Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 106 (2022) 209–214

www.elsevier.com/locate/procedia

9th CIRP Conference on Assembly Technology and Systems

Innovative safety zoning for collaborative robots utilizing Kinect and LiDAR sensory approaches

Peter Chemweno[a]*, Robbert-Jan Torn[a]

[a]*Chair of advanced manfacturing, Faculty of Engineering Technology, University of Twente, Enschede 7522 NB, The Netherlands,*

* Corresponding author. *E-mail address:* p.k.chemweno@utwente.nl

Abstract

Safe collaboration between a robotic and human agent is an important challenge yet to be fully overcome in a manufacturing set-up. Existing strategies, including safety zoning are sub-optimal since they seldom fully exploit the capabilities of the collaborative robot (for repetitive tasks) and highly cognitive tasks (best suited for the operator). The recently released ISO 15066 standard for collaborative robots proposes varying safeguards, including force, speed and distance limiting functions. The latter is particularly attractive as it allows the robotic agent to adapt its operating behaviour in proximity of the operator and in instances likely to lead to safety hazards. This paper discusses strategies explored for implementing dynamic zoning in shared workspaces, considering the input speed/force of the robot as dependent on the distance between human and robot. Two main strategies were modelled, for implementing zoning. The first strategy explored integrating a LiDAR sensor, and utilising LiDAR data to dynamically map the separation distances between the operator and robotic agent. The second strategy explores an experimental setup utilising the Microsoft Kinect V2 sensor for capturing 3D point clouds, and in turn, detecting objects/agents and the proximity distance. In both instances, objects/agents were detected up to a separation distance threshold, considering error sensitivity below values of 0.1 meters. Both use cases were demonstrated using a Yumi robot and form the basis of future work towards dynamic workspace zoning.

Keywords: Safety zoning; collaborative robots; dynamic separation distance; LiDAR sensor;

1. Introduction

Robots play a significant role in manufacturing processes, and this is expected to increase in the future. In recent years, there is a drive towards integrating collaborative robots for agile and intricate manufacturing tasks [1]. However, integration of collaborative robots in manufacturing cells is still limited because of safety concerns. Most safety hazards arise in shared workspaces, areas in which a human operator collaborates with a robotic agent [2]. In shared workspaces, an example of task allocation may include a division between cognitively demanding (best suited for human operators) and repetitive and physically demanding tasks (most optimally suited for collaborative robots) [3]. Still, facilitating shared workspaces requires safeguards to minimize the risk of injuring human operators.

Current safety standards for shared workspaces predominantly focus on static approaches for realizing safety zoning, including caging away the robot from human presence [4]. For instance, the ISO/TS 15066:2016 standard prescribes four safety requirements for robots and robotics devices, and more specifically, collaborative robots. These four conditions include [5]:

1. **Power and force limitation:** Limits the forces exerted by the robot manipulator to level below thresholds that would be harmful.
2. **Safety Monitored Stop:** Includes measures to a human's presence in a collaborative workspace.

10.1016/j.procir.2022.02.180

3. **Hand Guiding:** Hand guiding robot's motion is only possible using direct input of the operator.
4. **Speed and safe distance separation:** Influence the robot motion and adapts the manipulator speed when an operator enters the shared workspace.

In designing safe shared workspaces following the ISO/TS 15066:2016 standard, speed and distance separation is tantamount to implementing safety zoning. This can be realised in multiple ways, such as: a) by adding virtual barriers (laser scanners and motion detectors) which the robot is not allowed to cross, or b) the initiation of a safety monitored stop that lets the robot come to a full stop when a human comes within an unsafe defined proximity. In addition, in designing of collaborative robot operations, complementary safeguards can be embedded such as safety-rated axis manipulation of the robotic manipulator, force sensing capabilities, establishing safe force and power thresholds for the robotic manipulator. Overall, safety zoning integrates several safeguards mentioned in the ISO 15066, including safety-oriented positioning of collaborative robots in shared workspaces (clause 5.3), workspace access and safety clearance [2, 6].

In contrast to static approaches to safety zoning, literature in recent years has been moving towards embedding dynamic approaches to safety zoning, such as gesture-based human-robot interaction [7, 8]. In dynamic safety zoning a collaborative robot adapts its' behaviour depending on the proximity of the human operator. Implementing design safeguards may include, the initiation of a safety monitored stop whereof the robot comes to a full stop, or an adaptive decrease in operating speed depending on the separation distance between the robot and operator. While static safety zoning improves protection for operators, dynamic safety holds in addition great potential for minimizing downtime.

However, there remains a gap in the literature on guidelines for implementing dynamic zoning in practice. Especially, much uncertainty still exists in designing adaptive robot behaviour of industrial robots in shared workspaces. Specifically, the behaviour of robots in implementing design safeguards such as dynamic safe distance separation, and manipulator speed motion in shared workspaces. This paper addresses this challenge by proposing a sensory and computer vision approach for implementing dynamic safety zoning. An experimental set-up is established integrating 3D sensors, including depth cameras, LiDAR (for variable distance determination and 3D cloud-points object mapping). In the set-up, the sensors are embedded in a YuMi robot, and programming implemented via the 'open ABB-driver' and ROS (Robotic Operating System) [9].

The structure of this paper is as follows: Section 2 presents the current state of research, followed by a discussion of the methodology in Section 3. Implementation of the methodology and results are presented in Section 4, while discussion and aspects for future work is discussed in Section 5.

2. Current state of research

2.1. LiDAR point cloud triangulation for terrestrial mapping

LiDAR (light detection and ranging) point clouds for spatial workspace mapping and variable distance determination presents an interesting opportunity for workspace zoning. Usually, the point cloud (both 2D and 3D) consists of many data points, representing a position orientation between an object and position of the LiDAR sensor. The point clouds are beneficial since they allow triangulation of an object via machine learning approaches, including triangulation-based spatial clustering [10].

Since doing a full analysis of every point in the point cloud would result in high computation times, different methods are suggested for triangulating position and orientation of an object in the spatial workspace. Often, this is through removal of outliers and filter out 'point noise' especially from large 2Dor 3D LiDAR point clouds [11]. The removal consequently improves and speeds up image process and object triangulation are workspaces shared by an industrial robot and human operator.

One specific challenge relates to dynamic triangulation, especially, computing the distance between a moving robot and a moving person. This is because, relevant point locations for the robot and human operator are moving, hence the points lack a consistent reference location in different point clouds. Therefore, the focus of the analysis is on points that change with different point clouds, indicating dynamic movement of objects (operator and robot) in the spatial space. Other points may be considered irrelevant and filtered from the point clouds, prior to computing variable distances between objects.

To distinguish between moving localisation points in the point clouds, the pose map optimisation method is suggested for LiDAR cloud points [12]. In this method, via a 3D LiDAR Simultaneous Localization and Mapping (SLAM), the spatial environment is scanned, and relevant points triangulated from the cloud points. The SLAM is useful for orienting/triangulating the position of the sensor vis a vis a moving object such as a moving robotic manipulator, or operator. The pose map allows new distance measurements to be compared to a reference test set (zero measurement localisation) and locating non-coherent localisation points. The reference test set in this instance, may indicate localisation of the robot, without the human in view.

Shan, Li [13] applies a zero-measurement approach integrated with SLAM. In their study, the zero measurement for depth estimation. The 3D points together with their localisation coordinates references the position of an unmanned vehicle (UV) in the spatial environment, without presence of other agents or objects. For every point in a new point cloud as the UV moves, the distance to the 'zero-point' is measured. However, the measurement is challenging because of new objects moving into, or out of the spatial environment. For a collaborative environment, this may corelate to movement of an operator into the shared workspace (LiDAR sensory zone), while disappearing points, would corelate to points beyond the sensory capability of the depth camera, e.g., objects behind the operator or the robot.

A similar method is presented in an early research [14]. They discuss practical implications for three different systems: Airborne (ALS), terrestrial static (TLS) and mobile mapping systems (MMS). The TLS is closely related to safety zoning challenges, which is the focus of this paper and concerns indoor terrestrial mapping and localisation triangulation.

On the other hand, Bouali, Oommen [15] looks at stationary sensor applications of LiDAR, where differences in distances measured between different point clouds are mapped via a 'change detection' method. However, the approach is computationally intensive, hence less interesting for this study.

2.2. Approaches for distinguishing between different clusters in a point cloud

To determine the distance between human and a collaborative robot, it would be beneficial to distinguish in a point cloud, points that belong to the same object. For instance, distinguish between points belonging to a human operator and a robot. Wellhausen, Dubé [16] suggest an approach in which they determine distance between points in a measurement, and in this way, detect a unique cluster of points. A point cluster corelates to a grouping of points, and such point clusters are formed on places where objects are detected.

The premise of mapping point clusters is that since all points on an object occupy a similar relative distance to the sensor, the density of the point cluster is assumed as constant.

To measure the distance between point cluster, a threshold value is defined, to distinguish between point cluster depending on whether the distance measure exceeds or is below the defined threshold. This approach is also used in this study to distinguish between different objects, such as human and robot. A caution, however, the point cluster does not yet describe the object, i.e., whether it represents the robot or operator.

To address this challenge, and distinguish between objects based on point clusters, Ahmadi, Meghdari [17] suggested an approach where a 2D LiDAR sensor connected to a moving robot is used to retrieve spatial data and directed to a Person tracker module. Next, using a Kalman filter, measurement of point clusters is combined to retrieve more elaborate spatial features of the moving robot (via density measurements). However, if different angle views are required, the density measures may not be accurate, which is a draw back.

A different study reported in Kenk, Hassaballah [18] applies the point cluster measurement principle. A use case of a Hokuyo UST-20LX 2D Lidar sensor is illustrated. A Point Cloud Library (PCL) is applied based on data retrieved from a depth camera, with the PCL used as the basis of detecting and locating humans in the spatial environment of the robot. This study is similar to [16] where distance measurement relative to adjacent points in a cloud is applied.

Furthermore, to retrieve the PCL and determine point clusters corelating to a human or robot, the study of Kenk, Hassaballah [18] uses a depth vision camera. They first transform the 3D view of the point clusters, to a 2D view via the principal component analysis. In the 2D view, they detect objects from point cluster clouds, based on user-defined threshold (where points above the set-threshold corelates to an object is the terrestrial space). Cluster points below the threshold, are eliminated as noise.

After filtering the noise, the 2D point clusters are transformed back to 3D cloud points. This approach is implemented in this study. This is because the distance from the robot to the sensor is a known variable, and a spatial shared workspace area (m^2) can be estimated. Only the distance between the moving robot and human operator remains dynamic.

From the introduction and current state of the art, dynamic safe distance separation appears as an important need, especially for collaborative workspaces. Especially, an important consideration is having a feasible approach for estimating distance between agents and objects in the shared workspace. The approach where the distance between data points is estimated in a point cloud, and between point clusters is an important first step, which we implement in this study. This adds to the body of literature, for implementing safe distance separation, discussed in studies such as [19, 20].

Our contribution lies in implementation of safety zoning via utilising spatial data captured via the Kinect and LiDAR sensors.

3. Implementation for safety zoning

3.1. Test set-up

For safety zoning, the ABB YuMi robot is used, with 7 degrees of freedom manipulators. The YuMi is fitted with two sensors for range detection, the Microsoft Kinect V2 and LiDAR. The depth camera of the Kinect sensor is the main feature used for safety zoning. In the experimental set-up, the depth is measured with the Time-of-Flight (ToF) principle discussed in [21]. A single Kinect sensor is used with a horizontal and vertical field of view of 70.6 by 60 degree-angles.

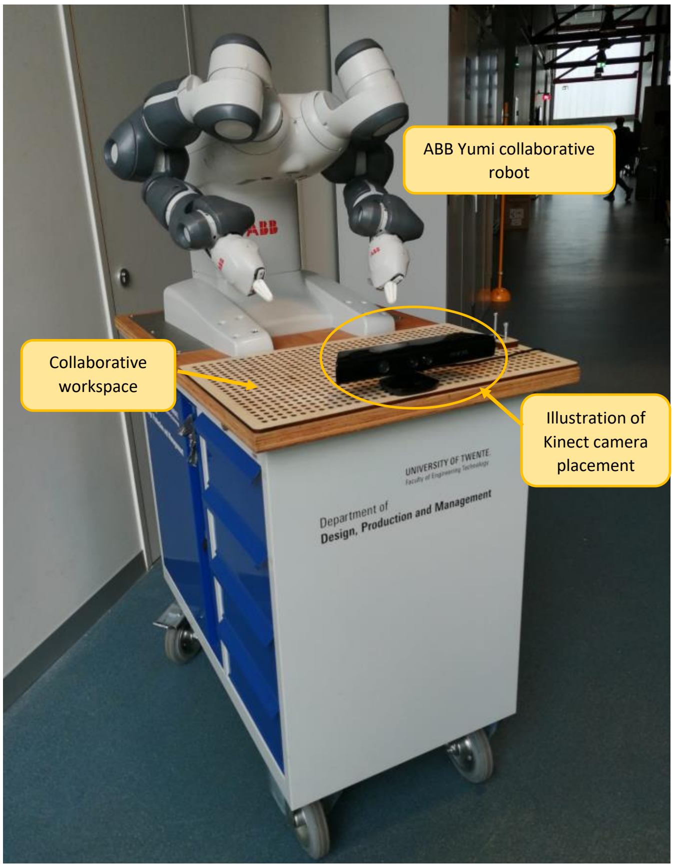
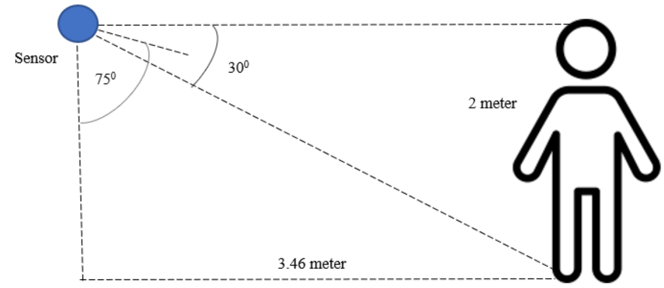


Fig. 1. Experimental set-up with the YuMI robot showing location of the Kinect Placement.



3.2. Implementation of the Kinect

The Microsoft Kinect V2 sensor generates 3D depth images, consisting of point clouds on which their RGB values are projected. The Kinect is programmed to communicate with Ubuntu using the Libfreenect2 library [22]. The point cloud can be visualized in RViz, a 3D visualizer in ROS. In Figure 1, the positioning of the Kinect and the mounting of the YuMi are shown. Ideally, the Kinect is placed on the workbench while careful not to obstruct view in the shared workspace (of operator and YuMi).

To reduce noise and the processing time, filters are applied to the point cloud using the Point Cloud Library (PCL) for both 2D/3D image and point cloud processing. Filters are applied with a segmentation algorithm used from the PCL library. Figure 2 illustrates the effects of applying the filters, displayed through the RViz plug-in ROS.

The first filter is a passthrough filter, which allows a user to specify a volume of points to be included or excluded from the total point cloud. This reduces the number of points in the cloud. In the simulation, the width of the point cloud corelates to the width of the walking area in front of the YuMi. The height is reduced to remove the floor and ceiling, in essence, significantly improving the performance of the segmentation algorithm, thus enhancing detection of obstacles.

Furthermore, three passthrough filters were used in sequence, consideration variations in distances in width (x-axis), height (y-axis) and depth (z-axis). The variations, determined as trial and error considering the size of the collaborative workspace, ranged between $-0.3<x<1.5$ meters, $-1<y<0.5$ metre, and $0<z<7$ metres.

Next, a voxel grid filter was applied to further reduce the number of points. This filter creates a grid of 3D boxes of the point clouds, where multiple points are replaced with single 3D boxes. By calibration and following a trial-and-error approach to visualise the objects in the workspace, width (x), height (y), and depth (z) of each box was set to 0.05 meters.

After the filtering, segmentation is applied via a plane model segmentation approach, which tries to find a group of points in the point cluster cloud that could represent a plane. In this study, multiple planes were detected, and distinguished by different assigned colours. A script was implemented for applying the segmentation and assigning specific colours to the point clusters [23]. This script was adapted to include the desired passthrough and voxel grid filtering, and to output the minimum distance to the closest cluster.

Figure 2 illustrates the steps applied from filtering, right through to segmentation and the related changes to the object views. Here, the point clouds and the effects of the filtering step are shown. The width (x), height (y) and depth (z) axis are

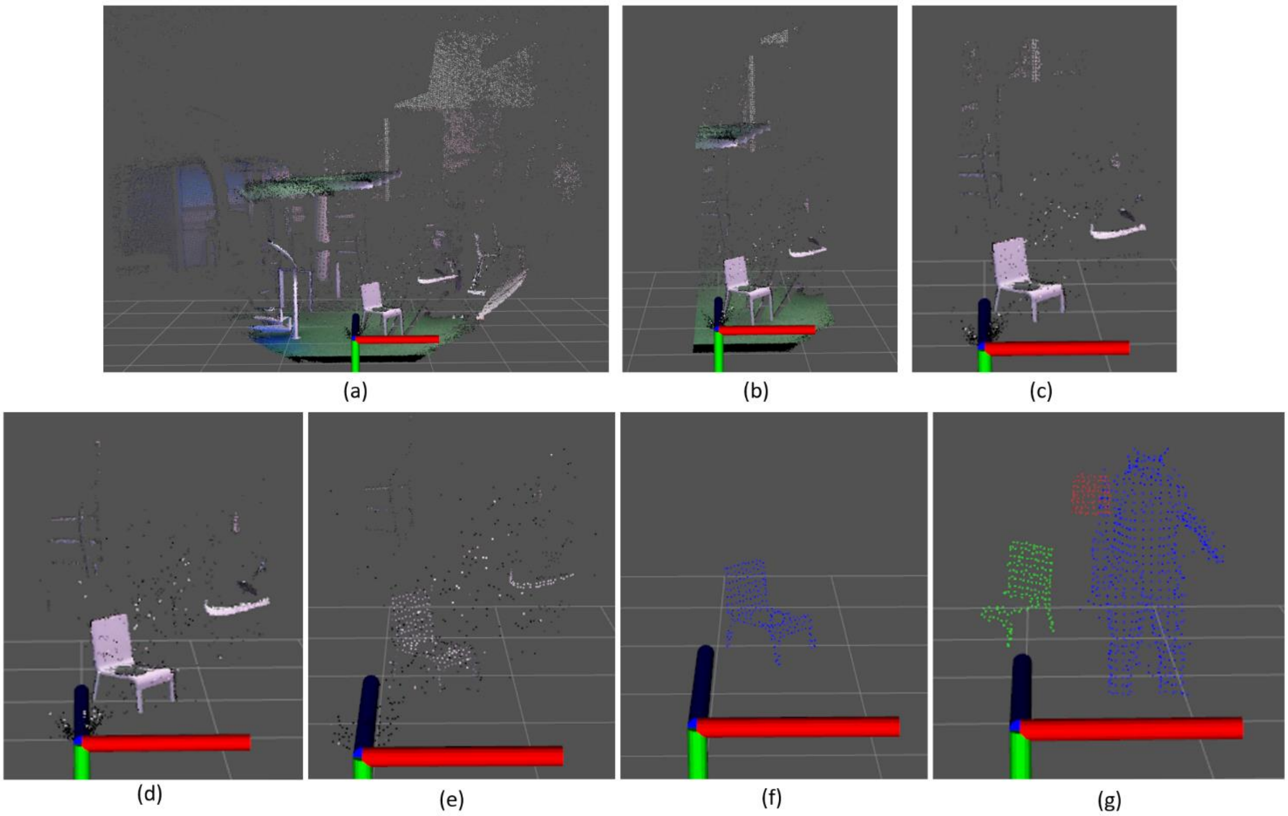


Fig. 2. Point cloud from the Kinect and the results from filtering: (a) represents the full point cloud, (b), (c) and (d) includes a passthrough filter for width, height and depth respectively, while (e) includes a voxel grid filter, (f) with inclusion of segmentation algorithm. In (g), additional objects are included in the detection zone and successfully detected.

represented by the red, green, and blue axis, respectively. Figure 2a is the full point cloud as received from the Kinect. Figure 2b applies the passthrough filter on the x-axis. Figure 2c applies the passthrough filter on the y-axis on the point cloud of Figure 2b. Figure 2d adds the z-axis passthrough filter. Figure 2e applies the voxel grid filter
after all the passthrough filters. In Figure 2f, the segmentation is added, and the chair is detected. In figure 2g another chair in the back and a person in front is added, which are indicated in separate clusters.

The minimum distance to the nearest cluster is published from the =pcl segmentation node as a single numerical value to the topic =minimum distance. This value can be used as described in Section 3.2. The minimum distance according to the Kinect is verified by a measurement tape. A cardboard box with a width, height, and depth of 0.4, 0.6 and 0.3 meters respectively, is used. The box is placed at a measured distance when the Kinect values are read. A box placed at 600 cm gives Kinect measurements between 594 and 599 cm. At 500 cm, the Kinect finds distances between 498 and 499 cm. At 400 cm, the values are between 394 and 401 cm. For a distance of 250 cm, the measurements are between 243 to 247.

A smaller distance of 180 cm results in most values between 163 and 177 cm, but some outliers are at 130 cm for one or a few measurements. To detect obstacles such as a human, the accuracy seems good enough. Because the control of the cobot was not successful, the implementation stopped here. To reduce fluctuations in the distance measurement, the average of several measurements can be used. This will increase the response time of the system and is discussed in Section 4.

The filtering and segmentation approach was implemented in ROS (Robot Operating Studio). ROS provides nodes, with different nodes implementing aspects of the filtering and segmentation. For instance, reading Kinect data (via the Libfreenect2 and the IAI Kinect2 bridge). Furthermore, this node allows data transfer of a point cloud. A segmentation node implements passthrough and voxel grid filtering, and after filtering, the node applies the segmentation algorithm to find the minimum distance to the closest cluster. The segmentation node further quantifies the minimum distance, containing a single numerical value representing the minimum distance to the nearest cluster.

Other nodes in ROS implementing the approach described in Section 3.1 the RViz, which displays filtered and segmented point cloud in 3D images. Different clusters (representing varying objects, or agents in the shared workspace) are assigned different colours, while a 'static transform' publisher node, are used to generate coordinate frames in the spatial space, with the RViz displaying the point cloud with the desired position and orientation [24].

4. Discussion

This study illustrated a methodology for implementing the LiDAR and Kinect V2 within ROS for mapping spatial distances between agents (human and robot), plus objects in collaborative workspaces. The sensors can detect and quantifying the distance between multiple objects, with a distance up to 6 meters, within an error of below 10 cm.

Research is still ongoing to find the range of sizes of objects the Kinect and LiDAR can detect, since this range is limited by the resolution of the Kinect, and further reduced by the voxel grid filter. Although a static use case is presented, a next step is implementing algorithms to dynamically track and label objects, and further, anticipate the direction the agent or object will move to avoid collisions. Studies in this direction, which could be form the focus for the next steps of this study, include intelligent path planning controllers potentially integrated in ROS [25].

As opposed to laser scanner, the Kinect and LiDAR sensing potentially allows distance measurement in a 3D spatial environment, characteristic of collaborative workspaces. This presents an opportunity for implementing dynamic detection, hence, more optimal adaptation of robot behaviour and avoid hazardous collisions in collaborative workspaces.

5. Conclusions

In this study, we propose an approach for implementing a dynamic safety zoning utilising Kinect V2 and LiDAR sensors and ROS. The latter is used to stream a 3D point cloud from the Kinect via passthrough filters, and segmentation. We note a detection of up to 6 meters, and feasible implementation of a user-defined minimum distance threshold to activate the robot to decelerate or stop. Future research will focus on quantifying optimal filter parameters, and location of the Kinect and LiDAR sensors. The study will also extend to integrating dynamic 3D image frames, as a static frame is currently implemented.

6. Acknowledgements

We would like to acknowledge Roy Burger and Bram van Eijk, Faculty of Engineering Technology, University of Twente for their contribution to this study, setting up the experiment and implementing the safety zoning approach.

References

[1] Pearce, M., Mutlu B, Shah J, Radwin, R. Optimizing makespan and ergonomics in integrating collaborative robots into manufacturing processes. 2018. **15**(4): IEEE transactions on automation science and engineering. p. 1772-1784.
[2] Chemweno, P., L. Pintelon, and W. Decre, Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. 2020. Safety Science. 129: p. 104.
[3] Dixon, J., B. Hong, and L.Wu. The robot revolution: Managerial and employment consequences for firms. Management Science. 2021.
[4] Kang, S. and K. Kim. Motion Recognition System for Worker Safety in Manufacturing Work Cell. in *2018 18th* International Conference on Control, Automation and Systems (ICCAS). 2018. IEEE.
[5] ISO 15066: Robots and robotic devices-Collaborative robots. International Standard Organisation. 2016.
[6] Matthias, B. ISO/TS 15066-Collaborative Robots: Present Status. in European Robotics Forum, Vienna, Austria. 2015.
[7] Neto, P, Simao, M., Mendes, N., Safeea, M. Gesture-based human-robot interaction for human assistance in manufacturing. 2019. International Journal of Advanced Manufacturing Technology. 101(1): p. 119-135.
[8] Mazhar, O., Navarro, B., Ramdani, S., Passama, R. and Cherubini, A. A real-time human-robot interaction framework with robust background invariant hand gesture detection. 2019. Robotics and Computer-Integrated Manufacturing, 60, pp.34-48.
[9] ABB, Robotics Open ABB.
[10] Kim, J. and J.. Cho, Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3D point clouds. 2019. Sensors. 19(18): p. 3926.

[11] Balta H, Velagic J, Bosschaerts W, De Cubber G, Siciliano B. Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. IFAC-PapersOnLine. 2018 Jan 1;51(22):348-53.
[12] Egger P, Borges PV, Catt G, Pfrunder A, Siegwart R, Dubé R. Posemap: Lifelong, multi-environment 3d lidar localization. In2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018 Oct 1 (pp. 3430-3437). IEEE.
[13] Shan Z, Li R, Schwertfeger S. RGBD-inertial trajectory estimation and mapping for ground robots. Sensors. 2019 Jan;19(10):2251.
[14] Gressin A, Mallet C, Demantké J, David N. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS journal of photogrammetry and remote sensing. 2013 May 1;79:240-51.
[15] Bouali EH, Oommen T, Vitton S, Escobar-Wolf R, Brooks C. Rockfall hazard rating system: Benefits of utilizing remote sensing. Environmental & Engineering Geoscience. 2017 Aug 1;23(3):165-77.
[16] Wellhausen L, Dubé R, Gawel A, Siegwart R, Cadena C. Reliable real-time change detection and mapping for 3d lidars. In2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR) 2017 Oct 11 (pp. 81-87). IEEE.
[17] Ahmadi E, Meghdari A, Alemi M. A socially aware SLAM technique augmented by person tracking module. Journal of Intelligent & Robotic Systems. 2020 Jul;99(1):3-12.
[18] Kenk MA, Hassaballah M, Brethé JF. Human-aware Robot Navigation in Logistics Warehouses. InICINCO (2) 2019 Jul (pp. 371-378).
[19] Marvel JA, Norcross R. Implementing speed and separation monitoring in collaborative robot workcells. Robotics and computer-integrated manufacturing. 2017 Apr 1;44:144-55.
[20] Byner C, Matthias B, Ding H. Dynamic speed and separation monitoring for collaborative robot applications–concepts and performance. Robotics and Computer-Integrated Manufacturing. 2019 Aug 1;58:239-52.
[21] Wasenmüller, O. and D. Stricker. *Comparison of kinect v1 and v2 depth images in terms of accuracy and precision*. in *Asian Conference on Computer Vision*. 2016. Springer.
[22] Echtler F. libfreenect2: Release 0.2. 2016.
[23] GITHUB, Plane segmentation script. Accessed on 7th August, 2021.
[24] Kam HR, Lee SH, Park T, Kim CH. Rviz: a toolkit for real domain data visualization. Telecommunication Systems. 2015 Oct;60(2):337-45.
[25] Liu Y, Jing H, Liu X, Lv Y. An improved hybrid error control path tracking intelligent algorithm for omnidirectional AGV on ROS. International Journal of Computer Applications in Technology. 2020;64(2):115-25.