Detection of Parcel Boxes for Pallet Unloading Using a 3D Time-of-Flight Industrial Sensor

Riccardo Monica[2], Jacopo Aleotti[1,2] and Dario Lodi Rizzini[1,2]
[1]RIMLab, Department of Engineering and Architecture
[2]CIDEA Centro Interdipartimentale per l'Energia e l'Ambiente
University of Parma, Italy
{riccardo.monica, jacopo.aleotti, dario.lodirizzini}@unipr.it

***Abstract*—This work presents a 3D vision system for automatic detection of cardboard parcel boxes of known size, located on the top layer of a pallet of known height. An industrial Time-of-Flight (ToF) sensor is adopted that can operate in different illumination conditions thanks to the use of an infrared light source. The perception system is intended for application in industrial warehousing for end-of-line operations, like robot depalletizing. The proposed method does not assume any predefined layout of parcels and, therefore, it can work even with an incomplete layer of misaligned cardboard boxes. The developed solution first extracts all possible object hypotheses, then an optimization problem is solved, based on a genetic algorithm, to exclude conflicts. Experiments have been performed on a real dataset including complex configurations of tightly packed parcels.**

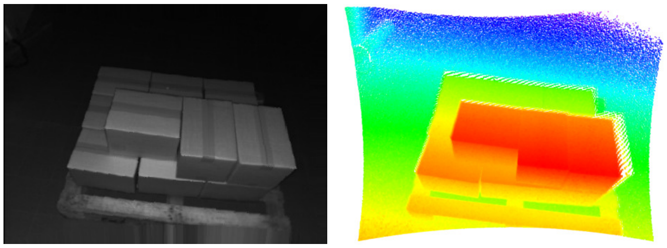

Fig. 1. An input intensity image R (left) and the point cloud P, where the color changes from red to blue as the distance from the camera increases (right).

I. Introduction

In end of line depalletizing, packed products are picked up from pallets stored in a warehouse and prepared for shipping, e.g. they are loaded on trucks or on other target pallets [1], [2]. Items are usually cardboard boxes, shrink-wrapped beverages, or other workpieces. This work presents a 3D vision system to detect cardboard parcel boxes of known size on the top layer of a pallet. It is also assumed that the height of the pallet is known in advance. Boxes must be recognized individually, so that each single parcel can be picked up one at a time according to some plan.

The proposed method adopts a Time-of-Flight (ToF) sensor, which can be fixed or carried by an automated guided vehicle (AGV) equipped with a robot manipulator. The approach does not assume any predefined layout of the parcels, i.e. the top layer of the pallet may be incomplete and the cardboard boxes can also be misaligned. The developed solution extracts all possible object hypotheses, then a genetic algorithm is executed to optimize the results. Experiments performed on a real dataset are reported to assess the efficiency and the accuracy, including complex cases with tightly packed parcels.

The closest work to ours is [3], where a similar labeling approach was presented based on a genetic algorithm. The main differences are that [3] adopted a gray scale camera, moreover, the sensor was placed above the pallet, whereas in our approach the camera does not need to be placed directly above the pallet. Prasse et al. [4], [5], [6] investigated a method to detect poses of parcels on a pallet using a ToF sensor and RFIDs, which were exploited to provide an estimated 3D model of the layer structure. In [7], [8], [9] methods for recognizing boxes of arbitrary size in cluttered environments were presented. However, the proposed solutions were based on 2.5D edge detectors that may fail to detect aligned boxes that are in contact with one another where two planar surfaces have a small angle between them. Moreover, data acquisition was performed by using a planar laser scanner, mounted on a robot arm in eye-in-hand configuration. Other applications of parcel detection were reported in [10], [11], [12], [13]. In [11] a system was presented to detect the size of parcels moving on a conveyor belt using RGB cameras. Doliotis et al. [12] proposed an autonomous robotic manipulation system, based on 3D vision, for detection and unloading of cardboard boxes from floor loaded shipping containers or semi-truck trailers.

II. Method

At each frame, the time-of-flight camera acquires an organized point cloud $P = \{p_{ij}\}$, where each p_{ij} is a point in 3D space with pixel coordinates (i, j). Moreover, the camera provides an intensity image R, which contains an intensity value r_{ij} for each pixel with the amount of light returned to the sensor (Fig. 1).

The goal of the proposed method is the detection of the observed boxes in each frame and the computation of their pose in 3D space with respect to the camera reference frame. We allow the camera to be tilted with respect to the top plane Ω of the highest layer of boxes. We obtain Ω with respect to the camera reference frame by repeatedly applying RANSAC to the range image to obtain all planes in the scene, and then by manually selecting the top plane.

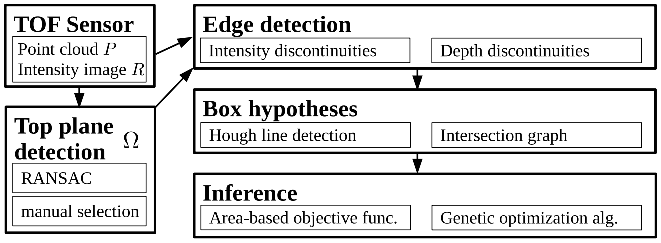

Fig. 2. The pipeline of the proposed approach.

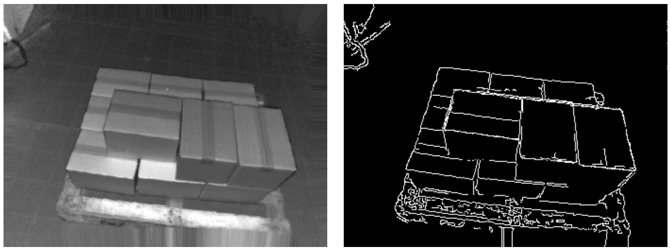

Fig. 3. The light-compensated intensity image R' (left) and the edge image E (right).

The pipeline of the proposed approach is illustrated in Fig. 2. First, discontinuities are detected using both the depth image and intensity image (Section II-A). Then, the Hough transform is applied to convert the detected discontinuities into 2D lines. Intersections between lines are then computed and box candidates are found by exploiting a graph (Section II-B). As a large amount of conflicting box candidates are found, in the inference phase a subset of non-conflicting boxes is selected, by maximizing an objective function (Section II-C) using a genetic algorithm (Section II-D).

A. Edge detection

The goal of the edge detection phase is to generate a binary image $E = \{e_{ij}\}$, where e_{ij} is 1 if the pixel may belong to a box edge, and 0 otherwise. Edges may be detected either from intensity discontinuities in the intensity image, or from depth discontinuities in the depth image. Intensity discontinuities allow box edges to be detected where boxes are tightly packed, and no depth discontinuity can be reliably detected.

Assuming that the only source of illumination in the infrared (IR) spectrum is the camera itself, the irradiance decreases quadratically with the distance of the observed object. As the distance $\|p_{ij}\|$ from the camera is known, a light-compensated intensity image R' can be generated (Fig. 3, left), where pixels r'_{ij} are given by

$$r'_{ij} = r_{ij}\,\|p_{ij}\|^2 \qquad (1)$$

Then, a standard Canny algorithm is applied to R', with upper and lower thresholds U_{canny} and L_{canny}, resulting in the Canny edge image e^C_{ij}.

To compute depth discontinuities, an image H is generated from the point cloud where each image pixel h_{ij} is equal to 1 (inlier) if the distance of $p_{i,j}$ from Ω is lower than a threshold

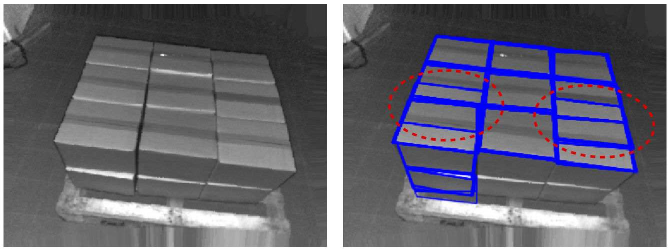

Fig. 4. Example scenario with cardboard boxes arranged in a grid (left) and the generated box hypotheses (blue rectangles, right). Adhesive tapes may cause the detection of spurious box hypotheses (highlighted by red ellipses) which can be eliminated in the inference phase.

th_{inlier}, and 0 otherwise (outlier). Then, a filtered image $H' = \{h'_{ij}\}$ is obtained by applying a closing operator with circular kernel and radius r_{close} to remove isolated 0 values caused by noise. An inlier pixel (i,j) is a depth discontinuity $e^D_{ij} = 1$ if there exists an outlier pixel in its neighborhood, in both the original and the closed image, i.e.

$$e^D_{ij} = h_{ij} \wedge h'_{ij} \wedge \left(\exists\,(i',j') \in N_4(i,j) \mid \neg h_{i'j'} \wedge \neg h'_{i'j'}\right) \qquad (2)$$

where $N_4(i,j)$ is the 4-neighborhood of pixel (i,j). Finally, the edge image E (Fig. 3, right) is computed as $e_{ij} = e^D_{ij} \vee e^C_{ij}$.

B. Box hypotheses

A set of 2D lines $L = \{l_m\}$ is computed in the edge image E using the standard OpenCV Hough transform, with distance resolution ρ_{hough}, angular resolution θ_{hough} and accumulator threshold th_{hough}. A set of intersection pixels $S_I = \{(i,j)\}$ is then found by intersecting each pair of lines (l_m, l_n), and then by finding the nearest pixel (i,j), if the intersection is in the image. The intersection pixels are taken as nodes of a connectivity graph. Pairs of nodes in the graph that correspond to pixels (i,j) and (i',j') are connected by an edge if (i,j) and (i',j') belong to the same line. An edge is accepted only if there is a ratio of at least th_{connect} pixels with $e_{ij} = 1$ on the corresponding segment.

Each cycle of length four in the connectivity graph is then searched, to identify a set of possible box hypotheses $\mathbb{B} = \{B_n\}$ as quadrilaterals in the image. A quadrilateral is accepted only if the size matches the known size $a \times b$ of a box top face, within a tolerance σ_{box}, and if the intersections are right-angled with an angular tolerance θ_{box}. Quadrilateral size is evaluated on the projected points p'_{ij} of points p_{ij} on plane Ω to improve accuracy. That is, given the quadrilateral $B_n = \{(i_0,j_0),(i_1,j_1),(i_2,j_2),(i_3,j_3)\}$, the following conditions must be met for all $k \in \{0 \dots 3\}$:

$$\begin{cases} \left| \left\| p'_{i_k j_k} - p'_{i_{k+1} j_{k+1}} \right\| - a \right| < \sigma_{\text{box}} & \text{if } k \text{ is odd} \\ \left| \left\| p'_{i_k j_k} - p'_{i_{k+1} j_{k+1}} \right\| - b \right| < \sigma_{\text{box}} & \text{if } k \text{ is even} \end{cases} \qquad (3)$$

$$\left| \frac{\left\langle p'_{i_k j_k} - p'_{i_{k+1} j_{k+1}},\; p'_{i_{k+1} j_{k+1}} - p'_{i_{k+2} j_{k+2}} \right\rangle}{\left\| p'_{i_k j_k} - p'_{i_{k+1} j_{k+1}} \right\| \left\| p'_{i_{k+1} j_{k+1}} - p'_{i_{k+2} j_{k+2}} \right\|} \right| < \sin(\theta_{\text{box}}) \qquad (4)$$

where $(i_4, j_4) \equiv (i_0, j_0)$ and $(i_5, j_5) \equiv (i_1, j_1)$.

C. Area-based objective function

The number of box hypotheses usually overestimates the true number of boxes located in the top layer of the pallet, as there are multiple overlapping box hypotheses which correspond to the same real box. Moreover, other false positives boxes may occur because of spurious lines that may be detected due to tags, labels, or adhesive tape strips on the cardboard boxes. As boxes are often arranged in a grid, the distance between spurious lines can indeed be equal to the box sizes (Fig. 4). Therefore, in the inference phase a subset $\mathbb{S} = \{S_n\}, \mathbb{S} \subset \mathbb{B}$, of box hypotheses is selected from the power set $\mathcal{P}(\mathbb{B})$ so that conflicts between the hypotheses S_n are minimized.

In particular, in the inference phase an optimization problem is solved to find the set of boxes $\mathbb{S} \in \mathcal{P}(\mathbb{B})$ which maximizes the area $F(\mathbb{S})$ of the image (in pixels) covered by exactly one of the top planes of the boxes:

$$\mathbb{S} = \underset{\mathbb{S}' \in \mathcal{P}(\mathbb{B})}{\text{argmax}} \; F(\mathbb{S}') \tag{5}$$

where

$$F(\mathbb{S}) = \sum_{S \in \mathbb{S}} A(S) - \gamma_O \sum_{S \in \mathbb{S}} O(S) - \gamma_I \sum_{S \in \mathbb{S}} I(S) + \\ - \gamma_C \sum_{S \in \mathbb{S}} \sum_{\substack{S' \in \mathbb{S} \\ S' \neq S}} C(S', S) \tag{6}$$

The first term $\sum_{S \in \mathbb{S}} A(S)$ in (6) is the sum of the quadrilateral area $A(S)$ (in pixels) of each box, penalized by three terms. The first penalty term is the area of the outlier pixels $\sum_{S \in \mathbb{S}} O(S)$ in image H, i.e. the number of pixels with $h_{i,j} = 0$. The second penalty term $\sum_{S \in \mathbb{S}} I(S)$ is the difference (in pixels) between the quadrilateral area (in meters) $A^r(S)$ and the expected area $a \times b$ of the top face of a box:

$$I(S) = \frac{|A^r(S) - ab|}{ab} A(S) \tag{7}$$

The double summation in (6) is a third penalty term that accounts for the overlap between all pairs of quadrilaterals, where function $C(S_1, S_2)$ is defined as the area, in pixels, of the intersection between two box hypotheses S_1 and S_2, i.e. $C(S_1, S_2) = A(S_1 \cap S_2)$. The Clipper library [14] was adopted to efficiently compute intersections and areas. Coefficients γ_O, γ_I and γ_C are parameters weighting each penalty term.

D. Genetic optimization algorithm

The exact optimal solution of (5) is usually computationally infeasible. Therefore, a genetic algorithm approach is adopted, based on OpenGA [15], using $F(\mathbb{S})$ as fitness. An individual in the population is defined as a subset $\mathbb{S} \subset \mathbb{B}$, represented by a dynamic array of box hypothesis indexes. The population size was fixed at G_{pop}. Evolution terminates when the best individual does not change for G_{stall} generations.

A Fill$(\mathbb{S}, \mathbb{B})$ operator is defined, which generates a superset $\mathbb{S}' \supset \mathbb{S}$ by repeatedly adding a random box hypothesis $B \in \mathbb{B}$ to $\mathbb{S}$, subject to $F(\mathbb{S} \cup \{B\}) > F(\mathbb{S})$. The Fill operator always produces a local (greedy) maximum, i.e. any additional box hypothesis added to $\mathbb{S}$ causes the objective function to decrease. By using Fill as the final step of every mutation and crossover operator, we ensure that each individual is always a greedy local maximum. Individuals of the initial population are initialized by using the Fill operator on the empty set, i.e. Fill$(\varnothing, \mathbb{B})$.

TABLE I
PARAMETERS CONFIGURATION.

Description	Symbol	Value
Canny upper/lower threshold	$U_{\text{canny}}/L_{\text{canny}}$	120/60
Plane inlier threshold	th_{inlier}	5 cm
Inlier image closing radius	r_{close}	2 px
Hough distance/angular resolution	$\rho_{\text{hough}}/\theta_{\text{hough}}$	1 px / 1 degree
Hough threshold	th_{hough}	15
Edge acceptance ratio	th_{connect}	0.5
Box size	$a \times b$	0.315×0.232 (m)
Box size/angular tolerance	$\sigma_{\text{box}}/\theta_{\text{box}}$	2 cm / 10 degrees
Outlier/area/overlap penalty coeff.	$\gamma_O/\gamma_I/\gamma_C$	2/8/2
GA population	G_{pop}	100
GA stall generations	G_{stall}	20
GA crossover/mutation probability	$P_{\text{C}}/P_{\text{M1}}/P_{\text{M2}}$	0.7/0.1/0.1

TABLE II
POSITION ACCURACY OF THE BOX DETECTION ALGORITHM ON 200 IMAGES FOR EACH CONFIGURATION WITH RESPECT TO GROUND TRUTH.

Config.	number of	avg. detected	false	pairwise distance [mm]	
	boxes	boxes	neg.	(avg)	(std.dev.)
A	9	8.99	2	6.70	8.65
B	6	5.94	12	7.72	9.54
C	7	6.83	34	7.11	8.71
D	3	3.00	0	6.77	7.93
E	4	3.99	2	5.20	6.08

Operator Mutate1$(\mathbb{S})$, with probability P_{M1}, removes up to three random elements $\{r_1, r_2, r_3\} \subset \mathbb{S}$:

$$\text{Mutate1}(\mathbb{S}) = \text{Fill}(\mathbb{S} - \{r_1, r_2, r_3\}, \mathbb{B}) \tag{8}$$

The Mutate2$(\mathbb{S})$ operator, with probability P_{M2}, removes a random element $r \in \mathbb{S}$ and replaces it with the one in $\mathbb{B}$ that maximizes the objective function:

$$\begin{gathered} \mathbb{S}^- = (\mathbb{S} - \{r\}) \\ S^\star = \underset{S' \in \mathbb{B}}{\text{argmax}} \, F\big(\mathbb{S}^- \cup \{S'\}\big) \\ \text{Mutate2}(\mathbb{S}) = \text{Fill}\big(\mathbb{S}^- \cup \{S^\star\}, \mathbb{B}\big) \end{gathered} \tag{9}$$

The crossover operator Crossover$(\mathbb{S}_1, \mathbb{S}_2)$ generates a new individual from two individuals $\mathbb{S}_1$ and $\mathbb{S}_2$ by using $\mathbb{S}_1 \cup \mathbb{S}_2$ as a pool from which Fill extracts elements:

$$\text{Crossover}(\mathbb{S}_1, \mathbb{S}_2) = \text{Fill}(\text{Fill}(\varnothing, \mathbb{S}_1 \cup \mathbb{S}_2), \mathbb{B}) \tag{10}$$

Each generation, the crossover operator is applied to a fraction P_{C} of the population.

III. EXPERIMENTAL EVALUATION

The experimental evaluation was carried out using an IFM Electronics O3D303 sensor: an industrial depth camera with

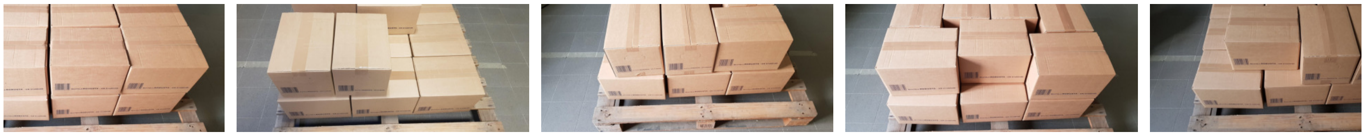

Fig. 5. Pictures of the six box configurations (A to E) used in the experiments (top row). Examples of image frames where all the boxes have been correctly detected (the top planes of the boxes are highlighted by a colored quadrilateral).

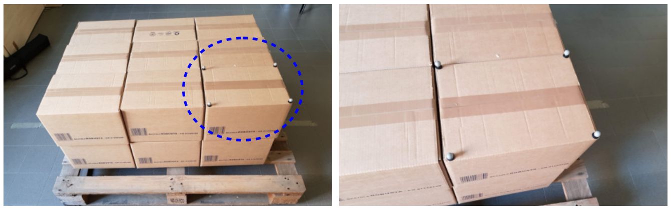

Fig. 6. Configuration A, with four OptiTrack markers temporarily placed on the box in the lower right corner, highlighted by a blue circle (left). Close-up detail of the box (right).

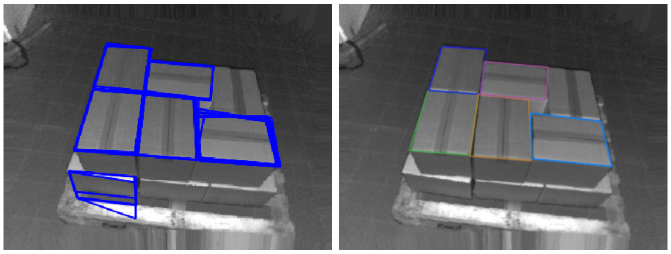

Fig. 7. Example of detection failure of a box (upper-right corner) in Configuration B (right). Failure is caused by the absence of any suitable box hypothesis for the box (blue rectangles, left).

TABLE III

AVERAGE COMPUTATION TIME [MS] AND STANDARD DEVIATION FOR EACH PHASE.

Phase	Computation time
Intensity discontinuities computation	0.3 ± 0.1
Depth discontinuities computation	1.4 ± 0.2
Hough line detection	3.9 ± 0.2
Intersection graph generation	216.3 ± 55.1
Box hypotheses generation	92.3 ± 49.9
Objective function computation	211.6 ± 390.3
Genetic algorithm optimization	26.0 ± 27.0
Total	551.9 ± 440.1

resolution 352 × 264 pixels. The sensor observes the scene consisting of a pallet base and two layers of boxes at a distance of about 110 cm and it detects the boxes on the top layer. The box size is 315 × 232 × 210 mm. Five different box configurations on the top layer were considered, and labeled from A to E, as shown in Fig. 5. For each configuration the position of each box center has been estimated by using an OptiTrack motion capture system. In particular, four markers have been temporarily placed on the top face corners of each box (Fig. 6) and the centroid position has been measured with respect to the OptiTrack reference frame. The pairwise distances between the box centroids have been used as ground truth values to assess the accuracy of the proposed detection algorithm.

The proposed algorithm was implemented in C++ and it has been executed on Ubuntu 18.04, on an Intel i7-6700 @ 3.40GHz CPU, 32 GB RAM. The parameters configuration is reported in Table I. Plane Ω was estimated once as discussed in section II. The dataset used in the experiments consists of 200 depth images for each box configuration A-E. Table II illustrates the outcome of the experiments. The average number of detected boxes is very close to the real one. Configurations B and C have the highest number of false negatives possibly due to their less regular arrangement with gaps and occlusions among the boxes (see Figure 5-C). Most false negatives are caused by the absence of a box hypothesis of correct size for a box (Fig. 7).

Accuracy has been estimated by matching and comparing the pairwise distances between the centroids of the detected boxes and the ground truth distances. Association among pairwise distances was computed by the Hungarian algorithm. The average distance error is about 7 mm. Overall, the error depends on both the detection algorithm and the estimated ground truth. The standard deviation value is comparable or slightly greater than the average distance error. The estimated accuracy complies to the requirements of warehouse manipulation tasks.

Table III reports the computation time for each phase, averaged over the five configurations with 200 frames each. The "intersection graph generation" phase is the most computationally expensive phase, due to the large amount of redundant similar lines generated by the Hough line detection. The high time of the "objective function computation" phase is due to the computation of all $C(S', S)$ in $F(\mathbb{S})$ (6), for every possible pair $S, S' \in \mathbb{B}$, as confirmed in Table IV, where for each configuration the number of box hypotheses $|\mathbb{B}|$ is reported alongside the "objective function computation" time. A larger number of box hypotheses corresponds to a higher computation time. Table IV also reports the number of generations executed by the genetic algorithm before convergence. More generations are required for more complex scenarios.

IV. CONCLUSION

This work presented a 3D computer vision approach for automatic detection and localization of cardboard parcel boxes of known size on the top layer of a pallet. The proposed method offers an efficient solution that does not assume any apriori knowledge on the layout of the boxes. The system

TABLE IV
AVERAGE NUMBER AND STANDARD DEVIATION OF BOX HYPOTHESES, OBJECTIVE FUNCTION COMPUTATION (OFC) TIME [MS] AND NUMBER OF GENETIC ALGORITHM GENERATIONS, FOR EACH CONFIGURATION.

Configuration	Box hyp.	OFC time	Generations
A	1020 ± 382	752.8 ± 597.5	47.0 ± 6.6
B	464 ± 192	177.1 ± 151.1	35.0 ± 4.5
C	218 ± 76	52.0 ± 28.4	35.3 ± 4.7
D	187 ± 96	37.6 ± 37.2	25.1 ± 2.5
E	195 ± 76	38.5 ± 28.1	27.4 ± 2.8

was successfully tested in real experiments, using an industrial ToF camera. In our future work we plan to further improve detection accuracy by analyzing multiple frames together. Moreover, the computer vision system will be integrated on an automated guided vehicle, equipped with a robot manipulator for depalletizing tasks, and it will be evaluated in an operative industrial environment.

ACKNOWLEDGMENT

This research was supported by the COORSA collaborative project (European Regional Development Fund POR-FESR 2014-2020, Research and Innovation of the Region Emilia-Romagna, CUP E81F18000300009). The authors would like to thank Elettric80 SpA. The authors would also like to thank all the partners of the COORSA project.

REFERENCES

[1] M. Kavoussanos and A. Pouliezos, "Visionary automation of sack handling and emptying," *IEEE Robotics Automation Magazine*, vol. 7, no. 4, pp. 44–49, Dec 2000.

[2] W. Echelmeyer, A. Kirchheim, and E. Wellbrock, "Robotics-logistics: Challenges for automation of logistic processes," in *2008 IEEE International Conference on Automation and Logistics*, Sep. 2008, pp. 2099–2103.

[3] M. Hashimoto and K. Sumi, "Genetic labeling and its application to depalletizing robot vision," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, Dec 1994, pp. 177–186.

[4] C. Prasse, S. Skibinski, F. Weichert, J. Stenzel, H. Müller, and M. ten Hompel, "Concept of automated load detection for de-palletizing using depth images and RFID data," in *IEEE International Conference on Control System, Computing and Engineering*, Nov 2011, pp. 249–254.

[5] C. Prasse, J. Stenzel, B. Rudak, F. Weichert, H. Müller, and M. ten Hompel, "Low cost contour check of loading units using PMD sensors," in *Seventh International Conference on Sensing Technology (ICST)*, Dec 2013, pp. 477–482.

[6] C. Prasse, J. Stenzel, A. Böckenkamp, B. Rudak, K. Lorenz, F. Weichert, H. Müller, and M. ten Hompel, *New Approaches for Singularization in Logistic Applications Using Low Cost 3D Sensors*. Cham: Springer International Publishing, 2015, pp. 191–215.

[7] D. K. Katsoulas and D. I. Kosmopoulos, "An efficient depalletizing system based on 2D range imagery," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, May 2001, pp. 305–312.

[8] D. Katsoulas, L. Bergen, and L. Tassakos, "A versatile depalletizer of boxes based on range imagery," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, May 2002, pp. 4313–4319.

[9] D. Katsoulas, C. C. Bastidas, and D. Kosmopoulos, "Superquadric segmentation in range images via fusion of region and boundary information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 781–795, May 2008.

[10] B. Zhang and S. B. Skaar, "Robotic de-palletizing using uncalibrated vision and 3D laser-assisted image analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3820–3825.

[11] R. T. Yunardi, Winarno, and Pujiyanto, "Contour-based object detection in automatic sorting system for a parcel boxes," in *International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*, Oct 2015, pp. 38–41.

[12] 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," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2016, pp. 262–267.

[13] H. Nakamoto, H. Eto, T. Sonoura, J. Tanaka, and A. Ogawa, "High-speed and compact depalletizing robot capable of handling packages stacked complicatedly," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 344–349.

[14] A. Johnson. (2010-2014) The Clipper library. [Online]. Available: http://www.angusj.com/delphi/clipper.php

[15] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "OpenGA, a C++ genetic algorithm library," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2051–2056.