Real-time Pallet Localization with 3D Camera Technology for Forklifts in Logistic Environments

Benjamin Molter, Johannes Fottner
Chair for Materials Handling, Material Flow, Logistics
Technical University of Munich, Department of Mechanical Engineering
Munich, Germany
{benjamin.molter, j.fottner}@tum.de

***Abstract*—This paper presents a novel approach for detection and localization of standardized euro pallets, which are orientated up to 90° in relation to the sensor plane. There is no a priori information about the pallets pose needed. We use a time-of-flight camera. Our algorithm is based on finding surfaces in the point cloud, which represent the three wooden blocks of a euro pallet. Different kinds of geometrical checks set up our detection pipeline, where no artificial markers on the pallets are needed. Since we perform the detection while driving a forklift, the algorithm must process the point cloud within a set time limit. The detection and localization result in the pallets position and orientation in relation to the camera coordinate system. This information can be provided to higher-level systems, like advanced driver assistance systems. The results show that the localization of pallets is possible in the scenario considered.**

Keywords—pallet detection, pallet localization, forklift, time-of-flight camera, Kinect camera, point cloud

I. Introduction

The detection and localization of pallets with sensors is a common task for different kinds of intra-logistical material handling purposes. The main objective is to detect whether any pallet is located in the field of view of the sensor and also its relative or absolute spatial orientation. Such positioning information is used mainly by automated guided vehicles (AGVs) or by human-operated forklifts to assist the pallet pick-up process.

In general, AGVs are designed for predetermined tasks, e.g. production supply or the simple transport of materials between two locations in-house. They determine their own absolute position in the operating area and acquire their destination for pallet pick-up or drop-off from a higher-level management system. Pallet detection itself is only performed when the AGV is more or less perfectly aligned to the front of a pallet. Whereas human-operated forklifts are used throughout the complete in-house transport process. The storage and retrieval of pallets from racks is one of its standard tasks. Usually, forklifts do not have any localization system enabling them to obtain their absolute position. Spatial orientation of the pallet is not known either and therefore this detection must be performed without any a priori knowledge. Pallet localization can be the basis to provide information to support the human operator during the pick-up process, in order to prevent any damages to goods or racks.

Object detection in general can be seen as a basic task in computer vision. We define object detection as the question "is the object in the recorded image?" and localization as "what is the spatial orientation in relation to the sensor?". In this paper, we introduce a novel approach for detecting and localizing pallets, which are orientated up to 90° in the direction of vehicle movement. Our method does not need any artificial markers to be placed on the pallet or any other modifications of the environment. For detection and localization purposes, a 3D camera with time-of-flight principle is used. Our work focusses on advanced driver assistance systems for forklift trucks.

II. Related Work

As already mentioned, the detection and localization of pallets can generally be divided into two groups: systems which can locate themselves and know the position of the target pallet, and systems which do not have any localization function and do not know the position of the target pallet. In addition, the known approaches to this problem can be distinguished by the type of sensor used. The research on pallet detection and localization to date has mainly been carried out for AGVs. For detection purposes, image-processing sensors are used primarily. Many works are based on 2D cameras, both in color and in monochrome variant. In some cases, the pallets must be equipped with artificial markers. Newer publications report the use of 2D laser scanners and 3D cameras, which have the advantage of direct depth values.

Already 20 years ago Garibotto introduced an autonomous forklift [1], where a 2D camera is mounted between the forks. An ultrasonic sensor determines the distance to the pallet. It is used to calculate the expected size of the openings for the forks in the current camera image. This calculated size is the basis for a search for dark areas of equal dimensions. If similar areas are found the centers of the openings can be determined. Byun and Kim follow the same approach [2]. They use a 2D camera and the back-projection method for the detection of such openings.

Lecking et al. show the detection of pallets for an AGV [3]. They use a 2D laser scanner, which delivers direct depth values. The wooden blocks of the target pallets are modified with artificial reflectors. The detection is based on comparison between the scanned environment with a known pattern of the wooden blocks. Bellomo et al. show a similar approach in [4].

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Wang et al. use a structured light sensor without any markers on the pallets to record the environment [5], but the general detection principle is the same.

The first use of a 3D camera with the time-of-flight principle is shown by Kleinert et al. 2012 as part of an advanced driver assistance system for forklifts [6]. The camera is integrated in the front end of one fork arm. The developed system gives the driver visual recommendations for the correct alignment of the forks to pick up the pallet. Pallet detection is based on finding planes inside the openings for the forks, when the camera has a good alignment to the front of the pallet.

In general, it can be stated that in the presented works the detection is performed only when the vehicle is already very well positioned in front of the target pallet. The free openings for the forks are always orientated in the direction of movement. Deviations of up to 50° appear possible, but are not mentioned clearly by the authors.

III. Problem Definition

All known approaches for pallet detection and localization want to find the free openings for the fork arms. The alignment between the sensor and the front of the pallet has a considerable impact on the success of the detection process. With these state-of-the-art approaches, it is not possible to detect pallets where the front is orientated up to 90° in relation to the sensor. But this scenario is very common whenever a forklift drives through an aisle in a pallet rack storage unit. The relative position between the forklift and the target pallet is necessary data for higher-level systems, such as advanced driver assistance systems. It is obvious that pallet detection and determination of relative position must be performed while the vehicle is in motion. This task must be processed so quickly that higher-level systems can respond to the results. This results in a real-time requirement for the algorithm. The typical vehicle velocity for unloaded forklifts moving along an aisle vary from the forklift model used and the prescribed guidelines of the company using the forklift.

IV. Pallet Detection And Localization

A. Camera hardware

We chose the Microsoft Kinect v2 camera, a 3D camera with time-of-flight principle, because it has certain advantages in comparison to 2D cameras. Logistical operation areas like a warehouse equipped with racks may have bad light conditions, especially at their lower levels. 2D cameras may need external illumination to work as expected. For object localization, some kind of depth values are needed. 2D cameras can not deliver these values directly. Whereas 2D laser scanners or 3D cameras acquire such data directly, due to the underlying physical work principle. 2D laser scanners have the disadvantage that their perception only operates on one level. 3D cameras provide a built-in infrared illumination system, which makes the use of external illumination obsolete. The Kinect v2 has a depth resolution of 512 x 424 pixel and a frame rate of 30 fps. The depth camera records a grayscale depth image, which is transformed into a 3D point cloud on the connected computer. The point cloud consists of voxel, which represents the x, y and z Cartesian coordinates of each point in the camera coordinate system. Additionally, the Kinect v2 has a 2D color camera but these images are not used in our procedure. The camera is mounted on top of the fork back (Fig. 1). It is tilted downwards about 10°. Otherwise the tips of the forks are not within the field of view. This is necessary in order to detect and locate the pallet just before the forks are placed in the free openings during the pick-up process. To process the point clouds, we use the Point Cloud Library (PCL), which was first released in 2011 by Rusu and Cousins [7]. The open source C++ library is released under the terms of the BSD license.

B. Point cloud preprocessing

The transformed point cloud consists of 217 088 points. To meet the real-time requirements, the number of points must be reduced as much as possible for downstream algorithms. Most of the PCL methods iterate through all present points. Hence, a lower number of points results in a faster iteration speed respectively computation time. The following steps describe the processing of one camera frame or rather one point cloud. Obviously, the steps are performed in a loop during reception of a live stream from the Kinect camera. Because of the physical function principle of the time-of-flight camera, different image areas, especially with different depths, have a different point density. To make the point cloud the same density, a voxel grid filter is applied. The filter divides the point cloud up into boxes with predefined dimensions and determines the centroid of all points included. The original points are replaced with the centroids. Compared to determine just the geometrical center, this filter leads to better representation of the underlying surface [8].

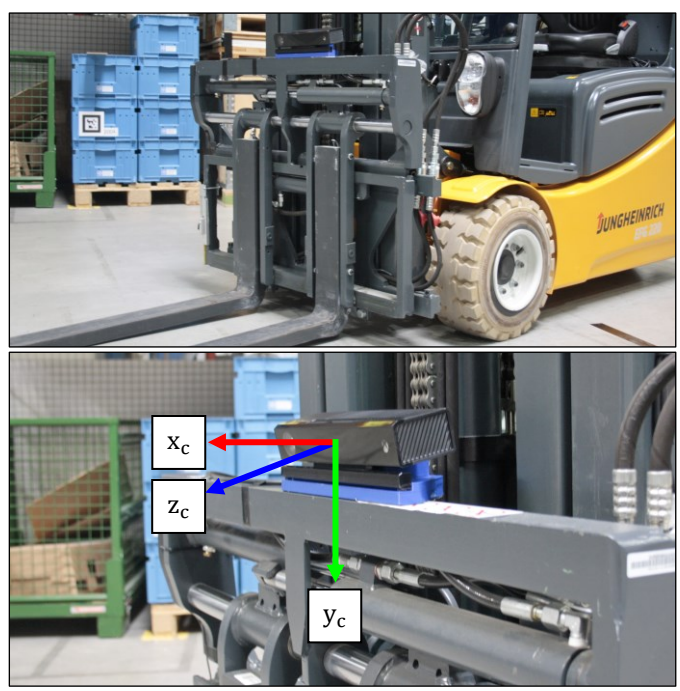

Fig. 1. Forklift with mounted Kinect camera and close-up view showing camera coordinate system.

In this paper, we focus on pallets, which are placed on the ground. Therefore, the next preprocessing step is to determine the ground plane and to remove all associated points from the point cloud. For plane detection, the well-known RANSAC algorithm is used. To prevent finding vertical planes like walls, the algorithm is configured to discard any planes with an angular deviation greater than 15° in relation to the axis z_c. The outputs of the plane detection process are the plane unit normal vector $\hat{n}$ and the distance p from the origin of the camera coordinate system (Fig. 1). The ground plane equation can be written in the Hessian normal plane form (1).

$$\hat{\mathbf{n}} \cdot \mathbf{x} = -\mathrm{p} \tag{1}$$

When all points representing the ground plane are known, they can be removed from the point cloud, so as to reduce the number of points. The plane equation itself is used for an additional reduction. The idea is to crop the point cloud vertically, so as to obtain only the points between the ground plane and a plane above and parallel to it (Fig. 2). The distance between the two planes is chosen so that one pallet fits well taking account of its height. We can use the point-plane distance equation (2) to obtain all the points with a distance d smaller than the height $d_{threshold}$ chosen (3).

$$\mathrm{d} = \hat{\mathbf{n}} \cdot \mathbf{x_0} + \mathrm{p} \tag{2}$$

$$\mathrm{d} \leq \mathrm{d_{threshold}} \tag{3}$$

Distance d is the shortest distance and therefore, the perpendicular distance between point x_0 and the plane. The resulting point cloud has the shape of a flat cuboid. These preprocessing steps are performed for every point cloud, because the height of the forks or the tilt angle of the mast can be changed by the operator. This can result in a new Kinect camera position and orientation with respect to the ground plane.

C. Geometrical pallet detection

Based on the preprocessing carried out, a geometrical pallet detection is performed. Our work is focused on standardized euro pallets [9]. All dimensions of the components are known. The main parts are wooden blocks and boards. We want to detect pallets, which are orientated up to 90° in direction of vehicle movement. This scenario represents a warehouse aisle with pallets left and right. When pallets are arranged near to each other, only the shorter pallet front side is visible. The short side of a pallet is characterized by three wooden blocks separated by two openings for the forks. Our approach is to detect the wooden blocks in the point cloud. Other parts of the pallet are not considered because they can be hidden by the load. Fig. 3 gives an overview of our detection pipeline.

The pallet detection starts with a segmentation of the point cloud. We use the region growing algorithm implemented in PCL. Starting with the preprocessed point cloud, the algorithm searches for related points representing a surface. It takes into account the point normals and the point curvatures. Both are determined by considering a given number of Cartesian neighbor points. The current point will be added to the temporary surface if the angle between two point normals is within the given threshold. The same procedure is used for the curvature.

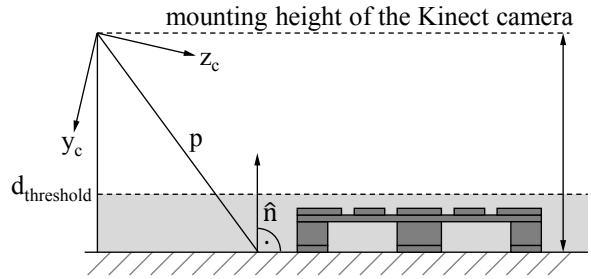

Fig. 2. Schematic of the Kinect v2 camera mounting position and the cropped ground plane.

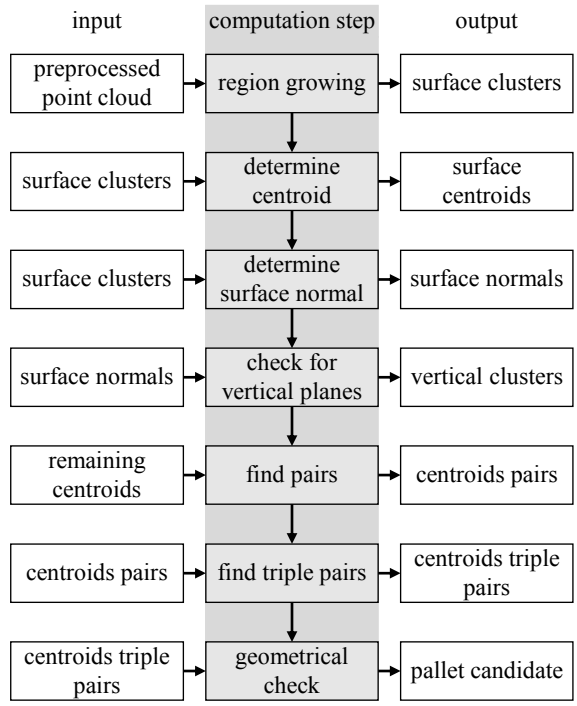

Fig. 3. Pallet detection and localization pipeline.

Points belonging to a determined surface are removed from the point cloud, so they are not considered in the next iteration. The results are point cloud clusters. Each cluster represents one found surface and is a subset of the original point cloud. The centroid, which can here be assumed as the surface center, and the surface normal of each cluster are needed for further processing. The surface normal is used to determine if a surface is perpendicular in the relation to the ground plane. The angle φ between two surface normals is calculated with the y-components of the normal vectors (4).

$$\varphi = \tan^{-1}\left(\frac{\mathrm{cluster_normal_y}}{\mathrm{n_y}}\right) \tag{4}$$

Since we want to detect the wooden blocks, all horizontal or inclined surfaces are discarded. Theoretically, a perfect perpendicular surface has a normal angle of 90°. Due to noise, in practice, this value must be extended to an interval ε_1, so that almost perpendicular surfaces are also allowed for further processing. The idea here is to find three centroids, which represents the front side surface of the three wooden blocks. The centroids determined are used for the first step of the

geometrical pallet detection. The Euclidean distance between all remaining centroids is calculated. The ideal distance between the outer blocks and the middle block is 0.350 m. Again, in practice, noise must be considered. Therefore, the hard threshold value is changed to a useful interval ε_2. To sum up, we have two necessary conditions (5) and (6), which need to be fulfilled until here.

$$\varphi \lesseqgtr 90° \pm \varepsilon_1 \tag{5}$$

$$d \lesseqgtr 0.350\ \mathrm{m}\ \pm \varepsilon_2 \tag{6}$$

The results are pairs of points, which have a distance that is within the interval. In these pairs of points, it is now necessary to find those with a common point. A common point is a good candidate for the centroid of the middle block. The constellations found are then grouped into three-point pairs. The next steps examine the three-point pairs geometrically. The first is derived directly from the dimensions of the pallet. Since we want to detect the wooden blocks, all cluster centroids should be situated more or less in the geometrical middle of the front side of the wooden blocks. This leads to the third condition; namely, that all three centroids have to be located on the front side of the respective wooden block. The fourth condition is that the centroids of one three-point pair must be almost the same height above the ground plane. All four conditions are shown in the schematic in Fig. 4. The gray areas visualize the possible interval for each condition.

In general, the clear localization of any rigid body in three-dimensional space requires three components of translation and three components of rotation. Hence, the rigid body has six degrees of freedom. The intra-logistical environment allows us to make some assumptions that reduce the degrees of freedom. The pallet lies directly on the ground, which is a flat surface, with no major bumps in it.

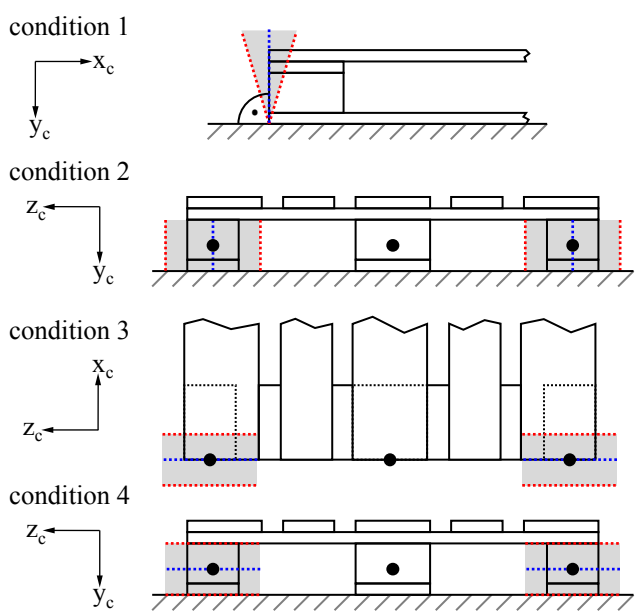

Fig. 4. All four geometrical conditions for pallet detection, when the pallet has an orientation of 90° to the Kinect camera. Gray areas indicate the range of valid values.

Thus, we can assume that the pallet has no roll or pitch angle. Also, the height perpendicular to the ground can be assumed to be zero. We chose the geometric middle of the pallet front side as our reference point for localization purposes. Therefore, the centroid of the middle wooden block surface is used for the remaining two components of translation. Since our algorithm works with a point cloud, the Cartesian coordinates can be read directly from the corresponding variable field. The yaw angle is calculated considering the two outer wooden blocks. Fig. 5 shows a schematic top view illustrating the relationships for the yaw angle calculations. Starting with the Cartesian coordinates from the outer wooden blocks the yaw angle is calculated with the arctangent function (7).

$$\Psi = \tan^{-1}\left(\frac{\Delta z_{\mathrm{pallet}}}{\Delta x_{\mathrm{pallet}}}\right) \tag{7}$$

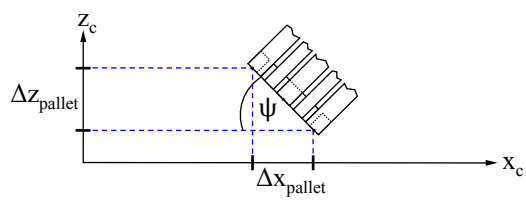

Fig. 5. Schematic showing for calculation of pallet yaw angle.

V. Test And Results

To prove our detection and localization pipeline we selected two scenarios. The first scenario is designed to compare our algorithm to other, state-of-the-art approaches. It consists of a single pallet, which is arranged at an angle of 45°. The second scenario consists of a single pallet, which is oriented at 90° in relation to the direction of movement. The second arrangement can be compared to the lowest level of a pallet rack. In both scenarios, there is no a priori information about the pose of the pallets. The pallets are unloaded and not modified with any kind of artificial reflector or marker. We have tested both scenarios in two settings. The first one is static, which means that the forklift stands still. In the second setting, we test our algorithm in a dynamic way, where we drove the forklift so that we approached the pallet in different distances x_c. The pallet was placed in the field of view of the Kinect camera.

The Kinect v2 camera is connected via USB 3.0 to a laptop running Ubuntu 16.04. The laptop is equipped with an Intel Core i7-6820HK CPU and 16 GB ram. The Kinect v2 camera driver [10] is configured to maximum depth values of 5.5 m. The Kinect driver uses the Nvidia CUDA pipeline for calculating the point cloud from the depth image. The laptop has a Nvidia GeForce GTX 1070 GPU.

In Fig. 6 the detection pipeline is shown with recorded point clouds. The first point cloud a) shows the 90° scenario. The ground plane found is colored blue. In the front area the forks are visible. The next point cloud b) is cropped down to the pallet. It shows the result of the segmentation. Every vertical plane found is visualized. The colorization is random and just for better visualization. Black points do not belong to any segments.

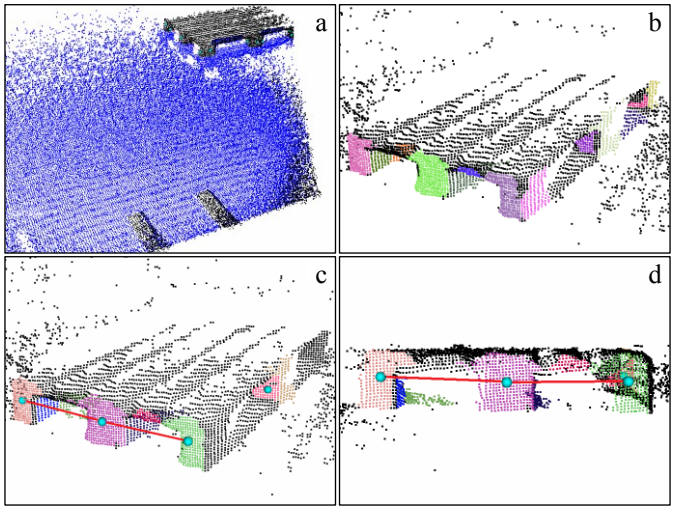

Fig. 6. a) point cloud of the 90° scenario with detected ground plane (blue), b) point cloud segmentation of vertical planes with region growing algorithm, c) and d) visualization of the geometrical pallet detection

The point cloud c) in the lower row shows the computed centroids as small light blue spheres on the planes found. Between the visualized centroids a red line is drawn, that shows a valid pallet by the geometrical pallet detection. The last point cloud d) is another view for the purpose of better understanding. All conditions from Fig. 4 are fulfilled. The black points around the pallet are noise. Some of them belong to the ground plane, but are not taken into account, because they do not match the criteria defined for the ground plane. Other points have their origin in the multipath effect, which is a known handicap of time-of-flight cameras [11].

The static localization results are shown in Table 1. The mean of all valid localizations and the standard deviation are shown. The latter is in the low centimeter region respectively below 1°, which are suitable values for our purpose. As we did not have a calibrated measuring instrument available to obtain the ground truth of our tests, our results are evaluated with the maximum depth error from the Kinect camera, as determined from Khoshelham and Elberink in [12]. The maximum depth error reaches 0.04 m at a distance of 5.0 m. Based on our use case of a pallet pick-up, a fork width of 0.1 m can be assumed. One opening hole has a width of 0.228 m, so that a gap of 0.064 m remains left and right, when the fork enters the opening hole in the middle (Fig. 7). The maximum Kinect camera error, combined with the values of the standard deviation, generally do not exceed the dimensions of the gap. There is just one violation of the limits, which is the case of the z standard deviation in the 90° scenario. However, this small excess can be neglected, because it is only 0.001 m and the maximum error mentioned for the Kinect camera is valid for a distance of 5.0 m. The error is smaller at shorter distances. Therefore, the accuracy of our localization approach is considered sufficiently good.

Table 2 gives an overview of the detection rate related to the number of recorded point clouds and the average calculation rate. In the first scenario, almost all pallets were detected. In the second scenario, the pallet was correctly localized in only 55 % of the recorded point clouds.

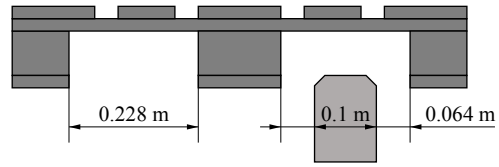

Fig. 7. Side view of a standardized euro pallet, which provides an overview of the dimensions of the opening hole, one fork and the free gap.

TABLE I. MEAN AND STANDARD DEVIATION OF THE PALLET LOCALIZATION FOR THE STATIC PALLET SCENARIOS

scenario	mean			standard deviation		
	x (m)	z (m)	Ψ (°)	x (m)	z (m)	Ψ (°)
45°	0.323	3.390	44.715	0.015	0.005	0.220
90°	1.886	3.716	90.506	0.018	0.025	0.871

TABLE II. LOCALIZATION RATE UND CALCULATION RATE FOR THE STATIC PALLET SCENARIOS

scenario	recorded point clouds	valid pallet localizations	localization rate (%)	calculation rate (fps)
45°	877	874	99.66	18.99
90°	853	470	55.10	18.80

This seems insufficient, but considering the depth image frame rate of 30 fps and the calculation rate measured, our algorithm delivers new localization values every 10.34 fps ≙ 96.71 ms. This is adequate for our purpose.

During movement with our dynamic setting the number of points obviously depends on the objects in the field of view. In Fig. 8 the influence of the number of points after the preprocessing on the calculation rate is shown. The number of points are plotted over time. As we did not have any other objects in the field of view, the rise shown in the black graph is the pallet. Additionally, the calculation rate is plotted over time. The calculation rate drops when the pallet enters the field of view, but did not drop permanently under 15 fps. The detection and localization is therefore considered to be real-time.

The detection and localization results for the dynamic setting can be seen in Fig. 9 and Fig. 10. The average velocity was 0.55 m/s for the 45° scenario and 0.42 m/s for the 90° scenario. The pallets drawn in the charts are not true-to-scale, but they are still helpful for a general overview. The charts show that our approach can detect and localize pallets while the forklift is in motion. The detection range in both axes is limited due to the field of view of the Kinect camera. Every marker stands for a valid pallet localization. Markers with the same shape and color represent one drive. For both scenarios six drives with variable distances in direction x_c were made. In the 90° scenario, the pallet could no longer be recognized if the distance x_c was greater than 2.0 m. The most valid detections were possible with the distances x_c of 0.5 m (45° scenario) and 1.0 m (90° scenario).

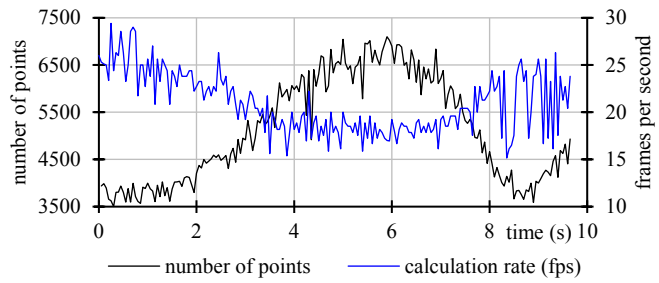

Fig. 8. Number of points over time (black) and calculation rate over time (blue) of the dynamic setting for the 90° scenario.

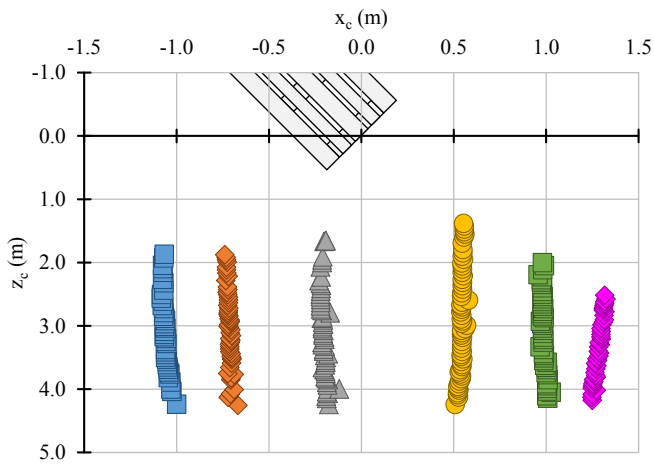

Fig. 9. Localization results for a single pallet in the dynamic 45° scenario.

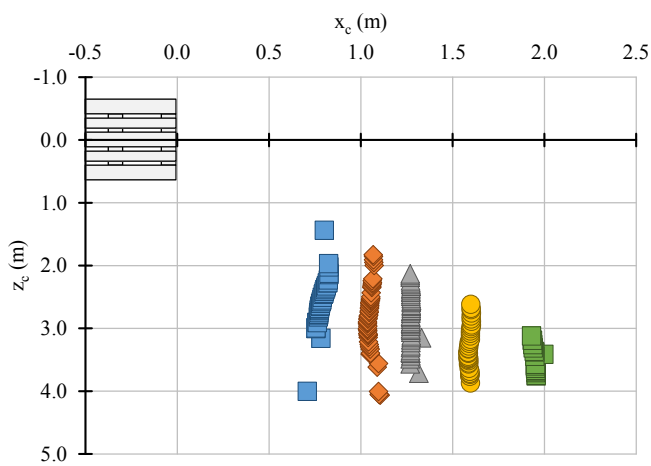

Fig. 10. Localization results for a single pallet in the dynamic 90° scenario.

VI. Conclusion And Outlook

In this paper, a new approach for detection and localization of euro pallets is proposed. We can detect pallets, which are orientated up to 90° in relation to the sensor plane. Our detection pipeline is based on geometrical features of the wooden blocks of the pallets. The Kinect v2 camera is used to record depth images, which are transformed into point clouds. The point clouds are processed with the help of the open source Point Cloud Library. Experiments showed that we can detect pallets under both static and dynamic conditions. Hence, a Kinect v2 camera was mounted on the front of a forklift. We showed that it is possible to detect and localize pallets with at least 15 fps, while driving the forklift.

Future work will focus on improvement of the calculation rate, in order to make higher forklift velocities possible. Also, the robustness of our algorithm could be improved if we can consider more points of the recorded point cloud at the same time. The algorithm will be extended so that palettes can be located at different levels as long as they are in the field of view of the camera.

Acknowledgment

The authors are grateful for the support of the Dr. Friedrich Jungheinrich Foundation from Hamburg, Germany.

References

[1] G. Garibotto, S. Masciangelo, M. Ilic, and P. Bassino, “Robolift: a vision guided autonomous fork-lift for pallet handling,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 656–663, 1996.

[2] S. Byun and M. Kim, “Real-time positioning and orienting of pallets based on monocular vision,” *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 2, pp. 505–508, 2008.

[3] D. Lecking, O. Wulf, and B. Wagner, “Variable Pallet Pick-Up for Automatic Guided Vehicles in Industrial Environments,” *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1169–1174, 2006.

[4] N. Bellomo *et al.,* “Pallet Pose Estimation with LIDAR and Vision for Autonomous Forklifts,” *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, vol. 42 issue 4, pp. 612–617, 2009.

[5] Z. He, X. Wang, J. Liu, J. Sun, and G. Cui, “Feature-to-Feature Based Laser Scan Matching for Pallet Recognition,” *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 2, pp. 260–263, 2010.

[6] S. Kleinert and L. Overmeyer, “Using 3D camera technology on forklift trucks for detecting pallets,” *Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST)*, pp. 55–62, 2012.

[7] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[8] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,” Phd Thesis, Technical University of Munich, Munich, 2009.

[9] EN 13698-1:2003, European Committee for Standardization, "Pallet production specification Part 1: Construction specification for 800 mm x 1200 mm flat wooden pallets", 2003.

[10] L. Xiang *et al.,* "libfreenect2: Release 0.2": Zenodo, 2016.

[11] D. Lefloch *et al.,* “Technical Foundation and Calibration Methods for Time-of-Flight Cameras,” in *Lecture Notes in Computer Science*, vol. 8200, *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications: Dagstuhl 2012 Seminar on Time-of-Flight Imaging and GCPR 2013 Workshop on Imaging New Modalities*, M. Grzegorzek et al., Eds., Berlin Heidelberg: Springer, 2013, pp. 3–24.

[12] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications,” *Sensors (Basel, Switzerland)*, vol. 12, no. 2, pp. 1437–1454, 2012.