

# EtherCAT-Enabled Depth Camera for Safe Human-Robot Collaboration

Peter Gsellmann, Tobias Steinegger, Christoph Buchner, and Georg Schitter

*Automation and Control Institute (ACIN)*

*TU Wien*

Vienna, Austria

gsellmann@acin.tuwien.ac.at

**Abstract**—This paper presents a novel integration approach to enable EtherCAT-bus connection for commercial depth cameras. Thereby, a microcontroller-based EtherCAT slave is designed which controls the data transmission from the depth camera onto the fieldbus. As a demonstration case, a human-robot collaboration application is designed, where the proposed camera module monitors a five-axis robotic arm, thus ensuring safety through timely obstacle detection, and execution of safety stops. Achieving data transmission rates exceeding 25 Mbit/s and detection times under 25 ms, the implemented system outperforms existing technologies in human-robot collaboration, which allows to reduce the minimal safety distance to 75 mm.

**Index Terms**—Programmable Logic Controller, vision-based systems, robotics.

## I. INTRODUCTION

THE range of applications that rely on 3D vision has grown significantly over the last few years [1], motivated by Industry 4.0. An example of the use of 3D vision is bin picking [2], [3], which is the robot-based separation of randomly provided objects in a bin. It is a core problem in computer vision and robotics, where depth data are used to guarantee a precise gripping process by means of pose estimation. Pose estimation based on 3D vision is also used in robotic depalletization [4], [5], where pallets loaded with cartons are unloaded one by one. These systems have increased steadily as a result of the growth of sectors such as logistics, warehousing, and supply chains. Another area of industrial 3D vision application is inspection [6], which is often used in industrial quality control, for example, to check certain characteristics of products. The inspection covers both functional aspects of a product, e.g. whether all components are fitted to an assembly, as well as aesthetic properties, e.g. checking for scratches on a polished surface.

Collaborative applications involving humans and robots also necessitate the integration of 3D vision within an industrial environment. Human-robot collaboration workspaces are becoming increasingly important to improve work efficiency, flexibility, and overall productivity in production facilities. State-of-the-art approaches in the field of collision avoidance use depth cameras to monitor the environment around robotic systems, enabling a safe and collaborative working environment. [7], [8]

The most common interface for depth cameras is the Universal Serial Bus (USB). However, disadvantages such

as a rather limited maximum cable length and susceptibility to electromagnetic interference [9] do not comply with the requirements of the industry, especially for safety applications. Camera interface solutions such as Ethernet or Wireless LAN lack of real-time capability or require additional redundancies, respectively.

However, the latest standard for data exchange in industrial automation between the field and control level is the fieldbus [10], [11]. For robot application implementation, low latencies in combination with a high data rate are among the most important specifications [12]. The EtherCAT (Ethernet for Control Automation Technology) fieldbus, disclosed in the IEC 61158 standard [13], meets these requirements. Furthermore, several robots, such as the modular robot system ATRO from Beckhoff [14], are by default supplied with a Ethernet for Control Automation Technology fieldbus connection from the base up to the tool-center point, which provides flexibility regarding camera placement.

This paper proposes an integration approach with the aim of enabling EtherCAT connectivity for of-the-shelf depth cameras. Thereby, a microcontroller-based EtherCAT slave is designed which orchestrates the data transmission from the camera onto the fieldbus.

The paper is structured as follows: In Section II, the utilized  $\mu$ C (microcontroller) and camera module are presented. Furthermore, the necessary scheme for the transmission of the depth data via EtherCAT is elaborated in Section III. Section IV delves into the performance analysis of the proposed approach and the subsequent discussion of the results. Finally, Section V concludes the work.

## II. SYSTEM OVERVIEW

With the aim of making depth data from a depth camera available on the Ethernet fieldbus EtherCAT, a system working as an EtherCAT slave is required, which consists of three components: a depth camera, a microcontroller device, and an ESC (EtherCAT SubDevice Controller). The depth camera transmits the data via a chip-level bus to a microcontroller. The microcontroller processes and transmits the depth information via a PDI (process data interface) to an EtherCAT SubDevice Controller. The EtherCAT SubDevice Controller is directly connected to the controller via the EtherCAT fieldbus. If a frame passes through the slave, the slave writes the data to the

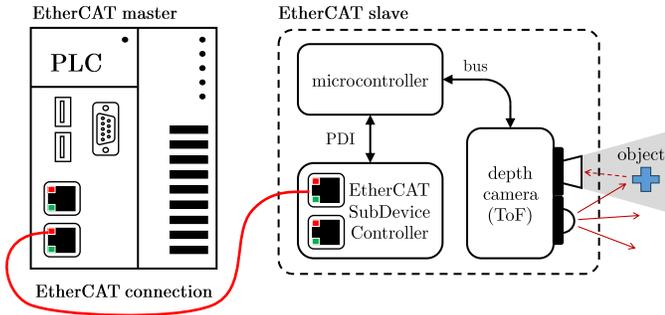


Figure 1: System overview of the proposed camera module as EtherCAT slave. The slave consists of three components: EtherCAT SubDevice Controller,  $\mu$ C and depth camera. The camera shares its information with a  $\mu$ C via bus communication. The  $\mu$ C processes the data and sends it via a process data interface to the EtherCAT SubDevice Controller, where it can be polled by EtherCAT frames to forward the data directly to the PLC.

frame and sends them back to the EtherCAT master device, such as a Programmable Logic Controller (PLC) or PC. An overview of this system is given in Fig. 1.

#### A. EtherCAT slave

The utilized EtherCAT slave consists of a microcontroller device, such as a Raspberry Pi 4 Model B. Furthermore, an EtherCAT SubDevice Controller, namely the EasyCAT HAT from AB&T Srl [15], is required to allow the microcontroller to become an EtherCAT capable slave. The communication between the single board computer and the EtherCAT SubDevice Controller takes place via SPI. The board has 4kB of DPRAM (Dual-ported Random Access Memory) to define the process interface of the EtherCAT slave. An SPI slave controller provides a synchronous slave interface with a low pin count. It allows access to the System Control and Status Registers, internal First-In-First-Out buffers, and memory. Bit lanes are supported with a clock rate of up to 80 MHz. The utilized microcontroller and EtherCAT SubDevice Controller setup are depicted in Fig. 2a.

#### B. Depth camera system

The requirements on the depth camera for the proposed system are a data transfer rate significantly below 100 Mbit/s, taking the EtherCAT limitations into account. Additionally, the preference is for a depth camera utilizing Time-of-flight (ToF) technology due to the advantages of its compact size, good mobility and easy and quick data extraction [16]. Compatibility with Linux-based operating systems on the Raspberry Pi is essential, which requires the availability of a suitable driver. Lastly, the camera should support connection options such as USB or Ethernet. An example of a ToF 3D imaging USB camera that meets these specifications is the flexx2 (cf. Fig. 2b) from pmd technologies [17].

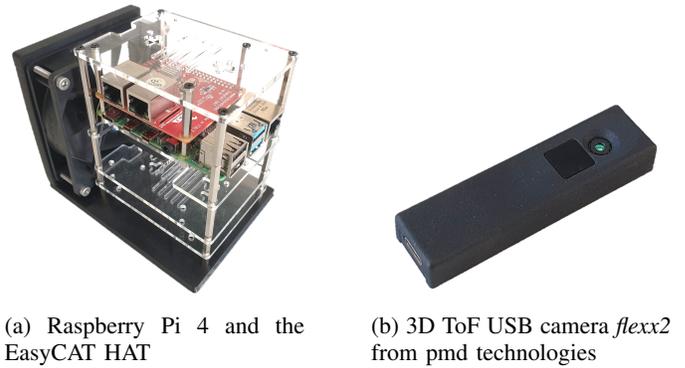


Figure 2: EtherCAT-capable depth camera module

The pmd flexx2 supports nine different pre-defined modi, that can be chosen depending on the use case. Thereby, frame rate and exposure time are varied between 5 fps-60 fps and 220  $\mu$ s-1500  $\mu$ s, respectively, in order to obtain a suitable image capture at a certain range.

### III. DEPTH DATA TRANSMISSION VIA ETHERCAT

Images and depth images have a high memory requirement depending on the resolution and the bits per depth pixel used. Hence, depth images of the utilized depth camera pmd flexx2 have 38.528 pixels [17] with a measurement range of 0.1 m – 4 m at a depth accuracy of  $\leq 1\%$  of the measured distance. Given by the camera drivers, 2 Byte per pixel are sufficient to define all the depth information provided by the pmd flexx2, which then results in a total byte count of 77 056 Byte per frame.

As stated in Section II-A, the EtherCAT SubDevice Controller has a Dual-ported Random Access Memory of  $DPRAM_{total} = 4096$  Byte. Due to the buffered mode this results in a maximum definable memory area for cyclic data exchange of

$$DPRAM_{available} = \frac{DPRAM_{total}}{3} = 1365 \text{ Byte.} \quad (1)$$

To transmit a depth image of the pmd flexx2 without loss, a method must be developed that allows 77 056 Byte per frame to be sent via the relatively limited Dual-ported Random Access Memory of  $DPRAM_{available} = 1365$  Byte. The considered approach, elaborated in the following, can be seen in Fig. 3.

The depth images sent by the flexx2 are transmitted as line-sorted 1D arrays to the Raspberry Pi via USB. Thereby, the 1D array is divided into segments of equal size. The depth data are then written to the EASYCAT HAT's Dual-ported Random Access Memory via SPI. In the EtherCAT SubDevice Controller, a two byte input variable is defined for each pixel in a segment. The definition of the input variable is given by the EEPROM (Electrically Erasable Programmable Read-Only Memory) configuration. If an EtherCAT frame passes the EtherCAT SubDevice Controller, the data of a segment are written to the EtherCAT frame, which is subsequently

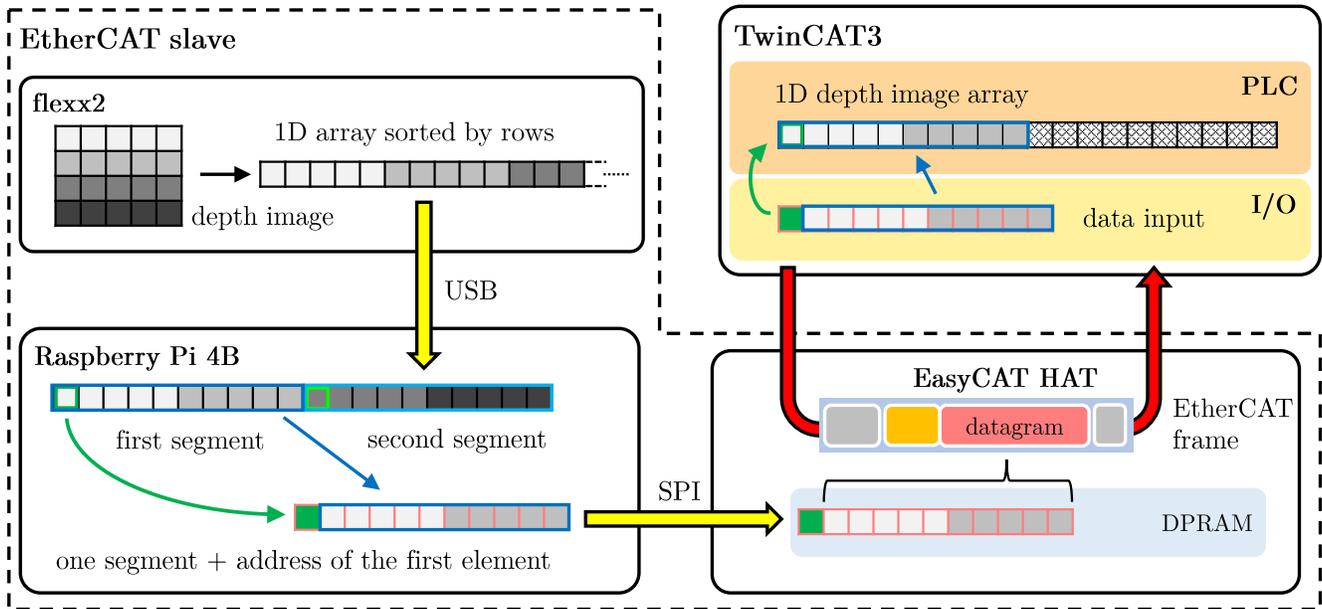


Figure 3: Transmitting depth data via EtherCAT: The depth camera sends a 1D array to microcontroller via USB, then segments and addresses are sent to EtherCAT SubDevice Controller using SPI. When an EtherCAT frame passes the slave, the data and address from Dual-ported Random Access Memory are read and sent to the controller. Subsequently, the depth image is reconstructed in an 1D array format at the TwinCAT soft PLC.

forwarded to the controller. During this reading process the data of the next segment are written into another buffer of the Dual-ported Random Access Memory. This is then read with the next EtherCAT frame. The proposed transmission protocol is repeated until the entire depth image has been transferred.

Within the controller, the input variables of the camera device in the I/O area are linked to a defined segment array in the PLC. To reconstruct the one-dimensional depth array, the position of the sent segment must be known. Therefore, the address of the first element in the segment is also sent with the depth data.

The attempted approach requires multiple EtherCAT frames per depth image. These are sent cyclically by the EtherCAT master, in this case Beckhoff TwinCAT3. The minimum cycle time of TwinCAT3 is 50  $\mu$ s. To keep the latency of the camera system as low as possible, the number of required frames is minimized, whereby a maximum of 1365 Byte per frame can be transmitted. In addition, the segmentation process should be as simple as possible. If the total number of bytes per depth image is divisible by the number of pixels per transmitted segment, this simplifies the implementation.

A data transmission of 301 pixels per segment results in an utilization of slightly more than 44% of the Dual-ported Random Access Memory as well as an easy implementation of the segmentation. The flexx2 has 172 lines per depth image. This results in exactly 128 required EtherCAT frames for a single depth image with a theoretical total latency of 6.4 ms. With the depth information of two byte per pixel plus two byte address information, a memory area of 604 Byte is defined in

the EtherCAT SubDevice Controller.

#### IV. EXPERIMENTS AND ANALYSIS

The safety distances for collaborative robots specified in the DIN ISO/TS 15066 standard [12] are determined in relation to the measuring system by the detection time, the position uncertainty of the operator in the collaboration space, and the penetration depth. In this regard, the position uncertainty and the penetration depth are determined by the calibration of the camera, the accuracy of the coordinate transformation between the robot and the camera system, the resolution of the camera and its depth resolution. The detection time results from the latency of the camera, including the transmission system and the frame rate. Data reduction methods are a way to enhance the performance of the transmission system. Moreover, it is essential for determining the safety distances in the collaborating workspace.

##### A. Evaluation of latency and frame rate

To determine the detection time, an infrared LED is mounted in the camera's field-of-view. This LED emits infrared light with the same wavelength of 940 nm as the flexx2 transmitter. If the LED emits light in the depth image, no depth data are available at the position of the LED due to interference with the modulated light from the depth camera. As a result, the sensor array cannot detect any phase difference between the emitted and captured light. Therefore, the distance at the LED position is measured as 0 m in the case it is turned on. If the LED does not emit any light, only the

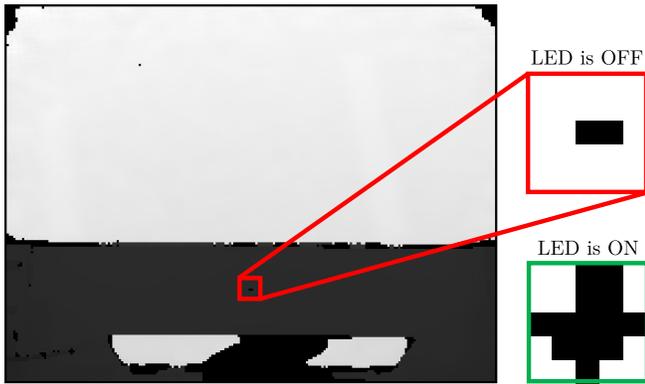


Figure 4: Measurement of the camera module latency: In the recorded depth image, the red window shows the area around the infrared LED when it is switched off. Two pixels remain black due to reflections on the surface of the LED. The green window shows the area around the LED when it is switched on, and thus emitting light. The camera detection time is derived from the measurement of the time between the turn-on of the LED and the detection in the depth image.

modulated reflected light is received by the depth camera, so the distance between the camera and the LED can be measured. An exemplary measurement is shown in Fig. 4. This measurement process is repeated with a uniform random time delay until a sufficient number of measurements is reached, in this case 10000 measurements.

The detection times of four different camera modes, 24 fps, 30 fps, 45 fps, and 60 fps are measured, and the results are shown in Fig. 5. The measured values show that the average detection time is significantly higher than the reciprocal of the frame rate of the camera. Furthermore, a high variance in the values can be noted. If the transmission latency is negligible, one would expect the histogram to resemble a uniform distribution on an interval corresponding to the reciprocal of the frame rate, and the mean value of all measured detection times to correspond to the value of half the frame rate. However, if the latency is not negligible but constant, all values of the uniform distribution would increase by the latency. If the latency time varies, the interval of the uniform distribution would extend. This can be observed in the second histogram of Fig. 5. Additionally, with higher frame rates, it can be seen that the measured detection time values are no longer uniformly distributed. This circumstance caused by the limited maximum transfer data rate of the SPI (Serial Peripheral Interface), which either can be solved by a lower frame rate or reduced amount of data.

### B. Data reduction methods

A trivial method to lower the data transfer rate of pmd flexx2 is to reduce the number of bits per pixel. The smallest number of bits an EtherCAT slave variable can have is eight. The largest unsigned integer that can be displayed is 255, which means that if a pixel is represented by just one byte, that is, a depth of 0 m to 2.55 m can be displayed with a depth resolution

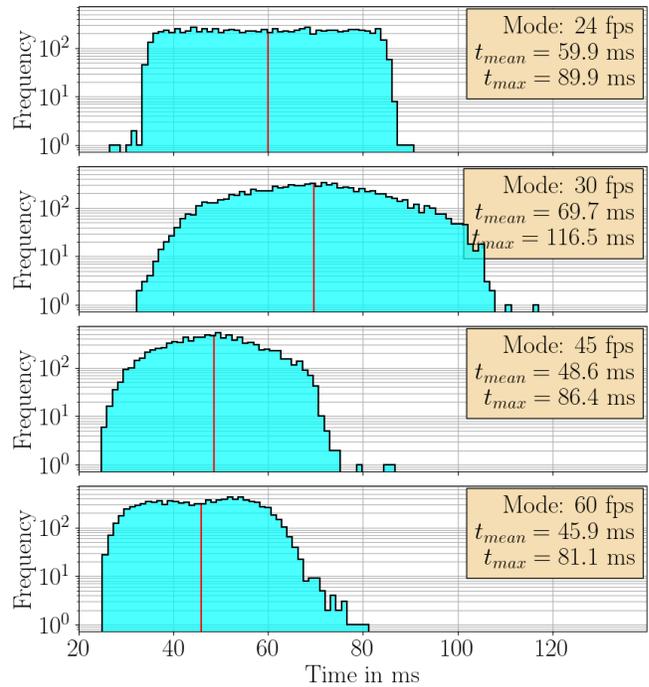


Figure 5: Measured detection times, defined as the time span between switching the LED on to detecting the LED in the depth image. The results of 10000 measured values from four different operating modes are displayed. The histograms contain the measured values from the 20, 30, 45, and 60fps mode. The red line shows the mean value of all measured values.

of 1 cm. Thus, this method sets all values greater than 2.55 m to 0 m. Therefore, the data of a depth image are reduced by a factor of two.

An alternative approach to reducing the data transfer of flexx2 is to decrease the number of pixels per depth image. Hence, a minimum pool operation is deployed before transferring data at the .

In this experiment, a  $2 \times 2$  window is used to iterate over the depth image, and the smallest value except 0 m is retained, the other pixels are ignored. Therefore, the resolution of a flexx2 depth image is reduced from  $224 \times 172$  pixels to  $112 \times 86$  pixels. This means that a  $2 \times 2$  pooling operation reduces the data volume of a depth image by a factor of four, a  $4 \times 4$  pooling operation by a factor of 16. The effect of the  $4 \times 4$  pooling operation is shown in Fig. 6, where the measurement of the detection times from Fig. 5 is repeated.

The results show a significant reduction of the detection time. The maximum detection time in the case of the 60 fps mode reduces from 81.1 ms by 65.5 % to 28.0 ms. The best results are obtained with a  $2 \times 2$  pooling operation at 60 fps which corresponds to 25 Mbit/s, making this performance enhancement viable for the use in human-robot applications.

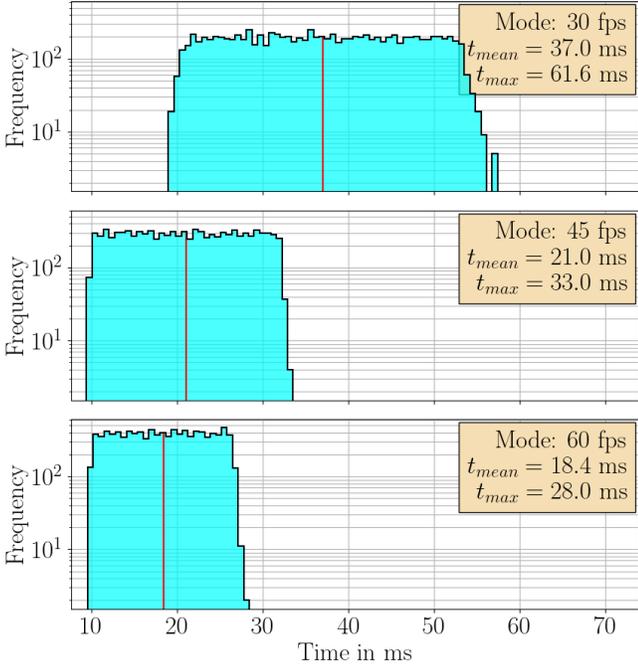


Figure 6: Measured detection times using the  $4 \times 4$  minimum pooling operation. The results of 1000 measured values from three different operating modes are displayed. The upper histogram contains the measured values with 30 fps, the second with 45 fps, and the last with 60 fps. The red line shows the mean value of all measured values.

### C. Human-Robot collaboration application

The desired application to test the EtherCAT-capable depth camera is a human-robot collaboration environment. The safety distances between robots and people in the collaboration space are evaluated according to the DIN ISO/TS 15066:2016 [12] standard based on the measured detection time of the camera system. This safety distance allows the system to be benchmarked and creates comparability with state-of-the-art applications. The safety distance,  $S_p$ , at the time  $t_0$  is described in the standard by the following equation:

$$S_p(t_0) = S_h + S_r + S_s + C + Z_d + Z_r \quad (2)$$

with the contribution of the position change of the person  $S_h$ , the contribution due to the response time of the robot system  $S_r$ , the contribution due to the stopping distance of the robot system  $S_s$ , the penetration distance  $C$ , and the position uncertainty of the operator and the robot system  $Z_d$  and  $Z_r$ , respectively. When determining the minimum safety distance, the camera system only has an influence on the parameters  $S_h$ ,  $S_r$ ,  $C$ , and  $Z_d$  of Equation (2) according to [12]. By assuming the velocity of a non-monitored person to  $1.6 \text{ m s}^{-1}$ , a constant value for  $S_h$  can be calculated with the detection time of the robot system  $T_r$ . Furthermore, the value for  $S_r$  is obtained by considering the maximum velocity of the tool center point  $v_r$

and the detection time  $T_r$ . For the used 5-axis robot system RL-DP-5 from igus [18], a value of  $v_r = 0.565 \text{ m s}^{-1}$  is calculated. For the contributions  $C$  and  $Z_d$  of Equation (2), the depth resolution of the pmd flexx2 is assumed at a maximum measuring distance of 2 m. The maximum penetration depth  $C$  and the positional uncertainty  $Z_d$  of a person with this camera system is half the diagonal of a voxel. A voxel is calculated from the depth resolution  $z$ , the x and y dimensions of a pixel at the maximum distance. The distance values calculated for the different methods are presented in Table I.

Table I: Safety distances according to standard DIN ISO/TS 15066 of measured detection times.

| method               | $f_r$<br>[fps] | $S_h$<br>[mm] | $S_r$<br>[mm] | $C$<br>[mm] | $Z_d$<br>[mm] | $S_p$<br>[mm] |
|----------------------|----------------|---------------|---------------|-------------|---------------|---------------|
| no<br>method         | 20             | 144           | 51            | 45          | 45            | 284           |
|                      | 30             | 186           | 66            | 45          | 45            | 342           |
|                      | 45             | 138           | 49            | 45          | 45            | 277           |
|                      | 60             | 130           | 46            | 45          | 45            | 265           |
| pool<br>$2 \times 2$ | 30             | 99            | 35            | 57          | 57            | 247           |
|                      | 45             | 53            | 19            | 57          | 57            | 186           |
|                      | 60             | 45            | 16            | 57          | 57            | 175           |
| pool<br>$4 \times 4$ | 20             | 84            | 30            | 83          | 83            | 280           |
|                      | 45             | 47            | 17            | 83          | 83            | 229           |
|                      | 60             | 39            | 14            | 83          | 83            | 217           |

The lowest safety distance is achieved with the data reduction method minimum pooling  $2 \times 2$  at a frame rate of 60 fps and is evaluated as  $S_p = 175 \text{ mm}$ . In order to create a comparison of the implemented system with the state of the art, the specifications of the human-robot collaboration system from Veo Robotics [19] are applied to the implemented system. The recognition time is specified as 100 ms and the resolution as 25 mm at a distance of 3 m. This leads to a safety distance of  $S_{p,Veo} = 250 \text{ mm}$ , exceeding the results of the proposed system and thus inferior in this human-robot collaboration task.

In summary, the presented integration approach enables the depth camera data transmission rates of 25 Mbit/s through the EtherCAT fieldbus. When comparing the integration approach within a robotic application with existing technologies, the minimum safety distance can be reduced from 250 mm to 175 mm, thus demonstrating the viability of the proposed systems.

## V. CONCLUSION

This paper targets the simplification of components in Industry 4.0 and their standardized integration into the field level. A depth camera, the flexx2 from pmd technologies [17], is converted for use with EtherCAT. A Raspberry Pi 4 microcontroller segments the depth data from the flexx2 connected via USB 3.0 and sends it to the EtherCAT Sub-Device Controller via SPI. The EtherCAT master, TwinCAT3, reassembles the segments sent via EtherCAT frames into a complete depth image. The cycle time is 0.1 ms to ensure that all segments of a depth image are transmitted as fast as

possible. Various modes of operation for the depth camera are evaluated, which differ in the range of depth information and the frame rate.

With the proposed system, depth images can be transmitted with a resolution of  $224 \times 172$  pixels, a depth information of 16 bit/pixel and a frame rate of 30 fps, equivalent to a data transfer rate of 25 MBit/s. For frame rates up to 60 fps, the data transfer rate is reduced to 17 MBit/s, due to the limited computing power of the Raspberry Pi 4 Model B on the SPI. The safety distances between humans and robots, as defined in the DIN ISO/TS 15066 standard [12] for collaborative robot systems, are up to 75 mm smaller with the implemented EtherCAT-capable depth camera in combination with various data reduction methods than with the reference system from Veo Robotics [19].

The maximum data transfer rate of the proposed system is limited by serial communication via Serial Peripheral Interface between microcontroller and EtherCAT SubDevice Controller. Therefore, a more powerful single-board computer or a Field Programmable Gate Array (FPGA) could ensure lower latencies. Parallel interfaces can improve the data transfer rate and reduce latencies, enabling the transmission of higher-resolution images and higher frame rates up to the maximum field bus capacities.

#### REFERENCES

- [1] V. Alonso, A. Dacal-Nieto, L. Barreto, A. Amaral, and E. Rivero, "Industry 4.0 implications in machine vision metrology: an overview," *Procedia Manufacturing*, vol. 41, pp. 359–366, 1 2019.
- [2] W. Yan, Z. Xu, X. Zhou, Q. Su, S. Li, and H. Wu, "Fast Object Pose Estimation Using Adaptive Threshold for Bin-Picking," *IEEE Access*, vol. 8, pp. 63 055–63 064, 2020.
- [3] J. Su, Z. Y. Liu, H. Qiao, and C. Liu, "Pose-estimation and reorientation of pistons for robotic bin-picking," *Industrial Robot*, vol. 43, no. 1, pp. 22–32, 1 2016.
- [4] P. Opaspilai, S. Vongbunyong, and A. Dheeravongkit, "Robotic System for Depalletization of Pharmaceutical Products with 3D Camera," *ICSEC 2021 - 25th International Computer Science and Engineering Conference*, pp. 422–427, 2021.
- [5] P. Doliotis, C. D. McMurrough, A. Criswell, M. B. Middleton, and S. T. Rajan, "A 3D perception-based robotic manipulation system for automated truck unloading," *IEEE International Conference on Automation Science and Engineering*, vol. 2016–November, pp. 262–267, 11 2016.
- [6] C. Koch, Z. Zhu, S. G. Paal, and I. Brilakis, "Machine vision techniques for condition assessment of civil infrastructure," *Integrated Imaging and Vision Techniques for Industrial Inspection: Advances and Applications*, pp. 351–375, 9 2015.
- [7] B. Schmidt and L. Wang, "Depth camera based collision avoidance via active robot control," *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 711–718, 10 2014.
- [8] Terabee, "Building TOF Sensor Arrays for Collision Avoidance Systems," 2022. [Online]. Available: <https://www.terabee.com/an-introduction-to-building-time-of-flight-sensor-arrays-for-collision-avoidance-systems-in-mobile-robots/>
- [9] M. Mačkowski, "The influence of electromagnetic disturbances on data transmission in usb standard," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 95–102.
- [10] J. P. Thomesse, "Fieldbus technology in industrial automation," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1073–1101, 6 2005.
- [11] S. K. Sen, *Fieldbus and Networking in Process Automation*. CRC Press, 4 2021.
- [12] DIN ISO/TS 15066:2016, "Roboter und robotikgeräte – kollaborierende roboter," 12 2017.
- [13] DIN EN IEC 61158-1, "Industrielle kommunikationsnetze - feldbusse - teil 1: Überblick und leitfaden zu den normen der reihen iec 61158 und iec 61784," 4 2020.
- [14] Beckhoff Automation GmbH, "ATRO," 2023. [Online]. Available: <https://www.beckhoff.com/de-at/produkte/motion/atro-automation-technology-for-robotics/>
- [15] AB&T Srl, "EtherCAT® & Raspberry." [Online]. Available: <https://www.bausano.net/it/hardware/ethercat-raspberry.html>
- [16] U. B. Himmelsbach, T. M. Wendt, and M. Lai, "Towards safe speed and separation monitoring in human-robot collaboration with 3d-time-of-flight cameras," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 197–200.
- [17] pmdtechnologies ag, "pmd flexx2 3D Camera Development Kit – pmdtechnologies ag," 2023. [Online]. Available: <https://3d.pmdtec.com/en/3d-cameras/flexx2/>
- [18] igus GmbH, "robotlink® RL-DP-5 | 5 Freiheitsgrade (DOF) | 790mm Reichweite." [Online]. Available: <https://www.igus.at/product/20239?artNr=RL-DP-5>
- [19] VEO ROBOTICS, INC, "Veo Robotics," 2022. [Online]. Available: <https://www.veobot.com/>