Technical Report

IRI-TR-20-03

Visual feedback for humans about robots' perception in collaborative environments

IRI Technical Report

Juan Cruz Gassó Loncan
Alberto Olivares-Alarcos
Guillem Alenyà

July, 2020

Institut de Robòtica i Informàtica Industrial

Abstract

During the last years, major advances on artificial intelligence have successfully allowed robots to perceive their environment, which not only includes static but also dynamic objects such as humans. Indeed, robotic perception is a fundamental feature to achieve safe robots' autonomy in human-robot collaboration. However, in order to have true collaboration, both robots and humans should perceive each other's intentions and interpret which actions they are performing. In this work, we developed a visual representation tool that illustrates the robot's perception of the space that is shared with a person. Specifically, we adapted an existent system to estimate the human pose, and we created a visualisation tool to represent the robot's perception about the human-robot closeness. We also performed a first evaluation of the system working in realistic conditions using the Tiago robot and a person as a test subject. This work is a first step towards allowing humans to have a better understanding about robots' perception in collaborative scenarios.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain
Tel (fax): +34 93 401 5750 (5751)
http://www.iri.upc.edu

Corresponding author:
Guillem Alenyà
tel: +34 93 401 0775
galenya@iri.upc.edu
http://www.iri.upc.edu/staff/galenya

Contents

1 Introduction

During the last years, major advances on artificial intelligence have successfully helped robots to perceive and be aware of the environment were they are performing their tasks. However, most of the research on human-robot collaboration (HRC) in shared spaces is focused on assuring human safety, forgetting about the collaborative aspect of the tasks. Of course, safety is a fundamental aspect of collaborative robotics, but it is also important to ensure a fluent and efficient collaboration between humans and robots.

In HRC, collaboration is defined as the joint activity of humans and robots in a shared workspace, working together on a set of shared tasks to accomplish a common objective [14]. Hence, HRC requires that the robot shows a certain degree of autonomy to adapt to a changing environment. To increase robots' autonomy, it is necessary that they understand the changes occurring in their workspace (perceive), for instance, the actions of humans. Of course, autonomous robots are less predictable and reliable, therefore humans would need to be aware of the current situation and understand robots' behaviours [14]. Indeed, to achieve a true and an efficient collaboration, it is needed that both robots and humans can perceive each other's intentions and interpret which actions they are performing. Reaching that level of awareness of each other, both of them could be able to plan and adapt their actions accordingly towards achieving their shared objective, while the safety is also assured [3].

Apart from understanding the actions that a robot is doing, it would be desirable that a human could also have access to the robot's interpretation of the environment, namely what the robot is perceiving. Therefore, the robot needs to communicate to the human how the robot's senses are considered, understood or interpreted. The real challenge is to find a way to do this without distracting the operator [8]. The communication needs to be intuitive, fluent and not invasive or annoying, so the humans can be concentrated on their tasks and not be concerned with the robot's movements [6].

As examples of such communication, from the HRC safety approach there are some approaches where the human obtains real-time visual information about the adaptive safety zones defined in ISO/TS 15066 [1]. Vogel et al. [13] proposed a Human-robot separation monitoring system of the shared workspace using tactile floor mat (to perceive the human presence) in combination with a visualisation system to project on the floor the safety zones, as well as a set of symbols to communicate the intended robot movement direction and the target position. A similar approach was presented by Hietanen et al. [7], replacing the tactile floor mat with an depth-vision sensor. A different approach of communication was studied by Casalino et al. [6], where human awareness of the robot's perception was addressed by using haptic feedback through a vibrotactile device to alert the user during critical phases of the collaborative task.

The examples above focused on informing the user whether the robot detected the presence of the operator or not. Both [7] and [6] used depth cameras as the main sensor to detect humans. Furthermore, the use of this 3D sensor technology, together with artificial intelligence (AI) techniques (such as OpenPose [5]), has also enabled the 3D detection of human pose. Works like [12, 9, 10] used a RGB-D sensor and human's joints 3D tracking techniques to measure the minimum human-robot distance to apply the speed and separation monitoring ISO 15066 based approach [1].

The aim of the current work is that the human can obtain knowledge about the robot's level of understanding of the environment they share, with special focus on the perception of the human presence. We concentrate our efforts on developing a tool for HCR applications to visualise and represent the human-robot closeness within a shared workspace. For this, a robot must be able to estimate the pose of humans and their body parts in the 3D space with respect to itself. In the following section (section 2) we defined the objective of this work. In section 3, it is detailed the tool presented while in section 4 are shown some results. Finally, in section

5 we present our conclusions, limitations and future work.

2 Objectives

The goal of this work is to *develop a tool for collaborative robotic applications to visualise and represent the human-robot closeness within a shared workspace*. For this, a robot must be able to estimate the pose of humans and their body parts in the 3D space with respect to itself. The main goal can be split into two specific objectives:

- Integration and adaptation of already existing approaches to human 3D pose estimation.
- Development of a virtual approach to represent and visualise the robot's interpretation about the closeness to a human.

3 Tool description

The tool presented in this work has three main entities that exchange information as shown Fig. 1. The artificial intelligence (AI) of the robot takes as input the information of the space and the human (H) pose through the subjective camera of the robot as RGB-D images. This AI then generates a set of points in the three-dimensional space with reference to the Tiago's camera, for each of the human joints. Finally, the interface module (I) takes these points and interprets them to generates a visual representation for the human (H) of what the robot is perceiving.

The proposed tool is composed of two main ROS nodes: rgbd-pose3d and iri_visualizer. The rgbd-pose3d corresponds to the AI entity described before, generating the body's joints 3D data from the RGB-D images. The iri_visualizer is the core of the interface (I), and where the graphic representation robot perception is generated. In Fig, 2, it can be seen their basic relationship and the main topics they publish and are subscribed to.

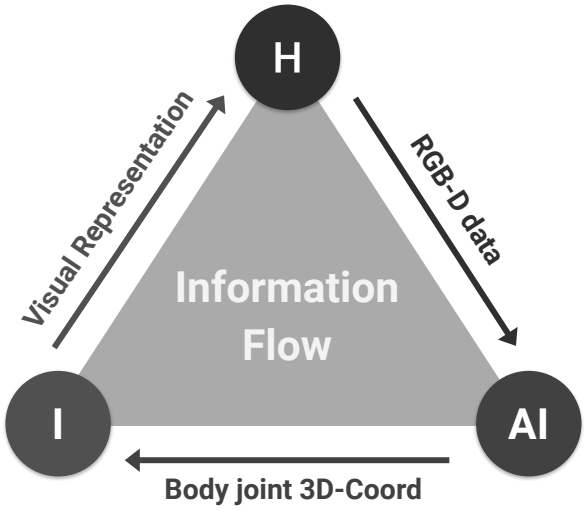

Figure 1: Information flow diagram. **H**: Human, **AI**: Artificial Intelligence, **I**: Interface. The AI use the space information acquired by RGB-D camera and reports the 3D coordinates of the human (H) joints. Finally the interface (I) takes this information, interprets it and generates a visual representation of the perception.

3.1 The rgbd-pose3d ROS node

This node aims to get body's joints 3D coordinates from the RGB-D images. In 2017, at the IRI's lab, Arduengo et al. [4] developed a ROS wrapper for people pose estimation in 3D from a single RGB-D camera, based on the well know OpenPose library [5]. Unfortunately, due to several updates in different layers of that library, the mentioned wrapper became obsolete and was no longer functional. From this moment on, and with no intention of redoing the previous work, the effort was put in finding the shortest way to obtain similar results to the previous one, implementing any relevant modification to be used with Tiago's subjective camera.

Our node is a minor modification of the rgbd-pose3d ROS wrapper developed by the Zimmermann et al. [15]. In this wrapper underlies a convolutional neural network (CNN) architecture that jointly uses colour and depth images in order to predicts 3D human pose in real world units. The CNN model makes use of the OpenPose library [5] to get the 2D key points, which are used together with the depth data as input to the VoxelPose net to generate the 3D points.

Their node, basically subscribes to the to colour and depth images provided from a Kinect v1 to feed the CNN architecture and generates the 3D data of the human joints. With this information the node publishes the 3D position of each joint as frames on the `\tf` topic. Additionally, it also publishes a `MarkerArray` message that can be used on Rviz to draw a line representation of the human on the 3D space. The problem is that there is no way to access the CNN result directly. Although positions can be obtained from the `\tf` topic, we do not know the confidence level of the CNN inference.

In response, we propose a new topic to publish the whole CNN outcome. Also, Tiago's RGB-D camera is an Asus Xtion Live Pro that required some software modifications to make it compatible. Lastly, a new configuration file was added to make the settings cleaner.

The new topic publishes the raw output of the CNN inference by a `Float32MultiArray` message, called `human_pose_inference`. The CNN model outcome consists of two arrays, one for the 3D coordinates of each human joint, and another one with the level of confidence in the detection. The first array, named `coord3d_mat`, has three dimensions and shape (**np**, 18, 3) where **np** is the number of people detected and 18 represents the joints. The last number (3) corresponds to the `(x, y, z)` coordinates. The second array, named `conf_mat`, has shape (np, 18), is a matrix in which each row represents a person and the columns are the level of confidence for each joint. As the multi-array message has a single flat vector to send the data, we created a new array from the concatenation of the previous two, getting a shape of (np, 18, 4). Hence, for each key point we have (x, y,z, *lc*), with *lc* as the *level of confidence* between 0 and 1. Finally, this new array is flatten getting vector with ($\mathbf{np} \times 18 \times 4$) elements.

To make the script compatible with the xtion, it was necessary to multiply the scale of the depth maps by a factor of 1000, simply because the depth map from the Kinect were published on millimeters, while the xtions depth map are in meters. In addition, a method was applied to

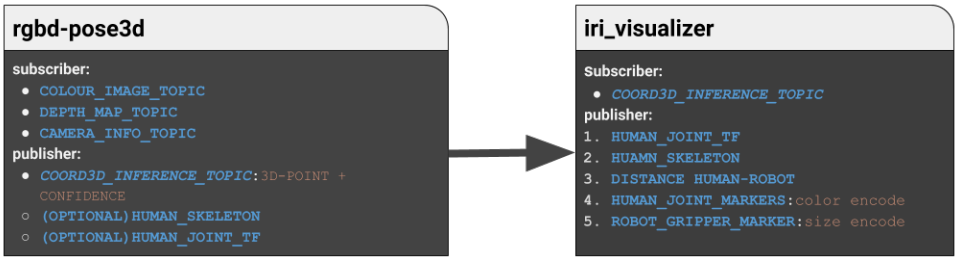

Figure 2: Nodes architecture where it can be seen their basic relationship and the main topics they publish and are subscribed to.

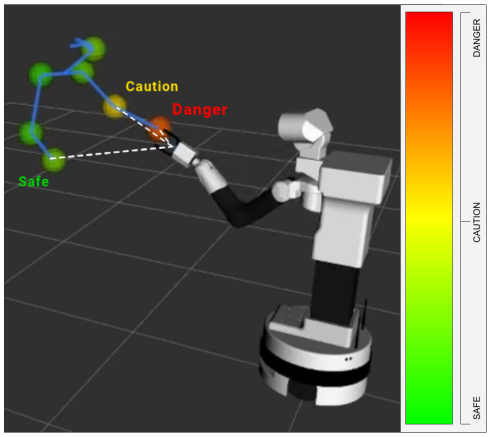

Figure 3: The image shows the colour scale (right side) used to represent the danger of being impacted by the robot gripper, which is determined by distance between the human joint and the robot's gripper. There are three basic states (left side), green for safety state (far), red for danger state (close) and between them it is the cautious state (middle point) identified by yellow.

ensure that any NaN value is replaced by a zero value.

Finally, all the settings parameters from the node script were migrated to a yaml file in which you can set the name of the subscriber topics from the camera, a set of parameters for the CNN model and the confidence threshold. Additionally, a group of optional flags can be be used to activate or not the *transform broadcasting* for the human joints and *publishing a marker* for a 3D line representation of the human detected. This options were implemented trying to minimise the computational cost and improve the node rate.

3.2 The iri_visualizer ROS node

This section comprises the explanation of the node for processing the 3D data published by the rgbd-pose3d node to finally publish visual messages that can be used to render markers on Rviz. The human-robot distance is calculated for each human joint and then used to modulate some properties of such markers, for instance the colour or size. These properties are used to give meaning to those markers. On one hand, the colour of the markers for the human joints is used to represent the danger of being impacted by the robot. On the other hand, the size of a marker on the robot's gripper represent the free space available for movement, increasing proportionally with distance.

As mentioned, the distance between the robot and the human is used to modulate certain properties of the markers. To make this possible, the sigmoid function or logistic curve was used (See Fig. 5). This function allowed us to smoothly map the entire distance domain into an arbitrarily limited range. For example, for the size of the gripper's marker, it is possible to define the maximum size of the marker and maintain a range where it scales proportionally with the distance.

This node is written in C++, based on the IRI catkin-scrips[1]. The node only subscribes to the `human_pose_inference` multi-array message from the `rgbd-pose3d` node. The first step is taking the message data and recovering the original two arrays generated by CNN model of the `rgbd-pose3d` node, one with the 3D points of the human joints (`coord3d_mat`) and the other with the level of confidence (`conf_mat`).

[1]Available here: https://gitlab.iri.upc.edu/labrobotica/ros/iri_core/scripts-catkin.git

Once the data its recovered, the node does the following:

1. Broadcasting a `transform` (TF) for each human point. For each 3D point[2] from the `coord3d_mat` is published a frame on the \`tf` topic.

2. Publishing a `LINE_LIST` visual marker to represent the human pose. The `LINE_LIST` it is defined by a set of 3D points pairs. Each pair define a line segment. For a segment to be drawn, the confidence level of both 3D points must be greater than the confidence level threshold.

3. Publishing the distance from the human joints TFs to the robot's gripper TF. The distance is calculated as the Euclidean distance. Here, the level of confidence is considered to filter out bad detection and forcing the distance to zero when the confidence is not enough. For the publishing, it was created a custom message with a standard Header and two arrays, one of strings for the joints names and other of floats for the corresponded calculated distance. At this point, it is able to select a subset of human joints.

4. Publishing a MarkerArray with `SPHERE` markers for the human joints. Their colour represent the danger of being impacted. Each of this spheres are coincident in position with the sub-set of human joints chosen on the previous point and coloured to represent the danger of being impacted by the robot gripper (see Fig 3). The colour goes in a continuous scale from full red for *danger* at close range, passing through orange until yellow for *caution* at mid range, greening up to a full green for *safe* at far distances. In addition, these markers turn grey when the confidence was not met (maintaining the position estimated by CNN, which is probably inaccurate).

5. Publishing a `SPHERE` marker on the Robot's griper. Its size represent the free space available for movement (see Fig. 4). To achieve that, the diameter of the marker varies proportionally to the distance, from zero until a predefined maximum. It is important to remark that it is not considered any other object that could be an obstacle, only the distances to the human. Specifically, it is taken the minimum calculated distances, different from zero [3].

The sigmoid function is used for the modulation of the marker's parameters, is defined by Eq. (1). As can be seen in the Fig. 5, it has an "S" shape, approaching asymptotically to 0 from the left and to a maximum from the right. By changing the values of M, S and I parameters, it is possible to adapt some of the characteristics of the shape to our needs. M defines the upper asymptote, that is the maximum. S defines the slope of the linear section of the "S" shape. This parameter can be used to delimit a domain section of interest, which will be affected in this linear part. For instance if S is lower than 1, the lineal part will cover more domain on x. Finally the I defines the inflection point. In other words, this parameter can be used to move the "S" shape along x axis.

[2] We talk about 3D points and no human joints, because there are also points for the nose, ears and eyes.
[3] recall that zero also means lack of confidence

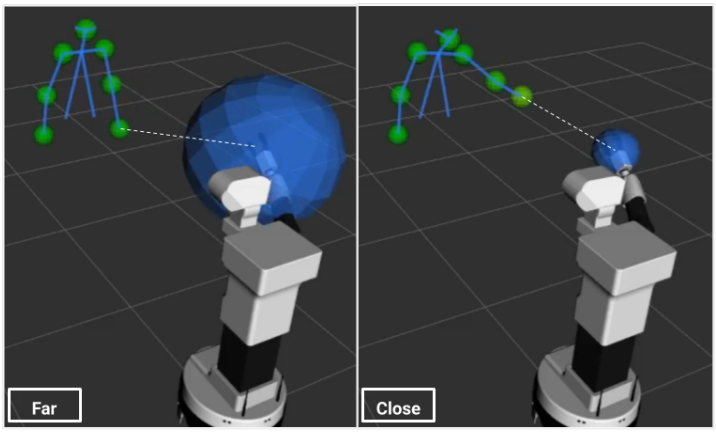

Figure 4: The image shows two different states for the size of the robot's gripper marker. On the left, the size of the marker is large, representing the large free space available, because the gripper was far from the nearest human joint (the left wrist). On the right, the human had extended his left arm towards the robot's gripper. You can see that the left wrist is a little bit more yellow, because it is closer than before. So, there is less free space available and therefore a smaller clamp marker.

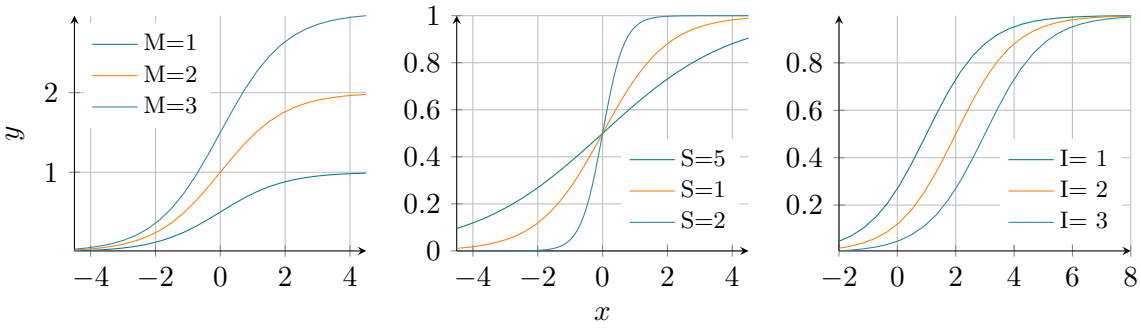

Figure 5: Sigmoid shape used on the modulation of the distance between the human and the robot, and its principals parameters effects.

$$y = \frac{M}{1 + exp(-S(x - I))}, \tag{1}$$

Where:

M : ≥ 0, defines the upper bound.
S : ≥ 0, defines slope of the middle region.
I : ≥ 0, defines the inflection point.

4 Results

In this section, we analyse the performance of the implemented tool in a case study with different situations where a human moves around a robot. Our aim is to identify the assets and possible

drawbacks of using our tool in such a realistic scenario. Specifically, we tested the performance of the human's pose estimation system in several situations with different degree of complexity. We recorded the human-robot interaction using `rosbags`, so that we could evaluate and visualise the tool's performance off-line[4].

4.1 Evaluation setup

In order to properly process almost every frame during the offline visualisation, we needed to play the `rosbags` at half of the velocity, 15 fps. The hardware used during this test was a laptop with a Intel I7-7700HQ, 8GB RAM, GTX 1060 3GB VRAM.

As for the `rgbd-pose3d` node, we disabled both optional publishers, so it was only dedicated to the inferences of the 3D pose. With respect to the CNN setting, we needed to set the value of several parameters. First, we defined the number of stage for the 2D part of the model (OpenPose net). Smaller number makes the network fast but less accurate. The value was set to 5. Second, one can choose between two types of architectures of the VoxelPose net, `fast` or `default`. They differ in the numbers of convolutional layers of the encoder blocks. The effect is the same as before, the first is faster but less accurate than the second. The architecture type was `fast`. Finally, GPU memory was limited to 95%.

Regarding to the `iri_visualizer` node, we needed to fix the value of a threshold that is used to decide whether a human limb is shown or not. Note that for a human segment to be drawn, the two joint points delimiting a limb must be inferred with a confidence greater than this threshold. The value was set to 0.2. With respect to the colour and size modulating, we needed to define two sigmoid functions. With regard the colour of the human joints, we set $M = 1, S = 5I = 0.5$. In this way, the active distance is between 0 and 1.5m. So it is *safety* state (green) around 150cm, getting yellow until it pass through the *cautious* state at approximate 50cm and increasingly red until it reaches the *danger* state near 0cm (See Figure 10). Finally, regarding the size of the blue marker on robot's gripper, the configuration was: $M = 1.2, S = 5, I = 1$, so the maximum sphere diameter is equal to M, being at 90% of the max size at a distance of 1.44m.

4.2 Studied situations

From the recorded human-robot interaction, we selected a set of interesting situations that allowed us to analyse different aspects of our tool. First, we studied situations in a free workspace where there is not any object between the human and the robot. In this case, we distinguished three different situations: no occlusion, human's self-occlusion and robot's self-occlusion. Finally, we examined two more situations with a shared working table in the workspace: standing human and sitting human.

For the cases, we analysed different visual and quantitative features which are depicted in a single figure per each case. First, we showed the colour (RGB) and depth frames. Secondly, the Rviz visualisation containing the robot, the human, and the markers used to indicate the human-robot closeness. Finally, we also showed a plot depicting the Euclidean distance between the robot's wrist and three human's body parts: nose, right and left wrists. Note that the human was never at a distance of zero meters, so when in the plots the distance is zero means that we lost the human's joints.

4.2.1 Free workspace: No occlusion

In this case, the robot had access to the whole body information of the human without any type of occlusions. Therefore, our system was able to estimate the pose of all the human body

[4]Video of the off-line analysis: https://drive.google.com/file/d/1ltQMcQ88-3J4vmBIaXUEAiXUl8Lu1knc/view?usp=sharing

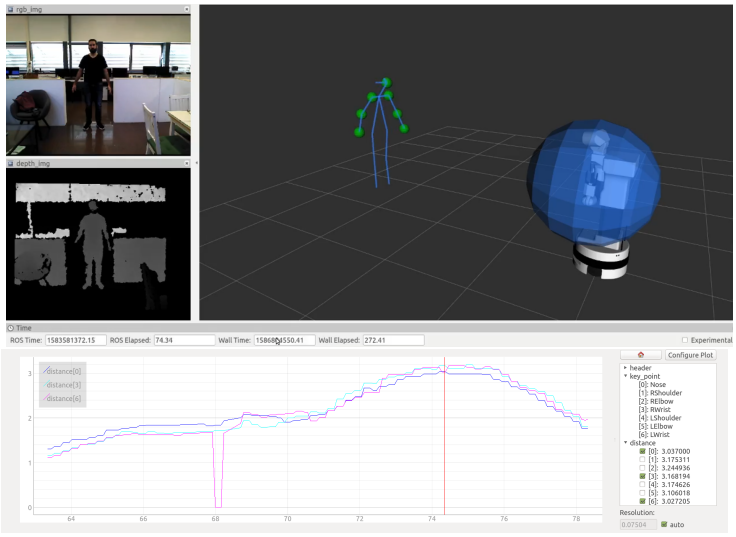

Figure 6: Free workspace: No occlusion - without relevant losses. The human body is completely detected, and all the upper body markers are green and the robot's marker is big because the human is far from the robot (safety state). There is an instantaneous loss at 68 sec ROS-time when the subject took off his glasses. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot.

joints (see Fig. 6). The distance between the robot and the human was around 3 meters (*safety* state), thus, the markers of the upper human joints were represented in green, and the marker of the robot's wrist was big (full freedom of movement). Nevertheless, the system showed some instabilities and we lost track of some of the joints and thus their markers (see Fig. 7). For visualisation purposes, whenever we lost any marker (confidence lack), we kept the current estimated position of the marker, but uncoloured, i.e. in grey. We would like to focus on the right wrist of the human (signal cyan in the plot). We lost the joint several times (distance equals to zero) even though there was not occlusions at all. This might be caused by the window at the background, the glass is a reflective surface that introduces a lot of zeros in the depth data. In conclusion, we saw that our system works pretty well even when the human-robot distance is large. We also noticed that the system loses some of the joints, but this is during short periods. Finally, we just corroborated that the system does not performs well with reflective backgrounds.

4.2.2 Free workspace: Human's self occlusion

In this situation, the robot could visualise the whole human body with exception of the legs which were out of frame. Nevertheless, in this case the subject was sitting on a chair with his hands resting over his thighs. Now the joints were much closer to each other and some body parts could overlap generating self-body occlusions. However, the system could infer the pose of all the human joints successfully (see Fig. 8). Even though the distance between human and robot's gripper was reduced to 1.2m approx., the upper body markers remained on *safety* state. Regarding the marker of the robot's wrist, it was indicating half freedom of movement, in response to the distance diminution. Note that the robot arm was moving towards an unfolded position, during which the system did not lose. If we focus on the distances plot, we can see that mean distances changed from 1.6m to 1.2m, because of the arm unfolding. The losses shown in figure 9 happened when the left arm was extended and almost aligned to the line of robot's view. Both wrist and elbow were occluded by the hand. Furthermore, the right elbow was very close

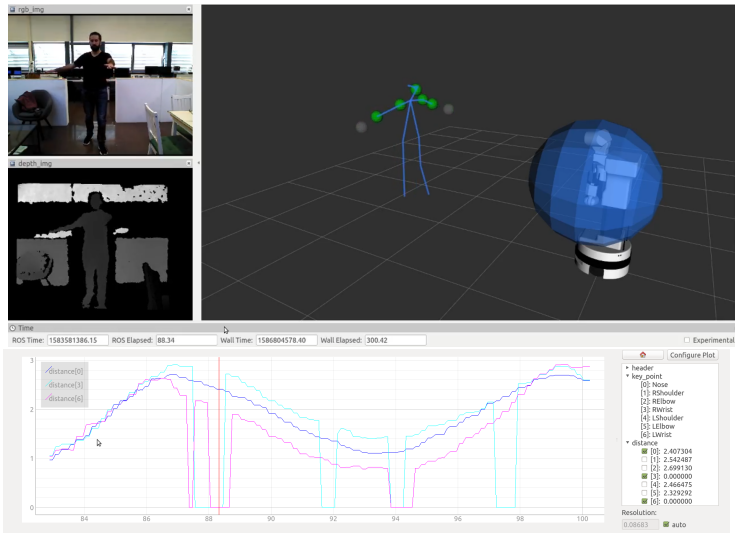

Figure 7: Free workspace: No occlusion - with relevant losses. The upper body markers are green and the robot's marker is big because the human is far from the robot (safety state). There are several losses of the human wrists (grey markers), specially of the right one (cyan line). The depicted frame corresponds to the ROS-time pointed by the vertical red line in the plot.

to the body. The left elbow was the worst estimation, but expected due to the almost complete occlusion of this joint. Finally, it is worth mentioning that the gripper marker was bigger than expected because the system lost two joints of the left arm, witch were the the closest body joints. Hence the size was defined by the current closest marker, probably the left shoulder. In conclusion, our system performs satisfactory well while the human was siting. The system loses were expected regarding the pose estimation of the left arm and the right elbow. We also noted that despite the lack of confidence due to the human self occlusion, the inference was very close. However, the fact that the size of the gripper's marker was clearly affected by the losses, it is something that must be attended.

4.2.3 Free workspace: Robot's self occlusion

This case is regarding to the occlusions produced by the robot itself. Due to the position of the onboard camera of Tiago, its own arm could occlude the vision of the human subject. Nevertheless, the system showed quite good performance during these situations as shows Fig. 10. Even though the robot's gripper was occluding most part of the left human's hand, the system managed to detect the left wrist. Furthermore, this frame shows an interesting case where we got the three basic visualisation states (*safety, cautious* and *danger*) simultaneously. It can be seen that the left wrist marker was red, the left elbow was yellow while all the rest were green. In addition, the marker on the robot's gripper was imperceptible (indicating no free space available) according to the *danger* state of the left wrist joint. Note that the robot arm was moving to an offering position, during which the system did not have any losses. In fact, if we pay attention to the plot, the system worked smoothly and stably, and at very short distance (even less than half meter) throughout this trail. As expected, the system lost track of some joints due to a total occlusion of a body part. Figure 11 shows a case where the robot's gripper occluded a considerable part of the human's right arm, mostly the wrist. This occlusion obviously impacted on the detection of the wrist, significantly compromising the level of confidence. However, if we

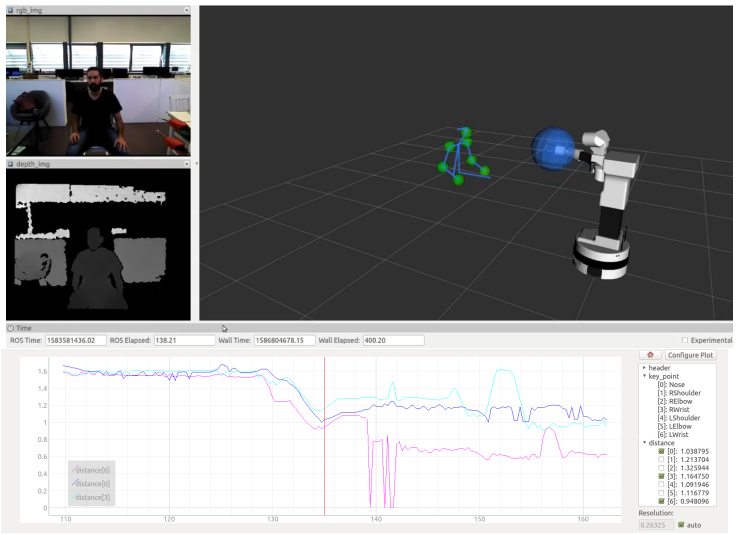

Figure 8: Free workspace: Human's self occlusion - no losses. Complete detection of the human pose on sitting position, with both arms resting on the thighs and exclusion of the legs which were out of frame. With a mean distance over 1m, all key points green (safety) but the gripper sphere is at half size, showing a reduced freedom of movement. The distance graph shows stable detection along the time window. The depicted frame corresponds to the robot arm unfolding movement at ROS-time of 138.21 sec indicated by the red vertical line.

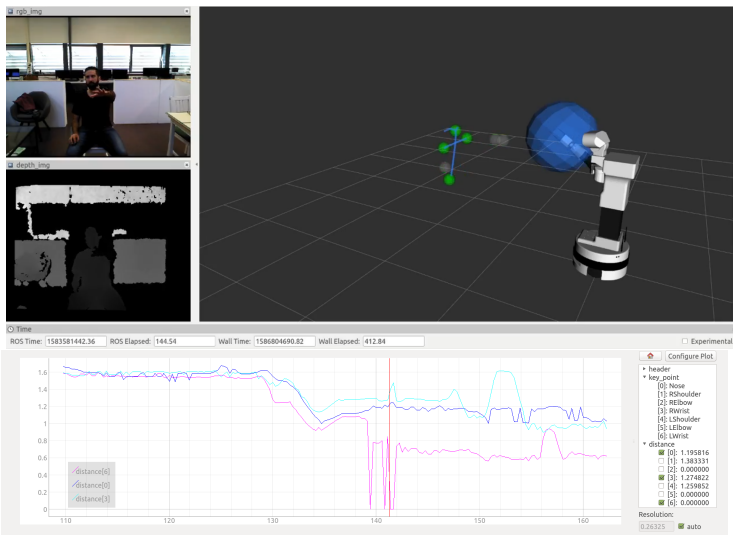

Figure 9: Free workspace: Human's self occlusion - expected losses. The image shows the loss of three markers since there were not enough confidence in the detection. The left arm markers for the elbow and wrist were lost probably because the arm was fully extended to the front with hand open, almost aligned to the line of view. Both wrist and elbow are occluded by the hand. The right elbow is flexed and close to the torso, which may explain the non-detection. Notice that the robot arm is fully unfold. Frame on ROS-time: 138.21 sec indicated by the red vertical line.

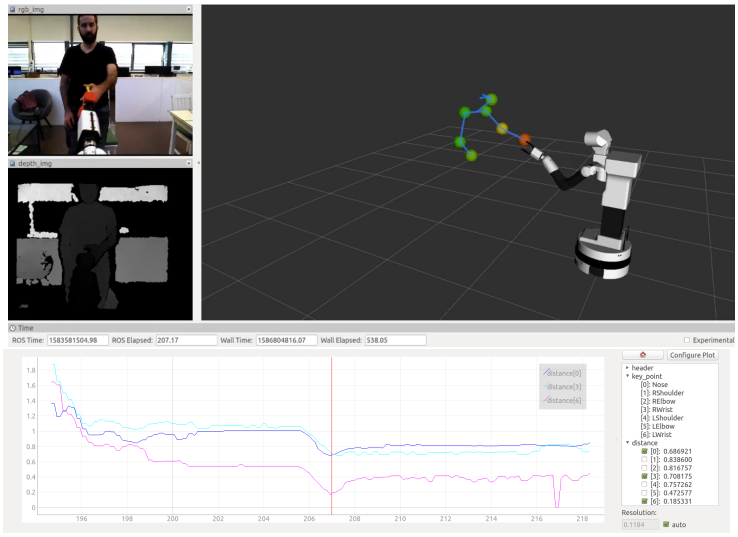

Figure 10: Free workspace: Robot's self occlusion - no losses (Tri-state). Full detection of both arms, despite that the gripper was occluding a significant part of the left hand. Three basic states simultaneously: red-danger (left wrist), yellow-cautious (left elbow) and green-safe the remaining. The gripper's sphere diameter was close to zero, due to the closeness of the left wrist. Frame in ROS-time: 207.17 sec. indicated by the red vertical line.

look at the grey marker, the estimation seems to be reasonable. In general, our system behaves reasonably well along these situations. Although the system logically lost confidence during total occlusions, the inferences were still well-estimated. In addition, we found situations where occlusion was important and yet the system managed to keep track of all human joints.

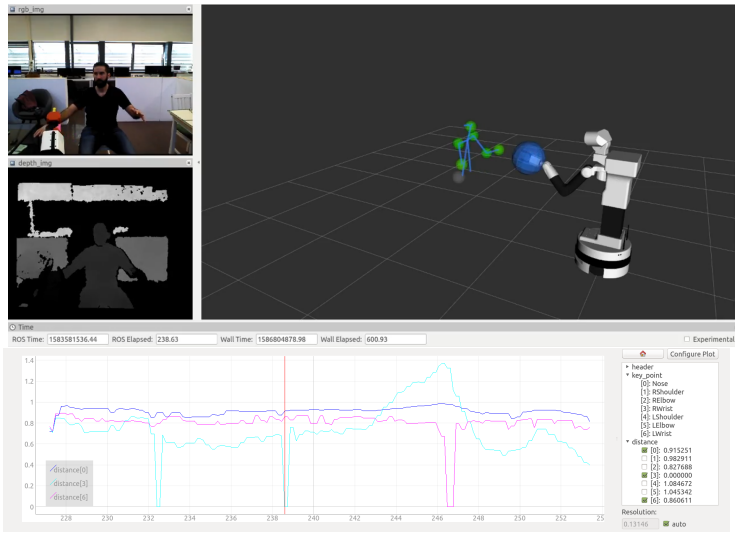

Figure 11: Free workspace: Robot's self occlusion - Expected losses. The Rviz scene shows that the gripper occluded the vision of the right wrist, and therefore the lack of confident (grey and distance cero). The plot shows three losses, the first two losses were similar, while the third was human's self-occlusion. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (238.63 sec.).

4.2.4 Shared working table: Standing human

We wanted to determine the system's behaviour when the subject was standing in front of a table. The lower body was occluded but the body is more visible than when the human was sat. In this scenario, we analysed three different cases, of the same time window of almost a minute without lack of confidence. During this period, the subject moved both arms and the robot's arm was also moved from an offering position to unfolded.

In figure 12 it can be appreciated that there were no problem with the detection from the hips to the head, even though the robot's grippers was in the middle. It is worth noting, that if you look at the 2D RGB image the robot's end effector was very close to (even partly occluding) the subject's right hand, and likewise the system was able to discriminate the gripper from the hand correctly.

Another example can be seen in figure 13, where the subject's right hand was reaching the robot's gripper while the left hand remains on the table. The right wrist marker was red-orange and the gripper size it nearly zero, implying that no movement was safe.

Going a bit further, figure 14 depicts an example where the subject was carrying the table, and the system did not lose any joint.

All three cases explained above correspond to the same 50-second. During this period, the graph shows that none of the markers was lost even during the movement of the robot's arm from the offering position to unfolded, as well as when the human carried the table.

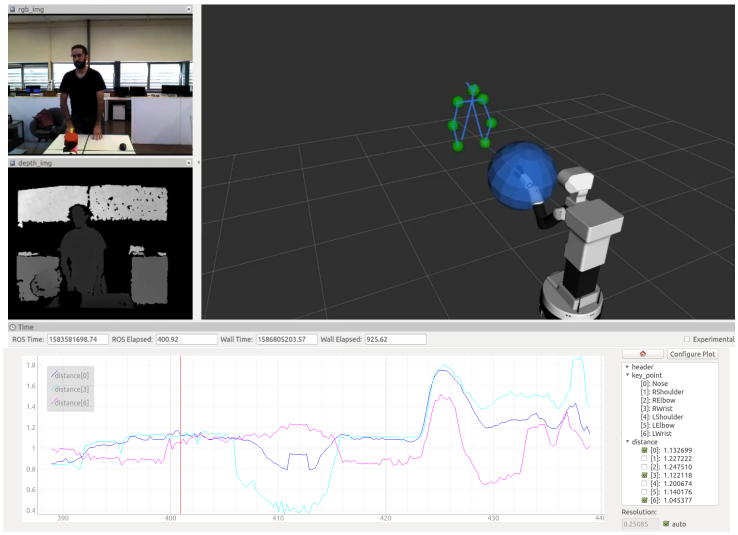

Figure 12: Shared working table: Standing human - example 1. Complete detection of the upper body from the hips to the head while the subject was standing behind a table. The subject remained stood with both hands resting on the table. Notice that even the robot's gripper was occluding part of the hand, there were no losses. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (400.92 sec.)

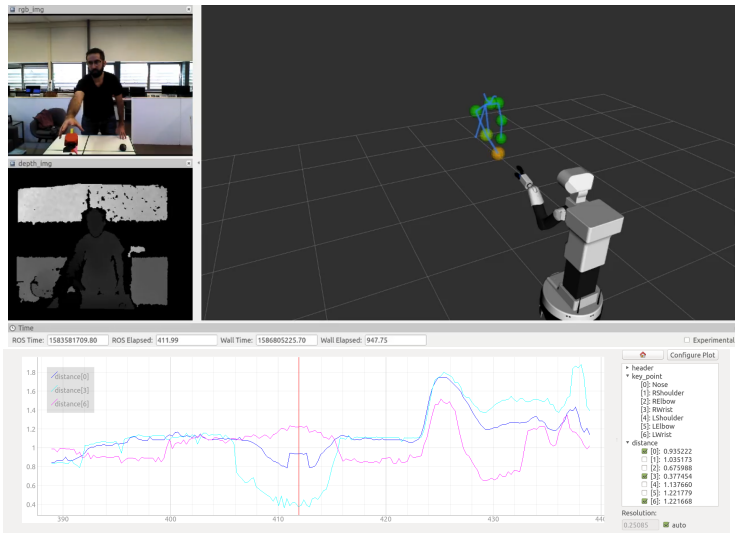

Figure 13: Shared working table: Standing human - example 2. The image shows the subject standing behind a table, reaching the robot's gripper. All human joints from hips to head were detected. The colour of the right arm markers went from green to an orange, from the elbow to the wrist, due to the progressive proximity to the gripper. The end-effector marker's diameter was nearly zero in due to the closeness. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (411.99 sec.)

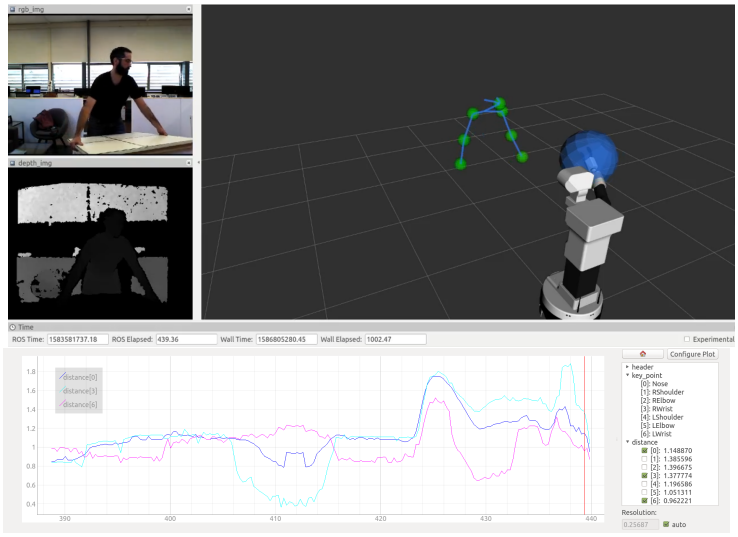

Figure 14: Shared working table: Standing human - example 3. The image shows a particular situation where the subject was carrying a table. During this kind of situation the system managed to detect all markers. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (439.36 sec.)

4.2.5 Shared working table: Sitting human

This case comprehends the human sitting in front of the robot, just behind a working table. We show different images which correspond to the same time window, and as it can be deducted from the plot the detection it was stable enough. As well with the standing position, the system managed the sitting position behind the table with no problem. Figures 15 and 16 are merely the same with the difference in the robot's arm position, first in offering position and then unfolded. This can be appreciated on the diameter of the gripper's marker. There were two losses which correspond to times where the robot's gripper occluded the visual of the subject wrist during the unfolding of the arm and then coming back to offer position. The first case was when the robot's gripper passes just over the subject right wrist, as it can be seen in figure 17. The same happened at the end of the graph with the left wrist. This case shows that the system can manage the human detection even with half of the body occluded by the table. Of course, the system lost all the joints below the table, but this did not affect the detection of both arms and head.

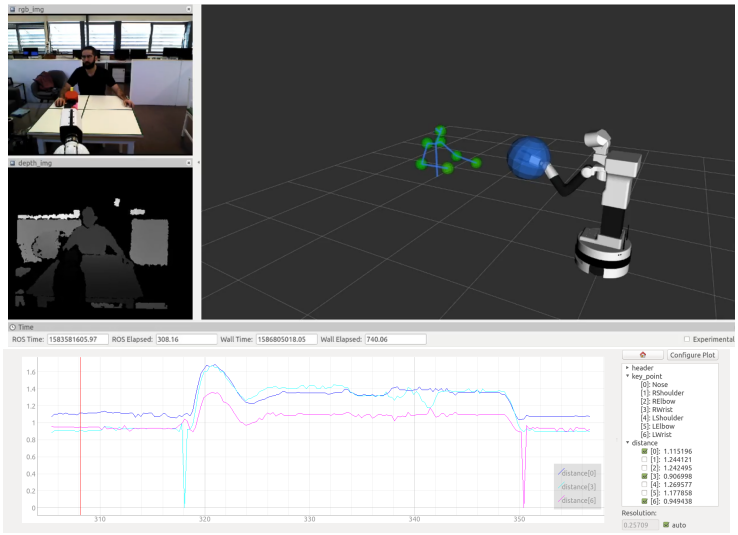

Figure 15: Shared working table: Sitting human - Robot's arm offering position. Complete detection of both arms and head while the subject was siting behind a table. The subject rested both arms on the table. The robot's gripper was in an offering position occluding part of the hand, however there were no losses. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (308.16 sec.)

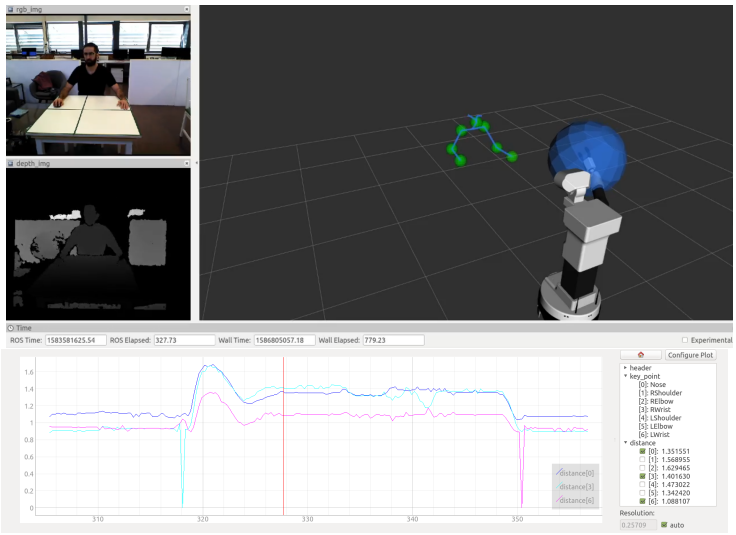

Figure 16: Shared working table: Sitting human - Robot's arm unfolded position. Complete detection of both arms and head while the subject was siting on a chair, behind a table. The subject rested both arms on the table. The robot's gripper was in unfolded position, increasing the distance and though getting a big gripper's marker. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (327.73 sec.)

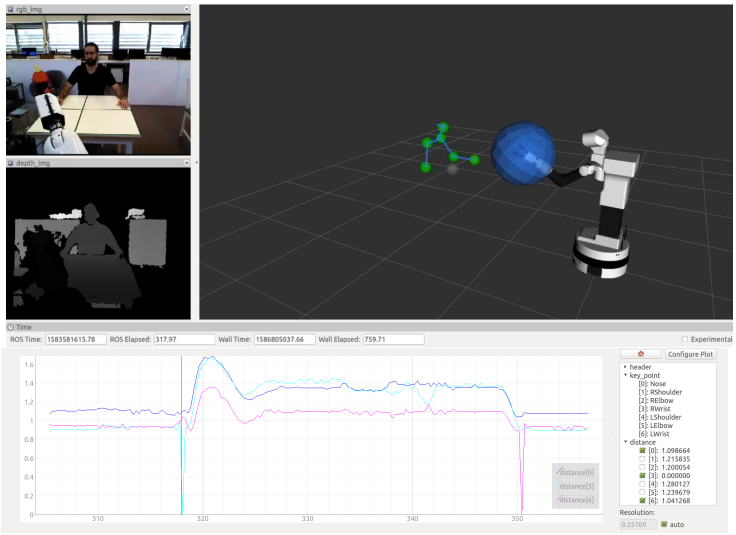

Figure 17: Shared working table: Sitting human - Robot's arm occlusion. The image shows one of the two losses cases. In both cases, the gripper passed over the wrist, occluding the visual and therefore causing the lack of confidence on the respective human joint. The depicted frame corresponds to the ROS-time pointed by the red vertical line in the plot (317.97 sec.)

4.3 Lesson learned

During the development of this work there were some problems that are worth mentioning. The first was with `rosbag`[5] recording of the Tiago Robot. Fist of all, it is important to point out that, although the notion of `rosbag` as a tool existed, but there was not enough experience, neither with real robots nor with such a complex ROS system in terms of number of nodes and topics. At the begging it was tried to record every topic published by Tiago. To do that, communication was established with Tiago through an isolated WiFi network. It was set to `ROS_URI` with the IP address assigned to the robot so that commands executed on an external computer are managed by the same `ROS_CORE` as Tiago. Once the communication was established, the following command was run:

```
$ rosbag record -a
```

But that did not work as expected. When the bag file was reproduced using:

```
$ rosbag play <FILE_NAME>.bag
```

When we tried to visualise the RGB-D image by reproducing the rosbag, there were no images at all (or only the first frame) at the respective topics. To discard any problem regarded the WiFi network, the process was repeated but within the Tiago system by SSH protocol, and then sending to the personal computer. The result was the same. Additionally it was added the command argument `-b 0` set the internal buffer to infinite in both cases but with no change.

The next trial was to reduce the amount of data recorded. We only selected the topics needed by the `rgbd-pose3d`:

```
COLOR_IMAGE_TOPIC : /xtion/rgb/image_raw
DEPTH_MAP_TOPIC : /xtion/depth_registered/image_raw
CAMERA_INFO_TOPIC : /xtion/rgb/camera_info
```

To select the topics, we opted to use *regular expressions* by using the `-e` argument.

```
$ rosbag record -b 0
  -e '^(/xtion)/(rgb|depth_registered)/(image_raw|camera_info)$'
```

In this way, the images were recorded correctly. However, we also needed other topics, such as those regard the position of the robot arm.

In response and following the same logic, we implemented a `roslaunch` file which brings up three nodes, each in charge of recording a different set of topics. One for the aforementioned `xtion` topics. A second one only to record the points cloud. Finally, a third node records everything except the topics containing the words `xtion|image|compressed|theora|points`, avoiding record almost all heavy data flows.

Lastly, we saw that using an Ethernet cable ensure a reliable and stable connection than WiFi, hence better rosbags recordings.

[5]Commandline documentation available in: http://wiki.ros.org/rosbag/Commandline

```
<launch> <arg name="path" default="." />
  <node pkg="rosbag" type="record" name="xtion_rgbd" args="-b 0 -e
    '^(/xtion)/(rgb|depth_registered)/(image_raw|camera_info)$' -O $(arg
    path)/xtion_rgb"/>

  <node pkg="rosbag" type="record" name="xtion_points" args="-b 0 -e
    '/xtion/depth_registered/points' -O $(arg path)/xtion_points"/>

  <node pkg="rosbag" type="record" name="tiago_but_xtion" args="-b 0 -a -x
    '(.*)(xtion|image|compressed|theora|points)(.*)' -O $(arg
    path)/tiago_but_xtion" />
 </launch>
```

Next, to synchronously reproduce the three rosbag files, that is to say that they keep the same time base, the following command was used:

```
$ rosbag play --clock tiago_but_xtion.bag
  xtion_rgb.bag xtion_points.bag
```

The argument `--clock` is used to publish the `rostime` clock time. Additionally it was set the `use_sim_time` to `true` as it was recommend in several ROSanswers like this[6] and this interesting discussion[7]. However, due to limited hardware resources at the time, it was modified to lower the computational cost during playback. The file with the point cloud data was discarded and the playback speed was reduced by half or less using the `-r` command.

[6] Full link: https://answers.ros.org/question/12577/when-should-i-need-clock-parameter-on-rosbag-play/

[7] At: https://discourse.ros.org/t/timestamps-and-rosbags-discussing-an-alternative-to-clock-and-use-sim-ti
3238

5 Conclusions

In this work, we developed a visual representation tool that illustrates in an intuitive way, the robot's perception of the space that is shared with a person. Specifically, we adapted an existent system to estimate the human pose, and we created a visualisation tool to represent the human-robot closeness.

The level of danger was computed as a function of the distance between the human and the robot and represented using two different ways. First, we showed some spherical markers on the upper joints of the human body. The human-robot closeness is represented by changing the colour of the markers: from green, through yellow, to red. Second, this same information is reflected in another sphere on the robot's manipulator tool. The diameter of the sphere varies with the proximity to the person, representing the corresponding freedom of movement of the robot (e.g. bigger when the human is far).

We also performed a first evaluation of the system working in realistic conditions, using the Tiago robot and a person as a test subject. A complete human-robot interaction was recorded from which we can extract scenes with different levels of complexity. Beyond the visual representation, we showed an acceptable stability of the perception system, which can be seen in the plots with the evolution of the distance calculated for three human parts: the two wrists and the nose.

Regarding the implementation, two ROS packages were implemented and are ready to be used. The first one is in charge of the inference of the human pose, and the second processes the human pose's information and represents it visually. Although the second one uses the output from the first one, both could work independently. We did some minor modifications of the package used for the human pose estimation [15]. Our aim was to facilitate the configuration of its main parameters, as well as the possibility of choosing whether this node is in charge of the visual representation of the skeleton or the broadcasting of the TF. The visualisation package, iri_visualization was developed completely from scratch, making use of the IRI's scripts. Its design was conceived with modularity in mind to ease future updates and improvements.

5.1 Limitations

As mentioned, the software performs well but has shortcomings and limitations. First there are some inherits problems from the rgb-d technology. The Xtion has a range of work from 0.8m to 3.5m [2], which of course limits the space of work. Figure 11, at ROS-time 255, shows instability due to the closeness to the camera. On the other hand, the reflective surfaces or transparent materials introduce much noise to the depth data.

Secondly, visual occlusions, either from the human body or from the robot's arm, are part of the problem and cannot be avoided. Despite that the model of [15] estimates the position of the joints even when they are not visible, in many cases the estimation is worse, and no method was implemented to solve this problem (e.g. using the poses from the previous frames to compare the estimation of the current pose in order to quantify the error. Another option could be to use this previous information, to estimate a velocity vector that represents the movement of the missing point, and use it to make a new rough estimate of the pose). In Fig. 7 and Fig. 9, the system lost confidence in the wrists position, yet the estimation was relatively good. The lost of confidence also implies that the distance was fixed to zero and, in consequence, the distance used to determine the size of robot's gripper marker was calculated from the nearest joint available with enough confidence.

Thirdly, we have only considered the distances between the subject's joints and the robot's gripper. This means that neither the body volume nor the rest of the volume of the robot arm is taken into account. The robot's elbow, for example, can also be a point of contact.

Finally, the CNN model demands a mid to high-end GPU with at least 4GB of VRAM to achieve real-time operation [15], which is a expensive and high power consumption hardware.

5.2 Future work

It is believed that the development carried out in this work could form a solid base for a more advanced interface, as well as be used as input in a path re-planning system, including in real time. However, a considerable amount of work remains to be done.

Firstly, it would be interesting to carry out more experiments in more controlled conditions, with dynamic movement of both parts (robot and human) to collect more data. A reasonable step would be to perform a set of experiments in some simple collaborative task, so that both qualitative and quantitative variables can be studied.

Second, regarding the visualisation, it could be added more detail with respect the human limbs, as well as colouring each of one with different criteria. For example, it is not the same if the robot's gripper it is approaching to the forearm or to the head, being the last one a much more sensible part of the body. Also, it could be a good to take into account more robot parts, not only the distance from the robot's gripper, but also every link of the robot. An approach like the developed in [11] could be worth studying.

Third, it would be possible to adapt the visualisation tool to inform the user in real time about the robot's interpretation of the closeness to the human. Some examples of technology: RGB LED arrays, sound or even synthesised voice to aid on the communications. Although, it is important not to leave behind the operator's opinion, or the level of distraction this could cause to.

Finally, with respect to the detection system robustness, without getting into the CNN model (which of course could be improved), there is an intrinsic problem due the position of the Tiago RGB-D sensor. Several times its own arm will occlude the visualisation of the human subject, and that is a problem that could be approach using the last known positions to infer the current position of the occluded limb. Some ideas with which this problem could be addressed, are with techniques used in video tracking systems or may be through a recurrent neural network (RNN) trained to predict the next position of a human movement given a set of previous points.

References

[1] Iso/ts 15066:2016 robots and robotic devices — collaborative robots. https://www.iso.org/standard/62996.html. Accessed: 11-05-2020.

[2] Xtion pro live: 3d sensor specifications. https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/specifications/. Accessed: 19-04-2020.

[3] Liliana Antão, João Reis, and Gil Gonçalves. Voxel-based Space Monitoring in Human-Robot Collaboration Environments. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2019-Septe:552–559, 2019.

[4] Miguel Arduengo and Sven Jens Jorgensen. Ros wrapper for real-time multi-person pose estimation with a single camera. Technical report IRI–TR-17-02, Institut de Robotica i Informatica Industrial, 2017.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[6] Andrea Casalino, Costanza Messeri, Maria Pozzi, Andrea Maria Zanchettin, Paolo Rocco, and Domenico Prattichizzo. Operator Awareness in Human-Robot Collaboration Through Wearable Vibrotactile Feedback. *IEEE Robotics and Automation Letters*, 3(4):4289–4296, 2018.

[7] Antti Hietanen, Roni-jussi Halme, Jyrki Latokartano, Roel Pieters, and Minna Lanz. Depth-sensor – projector safety model for human-robot collaboration. (1):3–6, 2017.

[8] Teegan Johnson, Gilbert Tang, Sarah R. Fletcher, and Phil Webb. Investigating the Effects of Signal Light Position on Human Workload and Reaction Time in Human-Robot Collaboration Tasks. In *Proceedings of the 5th Applied Human Factors and Ergonomics Conference*, pages 207–215. 2016.

[9] Nikolaos Nikolakis, Vasilis Maratos, and Sotiris Makris. A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace. *Robotics and Computer-Integrated Manufacturing*, 56(September 2018):233–243, 2019.

[10] Martin J. Rosenstrauch, Tessa J. Pannen, and Jörg Krüger. Human robot collaboration - Using kinect v2 for ISO/TS 15066 speed and separation monitoring. *Procedia CIRP*, 76:183–186, 2018.

[11] Mohammad Safeea, Pedro Neto, and Richard Bearee. Efficient calculation of minimum distance between capsules and its use in robotics. *IEEE Access*, 7:5368–5373, 2018.

[12] P. Svarny, Z. Straka, and M. Hoffmann. Toward safe separation distance monitoring from RGB-D sensors in human-robot interaction. 2018.

[13] Christian Vogel and Norbert Elkmann. Novel Safety Concept for Safeguarding and Supporting Humans in Human-Robot Shared Workplaces with High-Payload Robots in Industrial Applications. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, pages 315–316, New York, New York, USA, 2017. ACM Press.

[14] L. Wang, R. Gao, J. Váncza, J. Krüger, X. V. Wang, S. Makris, and G. Chryssolouris. Symbiotic human-robot collaborative assembly. *CIRP Annals*, 68(2):701–726, 2019.

[15] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3D Human Pose Estimation in RGBD Images for Robotic Task Learning. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1986–1992, 2018.

Acknowledgements

This work is supported by the Regional Catalan Agency ACCIÓ through the RIS3CAT2016 project SIMBIOTS (COMRDI16-1-0017) and the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (Institut de Robòtica i Informàtica Industrial) (MDM-2016-0656). Thanks to J.L. Rivero for designing the cover page.

IRI reports

This report is in the series of IRI technical reports.
All IRI technical reports are available for download at the IRI website
http://www.iri.upc.edu.