Depth-data-based object cluster tracking and velocity estimation in robot workspace

Peter Gsellmann, Christoph Buchner, Karin Egretzberger,
Martin Melik Merkumians, *Member, IEEE*, and Georg Schitter, *Senior Member, IEEE*
Automation and Control Institute
TU Wien
Vienna, Austria
gsellmann@acin.tuwien.ac.at

***Abstract*—Depth-data-based sensor systems, such as depth cameras or LiDAR systems, are gaining popularity in the field of robotics, especially in human-robot collaboration. To avoid collisions with humans or external objects, object detection and tracking in the workspace is needed. This paper presents an integrated object cluster tracking and velocity estimation method that is purely based on depth data. Therefore, a tracking heuristic based on similarity and the velocity of the object is used to enable the tracking of external objects. To obtain the velocity, a Kalman filter utilizing a constant velocity model is implemented. For experimental verification, a case study comprising two objects moving within the robot workspace is designed. The experimental setup allows for the initial tracking a maximal trackable object velocity of $9\,\mathrm{m\,s^{-1}}$, and for already tracked objects a velocity deviation of $3.4\,\mathrm{m\,s^{-1}}$ to correctly track both repetitive and arbitrary motions of the test objects, and thus constitutes the proposed integrated object cluster tracking approach as a foundation for collision avoidance strategies in robotic tasks.**

***Index Terms*—Object tracking, vision-based systems, robotics.**

I. Introduction

INDUSTRIAL robot manipulators are nowadays used in a variety of different areas in factory automation, where high precision, repeatability and reliability of operations are required. Typical applications, in addition to picking and placing objects, are milling [1], welding [2], or assembly tasks [3]. However, developments in recent years have shown that in addition to a high degree of reconfigurability of the manipulator, the capability of collaborating with human operators is required. Today, the industry focuses more on human-robot collaboration (HRC) [4], such as processing the same work piece or sharing the same workspace. Consequently, surveillance is needed to detect or even react to collisions of the robot with objects in its workspace. Thus, to not compromise the high-level flexibility of the robotic system, typically considered sensor systems, such as Time-of-Flight (ToF) cameras or LiDAR systems, are mounted on the robot system [5]. This configuration drastically reduces the number of external calibrations, as the sensor system is fixed on the robot manipulator. Therefore, repositioning the robot system also does not pose problems in configuration. To avoid collisions with humans or obstacles, object detection and tracking in the workspace is required.

Several approaches regarding object cluster tracking in robotic and autonomous driving applications are implemented using classic methods with 2D images using color information and image features [6], [7]. Methods based on Structure-from-Motion (SfM) show proficient tracking performance [8]. However, the extraction of feature points, the reconstruction of feature tracks, and the parallel calculations for 3D feature locations generate a high computational effort and, thus, are not applicable in real-time applications.

With RGB-D cameras gaining popularity, the first approaches of object tracking with the addition of depth data are conducted. Typically, color information is used for the recognition of objects to be tracked and only uses depth data as an additional source of information, e.g., to calculate the possible motion trajectory of tracked objects [9], or to obtain only the distance to objects [10]. Other approaches utilize RGB-D point cloud features to track occluded objects [11].

Since depth cameras or LiDAR systems in robotic or autonomous driving applications do not always have the option to fall back to color information, approaches relying purely on depth data are necessary. Implementations that provide an a priori model of the objects to be tracked have proven promising [12]. The downside to these approaches is that this is rarely applicable in the case of HRC or autonomous driving applications.

To remedy this, solutions are devised with object cluster tracking using only depth data [13], [14]. Therefore, the background subtraction method is used to record the initial scenery as background. Thereafter, the background is subtracted from the current depth image in order to obtain the clusters of all emerged objects, which are subsequently tracked. The main disadvantage in the context of robotics, especially HRC, or also for autonomous driving, is that the background shall not change during operation, which is hardly possible.

This paper proposes an integrated object cluster tracking and velocity estimation method that is based only on depth data. Thereby, a heuristic based on similarity and the object's velocity is used to enable tracking of the obstacles in a robot workspace.

The paper is structured as follows: In Section II, the robot utilized and the vision system are presented. Furthermore, the preliminary actions carried out in the vision system are

Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361

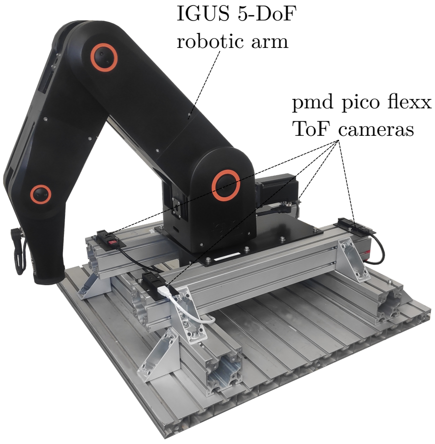


Figure 1: System overview: IGUS robolink RL-DP 5-Degrees-of-freedom (DOF) robot manipulator. In the robot base, four pmd pico flexx Time-of-flight (ToF) cameras are place around the manipulator in order to monitor the workspace.

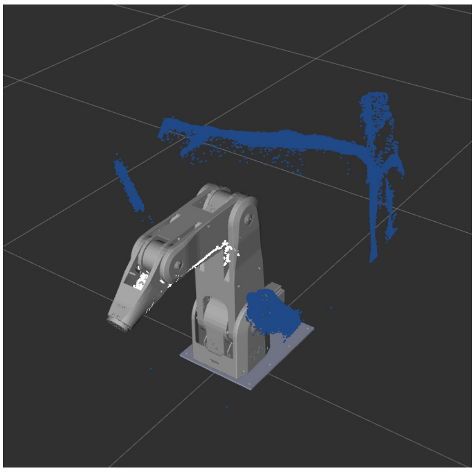


Figure 2: Example point cloud of the distinction between points on the robot (white points) and surroundings (blue points) via filtering of the robot's URDF model.

elaborated, which generates the basis for the object cluster tracking and velocity estimation approach. In Section III , the depth-data-based object cluster tracking and the subsequent velocity estimation approach are described in detail. This is followed by the presentation and analysis of experiments in Section IV. Finally, Section V concludes the work.

II. System Overview

The vision system, providing preliminary measures for the depth-only object cluster tracking, and the utilized robot system as demonstration application for the feasibility study are presented in the following in detail.

A. Vision System

The considered system consists of depth cameras mounted on the robotic system and the vision algorithm with preliminary steps for object cluster tracking and velocity estimation. The task of the vision system is to perceive the surroundings of the robot and subsequently process the data obtained for use in other applications, such as collision avoidance strategies.

To make the overall system as modular as possible, ROS is used as the middleware between robotic and vision systems. All software components used are executed on a PC equipped with an ASUS GeForce™ RTX 3070 GPU, an AMD Ryzen™ 9 3900X CPU, and 32GB DDR4 3200MHz memory.

1) Depth camera system: Since for applications such as HRC an all-round view of the robot has to be ensured, one possible configuration is to mount the cameras on the robot, though this leads to a high number of sensors. Especially in the context of modular robotics, this is problematic, as for the versatile structure and therefore the freedom to assemble the robot arbitrarily, an all-round view in every robot module is required, which enormously increases the data volume. To counteract this, a possible remedy for this is to place the cameras in the robot's base around the manipulator facing upward (see Fig. 1). For the purpose of this paper, four pmd pico flexx Time-of-Flight (ToF) cameras with a range of 0.1 - $4\,\mathrm{m}$ and a framerate of $45\,\mathrm{fps}$ are used, although the number of depth cameras can be arbitrarily selected and should be chosen based on the requirements of the application.

To correctly merge the point clouds, an extrinsic calibration with a calibration target is conducted [15], which yields the transformation between single camera frames. Subsequently, the recorded point clouds can be transformed and merged to one overall point cloud.

2) Robot model filtering: The overall point cloud perceived by the cameras lays the foundation for the surveillance of the robot's workspace. Since in configurations where the depth cameras are placed at an external position of the robot to perceive its environment, as well as in eye-in-hand configurations or in the present approach with depth cameras in the base of the robot, there is a high chance that the robot manipulator appears on that overall point cloud. Therefore, it is essential to know which points belong to the robot and which points belong to external objects in the workspace. A trivial approach to address this problem is to define a volume that contains the robot manipulator and subsequently marks every point in that volume as a point belonging to the robot. This consequently implies that all points outside the volume correspond to external objects. However, this also limits the usability of the system, since external objects in the volume cannot be detected, and thus no near approach of the robot is possible, which is vital in HRC applications, for example. A possible solution to overcome this is to use the a priori known robot model to extract the points that belong to the robot.

Modeled in the Unified Robot Description Format (URDF), this model's pose is updated via the current states of the joint encoders. Subsequently, all points inside and on the

Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361

model are removed, resulting in a point cloud with only all perceived external objects in the workspace. Furthermore, due to the range of the depth cameras used, also the room's ceiling is perceived, which can easily be discarded by setting a height threshold and remove all points above. An example of the implemented approach can be seen in Fig. 2, where a clear distinction can be made between the robot and external objects; in this example, a human working in proximity to the robot.

3) Point cloud clustering: After filtering the robot body, the remaining points in the point cloud represent external objects in the robot workspace. With the ultimate goal of tracking the separate objects, it is necessary to combine the points to clusters. For use in robotic applications, especially HRC, the number of clusters is unknown a priori, which already rules out methods such as k-means clustering [16]. A suitable technique without knowing the number of clusters is the agglomerative hierarchical clustering algorithm [17]. Therefore, a bottom-up approach is utilized, where each data point starts as its own cluster and then gradually merges with other clusters within a defined threshold based on their Euclidean distance. This yields a set of clusters with their associated points and the cluster center position.

III. Depth-data-based Object Cluster Tracking and Velocity Estimation

With the preliminary steps described in Section II done, a basis is given for the depth-data-based tracking of the object clusters and subsequent velocity estimation. This section is concerned with the proposed novel tracking method of the perceived object clusters and their 3D velocity estimation.

A. Cluster Tracking

The object cluster tracking algorithm based solely on depth data utilizes the cluster position $\mathbf{p}$ which is defined by the average cluster point position, the cluster velocity $\mathbf{v}$, the cluster volume V, the number of points in a cluster n for recognition. A pseudocode for this method is provided in Algorithm 1.

Initially, all clusters and their respective properties are stored. From the next sampling step on, a distinction between two cases must be made.

The first case (see Fig. 3) occurs if a cluster a_{k-1} is registered for the first time in the time step $k-1$. Thus, the velocity of this cluster cannot yet be estimated, since it needs at least two measurement points. For the recognition of the object cluster a_{k-1} in the successive time step k newly perceived clusters, in Fig. 3 a_k, b_n, and c_n, within a distance threshold $d_{th} = d_{th2}$ are considered. If the difference in volume dV and the difference in number of points dn for a cluster is within a set threshold $V_{th} = f_V V$ and $n_{th} = f_n n$, respectively, where f_V and f_n describe a threshold coefficient, the cluster a_k is found as a match to a_{k-1} and therefore the object cluster is tracked. Subsequently, the 3D velocity $\mathbf{v}_k$ of the cluster a_k is estimated.

The second case (see Fig. 4) is present if a cluster a_{k-1} is already registered in the time step $k-1$, and therefore has a

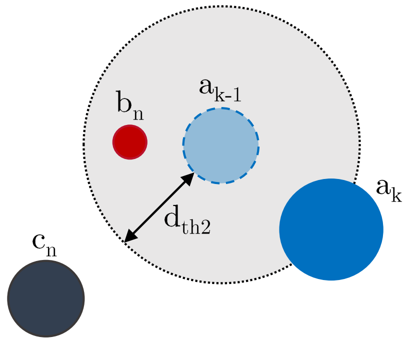


Figure 3: Object cluster tracking heuristic without prior information on object cluster's velocity. In order to recognize a object cluster for the first time, cluster size V, number of elements in the cluster n, and position $\mathbf{p}$ are considered. In this case, only the cluster a_k constitutes a suitable match to a_{k-1}, since the cluster b_n does not comply with the size threshold, and the cluster c_n is not within the distance threshold.

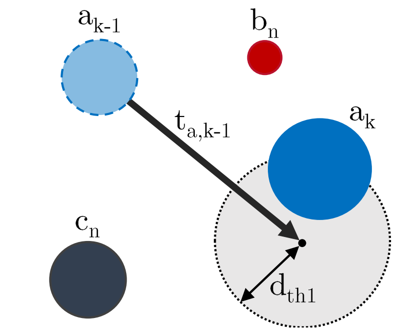


Figure 4: Object cluster tracking heuristic with prior information on object cluster's velocity. For already registered object clusters, the cluster size V, number of elements in the cluster n, position $\mathbf{p}$, and also the clusters velocity $\mathbf{v}$ are considered. In this case, only the cluster a_k constitutes a suitable match to a_{k-1}, since the clusters b_n and c_n are not located within the distance threshold.

current velocity estimation $\mathbf{v}_{k-1}$. Now, for the recognition of the object cluster a_{k-1} in the next time step k, its movement and therefore its approximate position can be extrapolated with the translation vector $\mathbf{t}_{a,k-1} = \mathbf{v}_{k-1}T$. With prior information on the velocity of the object cluster, the difference in volume dV, and number of points dn for a cluster within a set threshold V_{th} and n_{th}, a match is searched solely within a smaller distance threshold $d_{th} = d_{th1}$. If these requirements are met, the cluster a_k is found as a match to a_{k-1} and thus the object cluster is tracked. Finally, the 3D velocity is updated.

If multiple objects satisfy the tracking conditions, the first element is chosen as the match. If both cases do not apply, the perceived object cluster is registered for the first time, and

Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361

its properties are stored. As this process can be applied to several objects concurrently, the proposed algorithm is able to perform multi-object tracking.

Algorithm 1 Object cluster tracking heuristic

Input: List of clusters $\mathbf{C_k}$, list of registered clusters $\mathbf{R_{k-1}}$
Output: updated list of registered clusters $\mathbf{R_k}$

for each element r of $\mathbf{R_{k-1}}$ **do**
  for each element c of $\mathbf{C_k}$ **do**
    if r has valid $\mathbf{v}$ estimation **then**
      $d_{th} = d_{th1}$
    else
      $\mathbf{v}(r) = \mathbf{0}$
      $d_{th} = d_{th2}$
    end if
    $V_{th} = f_V V(r)$
    $n_{th} = f_n n(r)$
    $d \;\;= \|\mathbf{p}\,(r) - \mathbf{p}\,(c) + \mathbf{v}\,(r) \cdot T_S\|_2$
    $dV \;= |V\,(r) - V\,(c)\,|$
    $dn \;= |n\,(r) - n\,(c)\,|$
    if $(dV < V_{th}) \wedge (dn < n_{th}) \wedge (d < d_{th})$ **then**
      Cluster match found: c and r
      $\mathbf{R_k} \leftarrow$ Updated position, number of points and size of cluster
    end if
  end for
end for
Register all unmatched clusters in $\mathbf{R_k}$
return $\mathbf{R_k}$

B. 3D Velocity Estimation

For the proposed depth data object cluster tracking, a suitable velocity estimation is required. Kalman filters and its variants are usually used in tracking algorithms for the estimation of the motion [18]. For this purpose, a dynamic model of the target motion is required. Since in applications such as robot collision avoidance or autonomous driving, the type of motion of the external objects is unknown, a constant velocity (CV) model is used, which assumes that the velocity stays constant during the sampling.

For the velocity estimation of an object cluster, the state vector

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} \tag{1}$$

comprises the position vector $\mathbf{p}_k = [p_x, p_y, p_z]^\intercal$ and the velocity vector $\mathbf{v}_k = [v_x, v_y, v_z]^\intercal$ at the time step k. The utilized CV model is expressed as

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_{k+1}, \tag{2}$$

where $\mathbf{x}_k$ denotes the state at the time step k, $\mathbf{w}_{k+1}$ the process noise with the covariance matrix $\mathbf{Q}$, and $\mathbf{\Phi}$ the transition matrix, which is defined as

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_{3\times3} & T_S \cdot \mathbf{I}_{3\times3} \\ \mathbf{0} & \mathbf{I}_{3\times3} \end{bmatrix}. \tag{3}$$

Here, T_S defines the sampling time of the measurement system and $\mathbf{I}_{3\times3}$ the 3×3 identity matrix. To determine $\mathbf{Q}$, several strategies can be followed, such as the use of random velocity or random acceleration process noise models [18]. For the analysis of the proposed approach, the covariance matrix of the process noise is empirically determined and set to a diagonal matrix $\mathbf{Q} = \mathrm{diag}(0.1, 0.1, 0.1, 10, 10, 10)$. The measurement model is defined as

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k, \tag{4}$$

where $\mathbf{z}_k$ describes the measurement vector, $\mathbf{x}_k$ the measured position, and $\mathbf{v}_k$ the measurement noise with the covariance matrix $\mathbf{R}$. Within this paper, the latter is empirically determined and set to $\mathbf{R} = \mathbf{I}_{3\times3}$.

IV. Analysis and Discussion

In order to verify the proposed object cluster tracking algorithm, an experiment with two objects moving within the robot workspace is designed. For that, two types of motion are considered: a repetitive motion performed by a pendulum (diameter of pendulum body $d = 0.16\,\mathrm{m}$, pendulum length $L = 1\,\mathrm{m}$) mounted on the ceiling and an arbitrary motion performed by a wooden plank (length $l = 0.06\,\mathrm{m}$, width $w = 0.02\,\mathrm{m}$) moved by a human operator.

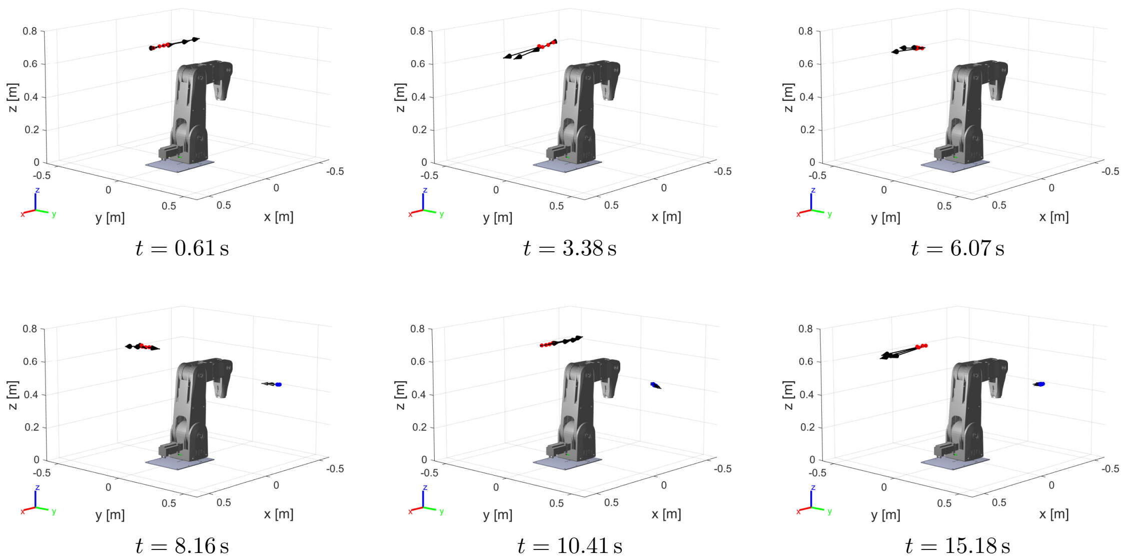
For this case study, the following parameters are chosen: The tolerance factors for the volume difference $f_V = 0.2$ and the difference in the number of points $f_n = 0.2$. The threshold for tracking without prior velocity information is set to $d_{th2} = 0.2\,\mathrm{m}$, whereas for the case of current velocity estimation the distance threshold is set to $d_{th1} = 0.075\,\mathrm{m}$.

An illustration of the test case is shown in Fig. 5 for six time stamps during the experiment. The pendulum swings from the beginning in an elliptical orbit, with its main component in the x-direction. The center position of the object cluster is marked with red dots that show the values for the last five measurements. At approximately $t = 8\,\mathrm{s}$ an object with arbitrary motion enters the workspace from the positive y-direction, with its cluster center position marked with blue dots. For both object clusters, the estimated velocity is indicated via the black vectors.

To illustrate the performance of cluster tracking and subsequent velocity estimation, the tracked position in x,y,z and the estimated 3D velocity of the pendulum are drawn over the elapsed time in Fig. 6. As can be seen in Fig. 6a, the repetitive motion of the pendulum is captured accordingly, with a pendulum frequency of approximately $f = 0.5\,\mathrm{Hz}$ which is consonant to the theoretical pendulum frequency $f_t = 1/(2\pi\sqrt{l/g}) = 0.498\,\mathrm{Hz}$. In Fig. 6b, the estimated velocity is also in accordance with the tracked position and has its main component in the x-direction.

Although the proposed novel depth-data-only object cluster tracking approach shows suitable results, some limitations are present. The tolerance factors set for the difference in volume f_V and the number of points f_n influence if a cluster is tracked successfully. Hence, information on the application

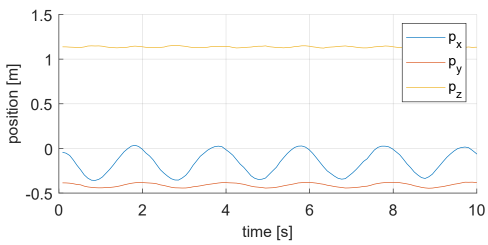
Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361



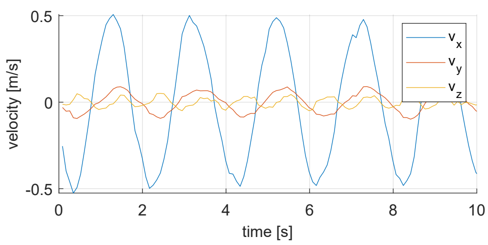
$t = 0.61\,\mathrm{s}$ $t = 3.38\,\mathrm{s}$ $t = 6.07\,\mathrm{s}$

$t = 8.16\,\mathrm{s}$ $t = 10.41\,\mathrm{s}$ $t = 15.18\,\mathrm{s}$

Figure 5: Analysis of the multi-object tracking performance in the robot workspace based on depth data only: A pendulum (cluster center marked with red points) swings next to the robot. At $t = 8\,\mathrm{s}$ an object with arbitrary motion enters the workspace from positive y-direction (cluster center marked with blue points). The arrows (black) represent the velocity vectors. Both the cluster positions and the velocity vectors for the different tracked objects are shown for the last five measurements.



(a) Tracked position



(b) Estimated velocity

Figure 6: Tracked position and estimated velocity in the case of the pendulum (see Fig. 5 points in red). The pendulum motion is visible in both the position and the velocity.

and the surrounding environment are necessary to tune them appropriately.

Moreover, tracking performance is also limited by the frame rate f or sampling time T_S of the vision system, and the set distance thresholds d_{th1} and d_{th2}. For the initial tracking, objects with a maximal velocity of $v_{max} = \frac{d_{th2}}{T_S}$ are trackable, which results for the present system with a sampling time of $T_S = 222\,\mathrm{ms}$ to $v_{max} = 9\,\mathrm{m\,s^{-1}}$. If an object is already tracked, it is of interest how much it can deviate from the considered constant velocity model. Thereby, the object's maximum velocity deviation which still allows tracking results to $\tilde{v}_{max} = \frac{d_{th1}}{T_S}$. In this case, this amounts to a maximum velocity deviation of $\tilde{v}_{max} = 3.4\,\mathrm{m\,s^{-1}}$.

In summary, the integrated object cluster tracking and velocity estimation method based solely on depth data is able to process repetitive and arbitrary motion of external objects, and thus is considered suitable as a basis for collision avoidance strategies in HRC applications.

V. Conclusion

This paper presents the development of an integrated object cluster tracking and velocity estimation method that is based on depth data only and is experimentally verified in use within a robot workspace. Depth cameras mounted in the robot's base are perceiving the exterior. By removing the robot from the point cloud only depth information on external objects remains. With an agglomerative hierarchical clustering approach, the object clusters, and thus their properties are obtained. By using information on the volume, number of points, position, and velocity of the cluster, the proposed

Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361

tracking method enables object cluster tracking solely based on depth data. Subsequently, a Kalman filter in combination with a constant velocity model is utilized to estimate the object cluster's 3D velocity. For verification, an experiment with two objects moving within the robot workspace is designed. The proposed algorithm correctly tracks both repetitive and arbitrary motion of external objects with an maximal trackable object velocity of $9\,\mathrm{m\,s^{-1}}$ for initial tracking, and a velocity deviation of $3.4\,\mathrm{m\,s^{-1}}$ for already tracked objects for the considered experimental setup. Therefore, the proposed tracking approach constitutes a foundation for collision avoidance strategies in robotic tasks.

For future work, the feasibility of this approach in collision avoidance applications within robot environments will be evaluated. In such applications systems often need information on objects moving in the workspace, which requires a reliable method for tracking external objects.

References

[1] E. Abele, M. Weigold, and S. Rothenbücher, "Modeling and Identification of an Industrial Robot for Machining Applications," *CIRP Annals*, vol. 56, no. 1, pp. 387–390, 2007.

[2] S. Erkaya, "Investigation of joint clearance effects on welding robot manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 4, pp. 449–457, Aug. 2012.

[3] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat, and F. Wallhoff, "Joint-action for humans and industrial robots for assembly tasks," in *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, Aug. 2008.

[4] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.

[5] Y. He and S. Chen, "Recent Advances in 3D Data Acquisition and Processing by Time-of-Flight Camera," *IEEE Access*, vol. 7, pp. 12 495–12 510, 2019.

[6] H. Wang, N. Sang, and Y. Yan, "Real-Time Tracking Combined with Object Segmentation," in *2014 22nd International Conference on Pattern Recognition*. Stockholm, Sweden: IEEE, Aug. 2014, pp. 4098–4103. [Online]. Available: http://ieeexplore.ieee.org/document/6977415/

[7] Z. Han, T. Xu, and Z. Chen, "An improved color-based tracking by particle filter," in *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, Dec. 2011, pp. 2512–2515.

[8] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4553715/

[9] D. Damen, A. Gee, W. Mayol-Cuevas, and A. Calway, "Egocentric Real-time Workspace Monitoring using an RGB-D camera," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2012.

[10] I. Yamamoto, K. Nakamura, N. Matsunaga, and H. Okajima, "Crowd Tracking of Electric Wheelchair using RGB-D Camera with Median of Candidate Vectors Observer," in *2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)*, Sep. 2022, pp. 174–178.

[11] Y. Xie, Y. Lu, and S. Gu, "RGB-D Object Tracking with Occlusion Detection," Dec. 2019, pp. 11–15.

[12] J. Issac, M. Wuthrich, C. G. Cifuentes, J. Bohg, S. Trimpe, and S. Schaal, "Depth-based object tracking using a Robust Gaussian Filter," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2016.

[13] C. Ramer and J. Franke, "Work Space Surveillance of a Robot Assistance System Using a ToF Camera," *Advanced Materials Research*, vol. 907, pp. 291–298, apr 2014.

[14] C. Ramer, "Arbeitsraumüberwachung und autonome Bahnplanung für ein sicheres und flexibles Roboter-Assistenzsystem in der Fertigung," Ph.D. dissertation, Jan. 2018.

[15] M. Ruan and D. Huber, "Calibration of 3D Sensors Using a Spherical Target," in *2014 2nd International Conference on 3D Vision*. IEEE, dec 2014.

[16] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, jul 2002.

[17] A. Fernández and S. Gómez, "Solving Non-Uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms," *Journal of Classification*, vol. 25, no. 1, pp. 43–65, jun 2008.

[18] K. Saho, "Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design," in *Kalman Filters - Theory for Advanced Applications*. InTech, feb 2018.

Post-print version of the article: P. Gsellmann, C. Buchner, K. Egretzberger, M. Melik Merkumians, and G. Schitter, "Depth-data-based object cluster tracking and velocity estimation in robot workspace," *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2023. DOI: 10.1109/IECON51785.2023.10312361