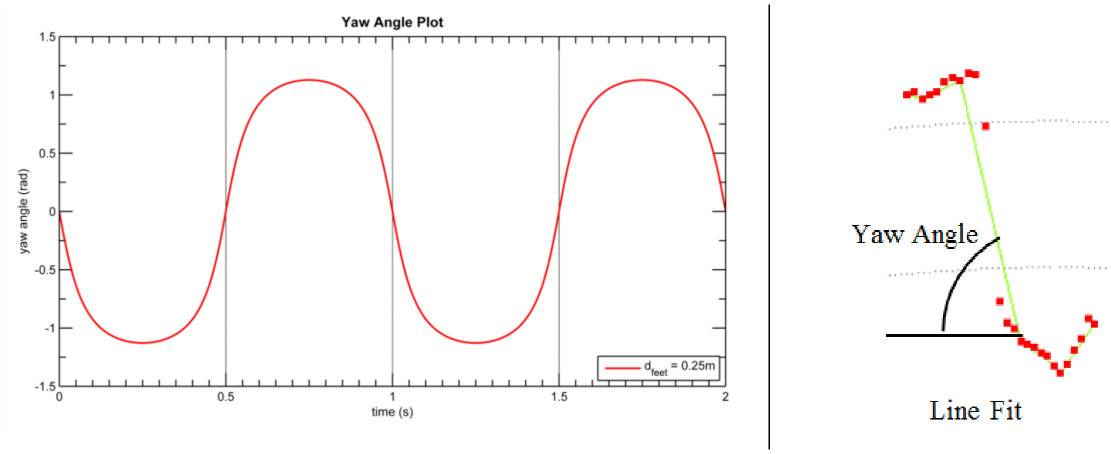

Figure 6: left: mathematical plot of the angle between the two human legs, assuming the green motion model from Figure 5; right: estimation of the angle from the scan points

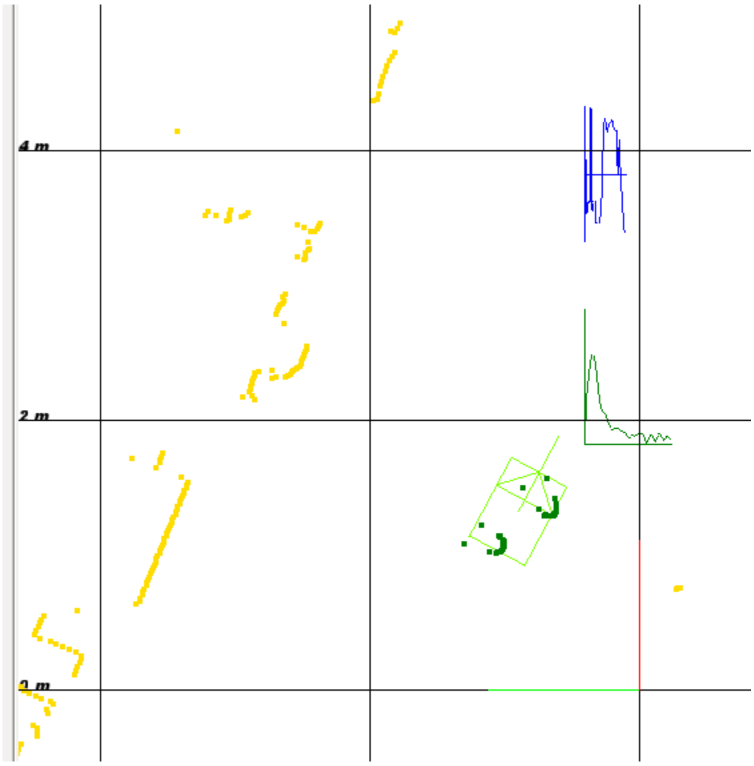

Figure 7: Object classified as human based on its motion; blue: plot of the angle between the two legs over time; green: Fourier transform of the curve

3.3. Testing

This section describes the tests of the developed tracking system on 2D laser scans, which were performed on a mockup of the MRP that was built up at SICK in Hamburg.

3.3.1. Setup

To be able to test the developed algorithms in a setup similar to the one in the use cases, we constructed a mockup of the MRP with similar dimensions and two safety laser scanners SICK microScan3 on opposite corners.

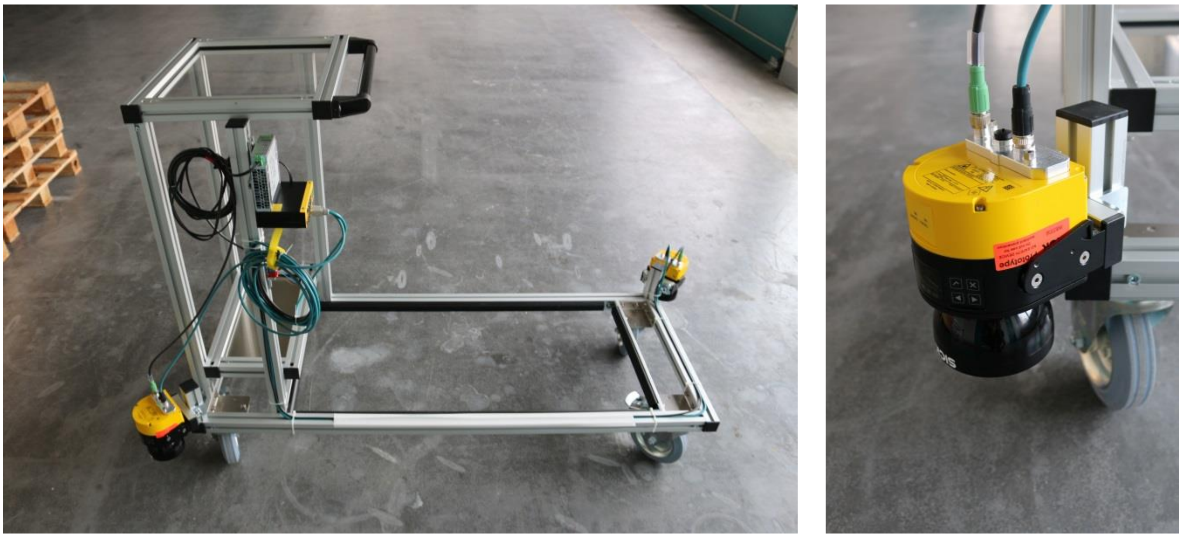

Figure 8: left: mockup of the MRP that was used to test the 2D tracking system; right: SICK microScan3

3.3.2. Stationary MRP

For each of the two laser scanners, the tracking system from Section 3.2 is deployed, enabling a 360° tracking of humans around the MRP. First tests were performed in a stationary scenario in which the MRP was not moving. A human approached the MRP and walked around the MRP. The tracking was able to detect the object and classify it as human. A snap-shot is depicted in Figure 9.

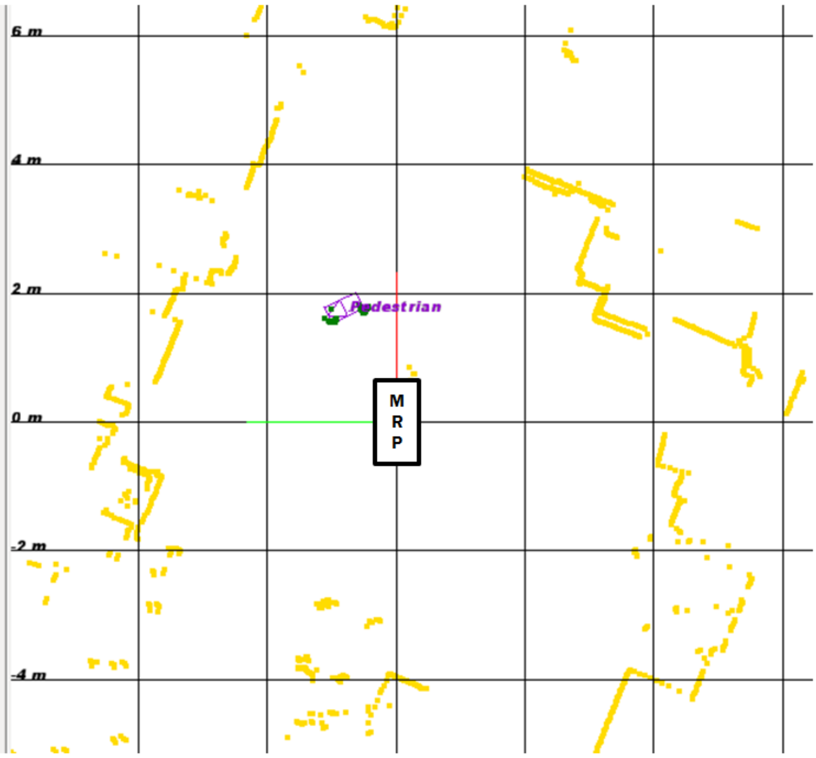

Figure 9: Detection of the human ("Pedestrian") walking around the MRP

For each detected human, several characteristics are estimated. The most import ones are:

- The position of the human.
- The absolute velocity of the human.

- The direction of movement of the human.
- The classification certainty that the object really is a human.

3.3.3. Dynamic MRP

If the sensor, that provides the measurement data for a tracking system, is moving, measures need to be taken to compensate the motion of the sensors. Thus, information about how the sensor is moving needs to be available. As described in Section 3.2.2, the laser scan measurement data itself is used to compute an estimate of the sensor's movement for each time step. Still, the motion of the sensors introduces further uncertainties that have influence on most parts of the tracking system, mainly the motion and background filtering and the prediction step of the Kalman filter. One key assumption from Section 3.2.2 is, that only detections, that are estimated to be moving by the motion filter, are initialized as new targets. One example for a situation for which a new target was initialized erroneously is shown in Figure 10. The green target represents a human that is tracked correctly and has already been classified as a human. The red target represents an unclassified target that was erroneously initialized due to a false detection of motion in the scan. As the laser scanner is a 2D sensor and only scans at one particular height, it is prone to the phenomenon of "mixed pixels", i.e. a mixture of different ranges for the same scan point, which is one of the main reasons for false detections.

Since false detections can not be avoided due to the nature of the sensor, further measures need to be taken to mitigate their effect on the overall tracking performance. Four our system, these include the post-classification of the targets as humans and a minimum walking distance/observation time.

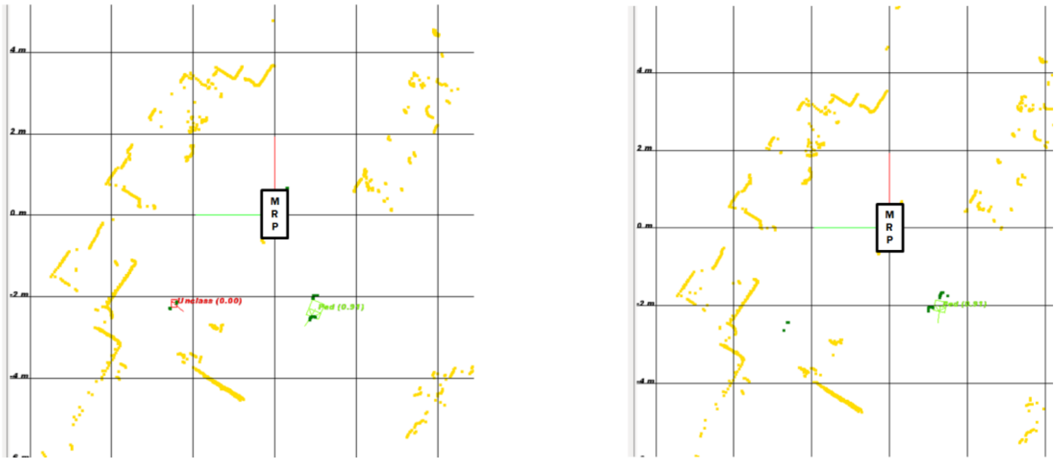

Figure 10: Example for false detections due to ego motion

3.4. Conclusion

The prototype of the 2D laser scanner tracking system has been developed with special focus on robustness and modular usage. First tests were carried out on a mockup of the MRP that show promising results regarding the tracking of humans in the closer environment of the MRP.

4. 3D BASED HUMAN DETECTION

Working in a collaborative environment where a robot is to close with human operators implies an augmented safety system. In the most cases these problems are solved closing the work area of the robot, but in that case is not possible describe as collaborative. For our prototype we have been testing an infrastructure that uses external cameras to see the work space and act in case when is required to avoid risky situations.

Figure 11: Camera configuration on Tecnalia laboratory

The configuration of our workspace monitoring system is composed with three KinectV2 cameras from Microsoft. These cameras are positioned creating an area where all the workspace is checked to get the best realistic status of the environment. This allows not only to detect dangerous situations, it also helps to get the best dynamic resolution on planning and execution for the different configurations

The multiple camera configuration implies a different approach that must take in account, the data received from they can be overlapped causing some inconsistences on the detected workspace. This overlapped information is useful in order to avoid blank spots produced by the objects. Also, it must be noticed that the individual cameras produce a FOV that is provided as a cone shape, this means that the visualization starts in one point and it opens gradually. This type of visualization is not very useful in order to delimitate the workspace of the robot, because of the irregular shape of it.

In order to solve these particularities of the usage of multiple cameras, a special filter has been created. This filter is able to get the point cloud generated from the cameras and combine it on a new point cloud where all the overlapped points are eliminated. On the other side, it also limits the valid point cloud to the workspace that was delimited for that work. So finally, the filter takes a multiple input point cloud and transform they on a combined and workspace limited unique point cloud.

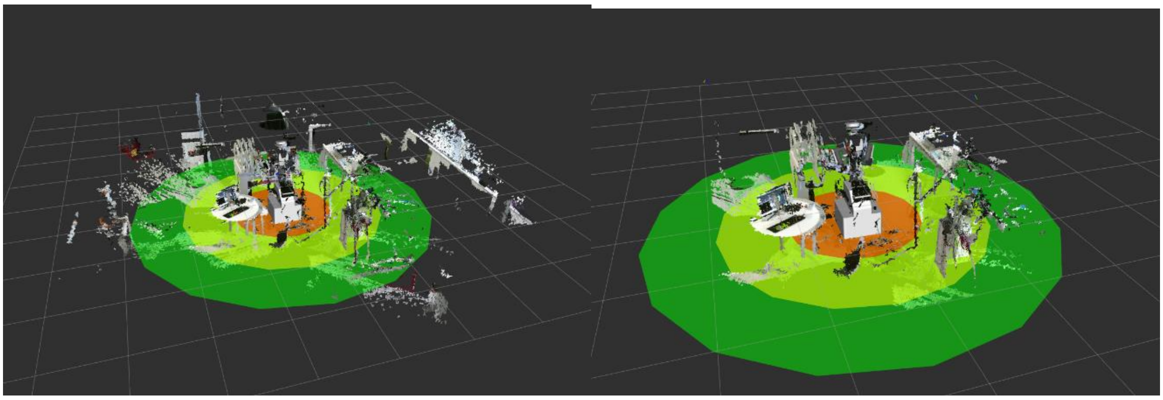

Figure 12: Point Clouds received from cameras, non-fused vs fused

The human detection is based on PointCloud Library (PCL). This information is retrieved from the cameras that control the workspace. For the human detection system two approach has been taken in order to get the best results. These two approaches are based on which information is used to use in the detection, the generated point cloud by each camera individually or the filtered and combined point cloud. Each approach has it owns advantages and disadvantages. The usage of the individual point cloud for each camera is more direct because it takes the original input, and it is also valid to track the human operators the enter and the exit from the workspace. On the other side, the usage of filtered point cloud is faster and creates and unique human detection source of output.

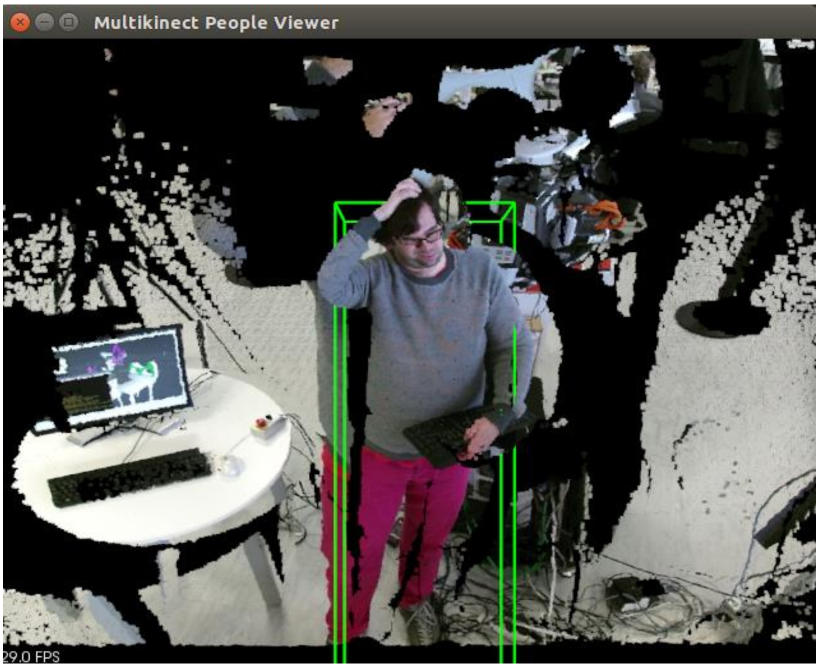

Figure 13: Detection of human by one of the cameras

In addition to this human detection system, some safety areas are included to have a more visual situation of the operator around the workspace. By default, the point cloud of the detected human show it as a new point cloud, but this cannot be enough to understand how risky is the current situation. With these markers is easier to understand where is the detected human and with it the behaviour of the robot in that cases.

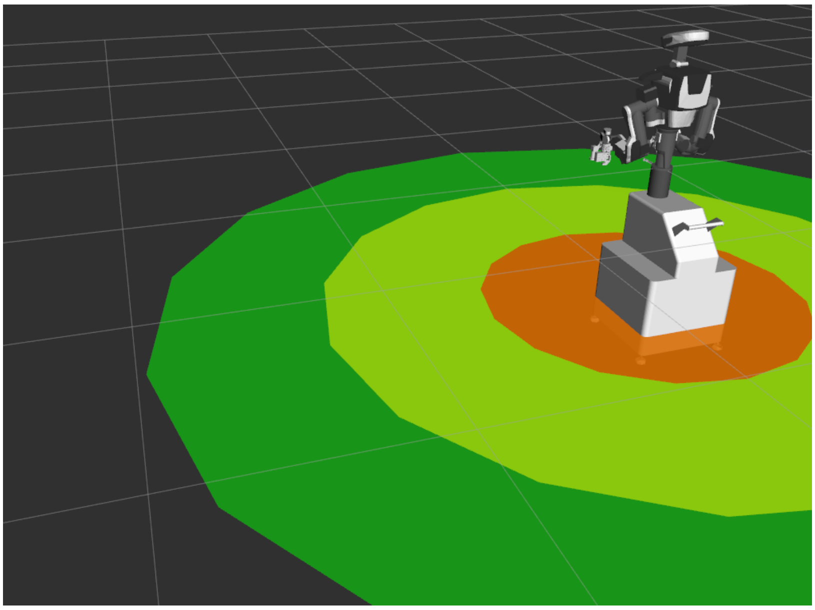

Figure 14: Workspace safety areas

4.1. Implementation

The system implementation is based on ROS indigo running on Ubuntu 14.04 using the ROS packages related with this version of the cameras. These packages are iai_kinect2 and libfreenect2, they are used to get all the required information related with the cameras. These packages are like the old kinectV1 OPENNI, but due to the sensors are not the same they work in other way.

The setup of the workspace monitoring is composed with three cameras that they are positioned shaping a triangle around the robot. In this particular case, the robot is a Hiro model manufactured by Kawada. This robot has a dual-arm configuration and it can manipulate multiple object using the two arms at the same time. Even so, the robot is fixed on the ground and it cannot move from it position. Taking these features of the environment into account, the workspace is fixed in the same position during all the tests. In the case of Thomas, where the safety area moves with the robot, this area can be changed in order to be more dynamic and changes in accordance with the movement of the robot.

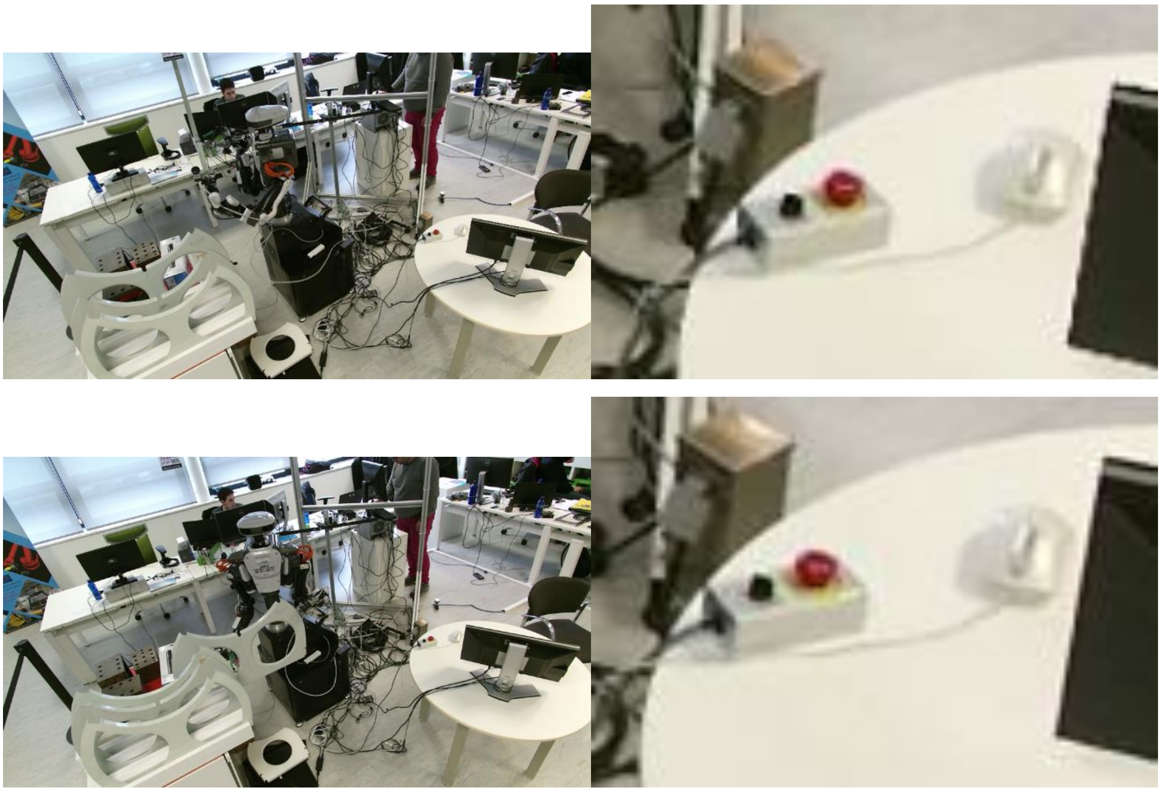

Figure 15: Quality of the camera comparison (SD, qHD, HD)

The cameras are focused directly to the centre of the safety area, where the robot is placed. The distance between the robot and the cameras are from 2 to 3 meters, and they are almost at the same high. The taken resolution from the cameras is qHD, 960px X 540px, in order to get the best balance between the quality of the point clouds and the lower overload of the system. Due to the cameras use USB 3.0 as interface, it was necessary add more USB 3.0 card to the controller PC of the workspace monitoring, this is related with the configuration limitations appeared with these cameras.

4.2. Testing

In order to get the most useful information about how the system works, several tests has been realized. These tests are based on how fast and reliable are the monitoring and detection gathered information. These test configurations were the following:

1. Only one camera: the same camera is used for workspace monitoring and human detection. In this case, the results are not enough to get a good safe status, a lot of blind spots and shadows exists in the workspace.

   In this experiment the overload of the computer is very low and the obtained result are good taking into account the limitations seen before. This is a very particular case where the same point cloud is used to make the workspace monitoring and the human detection.

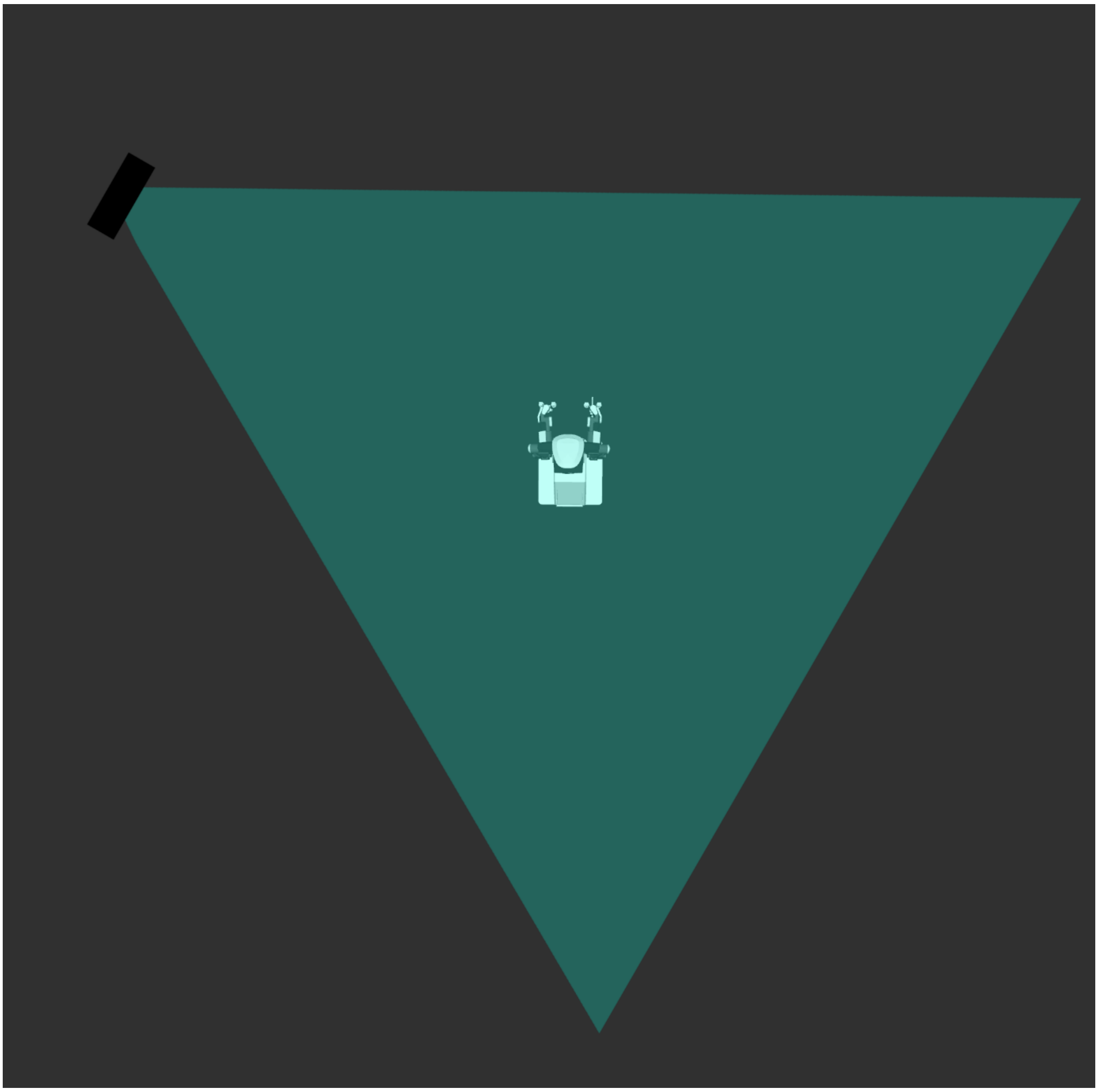

Figure 16: One camera configuration

2. Two cameras individually: the cameras work together to get the point cloud of the workspace, but the human detection is done by each camera. In this case, the workspace monitoring is more reliable due to the usage of the second camera to avoid blind spots. On the other hand, the human detection is divided in the two cameras, that produces multiple detection information that must be advertised in order to get the safest situation. With this detection in the two cameras, the input is direct from the output and they don't suffer any lag getting the point cloud. To monitor the workspace, the best way is put one camera in front of the other, this useful for the avoidance of the blind spots.

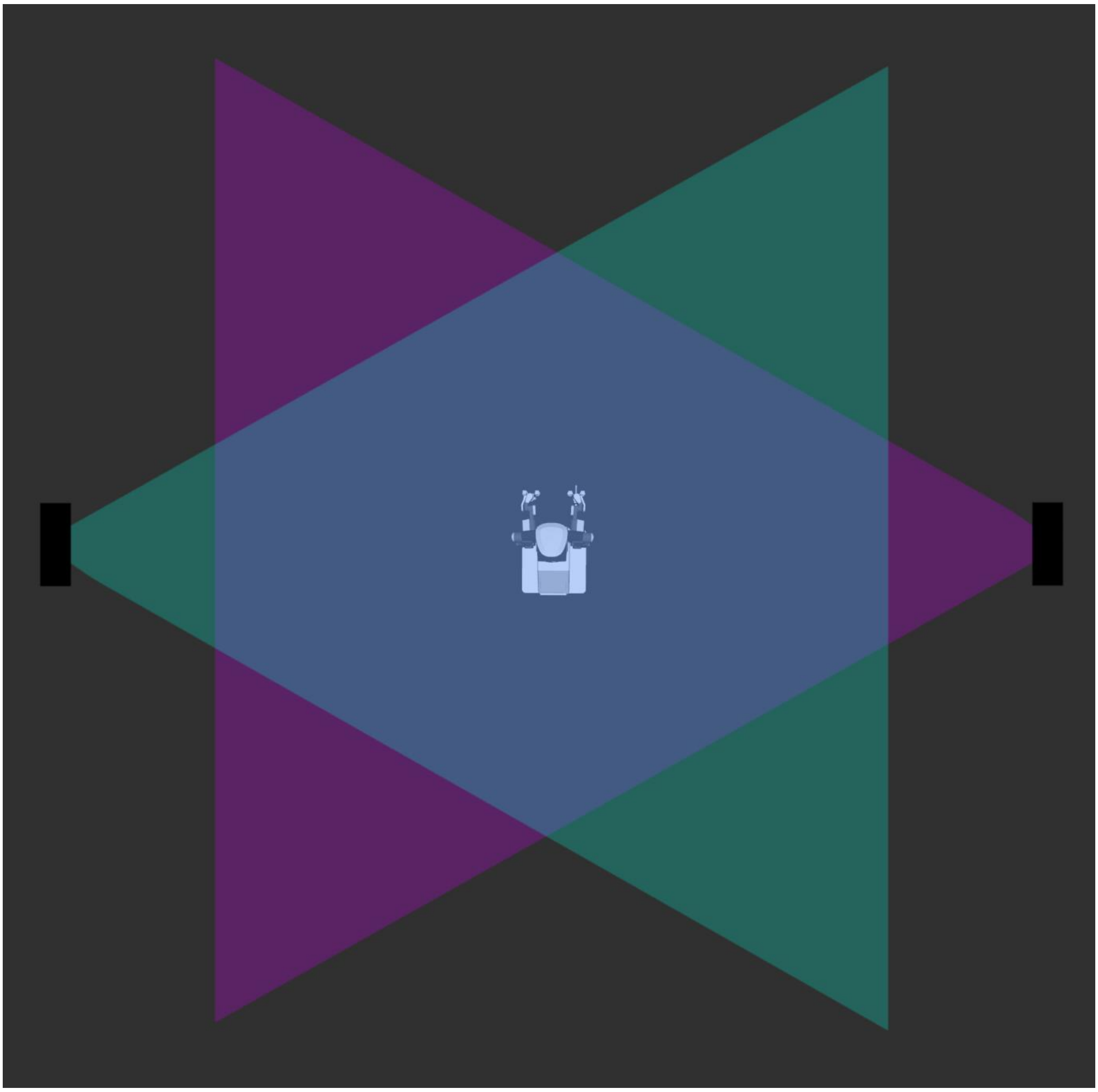

Figure 17: Two camera configuration

3. Two cameras together: the point cloud of the cameras is used together for the workspace monitoring and for the human detection. In this case the information for the human detection is received from the fused point cloud of the cameras, this provide a unique input for the system but is not as fast as the individual input, because some delay is added when the data is fused. As in the previous experiment, the best position for the cameras is one in front of the other.
4. Three cameras individually: as the case of the two cameras individually, the cameras work together to monitoring the workspace, but divided for human detection. This case has a better improvement for the workspace monitoring due to the better positioning of the cameras, but the same problems for the human detection related with this division stays.

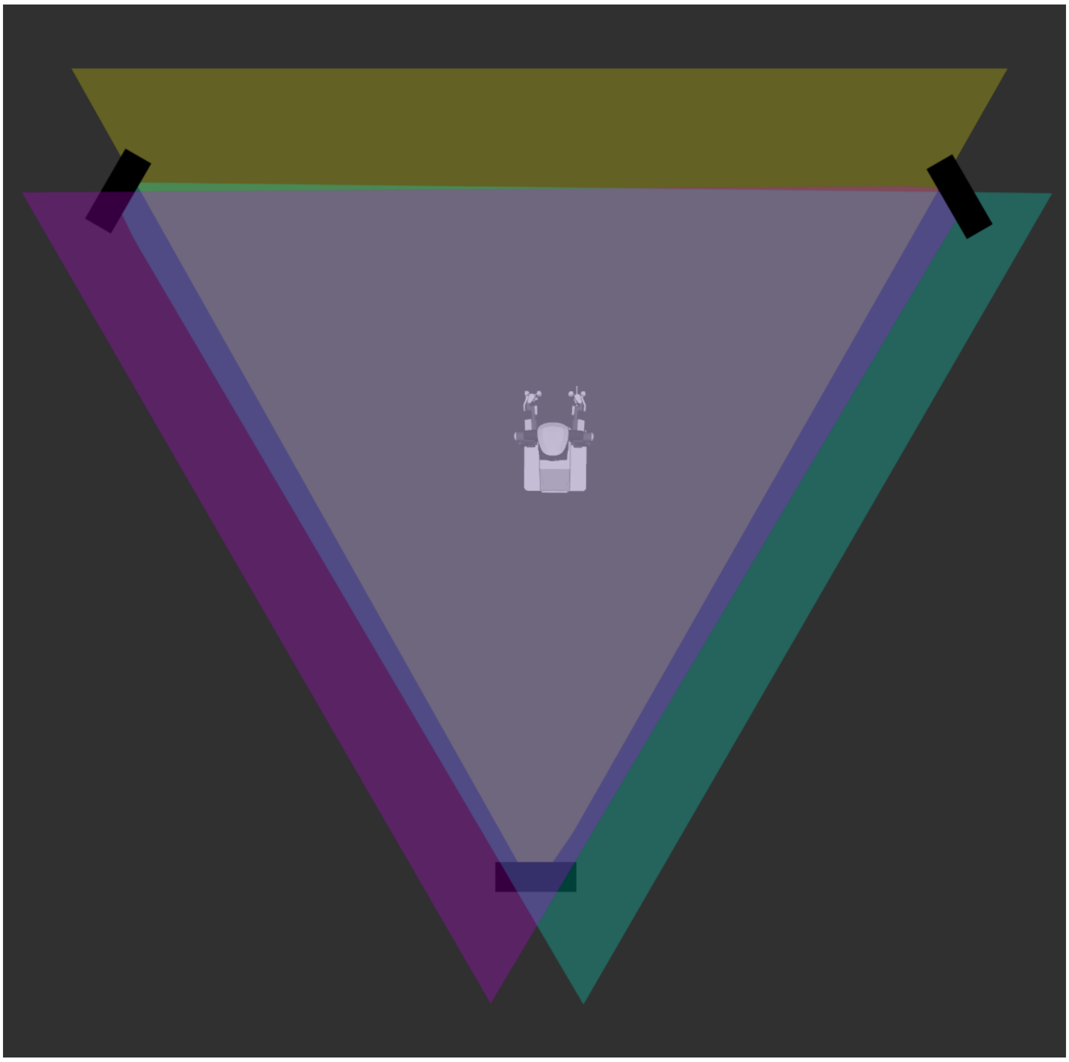

Figure 18: Three camera configuration

5. Three cameras together: like in the test with two cameras together, the fused point cloud generated from these three cameras is used as input for the human detection. The overload of the system to work properly with this amount of generated information induce a very several delay situations on the human detection. This delay can be seen when echoes from the detected human keep in on the detection area even when he has leave it.

According to these multiple test scenarios, the best scenario is the usage of more cameras is the best option in order to get the most possible information from the environment. But for the point of view of the human detection, the individual detection is more effective than the use of the fused point cloud. In all the cases, when a human is detected, the warning information works properly even if the visualization has problems cleaning the point cloud.

5. HUMAN-ROBOT INTERACTION

In the human detection and tracking mechanisms are also included the HRI technologies that will be used to facilitate the communication of the human with the MRP and the overall system, including him/her in the execution loop. As already described in D2.1 “Perception for HR interaction – Design” a number of sensors, buttons and wearable devices will be introduced and integrated to the overall architecture, sending different signals to the station controller and the robot resources available in the shopfloor. In the following sections the initial prototype of the smartwatch application is described, since the initial prototype of the gesture detection application has already been described in D2.1 “Perception for HR interaction – Design”.

5.1. Wearable device application

As explained in the previous deliverable, namely D2.1 “Perception for HR interaction – Design”, in THOMAS project a set of human interfaces will be introduced. These interfaces are divided as follows:

- Direct robot control interfaces
    - Contact/proximity sensors
    - Enabling devices for manual guidance operation
    - Physical buttons (e.g. for emergency cases)
- Human Machine interfaces
    - Wearable devices (e.g. smartwatch)
    - Physical buttons (e.g. to reset the cell)

This deliverable focuses on the development of an initial prototype of human machine interface and more specifically the application of a smartwatch wearable device.

5.1.1. Smartwatch application

Smartwatches are small devices that combine the small and easy to use menu with processing and connecting power of a small computer. Their size makes them ideal devices for an operator who doesn’t want to wear or carry heavy or bulky electronic devices, being at the same time, constantly connected with the central station controller and exchanging information. In the first prototype, specific buttons and lists of data have been developed, that will enable the operator to:

- Connect with the station controller
- Select the ID of the human resource that wears the smartwatch
- Declare if and when a task has been completed, from the operator side
- Take over a specific task from a specific resource, that are selected through dynamically updated lists, when needed

The application has been developed in Android Studio, using API 25 and higher. An overview of the different interfaces that implement the aforementioned functionalities is presented in the following figure (Figure 19).

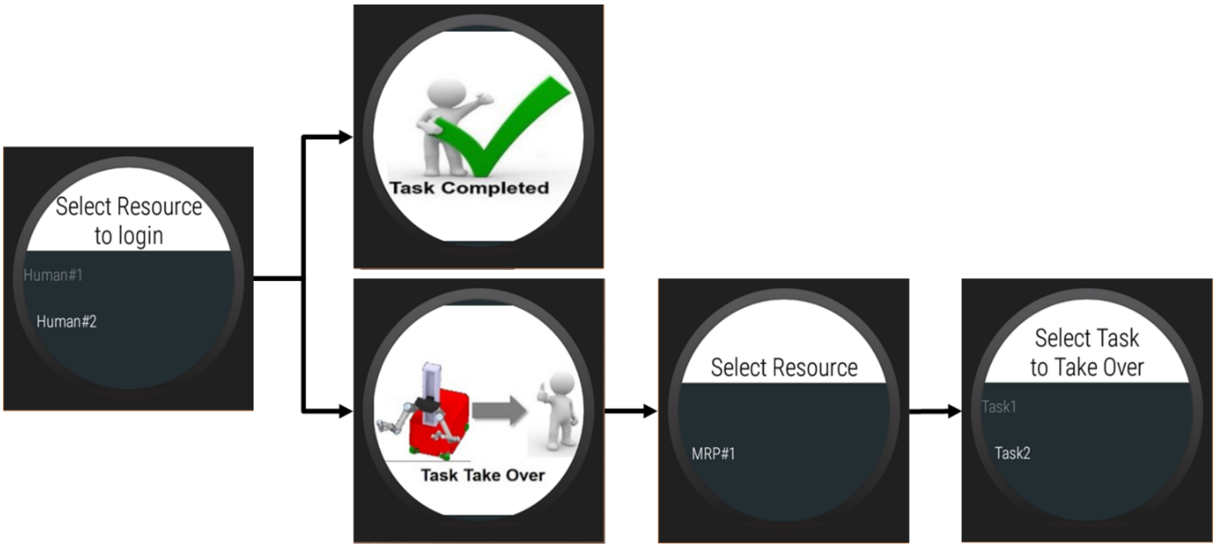

Figure 19: Smartwatch interfaces

When the application is activated, the application connects to the station controller and using a predefined service, request the list of available human resources in the system. This information then is displayed in the form of list to the operator, who selects the ID that belongs to him/her. This step is necessary, because it helps the application recognize the operator ID of the human that is using the smartwatch, in the messages that the application receives from the different topics. In this way, the application would handle only the messages that are intended for the specific operator, ignoring the rest.

After selecting the operator ID, the application enters the main menu which contains the buttons that activate the different functionalities of the application. In the first prototype the following buttons have been implemented, with plans of further enhancing them in the next versions:

- Task Completed button
- Task Take Over button

For the first button, the operator uses it to let the system know when he/she has finished with his/her application. Upon receiving a status message from the station controller, that contains information about the status of an operation, the smartwatch checks if this message is destined for the specific operator. If yes, then stores the operation ID to a buffer. When the operator finishes his/her operation, he/she presses the “Task Completed” button and calls a service which contains as arguments the operation ID, obtained from the status message, the resource ID, obtained in the initialization phase and the status completed code (hardcoded in the system). The response from the service is a Boolean variable to les the application know for the successful completion of the operation.

For the second button, more steps are followed. When the operator presses the “Task Take Over” button, a service is called that provides to the smartwatch the robot resources that are running in the shopfloor. Upon the selection of the resource by the human, a second service is called, using as an argument the resource ID of the previous selection. The second service provides a list of the tasks that have been assigned to this resource. Finally, once the operator chooses the task that he wants to take over, a third service is called providing the operator ID, the robot resource ID and the task ID as arguments. The response from this service is a Boolean variable that informs the application if the takeover functionality has been executed correctly or not.

The aforementioned message exchange sequence has been visually described in the following figure (Figure 20).

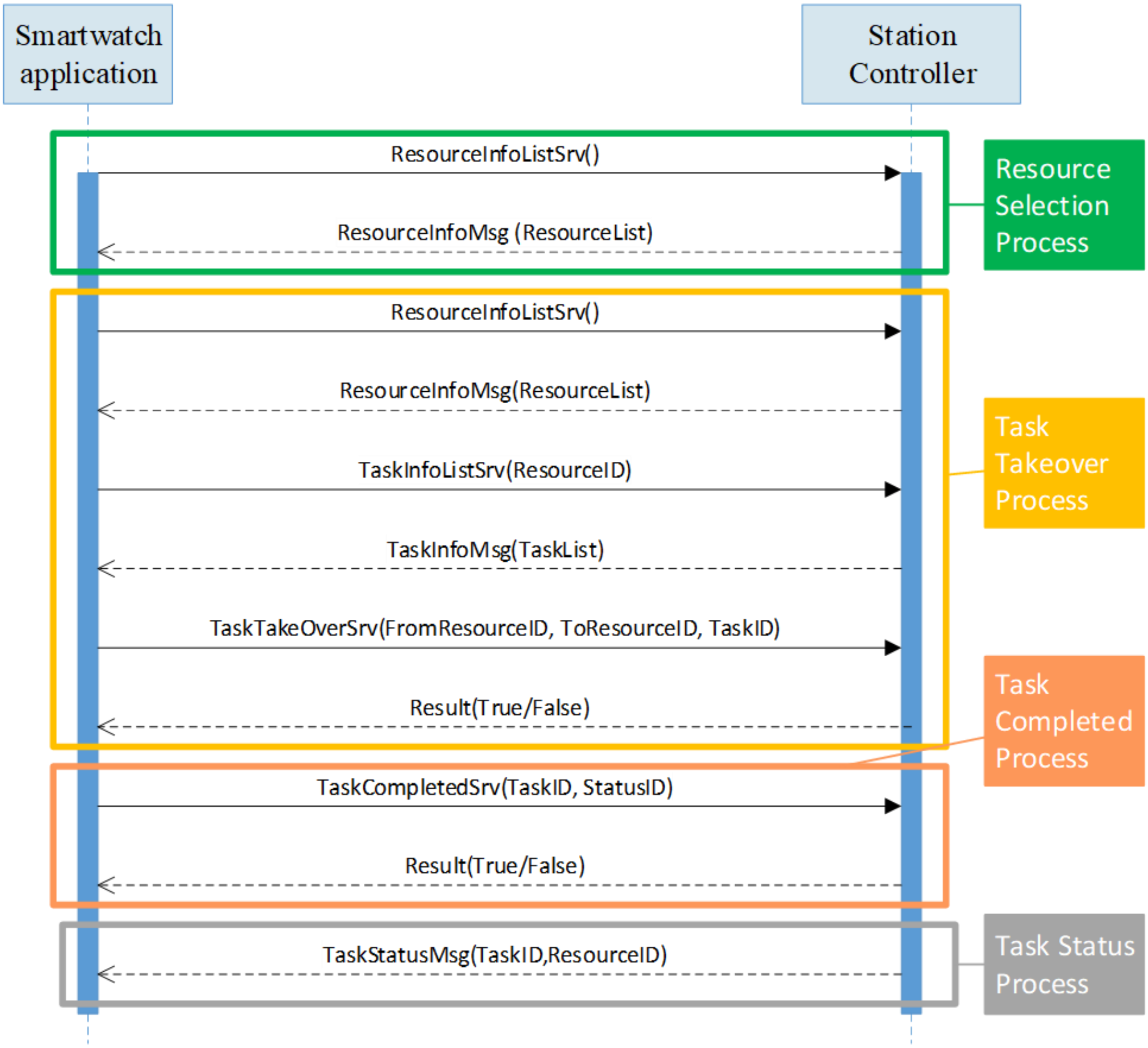

Figure 20: Smartwatch application – Message exchange sequence diagram

5.1.2. Integration in the THOMAS overall system architecture

An important aspect of the smartwatch application is its integration to the THOMAS system architecture. The application is depended on the station controller to which is connected, receives messages and uses its services (Figure 21). In other words, the human machine interface communicates with the station controller to exchange information depending the case.

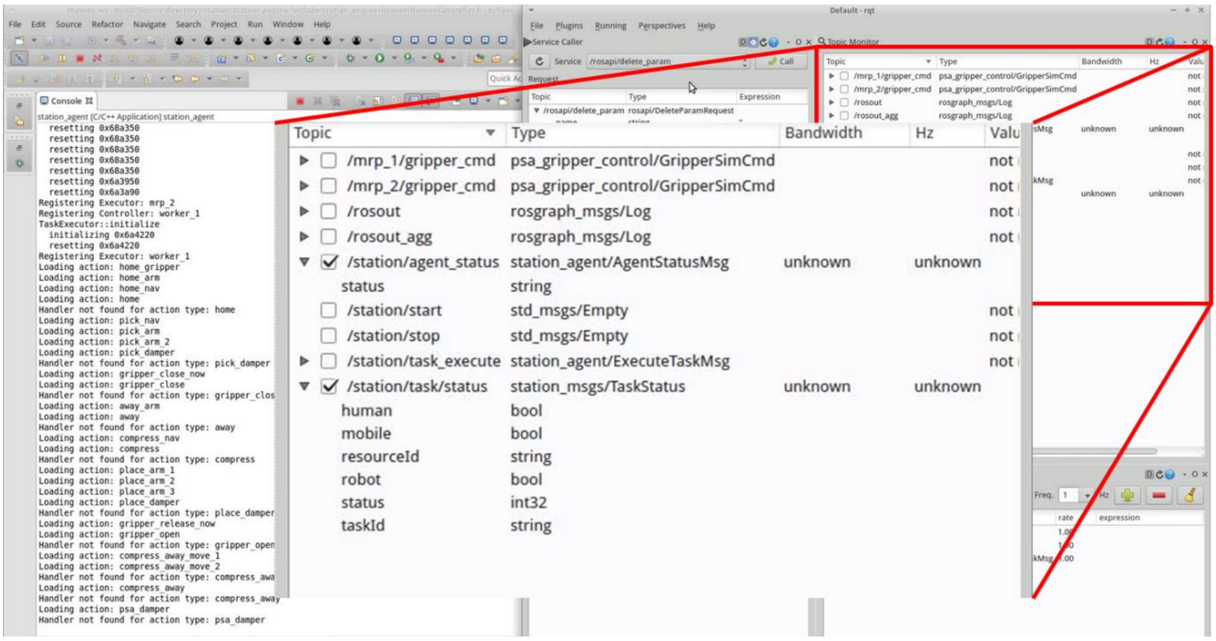

Figure 21: Station Controller with the available topics

In the following table are described the services exposed by the station controller and their type.

Table 1: Main services used by smartwatch application

Topic	Description
/ResourceInfoListSrv	Request the list of available, both human and robotic, resources
/TaskInfoListSrv	Request the list of assigned tasks to a specific resource
/TaskTakeOverSrv	Switch a specific task from a robot to a human resource
/TaskCompletedSrv	Inform the station controller that a specific task assigned to the human is completed

In addition to this, the application subscribes to the “Status” topic which informs the subscribed devices about the status of the different tasks that are running in the execution list. The status is declared in the form of enumerator. The description of the different status IDs are provided in the following table.

Table 2: Description of status IDs

ID	Description
0	NOT_STARTED
1	STARTED
2	PAUSED
3	RESUMED
4	COMPLETED
5	CANCELLED
6	ERROR

A visual representation of the HRI integration architecture is provided in Figure 22. Both the robot's controller and the smartwatch application are connected on the same network with the central PC where the station controller is running, exchanging information. The smartwatch application has been connected to the station controller through ROS, using the rosbridge API which contains a protocol that allows applications in different devices, that don't contain a ROS Master, to connect to the ROS Master using simple TCP/IP connection and custom-formatted messages.

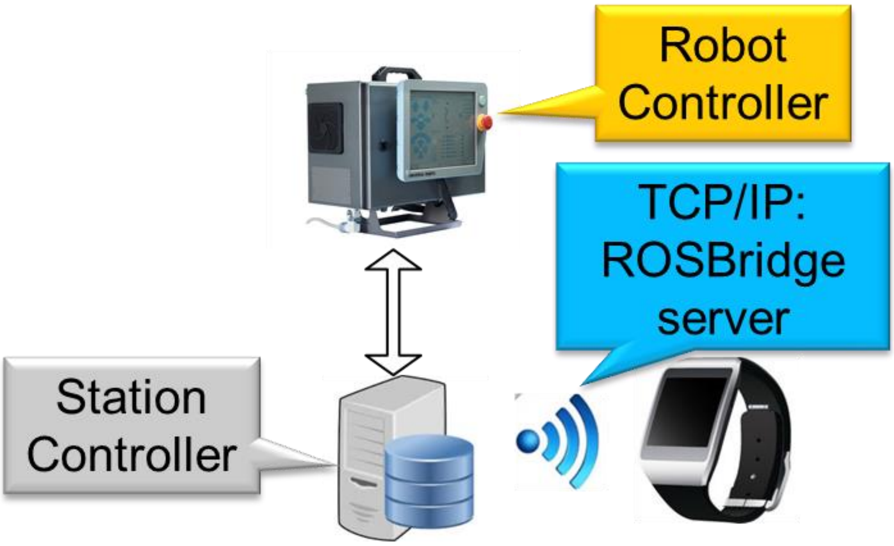

Figure 22: HRI integration architecture

5.1.3. Prototype implementation

The aforementioned application has been tested in the Laboratory for Manufacturing Systems (LMS) premises, using a Motorola Moto360 2nd edition smartwatch device in which the application was running. In addition to this, a prototype of the station controller was running on a PC, which contained a demo task sequence. Both devices, smartwatch and PC, were connected on the same network. The steps that were tested are described in the following figure (Figure 23) and tested all the functionalities described in the previous section. Initially the operator "signed-in" selecting his ID from the provided list. Then he took over a task from a robotic resource (Figure 23) and finally he pressed the task completed button to inform the system that the task has been completed and to proceed to the next one (Figure 24). The station controller provided the services and topics used by the smartwatch and forward as expect the commands from the operator to the rest of the system.

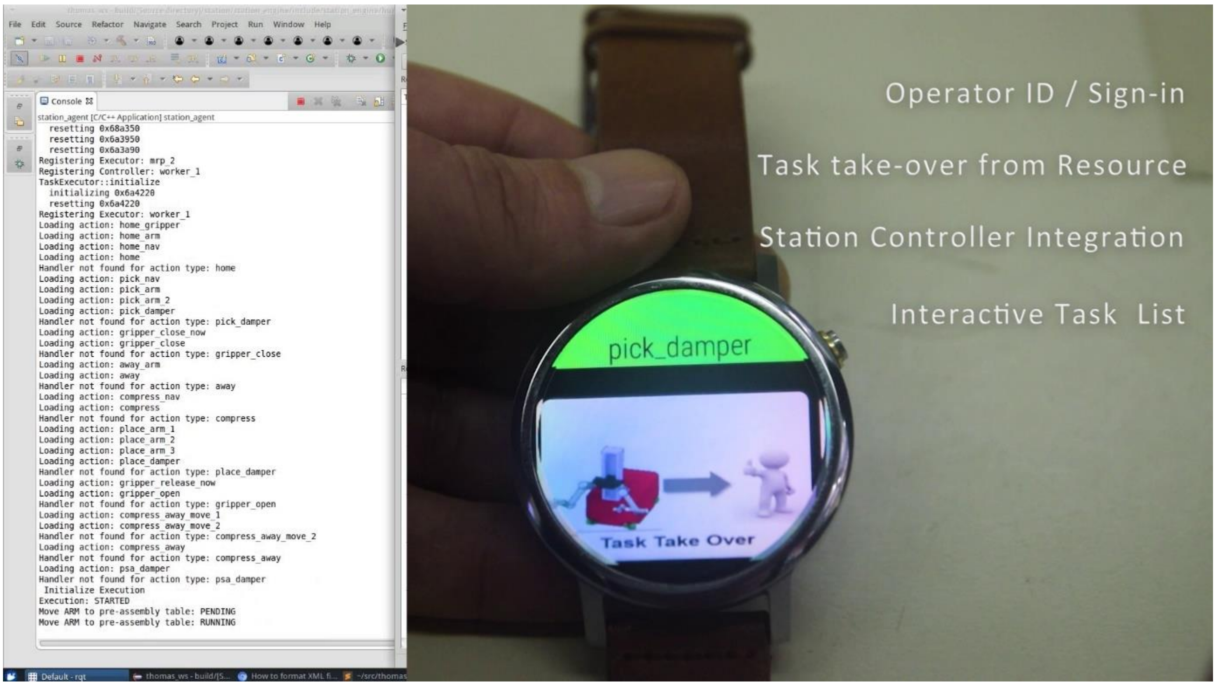

Figure 23: Smartwatch application integrated to station controller – Testing task takeover button

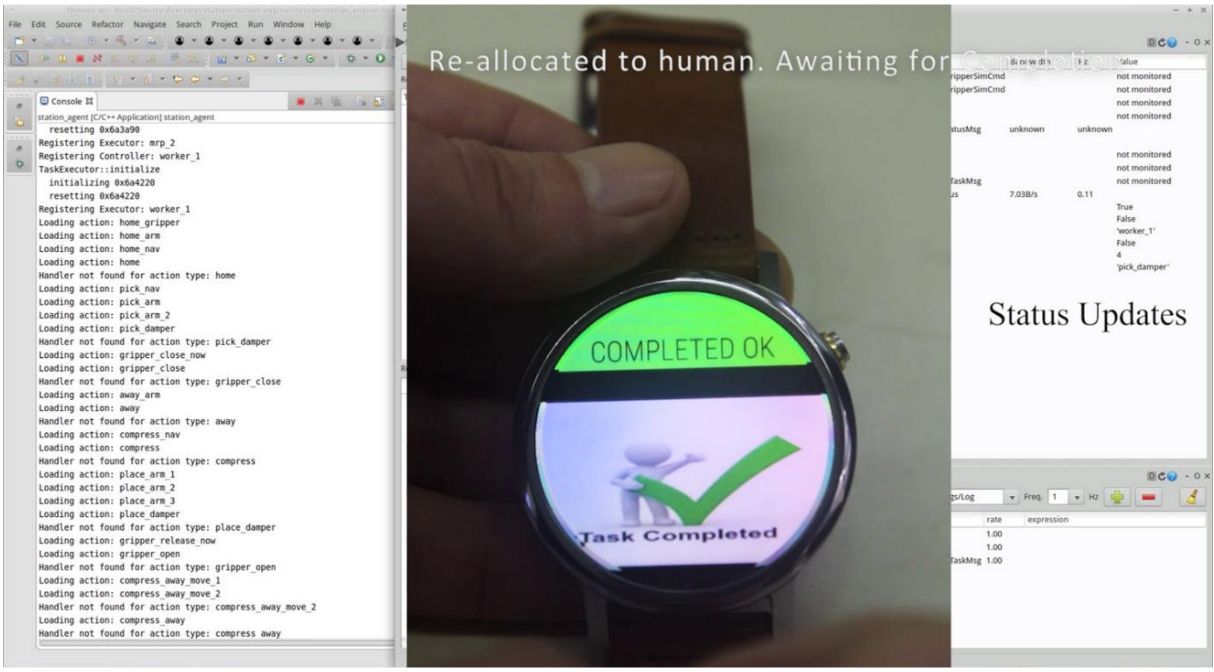

Figure 24: Smartwatch application integrated to station controller – Testing task completed button

In the following figure is displayed the update of the status topic that takes place when the operator presses the task completed button. As shown, the status ID from 1 (operation STARTED) is turned to 4 (operation COMPLETED). With this information the station controller proceeds dispatching the next operation to the assigned resources.

5.2. Augmented Reality

Another important part of the Human Robot Interaction explained in D2.1 “Perception for HR interaction – Design” is the visualization of information to the human operators. The functionalities that will be included in the AR application are as follows:

- Visualization of assembly instructions in text format
- Visualization of assembly instructions in 3D model representation of the parts
- Visualization of robot’s current execution task
- Visualization of alert messages for potential hazards on the shopfloor
- Visualization of general information of the production line status

5.2.1. AR Operator Support application

The AR application has a more passive behaviour compared to that of the smartwatch. Its main target is to receive information from the station controller depending the status and visualize it in operator’s field of view, without distracting him from his main job.

Initially, when the application starts, it connects to the station controller and subscribes to predefined topics. Using a default user ID, which later won’t be hardcoded in the code but would be dynamically assigned by each operator, the application can distinguish between the different messages that are published in the topic, which refer to its operator and need to be visualized to him and which are not. In this first prototype, the assembly instructions, the 3D model representation of the parts and the alerts for potential hazards on the shopfloor have been implemented and explained below.

When the robot is starting to move, the station controller sends the corresponding command to robot’s controller and in parallel publishes to a specific topic an alert in the form of text that the robot starts to move, to let the operator know of robot’s activity even if the operator is not looking at it.

When the robot completes its actions and the operator needs to perform an operation, instructions are sent to the operator either in textual format or in 3D model representation or both. The operator is informed about the imminent activity that needs to be performed and when he/she finishes, presses the corresponding button on the smartwatch, as explained in the previous section, and the visualized information disappears.

The message exchange sequence has been visually described in the following figure (Figure 25).

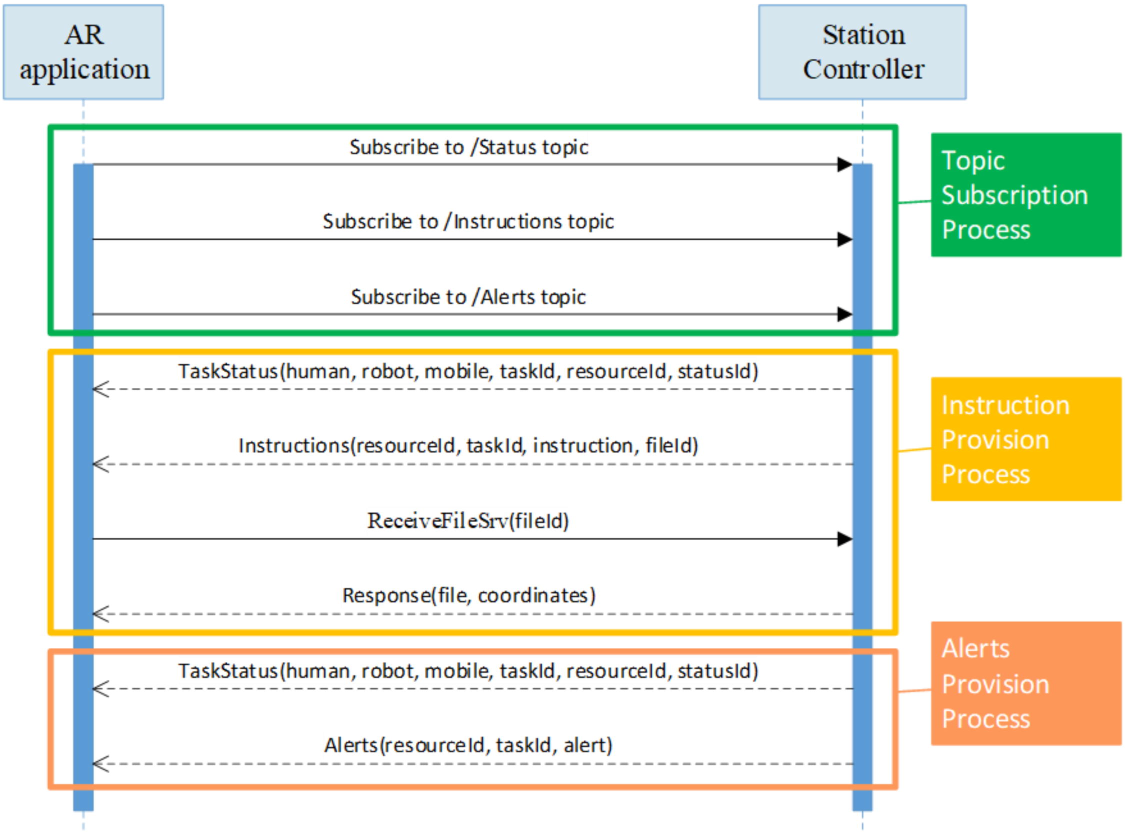

Figure 25: AR application – Message exchange sequence diagram

5.2.2. Integration in the THOMAS overall system architecture

In order for the AR application to function properly, it should be integrated to the THOMAS system architecture. The application is depended on the station controller to which is connected, receives messages and uses its services, depending the case. In the following table is described the service exposed by the station controller and its type that is used by the AR application.

Table 3: Main service used by smartwatch application

Service	Description
/ReceiveFileSrv	Request the file and its coordinates that need to be visualized based on the fileId received from the /Instructions topic

In addition to this, the application subscribes to the following topics.

Table 4: Main topics used by smartwatch application

Topic	Type	Description
/Status	station_msgs/TaskStatus	Topic to inform the subscribed devices about the status of the different tasks that are running in the execution list
/Instructions	station_msgs/Instructions	Topic to provide to the subscribed devices the instructions assigned to the specific operation
/Alerts	station_msgs/Alerts	Topic to provide to the subscribed devices informative messages for potential dangers in the cell

The description of the different status IDs, for the /Status topic, is provided in the following table.

Table 5: Description of status IDs

ID	Description
0	NOT_STARTED
1	STARTED
2	PAUSED
3	RESUMED
4	COMPLETED
5	CANCELLED
6	ERROR

A visual representation of the AR application integration architecture is provided in Figure 26. Both the robot's controller and the AR application are connected on the same network with the central PC where the station controller is running, exchanging information. Similar to the smartwatch application, the AR application has been connected to the station controller through ROS, using the rosbridge API which contains a protocol that allows applications in different devices, that don't contain a ROS Master, to connect to the ROS Master using simple TCP/IP connection and custom-formatted messages.

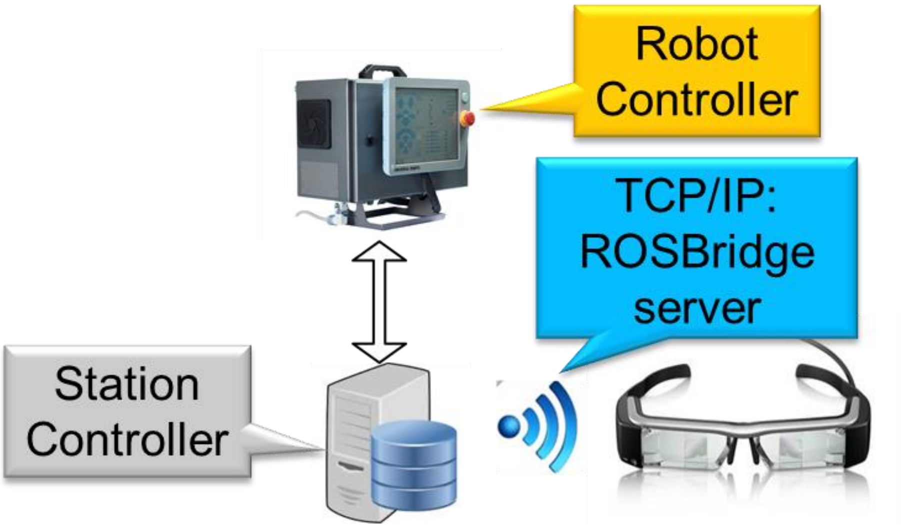

Figure 26: AR application integration architecture

5.2.3. Prototype implementation

The aforementioned application has been tested in the Laboratory for Manufacturing Systems (LMS) premises, using a EPSON Moverio BT200 device in which the application was running. In addition to this, a prototype of the station controller was running on a PC, which contained a demo task sequence. In Figure 27 are displayed the steps followed during the execution of a human operation. The station controller publishes through the /Status topic that a human operation has started and through the /Instructions topic the text, that would appear in operator's field of view, and the fileId of the 3D model that need to be visualized. Upon receiving the fileId, the AR application uses the /ReceiveFileSrv to get the model file and the coordinates that will be used, to visualize the model based on the marker. When the operator performs the indicated actions, presses the corresponding button on his smartwatch and a new /Status message is published. The AR application translates this message and if the status of the previously started operation is COMPLETED, then removes the overlaid information.

Similarly, to the above approach, in the case no 3D model is available, a textual message only appears in operator's field of view, letting him know what to do. The rest of the actions and message exchange are the same with the aforementioned steps (Figure 28).

Last but not least, when a robot operation begins, the station controller again publishes a message to the /Status topic to inform the resources about the new operation, but it also publishes to the /Alerts topic that a predefined message that is visualized, flashing, in operator's field of view as shown in Figure 29. This informs the operators that a robot operation is about to begin and increases his/her awareness of the imminent movement, even if the operator is not looking directly the robot.

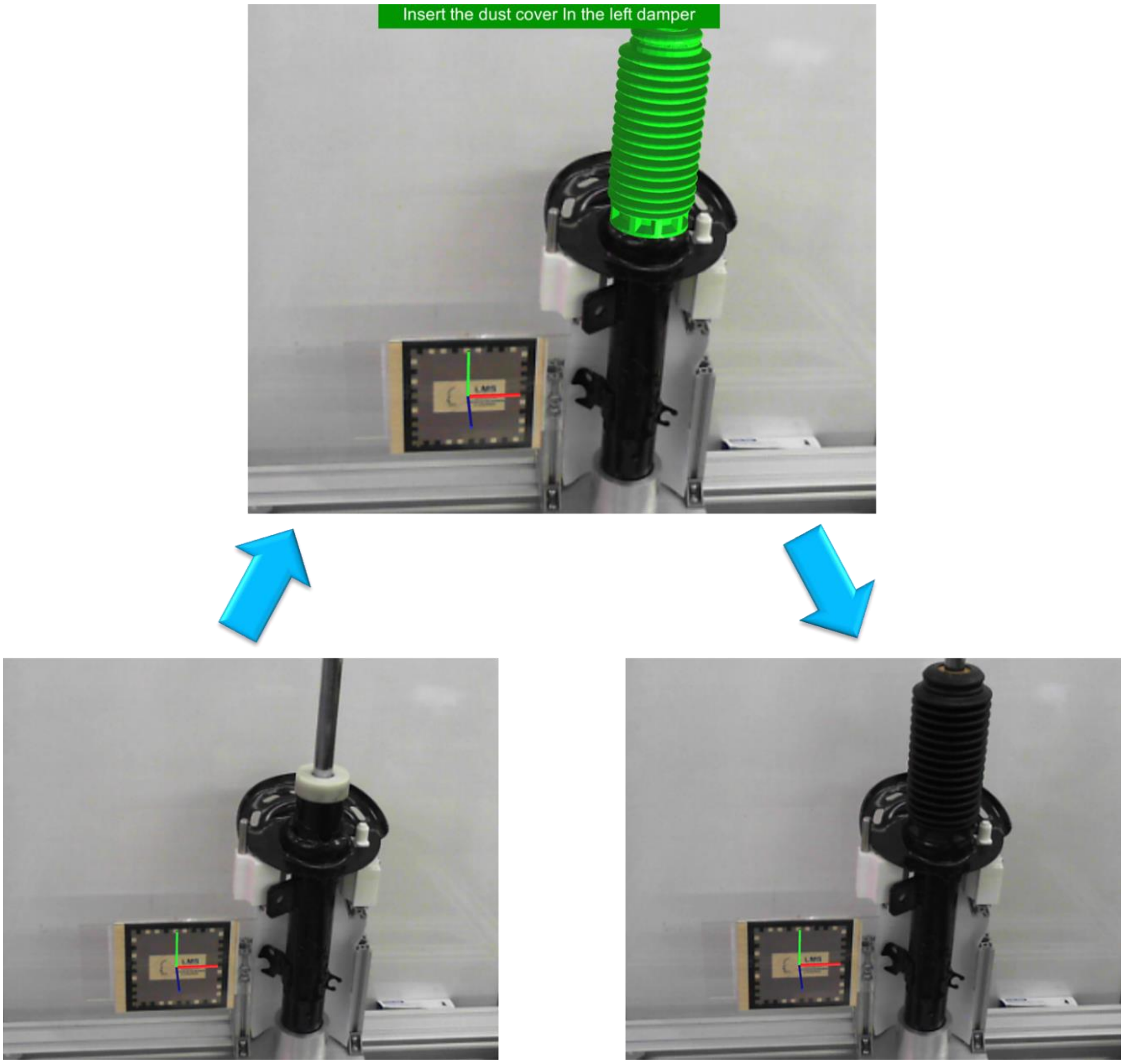

Figure 27: Assembly operation steps – Instructions and 3D model

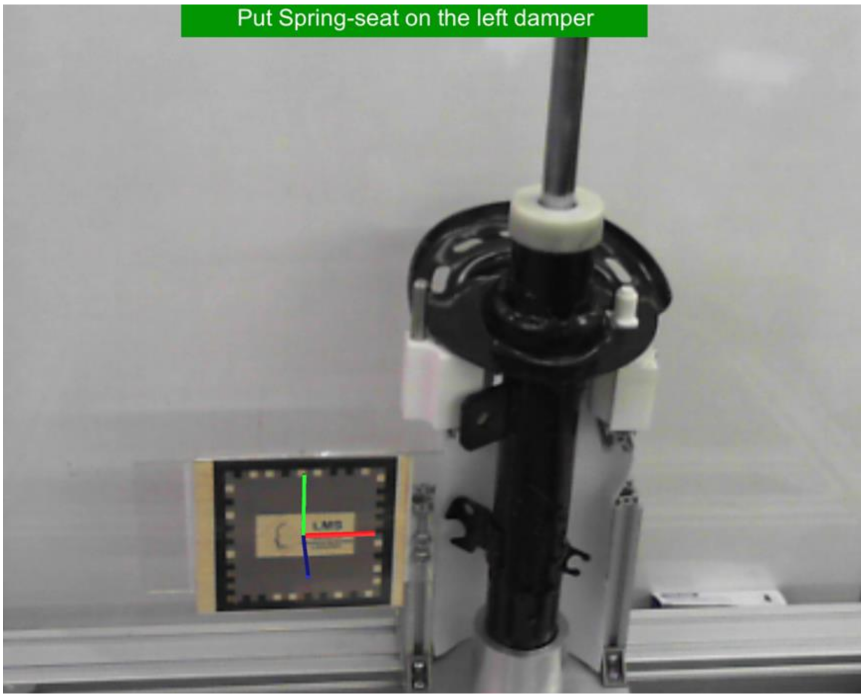

Figure 28: Assembly operation– Instructions only

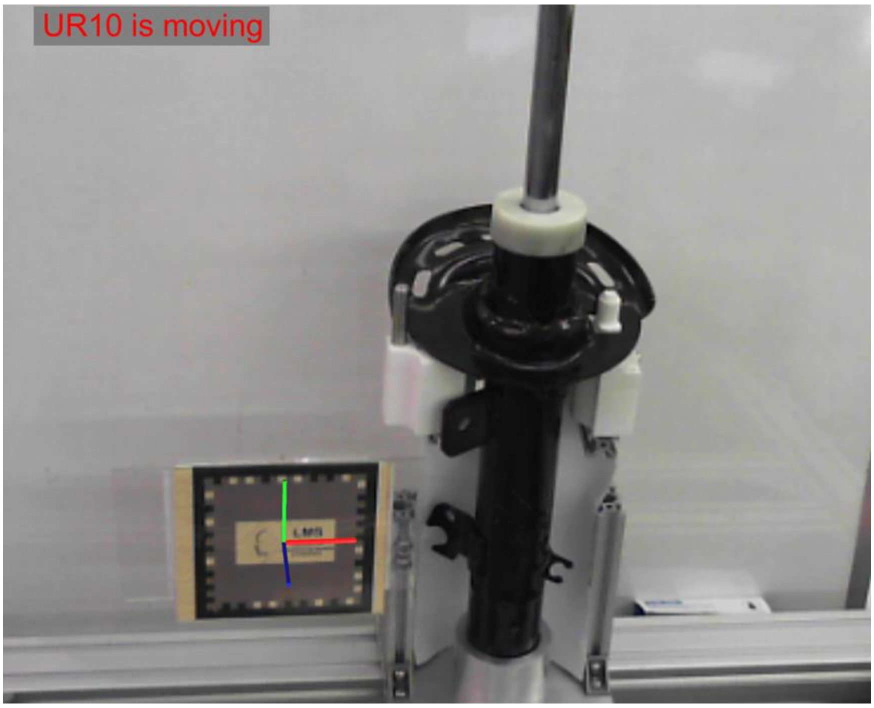

Figure 29: Alert message

5.3. Human Gestures / Posture Recognition for direct MRP guiding

Under this section the prototype application of human, gesture based, interaction with MRP will be presented. In particular, the developed application is responsible for detecting human presence as well as his posture and translating it into robot command allowing the user to directly instruct the MRP. In that way, the human – MRP "communication" during task execution is enabled. The design of this application as well as an initial prototype version has been documented in D2.1 that was submitted on M12 of the project. In this document, an enhancement of the application providing extra features will be reported.

5.3.1. Gesture recognition application

Human body is the main input that provides all the data for the human – MRP interaction. The developed application uses Kinect for capturing human kinesiology. In addition, extra sensors such as stereo cameras have been strategically positioned on the MRP and their input may be exploited in the next prototypes for better coverage of the area. The steps that the gesture detection follows are described below:

- **Human presence detection** – Using OpenNI and OpenNI Tracker libraries, raw depth data are captured from the Kinect and translated into TF frames accordingly. These libraries facilitate the gesture handler module receive 15 frames from different joints of the body and calculate their coordinates based on any other frame in the list. So, when the operator stands in front of the Kinect, these libraries start publishing his/her joint coordinates in the TF tree.
- **Human body type classification** – The main functionality of the gesture handler in order to detect which gesture is performed by the operator is based on the calculation of the relative position of the exploited frames to each other. Nevertheless, there should be some sort of classification of the detected skeletons, since there are different body types, to three main categories, namely small, medium and large, to increase the efficiency of the detection algorithm. In order to do this, the operator stands in front of the Kinect, with his/her hands extended in a T-shape position and for 10 seconds the algorithm measures the distance between his 2 hands. Based to human anatomy, the height is about the same with the arm span and this is why this gesture was selected for this measurement. After this process takes place, the operator is classified to one of the above categories:
    - Small - The calculated distance is less or equal than 1450 mm
    - Medium - The calculated distance is between 1450 and 1649 mm
    - Large - The calculated distance is more than 1650 mm

Figure 30: Stance for Body classification

- **Gesture recognition** – As already described in D2.1 “Perception for HR interaction – Design”, in the initial demo of the gesture handler, the following gestures have been recognized
    - Move left arm up
    - Move left arm down
    - Move left arm left
    - Move left arm right

  This first prototype has been enhanced with controlling both robot arms, based on the gesture performed as well as with the skeleton classification. The following table summarizes this assignment between human gestures and MRP movement:

Table 6: Gesture - Robot arm movement assignment

Hand gesture	MRP arm movement
Left hand fully extended upwards	Left arm moves up
Right hand fully extended upwards	Right arm moves up
Left hand fully extended left	Left arm moves left
Right hand folded on the elbow to the left	Right arm moves left
Left hand folded on the elbow to the right	Left arm moves right
Right hand fully extended right	Right arm moves right
Left hand folded on the elbow downwards	Left arm moves down
Right hand folded on the elbow downwards	Right arm moves down
Both hands extended down	Idle

The operator initially should perform the “Idle” gesture to notify the system of the human presence and put the robot in “Interaction mode” using gestures. After the robot pauses any performed activity, it is ready to receive gesture commands from the operator.

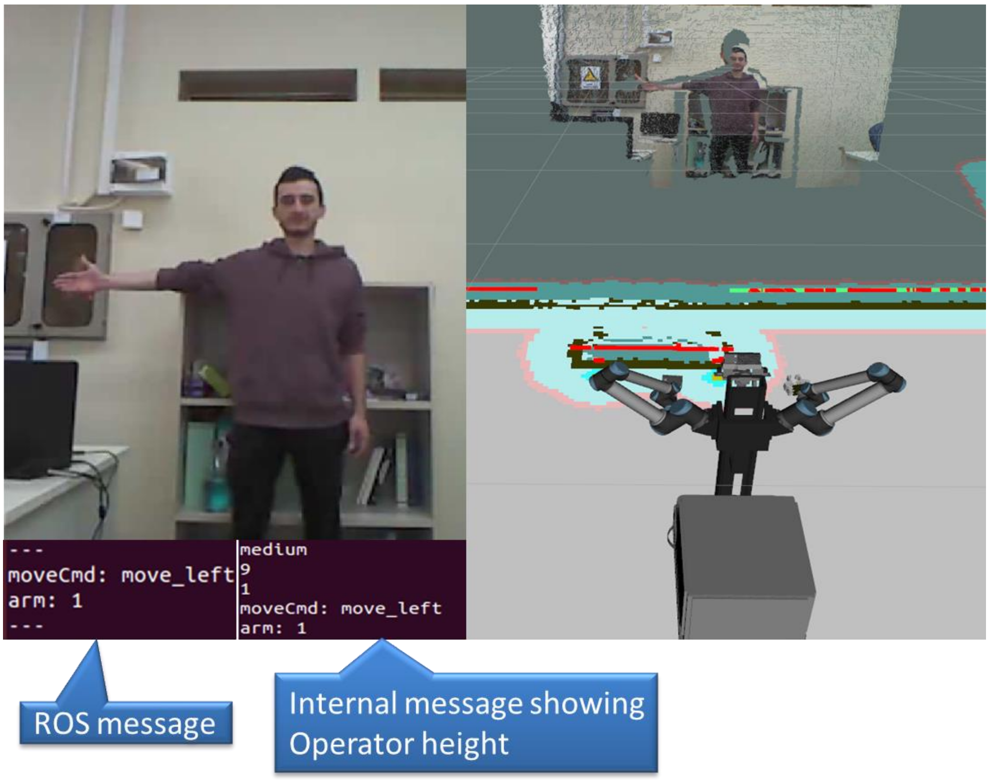

Figure 31: Medium Size Operator Gesture

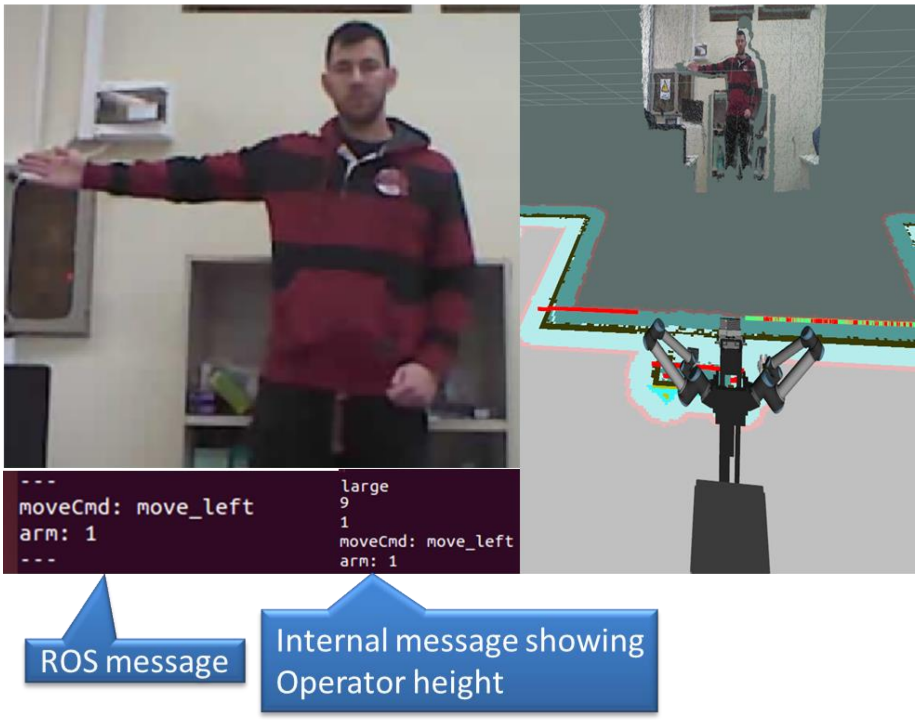

Figure 32: Large Size Operator Gesture

- **Recognized gesture message creation** – After recognizing the gesture performed by the operator, a predefined message is constructed and sent to the station controller for execution. During this phase, certain mechanisms have been introduced to eliminate false positive recognitions, as well as accidentally sending multiple times the same gesture to the station controller.

Figure 33 Ros Message

- **Exiting Interaction Mode** – When the operator finishes, leaves Kinect field of view and the gesture handler sends an exit command to the station controller which continues with the execution of the task flow

5.3.2. Integration in THOMAS overall system

Crucial to the gesture detection mechanism is the communication with the THOMAS architecture. The application is based on the station controller in order to consume the exposed topics and services. The published messages will be made available to the master node who will notify anyone who has subscribed to the topic. The station controller will receive these messages and interpret them into further commands. In the following table is presented the topic used by the gesture handler

Table 7: Main topics used by gesture recognition application

Topic	Type	Description
/GectureCommand	station_msgs/TaskStatus	Topic to inform the station controller about which gesture is performed

A visual representation of the gesture application integration architecture is provided in Figure 34. The gesture recognition application runs on the same PC where the station controller is setup. Similar to the other interaction mechanisms, robot's controller and the station controller are connected on the same network enabling the later one to send commands to the first one.

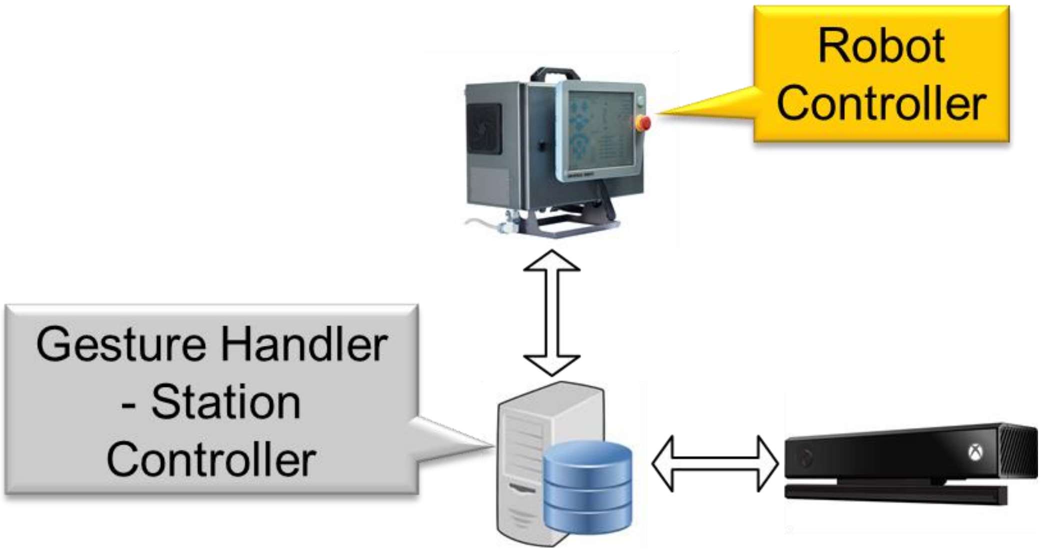

Figure 34: Gesture application integration architecture

6. CONCLUSIONS

As has been shown, the prototypes for the human detection and tracking system remain in an early development stage but showing good overall performance. So far, focus has been put mainly in achieving working systems, with less emphasis in reliability.

For the 2D laser-based detection, the prototype is able to detect, track and classify moving objects as humans from both stationary and moving sensors. False positives are still present – especially during the movement of the sensors - but several strategies are currently under testing to reduce their influence as much as possible. The ongoing activities focus on increasing the robustness of the developed prototype regarding the continuous tracking of the objects and the classification of the detected objects as humans.

The 3D based detection prototype is able to detect human presence in the monitored space, with single sensors or with a combination of them. The prototype is able to provide pose information to ROS based systems and has configurable safety areas for different warning levels. Current focus is put in improving robustness and the individual human pose tracking, which are still lacking reliability.

From the HRI side, three prototypes have been presented. Each one realizes a different type of interaction, while all the initial prototypes for the applications described in the D2.1 "Perception for HR interaction – Design" have been implemented. More specifically, initially the wearable device prototype is currently able to allow successfully a basic interaction between the human and the system, with the wearer being able to receive information, monitor and validate the operation flow. Additionally, gesture module enhanced from the initial version described in D2.1, is now able to support different skeleton types, control both robot arms and include more gestures. Finally, all the aforementioned prototypes have been integrated in the THOMAS framework. Next steps will include further enhancements and testing with the real MRP.

While all the sub-systems are successful to some level, the development of the prototypes is still work in progress. Future versions will focus on improving the overall robustness of the system and the integration in the THOMAS framework in order to provide an integrated source of human tracked poses.

7. GLOSSARY

MRP	Mobile Robot Platform
MPP	Mobile Product Platform
HRC	Human Robot Collaboration
HRI	Human Robot Interaction
AR	Augmented Reality

8. REFERENCES

Arbuckle, D., Howard, A., & Maja, M. (2002). Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 409-414 vol.1).

Elfes, A. (1989, June). Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, pp. 46-57.

Fuerstenberg, K. C. (2009). Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner. *Open Access Repositorium der Universität Ulm*.

Kohler, M. (1997). *Using the Kalman Filter to track Human Interactive Motion* -- *Modelling and Initialization of the Kalman Filter for Translational Motion.*

Leigh, A., Pineau, J., Olmedo, N., & Zhang, H. (2015). Person tracking and following with 2D laser scanners. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 726-733.