Industrial Environment Mapping Using Distributed Static 3D Sensor Nodes

Atle Aalerud[1], Joacim Dybedal[1], Erind Ujkani[1], and Geir Hovland[1].

[1]Mechatronics Group, Faculty of Engineering and Science, Department of Engineering Science, University of Agder, N-4898 Grimstad, Norway. atle.aalerud@uia.no

Abstract **– This paper presents a system architecture for mapping and real-time monitoring of a relatively large industrial robotic environment of size 10 m × 15 m × 5 m. Six sensor nodes with embedded computing power and local processing of the 3D point clouds are placed close to the ceiling. The system architecture and data processing is based on the Robot Operating System (ROS) and the Point Cloud Library (PCL). The 3D sensors used are the Microsoft Kinect for Xbox One and point cloud data is collected at 20 Hz. A new manual calibration procedure is developed using reflective planes. The specified range of the used sensor is 0.8 m to 4.2 m, while depth data up to 9 m is used in this paper. Despite the fact that only six sensors are used and that the Kinect sensors are operated beyond the specified range, a benchmark of the accuracy compared with a Leica laser distance meter demonstrates an accuracy of 10 mm or better in the final 3D point cloud.**

A.1 Introduction

The use of 3D sensors in robotics and coordinated machine control applications is growing. Some applications are the detection of humans in hazardous areas as well as collision detection and avoidance for multi-machine control. Many state-of-the-art 3D sensors generate large datasets in forms of point clouds. For example, the Microsoft Kinect for Xbox One (Kinect V2) sensor and the RealSense camera from Intel. These sensors have a limited range and field of view. Hence, in order to cover a larger volume, several of these sensors must be used to cover the entire volume, see for example [1]. One challenge with 3D sensors and point clouds is the large datasets generated. For example, the Kinect V2 sensor requires a bandwidth of several gigabits per second, and there is a limitation on how many such sensors can be connected to one computer via USB3. Hence, there is a need for embedded solutions which process data locally at the sensor nodes. Then, these nodes only transmit reduced, processed data to a central node which in turn collects the information from all the sensor nodes. The central node can then build up the global 3D map and perform the application-specific tasks.

3D sensors based on structured light are not suitable for use in applications requiring multiple sensor nodes, because of interference of the emitted structured light between the sensor nodes. In [2] a solution was presented where vibration motors were attached to each sensor to mitigate the problem. However, this solution reduces the accuracy of the sensors as well as increases complexity. In our work 3D sensors based on structured light are avoided and sensors based on the time-of-flight principle are used instead.

3D mapping using multiple depth sensors is an intensively active area of research. However, much of the research is currently focused on non-stationary sensors, i.e. typically for automotive use such as [3]. Other proposed methods consist of fixed position sensors where only the orientation is updated to increase the total field of view of the sensor such as [4], where a 2D Light Detection And Ranging (LiDAR) sensor is reoriented during capture. However, increasing field of view in this manner also reduces the frame rate. Multiple RGB-D cameras at fixed positions and orientations for environment mapping at short range are used by [5] and [6].

In [7], a potential industrial use-case is described for the work presented in this paper. Offshore drilling rigs are using increasingly more automated control systems to reduce the number of operators needed on the drill floor. The distance between the well centers such as described in [7] is typically ten to twelve meters, which matches well with the distances considered in this paper. Accurate and real-time 3D point cloud data would enable the development of new applications. For example, people detection on the drill floor, detecting failure of encoders used for machine positioning and thus redundancy for autonomous control systems.

This paper will use RGB-D sensors beyond specified recommended range to see if they still provide enough accuracy for low-resolution industrial safety applications.

A.2 Experimental Setup

A.2.1 Environment

In this paper, the industrial environment to be mapped is one of the Robotics Labs at the University of Agder depicted in Figure C.1. This lab functions as a large robotic cell where both industrial robots and humans need to operate. Thus, it is crucial that the positions of the human workers are known so that safety systems can be developed for the robots. A map of the space without equipment is shown in Figure A.4 where it can be seen that the floor space is approximately $10\,\mathrm{m} \times 15\,\mathrm{m}$. Subtracting walls yields a polygon shaped floor surface of approximately $137\,\mathrm{m}^2$. The ceiling height is above five meters, but the focus of this paper is to detect personnel at walking height. Thus, the volume from the floor up to approximately two meters should be fully mapped. To cover the entire worker if arms are raised, or if the person is jumping, climbing or similar is currently not part of the scope.

A.2.2 Hardware

To map the volume described above, multiple sensors must be used. In this experiment, six Kinect V2 are used. These are RGB-D cameras that give depth measurements based on an active Infra Red (IR) sensor using the time-of-flight principle. The depth cameras have a specified operating range of $0.8\,\mathrm{m}$ to $4.2\,\mathrm{m}$, an optical field of view of $70°(\mathrm{H}) \times 60°(\mathrm{V})$ and an image resolution of $512\,\mathrm{px} \times 424\,\mathrm{px}$ [8]. Even if it is well beyond the specified range, the sensors are used with a range up to nine meters in the experiment presented in this paper so that the entire required volume can be mapped. Note that the specified range of the Kinect V2 sensor applies to reliable human joint tracking. However, in [9] it

Figure A.1: A Robotics Lab at the University of Agder is used as the example case of an industrial environment. The lab consists of two rail-mounted ABB IRB4400 robots, one ABB IRB2400 robot mounted on a GÜDEL gantry and a processing facility.

is documented that point cloud data is obtainable even at larger distances. Further, [9] states that the standard deviation of the depth measurements increase rapidly from nine meters and that darker colors yield lower depth accuracy than brighter colors in general, as expected for this type of sensor.

Each of the described cameras are connected to an NVIDIA Jetson TX2 Development board. These boards process the sensor data before sending a compressed stream via Gigabit Ethernet to a Personal Computer (PC). This PC is equipped with an Intel Core i5-2500 Central Processing Unit (CPU), 8 GiB system memory and an NVIDIA GeForce GTX TITAN Graphics Processing Unit (GPU), which in turn has 2688 CUDA Cores and 6 GiB memory.

A.2.3 Software framework

The communication between the sensor nodes and the central PC is handled by the open-source Robot Operating System (ROS) [10] running on Ubuntu 16.04. An illustration of how the ROS topics and nodes are connected is given in Figure A.2. The ROS package IAI Kinect2 [11] handles the sensor input in the node */snN* where *N* is the sensor number. It uses the rectified depth image- and camera information topic to generate the point cloud topic */snN/sd/points_nocolor*. This point cloud topic is read by a compression node, */snN_compression*. As described in [12], the compression node transforms the point cloud into the global coordinate system using the given position and orientation of the IR camera. Further, it filters the data using a crop filter to remove non-relevant points before applying an octree-based compression scheme. This returns the topic */snN/wp3/pc2_compressed* which is a serialized point cloud that also contains an intensity value representing the

point density in the octree leaf nodes. On the master node, all the compressed streams are received by the */decompressor* node. This node reconstructs the point clouds including the intensity value. These point clouds are output as the topics */master/snN/pc2_decompressed*.

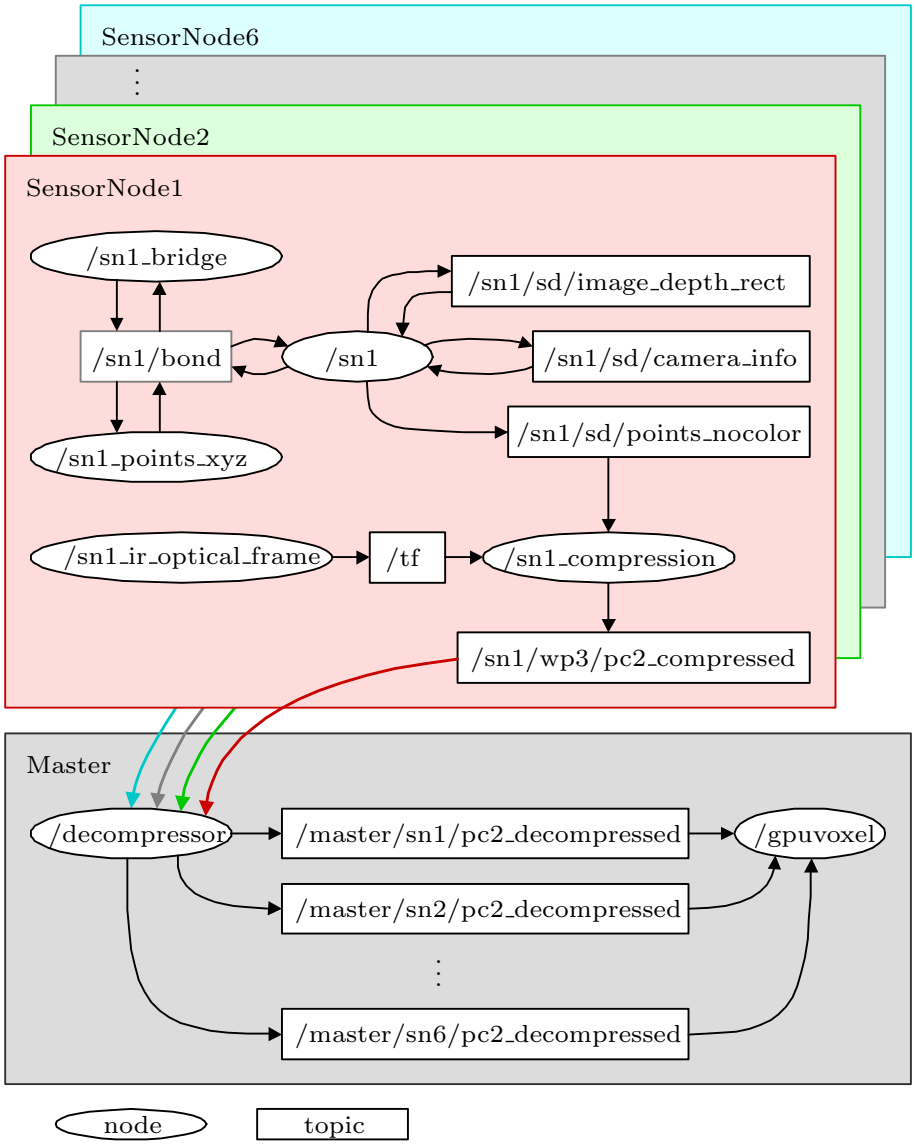


Figure A.2: Software setup using ROS architecture.

A.3 Sensor Placement and Calibration

The local coordinate system used for the Kinect V2 in this paper has its origin in the center of the IR sensor. The X-axis increases to the sensor's right, Y increases down and Z increases out from the sensor. Traditionally, the rotations around the X-, Y- and Z-axis are Roll, Pitch, and Yaw respectively. However, it is more intuitive seen from the local camera coordinates to use the order Pitch, Yaw, and Roll as the Z-axis is pointing forward. Therefore, to avoid confusion it is decided to only use the terms RotX, RotY and RotZ

for rotations around the X-, Y- and Z-axis of the Kinect V2 body frame as shown in Figure A.3.

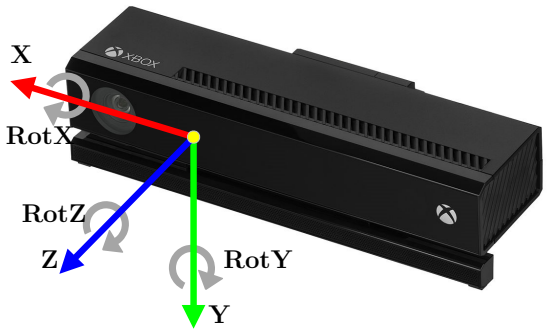


Figure A.3: The Kinect V2 coordinate system used. Illustration based on Xbox-One-Kinect, courtesy of Evan-Amos, Wikimedia Commons 2014.

The RGB-D Sensors are mounted with an approximate initial orientation of zero degrees RotZ and -150 degrees RotX. That is, facing 60 degrees downwards. Initial RotY is set dependent on sensor position. This initial sensor position (X,Y,Z) is measured using the Laser Distance Meter (LDM) Leica Disto D4a BT. With all sensors tilted 60 degrees down from horizontal, a vertical field of view between -90 and -30 degrees is obtained. Assuming a horizontal distance between two facing sensors of ten meters yields sensor overlap after five meters horizontally. Thus, the highest point of the intersection between the two cameras is

$$h = Z_s - \frac{10\text{m}}{2} \cdot \tan(30^\circ) = 5\,\text{m} - 2.887\,\text{m} = 2.113\,\text{m} \tag{A.1}$$

where Z_s is the mounting height of the sensors and h is the fully covered height. This height is sufficient to cover the described volume. However, to increase the coverage of the central working area between the robots, most sensors are tilted up approximately ten degrees giving a RotX of approximately -140 degrees.

Figures A.4 and A.5 show how the sensors cover the described volume using the updated RotX. All sensors are mounted at near five meters height and tilted down. Thus, the colored shapes in Figure A.4 indicate the approximate coverage of the individual sensor disregarding equipment shadows as seen from above. An arc and a line illustrate the far and near floor coverage limits respectively. The sensors are limited to a measurement depth of nine meters and the arcs show where this distance is reached at zero elevation. The lines close to the sensors illustrate where the lower field of view reaches the floor. The same field of view is illustrated vertically for sensor 1 to 4 in Figure A.5.

Calibration of the intrinsic parameters for all the cameras is covered in [13] where also an automatic calibration scheme for extrinsic parameters is developed. Meanwhile, calibration of the extrinsic parameters is performed manually. The motivation is to align the point clouds received from the cameras with each other and with the global reference system. Thus, the following manual calibration steps have been performed for each camera:

- Set initial transformations to values measured with the Leica LDM and set all rotations to zero.

- Place at least three white tables, or similar reflective surfaces with a known offset from the floor, behind each other at the bottom, middle and top of the camera's field of view. The planes should be horizontally centered in the camera image.
- Tune the RotX so that the depth measurements of the table surfaces are horizontally aligned. Then adjust the Z offset so the measurements match the actual table heights.
- Move the tables beside each other on the left, center and right of the camera's field of view. The planes should preferably be vertically centered in the image.
- Tune RotY so that the planes align horizontally. Verify that the heights of the tables are still correct and adjust Z offset if needed.
- Update RotZ, X, and Y to align with known coordinates and lines in the environment. An example can be seen in Figures A.7 and A.11 where yellow lines correspond to X=2 m, Y=2 and X=7 at height Z=0 in global coordinates.

Note that the RotX, RotY, and RotZ will affect each other unless calibrated in the described order. When all cameras are calibrated, the tables and other known geometries are used for verification of point cloud alignment between the sensor nodes.

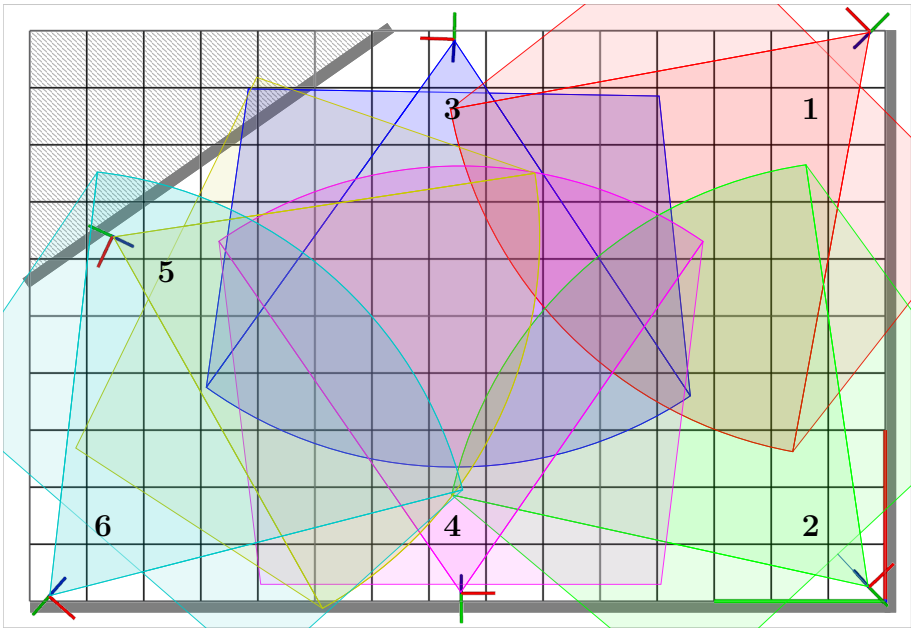


Figure A.4: Orthogonal map illustrating final placement and orientation of sensor 1 to 6. The global origin is located at the bottom right and the grid size is 1 m.

A.4 Experimental Results

It is possible to map most of the described volume using only six depth cameras limited at nine meters. The chosen placements and orientations yield acceptably small blind spots

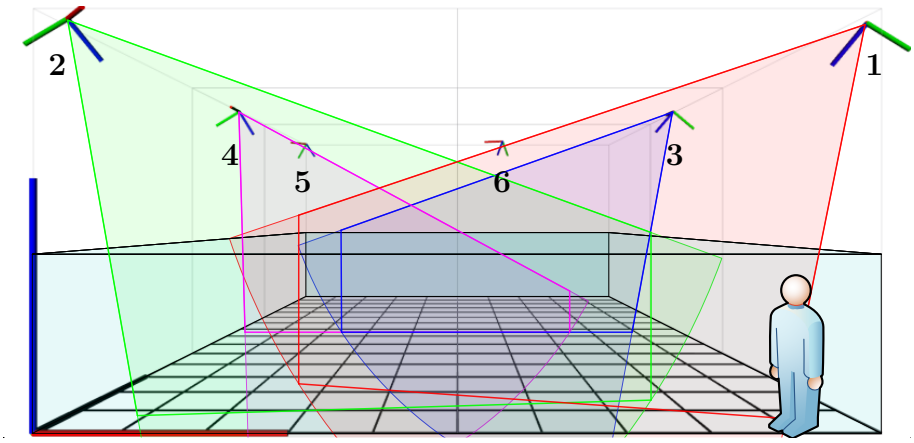


Figure A.5: A perspective of the mapped volume illustrating vertical coverage of sensor 1 to 4. Sensor 5 and 6 are omitted from the illustration for clarity. The blue area indicates the height found in Equation (A.1). A 1.8 m tall person is included for reference.

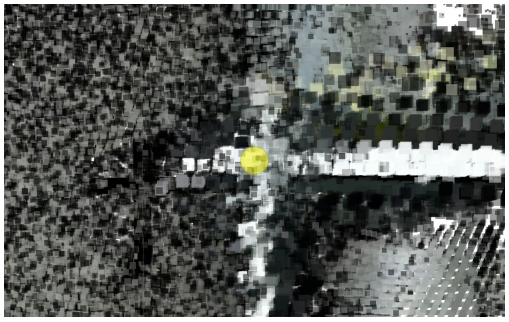
below the sensors. This can also be seen in Figures A.4 and A.5. The complete list of calibrated camera poses is given in Table A.1. Note that ROS static transform rotation order is RotZ, RotY, RotX according to Fig. A.3

Table A.1: Manually calibrated sensor positions in meter and orientations in degrees.

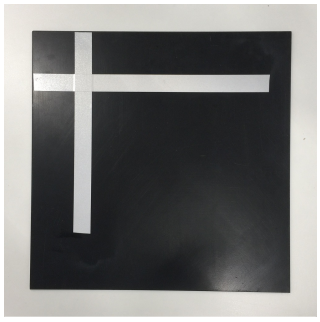
N	X	Y	Z	RotZ	RotY	RotX
1	9.975	0.260	4.898	44.977	-0.859	-138.598
2	0.247	0.285	4.960	-46.696	-1.146	-139.573
3	9.850	7.555	4.950	88.064	-1.375	-139.630
4	0.138	7.437	4.930	-89.668	-0.573	-148.396
5	6.395	13.550	4.954	154.870	-0.745	-138.255
6	0.087	14.660	4.990	-131.207	0.000	-134.645

The Kinect V2 cameras have rather noisy depth measurements at the implemented range. Thus, only a visual estimation of the experienced overall system accuracy is made. Here, the distance between the two gantry tracks and the two robot tracks are measured in both ends using the Leica LDM. These four distances cover multiple sensors and are in the region of particular interest. The same distances are evaluated in the point cloud data. Figure A.6 shows how the measurement positions are read from the combined accumulated point clouds. A cross, made with a two-centimeter wide retroreflective tape on a black plastic plate, is placed at the point to measure. As the black background is mostly invisible to the depth cameras, the retroreflective cross is easily found by visual estimation of the two crossing lines. It is also noted that the high reflectivity of the tape reduces noise in the depth measurements. This is observed as stable repeatable points at the tape position whilst other surfaces have more noise. The plates are also visible in Figures A.11(**c**) to A.11(**f**).

In Table A.2, known distances measured with the Leica LDM are shown. Further, the distance calculated between the estimated points in the combined accumulated point cloud is given. The error between the measured distance and the estimation is between two



(**a**) Point clouds



(**b**) Photo

Figure A.6: A reflective cross on a black background. Colorized combined point cloud with a yellow dot that indicates current read point in (**a**) and a photo in (**b**).

and ten millimeters. This is less than the accuracy of the individual sensor and is likely a result of visually estimating the coordinates in the combined point cloud as described. Nevertheless, it shows that the sensors are fairly well aligned with each other as sub-pixel accuracy is obtained.

Table A.2: Actual distance, D_{LDM}, estimated distance, D_{PC} and error in millimeters.

Description	$\mathbf{D_{LDM}}$	$\mathbf{D_{PC}}$	**Error**
Distance between gantry tracks start	6732	6722	10
Distance between gantry tracks end	6684	6686	-2
Distance between robot tracks start	1476	1484	-8
Distance between robot tracks end	1555	1548	7

The angular resolution between measurements is given by

$$\phi = \frac{\alpha}{s} \tag{A.2}$$

where α is the optical field of view and s is the amount of measurements captured by the image sensor. Thus, the distance between two measurements forming an isosceles triangle from the sensor is calculated by

$$r(z) = 2z \cdot \sin(\tfrac{\phi}{2}) \tag{A.3}$$

where $r(z)$ is the space between measurements and z is the distance from the sensor. Examples are listed in Table A.3.

Table A.3: Resolution as described by Equations (A.2) and (A.3). $r(z)$ is shown at maximum rated and maximum used distance.

Direction	$\boldsymbol{\alpha}$	**s**	$\boldsymbol{\phi}$	$\boldsymbol{r}$**(4.2 m)**	$\boldsymbol{r}$**(9 m)**
Horizontal	70°	512 px	0.137°	10.0 mm	21.5 mm
Vertical	60°	424 px	0.142°	10.4 mm	22.3 mm

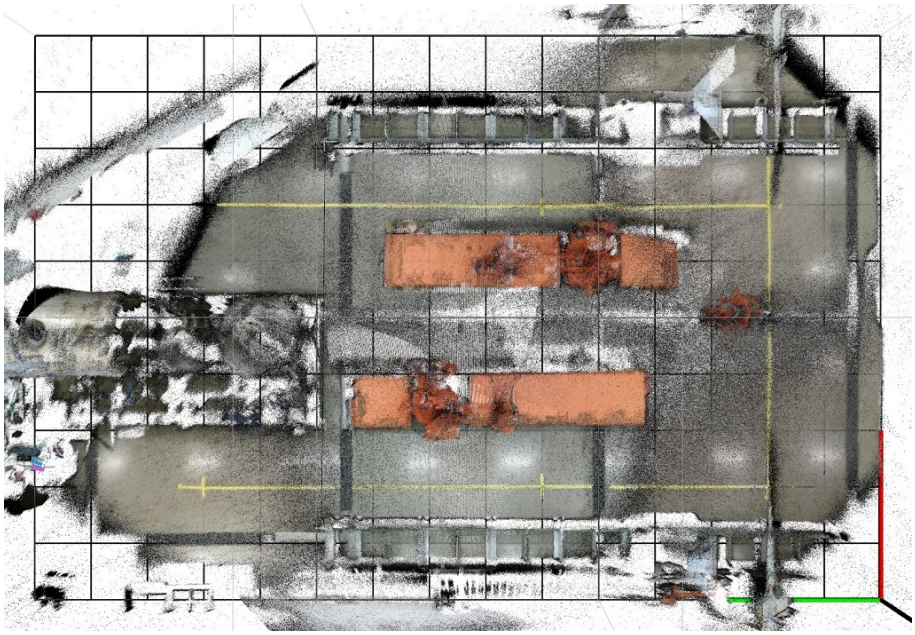


Figure A.7: The combined (accumulated) point clouds of all sensor nodes with colorized points. All six sensors are successfully aligned.

A.5 Discussion & Conclusions

In this paper, a relatively large industrial robot cell was mapped and monitored using six 3D sensor nodes. The sensor nodes were calibrated manually using white table planes in various positions, and accuracy was benchmarked using retroreflective tape on black background against measurements from a Leica laser distance meter. In [13], the sensor calibration in the same volume was performed pairwise using the Iterative Closest Point (ICP) algorithm. It turns out that the manual calibration procedure presented in this paper results in a better overall calibration (by a factor of approximately 10, ie. approximately 10 mm vs 10 cm.) compared with an automatic ICP based procedure. One reason for this result is the fact that the sensors have little overlap due to the relatively large volume and

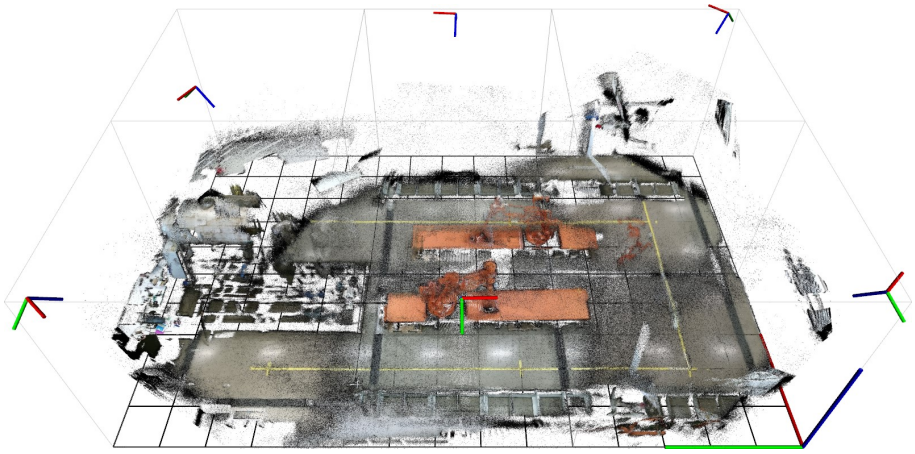


Figure A.8: The combined (accumulated) point clouds of all sensor nodes with colorized points. (Orientation is similar to Figure A.11(**d**).)

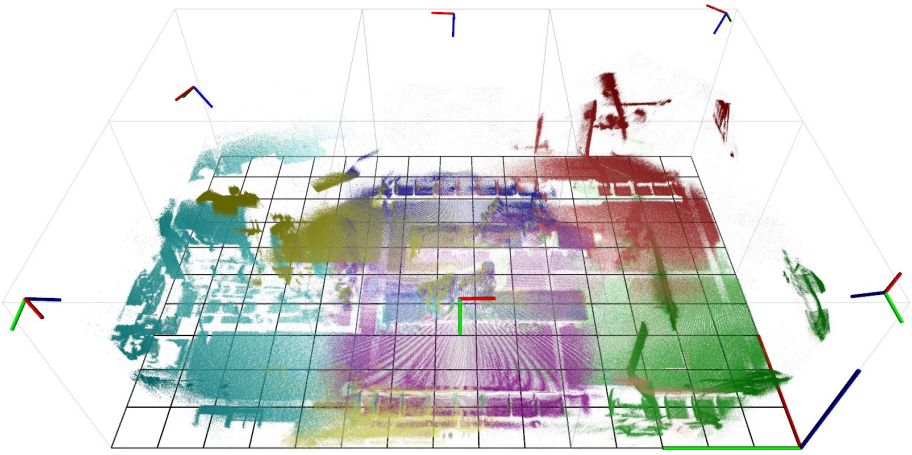


Figure A.9: The combined (accumulated) point clouds of all sensor nodes where points from each node have a separate color.

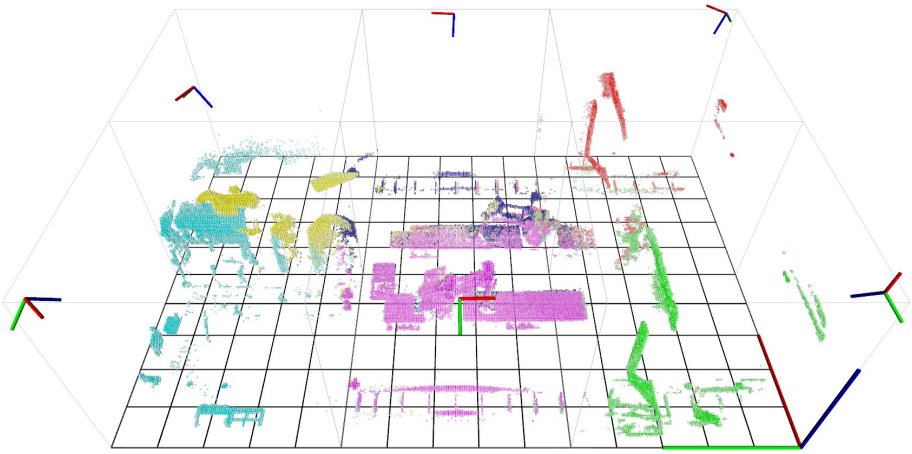
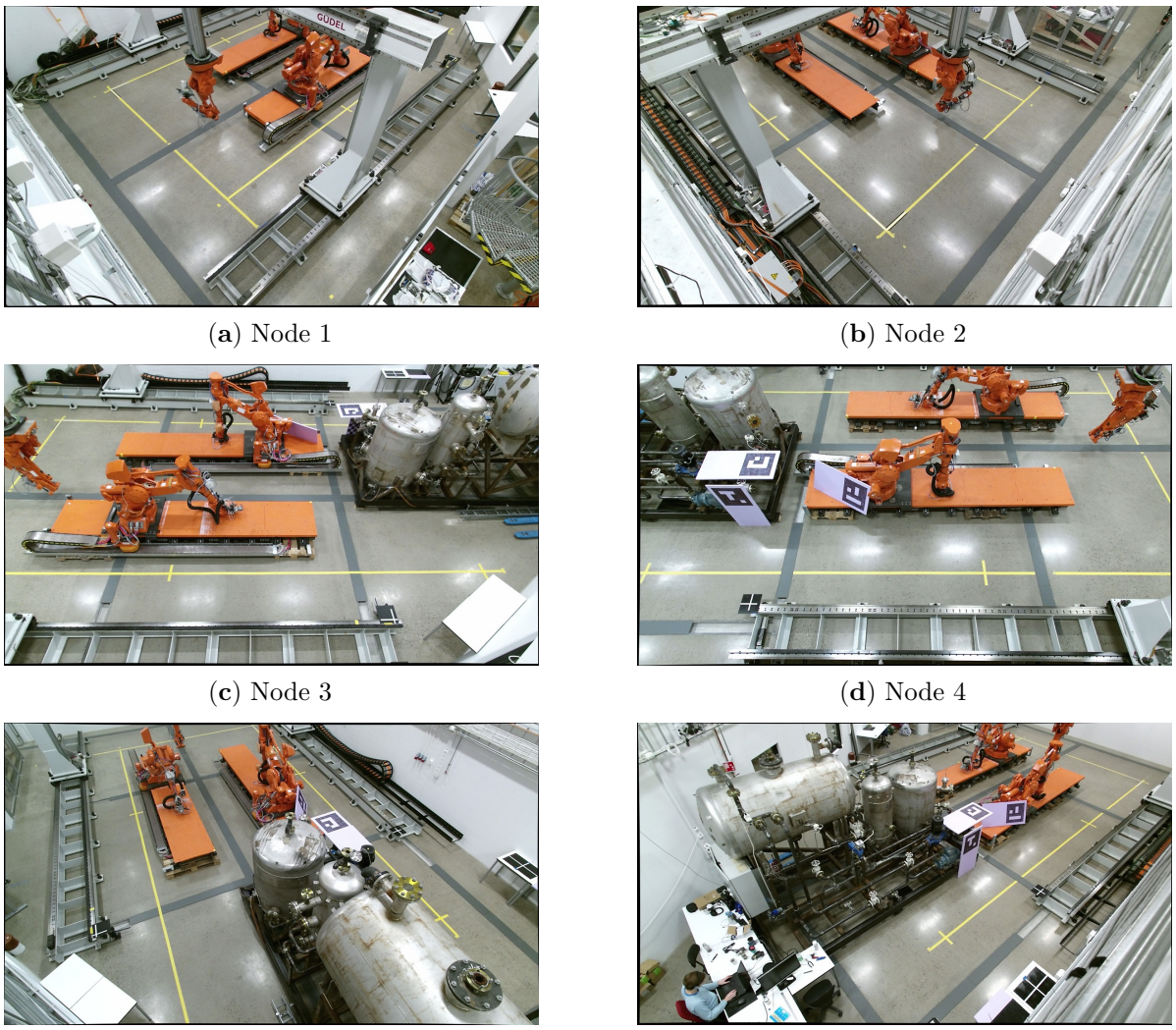


Figure A.10: The combined filtered (accumulated) point clouds of all sensor nodes where points from each node have a separate color. Intensity illustrates the number of points within the 4 cm × 4 cm × 4 cm grid.

overlap is required by the ICP algorithm to achieve good calibration. Another reason is the fact that the ICP algorithm works best when the angle between the sensor pair to be calibrated is relatively small, while in the work presented in this paper this angle is large (up to 180°).

Manual calibration is time-consuming. In the work presented in this paper, the manual calibration procedure took approximately two hours per sensor. In order to be able to scale up the system to include many more sensor nodes an automatic or semi-automatic procedure is desirable. In [12] it was found that the embedded processing solution of local 3D point cloud data was scalable up to a total of 440 sensor nodes using Gigabit Ethernet. One approach to reduce the overall time required for calibration would be to use an ICP based automatic calibration procedure as a starting point for the manual fine-tuning.

In this paper, the placement of the sensors was selected based on a manual judgment and not based on an optimization procedure. Optimal placement of 3D sensors in a volume is an area that requires more research. Especially if there are requirements such as redundancy where certain areas must be covered by at least two sensors. For example,



(**a**) Node 1 (**b**) Node 2

(**c**) Node 3 (**d**) Node 4

(**e**) Node 5 (**f**) Node 6

Figure A.11: RGB image as seen from the respective sensor nodes

when the large GÜDEL gantry moves towards the middle of the workspace it casts large shadows in the depth data. In order to overcome shadows, redundancy requirements must be specified for the regions of particular interest. In [1] an optimization method based on mixed integer linear programming and discretization of the volume into smaller cubes was used. That solution suffered from a large number of variables when linearizing nonlinear constraints.

Future work will focus on 3D sensor placement optimization, fast and accurate sensor calibration, and use of the 3D point cloud data in applications such as collision detection and avoidance.

Acknowledgment

The research presented in this paper has received funding from the Norwegian Research Council, SFI Offshore Mechatronics, project number 237896.

References

[1] J. Dybedal and G. Hovland, "Optimal Placement of 3D Sensors Considering Range and Field of View," in *IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics, AIM*. IEEE, 7 2017. doi: 10.1109/AIM.2017.8014245. ISBN 9781509059980 pp. 1588–1593.

[2] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'sense: reducing interference for overlapping structured light depth cameras," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. Austin, Texas, USA: ACM Press, 2012. doi: 10.1145/2207676.2208335. ISBN 9781450310154. ISSN 145031015X pp. 1933–1936.

[3] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robotics and Autonomous Systems*, vol. 83, pp. 299–311, 2016. doi: 10.1016/j.robot.2016.06.007

[4] P. Olivka, M. Mihola, P. Novák, T. Kot, and J. Babjak, "The Design of 3D Laser Range Finder for Robot Navigation and Mapping in Industrial Environment with Point Clouds Preprocessing," in *Modelling and Simulation for Autonomous Systems. MESAS 2016. Lecture Notes in Computer Science*, J. Hodicky, Ed., vol. LNCS 9991. Springer, Cham, 6 2016. doi: 10.1007/978-3-319-47605-6_30. ISBN 9783319476049. ISSN 03029743 pp. 371–383.

[5] J. Miseikis, K. Glette, O. J. Elle, and J. Torresen, "Multi 3D camera mapping for predictive and reflexive robot manipulator trajectory estimation," in *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 10 2017. doi: 10.1109/SSCI.2016.7850237. ISBN 9781509042401

[6] E. Ujkani, P. S. Eppeland, A. Aalerud, and G. Hovland, "Real-time human collision detection for industrial robot cells," in *2017 IEEE Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 11 2017. doi: 10.1109/MFI.2017.8170368. ISBN 978-1-5090-6064-1 pp. 488–493.

[7] E. A. Engum, E. Haavind, S. Enes, and J. Holck, "Multi-Machine Control Leads to Improved Rig Layout," in *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers, 3 2014. doi: 10.2118/168021-MS. ISBN 978-1-61399-296-8 pp. 4–6.

[8] J. Sell and P. O'Connor, "The Xbox One System on a Chip and Kinect Sensor," *IEEE Micro*, vol. 34, no. 2, pp. 44–53, 3 2014. doi: 10.1109/MM.2014.9

[9] S. Zennaro, "Evaluation of Microsoft Kinect 360 and Microsoft Kinect One for robotics and computer vision applications," 2014.

[10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.5, Kobe, May 2009, p. 5.

[11] T. Wiedemeyer, "IAI Kinect2," Institute for Artificial Intelligence, University Bremen, 2014 – 2015, accessed January 23, 2018. [Online]. Available: https://github.com/code-iai/iai_kinect2

[12] J. Dybedal, A. Aalerud, and G. Hovland, "Embedded Processing and Compression of 3D Sensor Data for Large Scale Industrial Environments," *Sensors*, vol. 19, no. 3, p. 636, Feb. 2019. doi: 10.3390/s19030636

[13] E. Ujkani, J. Dybedal, A. Aalerud, K. B. Kaldestad, and G. Hovland, "Visual Marker Guided Point Cloud Registration in a Large Multi-Sensor Industrial Robot Cell," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, Jul. 2018. doi: 10.1109/MESA.2018.8449195. ISBN 978-1-5386-4643-4 pp. 1–6.