A Combinatorial Approach to Detection of Box Pallet Layouts

A Combinatorial Approach to Detection of Box Pallet Layouts

Ernesto Fontana*, Dario Lodi Rizzini*† and Stefano Caselli*†
* Robotics and Intelligent Machines Laboratory (RIMLab)
Department of Engineering and Architecture
† Centro Intradipartimentale per l'Energia e l'Ambiente (CIDEA)
University of Parma, 43124, Parma, Italy
{ernesto.fontana, dario.lodirizzini, stefano.caselli}@unipr.it

***Abstract*—This paper presents an algorithm for detection and pose estimation of cardboard parcel boxes in depth images based on combinatorial enumeration. It is designed for sensor-driven manipulation of pallets consisting of stacked planar layers chiefly using 3D range measurements. The proposed method initially detects the planar top layer of the pallet and its polygonal contour, possibly containing holes. Then, it enumerates the hypotheses about the layout of the pallet layer and estimates the best matching configuration. Experiments on a real dataset assess the feasibility of the proposed approach.**

***Index Terms*—depth camera, object detection, pose estimation, industrial application**

I. Introduction

The end-line management of product packages is an important part of industrial logistics. Packages are usually arranged into stacks of pallet layers for storage and shipping to the desired facilities. The operation of sorting packages into pallets is called palletizing. The inverse operation is named depalletizing and is an essential task for the customization of the pallet format according to customer requirements. In order to ensure stability, pallets are generally arranged as a stack of layers and both palletizing and depalletizing are performed by inserting or removing the parcel boxes from the top layer. Automated operations on pallets are crucial for efficient management of warehouses [1] that still rely on sensor-less manipulation with part feeding mechanisms, assumptions about object poses and constrained motion. However, the ongoing trend in industrial logistics aims at increasing flexibility and adaptability, with the goal of being able to manage different pallet and package formats. The palletizing station is a new warehouse component consisting of one or more robot manipulators and a proper perception system to re-arrange an input pallet with unknown layout into a new pallet format. Sensor-driven manipulation is required to automatically adapt to production demands, without explicit re-programming, and allowing considerable cost savings when similar applications are required for different manipulators.

Computer vision and sensor technology have been used for long time in automation (e.g. in quality control inspection), but their extensive use for end-line manipulation and logistics is more recent. In particular, depth cameras enable reliable 3D geometric representation of the scene in the form of point cloud data. Such sensors are based on different technologies such as time-of-flight (ToF), structured light or stereo vision. Range data are at times matched with RGB or grey color, but the latter are not always available nor reliable. For example, depth cameras such as *IFM o3d3xx*, that are designed for industrial applications, acquire both robust range measurements and intensity images. The raw intensity image only provides a complementary representation of the scene in the form of a low resolution grey-scale image. Thus, detection and pose estimation algorithms should rather focus on the point cloud.

In this paper, we present an algorithm for detection and pose estimation of product boxes in depth images based on combinatorial assessment of pallet layout. The method is designed for a palletizing station that recognizes the configuration of pallet layers chiefly using the range measurements of the cameras. It assumes parcel boxes belong to a limited set of formats with known dimensions and that the boxes' top sides lie on the same plane. The proposed approach initially detects the co-planar points lying on the top pallet layer and finds the polygonal contours enclosing the boxes. Since the boxes of a layer are organized according to planar patterns, the algorithm selects the points belonging to the top plane and finds the polygonal curves, possibly with inner "holes", that contain each connected component in 2D space. A crust technique [2] is exploited for robust detection of the polygons from noisy points with slight displacement w.r.t. the plane. The main original contribution of this paper lies in the combinatorial algorithm to infer the layout of the pallet layer. The proposed procedure is essentially a geometric bin packing solver that iterates through the different possible box configuration hypotheses by placing them inside the layer area. This means that each hypothesis corresponds to a box layout configuration and is generated from an orthogonal vertex and expanded by adding stacked boxes on the existing ones. The estimated pallet layout corresponds to the best coverage found, and this output can be enforced and refined through registration. Preliminary experiments on a real dataset assess the feasibility of the proposed approach.

This paper is organized as follows. Section II presents relevant literature on object detection for industrial applications.

978-1-6654-0976-6/21/$31.00 ©2021 IEEE

Section III presents the procedure for detection of pallets and of their contour, whereas section IV describes the algorithm for the estimation of the parcel box configuration. Finally, section V reports the results of the experiments.

II. Related Work

While scientific literature on object detection and pose estimation with depth cameras or computer vision is extensive, occasional works about perception-driven manipulation in manufactory are often strictly coupled with a given application or too generic. An early work [3] presents a genetic algorithm to detect and label parcel boxes arranged in multi-layer pallets using a gray scale camera. Prasse et al. [4]–[6] illustrate a method for pallet scene understanding using a ToF sensor and RFIDs. The work in [7] addresses the product unloading problem, comparing the effectiveness of different range sensors. More recently, some model-based methods have been proposed for depalletization using depth sensors with or without color data. The technique proposed in [8]–[10] combines box contour detection and general pallet layout detection based on genetic optimization. Deep learning algorithms have demonstrated their effectiveness in recognition of objects with different shapes, notably in the Amazon Picking Challenge 2017 [11]. Their success often depends on careful preliminary training, which is done on specifically acquired and labeled dataset. Moreover, manipulation requires the estimation of object poses and is generally solved by combining detection based on convolutional networks and geometric registration [12], [13]. Research projects like ILIAD [14] investigated more structured procedures for product manipulation driven by deep learning in industrial logistics, such as Object-RPE [15]. Several of the mentioned algorithms require both range measurements and images, which in many industrial settings are not available or do not guarantee sufficiently robust detection.

Box layout estimation in pallet displays similarities with 2D rectangle packing [16], [17] and, more generally, of rectangular knapsack packing problem. Such problem is $\mathcal{NP}$-hard and several deterministic and non-deterministic approaches [18]–[20] have been proposed for industrial applications. Other works make use of genetic algorithms [21], often with heuristics that address the large computational complexity. Finally, learning-based techniques such as [22] have recently appeared, some of which also have the goal of solving the 3D formulation of the problem [23], [24]. However, bin packing problem and pallet layout estimation are different in input data and goal. Bin packing problem operates on extact geometric description of the region to be filled in order to maximizes, whereas layout recognition searches the best arrangement of the given items into the available noisy data without hard geometric constraints.

III. Contour Detection

The first part of the algorithm presented in this paper deals with the detection of the upper contour for the highest layer of boxes standing on the pallet. Such contour is then used as input for the box layout estimator. The contour detection procedure incrementally filters the points that do not belong to the contours of the boxes from the initial point cloud. The geometric shape of the pallet layer contour, which is the output of this procedure, allows the geometric reasoning presented in section IV.

First, the contour detection algorithm extracts the dominant plane of the scene, which corresponds to the upper side of the top layer. Among the different possible approaches to this task, we chose the well-known consensus method RANSAC. The coordinates of the points are changed to a new reference frame where aces are consistent with the plane tangent to the pallet top layer. This frame is s.t. axis $\hat{\mathbf{z}}$ is oriented like the normal of the dominant plane, whereas axes $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ lie on it. The points belonging to the plane can be then easily found using passthrough filtering on the point coordinate $p_{i,z}$, with a given tolerance on the coordinate value.

After filtering out all the points that do not fit into the set gap, the remaining point cloud contains only the points belonging to the boxes top side of the highest layer. From this point on, the algorithms focuses on the detection of the planar pattern of the boxes and operates only on their x and y coordinates. Their z coordinate is close to zero and negligible. This algorithmic choice follows from assumptions derived from the industrial context and reduces the complexity of a fully 3D problem. Next, the goal is the extraction of the border points of the planar cloud representing the layer. The first performed operation is the removal of points with complete neighborhood, i.e. points surrounded by other points belonging to the pallet layer. We use the 8-neighborhood in the depth image or, equivalently, the organized point cloud, which is faster to visit using row and column indices. The algorithm marks as border points those with at least one neighbor not on the layer.

At this point of the processing pipeline, two problems must be addressed to estimate the planar contour of the pallet layer. The first one straightforwardly derives from the aim to find one or more closed polygonal curves: the algorithm should sort the points s.t. each one has both a predecessor and a successor. Such order cannot be directly derived from the indices of input depth image, since the pallet top layer plane and the camera plane do not correspond. Hence, neighbor pixels in the image may corresponds to different ranges and parts of the objects. As a result, the border detected with index reasoning leads to overlap of different chains around the layer. Thus, correct contour estimation requires to address a 2D curve reconstruction problem. The second problem stems from the presence of noisy and incomplete measurements. Cameras like IFM O3D303 acquire about 2% invalid measurements that must be filtered out to avoid detection of false contour points.

The initial approximate order is obtained from the topolog-

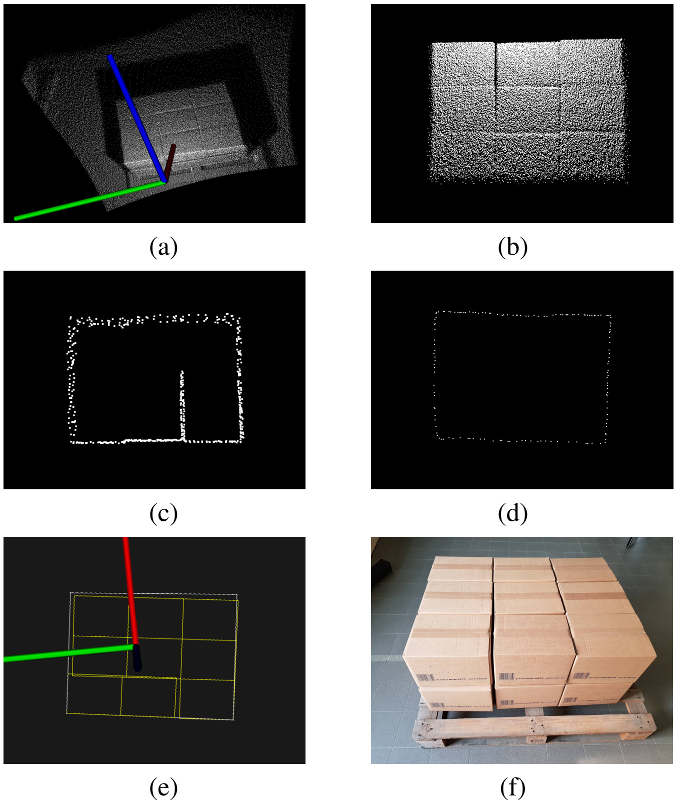


Fig. 1. Main steps of the algorithm: (a) the input point cloud representing the pallet, (b) the points corresponding to the top layer, (c) the noisy border points with unsorted measurements overlapping on neighbor pixels, (d) the cleaned contour points, (e) the contour and box layout configuration estimated respectively by the crust algorithm and the proposed combinatorial algorithm, (f) the pallet layer to be detected.

Algorithm 1 Find the best Layer Box Layout

function FindBoxLayout($\mathcal{C}$, $\{(w_f, h_f)\}_{f=1\ldots n_f}$)
    // Places boxes on orthogonal vertices of contour
    $\mathcal{Q} \leftarrow \emptyset$;
    for $\mathbf{v}_i \in OrthogonalCorners(\mathcal{C})$ **do**
        for $f \leftarrow 1 \ldots n_f$ **do**
            $B_{init} \leftarrow InitVertexBox(\mathcal{C}, \mathbf{v}_i, w_f, h_f, true)$
            if $B_{init} \neq null$ **then**
                $\mathcal{H}_{init} \leftarrow \{B_{init}\}$
                $score(\mathcal{H}_{init}) \leftarrow Area(B_{init} \cap \mathcal{C})$
                $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathcal{H}_{init})$;
            end if
            $B_{init} \leftarrow InitVertexBox(\mathcal{C}, \mathbf{v}_i, w_f, h_f, false)$
            if $B_{init} \neq null$ **then**
                $\mathcal{H}_{init} \leftarrow \{B_{init}\}$
                $score(\mathcal{H}_{init}) \leftarrow Area(B_{init} \cap \mathcal{C})$
                $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathcal{H}_{init})$;
            end if
        end for
    end for
    // Visits and expands the hypotheses in the queue
    while not $empty(\mathcal{Q})$ **do**
        $\mathcal{H}_{cur} \leftarrow pop(\mathcal{Q})$
        // Finds all the valid boxes expanding $\mathcal{H}_{cur}$
        $\mathcal{B} \leftarrow \emptyset$
        for $B_{in} \in \mathcal{H}_{cur}$ and all box formats (w_f, h_f) **do**
            $\mathcal{B} \leftarrow \mathcal{B} \cup FindStackedBoxes(B_{in}, \mathcal{C}, w_f, h_f)$
        end for
        remove all $B_{new} \in \mathcal{B}$ s.t.
            $Area(B_{new} \cap \cup_{B_j \in \mathcal{H}_{cur}}) > A_{max}$
        // Generates new hypotheses
        for $B_{new} \in \mathcal{B}$ **do**
            $\mathcal{H}_{new} \leftarrow \mathcal{H}_{cur} \cup \{B_{new}\}$
            $score(\mathcal{H}_{new}) \leftarrow score(\mathcal{H}_{cur}) + Area(B_{new} \cap \mathcal{C})$
            $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathcal{H}_{new})$;
            if $score(\mathcal{H}_{new}) > score(\mathcal{H}_{max})$ **then**
                $\mathcal{H}_{max} \leftarrow \mathcal{H}_{new}$
            end if
        end for
    end while
    return $\mathcal{H}_{max}$
end function

ical algorithm [25] implemented by function *findContours()* of library *OpenCV*. The function can only be used to find connected components and hierarchy due to the problem with the order described before. The contour reconstruction is performed using a *curve crust method* [26]. First, the Delaunay triangulation is computed from the candidate contour points. The geometric graph corresponding to the triangulation is pruned by removing the edges whose length is greater than threshold d_{max}. Second, the algorithm lists the points on the convex hull in order to provide anchor vertices. Finally, the anchor vertices are connected by searching minimum paths on the graph. The joined paths are used to obtain polygonal curve rings which correspond to the contour curves. In industrial practice, some layouts of pallet layers have inner holes in order to increase the friction between boxes of consecutive layers. Thus, the layer contours may be polygons with genus greater than 0, i.e. polygons with holes delimited by inner rings inside an outer ring. A graphical representation of the results obtained at different steps of the proposed algorithm is in Figure 1.

IV. Box Layout Estimation

The proposed algorithm estimates the layout of the pallet layer starting from its polygonal contour $\mathcal{C}$, which is computed according to the procedure presented in section III. The polygonal contour may include holes deriving from some filling pattern recurrent in industrial practice. In the infrequent case of unconnected parcel boxes, our algorithm can be applied separately to each contour.

The combinatorial procedure visits all the box coverage hypotheses. A box is represented by a tuple $B = (\mathbf{c}, \theta, w, h, \mathbf{d})$, where $\mathbf{c}$ and θ are respectively the origin and the orientation angle of the box reference frame, w and h ($w > h$) the width and height of the box, and $\mathbf{d}$ is the pivot direction. The reference frame is placed on the centroid of the box, with the axis x aligned with the longest edge. The frame coordinates

Algorithm 2 Try to place a box on vertex

```
1: function InitVertexBox(𝒞, v_i, w, h, onPrev)
2:     e_p ← (v_{i−1} − v_i)/‖v_{i−1} − v_i‖, e_n ← (v_{i+1} − v_i)/‖v_{i+1} − v_i‖
3:     if not onPrev then
4:         swap e_p and e_n
5:     end if
6:     c ← v_i + e_p w/2 + e_n h/2
7:     θ ← direction(e_p)
8:     d ← [sign(c_x − v_{i,x}), sign(c_y − v_{i,y})]^⊤
9:     B ← Box(c, θ, w, h, d)
10:    if Area(B ∩ 𝒞) > A_min then
11:        return B
12:    else
13:        return ∅
14:    end if
15: end function
```

are referred to a 2D frame in the pallet layer plane. The pivot direction vector $\mathbf{d}$ points the direction of expansion of new stacked boxes. The direction $\mathbf{d}$ is opposite to the original vertex from which the configuration hypothesis originated (see the following discussion of hypothesis initialization).

A box layout hypothesis (called with symbol $\mathcal{H}$. and proper pedix) consists of a set of placed boxes. Every hypothesis has an associated score equal to the sum of the intersection areas of each box with the layer surface, i.e.

$$score(\mathcal{H}) = \sum_{B \in \mathcal{H}} Area(B \cap \mathcal{C}) \quad (1)$$

The initial hypotheses are obtained by placing valid boxes on each convex orthogonal vertex of contour $\mathcal{C}$. A new child hypothesis $\mathcal{H}_{child}$ is obtained by adding a new valid box to a parent hypothesis $\mathcal{H}_{par}$. A new box B_{new} is valid if two conditions hold:

1) B_{new} lies on the layer, i.e. the area of its intersection with the contour is greater than a threshold A_{min}

$$Area(B_{new} \cap \mathcal{C}) > A_{min} \quad (2)$$

2) B_{new} does not conflict with the previously placed boxes, i.e. the area of its intersection with the previously placed boxes is less than a threshold value

$$Area(B_{new} \cap \cup_{B_i \in \mathcal{H}_{par}} B_i) < A_{max} \quad (3)$$

The tolerance parameters A_{min} and A_{max} are required, since the contour is detected from noisy data. Their value is proportional to the area of the box (in our experiments $A_{min} = 0.9\ Area(B)$ and $A_{max} = 0.1\ Area(B)$).

Algorithm 1 illustrates the procedure for the layout estimation. The input data consist of the contour $\mathcal{C}$ and the dimensions of the n_f box formats of the pallets, i.e. width w_f and height h_f. In most cases there is only a single product type $n_f = 1$, but the procedure can handle different formats. The hypothesis queue $\mathcal{Q}$ is the main data structure that stores the hypotheses to be expanded. Lines 4-19 provide the initialization of the hypotheses. The new boxes are placed

Algorithm 3 Expand the given layout hypothesis $\mathcal{H}$ by adding stacked boxes

function FindStackedBox(B_{in}, $\mathcal{C}$, w, h)
  $\mathcal{B} \leftarrow \emptyset$
  // Tries to stack a new box on edges of B_{in}
  B_{in} parameters: $(\mathbf{c}_{in}, \theta_{in}, w_{in}, h_{in}, \mathbf{d}_{in})$
  $\mathbf{c}^{(1)}_{new} \leftarrow \left[d_{in,x}\ \frac{w_{in}-w}{2}, -d_{in,y}\ \frac{h_{in}+h}{2}\right]$, $\theta^1_{new} \leftarrow 0$
  $\mathbf{c}^{(2)}_{new} \leftarrow \left[d_{in,x}\ \frac{w_{in}-h}{2}, -d_{in,y}\ \frac{h_{in}+w}{2}\right]$, $\theta^2_{new} \leftarrow \frac{\pi}{2}$
  $\mathbf{c}^{(3)}_{new} \leftarrow \left[-d_{in,x}\ \frac{w_{in}+w}{2}, d_{in,y}\ \frac{h_{in}-h}{2}\right]$, $\theta^3_{new} \leftarrow 0$
  $\mathbf{c}^{(4)}_{new} \leftarrow \left[-d_{in,x}\ \frac{w_{in}+h}{2}, d_{in,y}\ \frac{h_{in}-w}{2}\right]$, $\theta^4_{new} \leftarrow \frac{\pi}{2}$
  for $i = 1 \ldots 4$ **do**
    if $i < 2$ **then** $\mathbf{d}_{new} = [-1, 1]$
    else $\mathbf{d}_{new} = [1, -1]$ **end if**
    $\mathbf{c}_{new} \leftarrow \mathbf{R}(\theta_{in})\mathbf{c}^{(i)}_{new} + \mathbf{c}_{in}$, $\theta_{new} \leftarrow \theta^{(i)}_{new} + \theta_{in}$
    $B_{new} \leftarrow Box(\mathbf{c}_{new}, \theta_{new}, w, h, \mathbf{d}_{new})$
    if $Area(B_{new} \cap \mathcal{C}) > A_{min}$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{B_{new}\}$
    end if
  end for
  return $\mathcal{B}$
end function

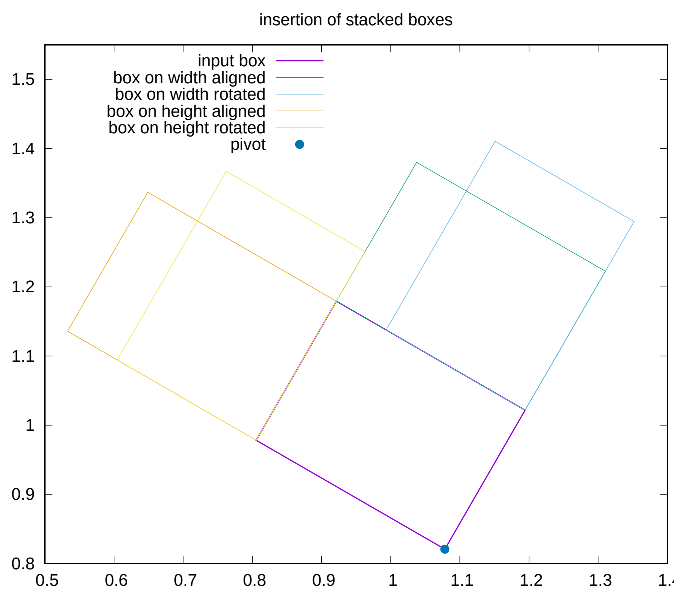


Fig. 2. Examples of insertion of new boxes. The new boxes are stacked on the width or on height edges opposite to pivot point of input box, either aligned or rotated by 90 deg.

on the orthogonal vertices $\mathbf{v}_i$ of the contour either aligning their longest edge with the previous $\mathbf{v}_{i-1}\mathbf{v}_i$ or with the next $\mathbf{v}_i\mathbf{v}_{i+1}$ edge. The box initialization on a vertex is described by Algorithm 2. After computing the box parameters (poses and dimensions), the validity of the candidate box is tested using the criterion of eq. (2) at line 10. After calling function *InitVertexBox()*, it creates and scores a new hypothesis $\mathcal{H}_{init}$, with a single valid box, and inserts it in the queue $\mathcal{Q}$.

The queue expansion is illustrated at lines 21-39 of Algorithm 1. The procedure iteratively extracts an hypothesis $\mathcal{H}_{cur}$

and tries to expand it, starting from the boxes $B_{in} \in \mathcal{H}_{cur}$. Algorithm 3 implements the addition of new boxes by stacking them on each B_{in}. Figure 2 illustrates with an example the four ways (lines 5-8) to stack a new box on a given box. The candidate stacked boxes are tested w.r.t. criteria of eq. (2) in *FindStackedBox()* (line 14) and of eq. (3) in *FindBoxLayout()* (line 28). At lines 31-38 of Algorithm 1, the new hypotheses derived from the added valid boxes are scored and added to the queue. After this insertion, the algorithm checks if the new hypotheses achieve a better score and, in such case, it keeps track of the best hypothesis $\mathcal{H}_{max}$. When the queue is empty, and the exploration of hypotheses is finished, the algorithm returns the best box layout.

The proposed algorithm is based only on tessellation of the layer region and on geometric scoring of the configuration hypotheses. A future improvement could take into account the edges obtained from the intensity image that sensors like IFM O3D303 provide with the depth image. While the raw intensity image is not very accurate, it could help in order to disambiguate between symmetric hypotheses, e.g. scoring the hypotheses according to their matches with the extracted edges. Our method efficiently estimates the layouts occurring in most real industrial cases. For more complex arrangements, optimizations to limit computational complexity when dealing with the most recurrent pallet layouts could be made: an example of this would be, when dealing with pallet layouts that are evaluated as highly symmetric and populated, to run the algorithm only along the edges of the extracted contour, and then recur to dynamic programming for iterating the results obtained on the edges across the interior of the layer.

V. Experiments

The experimental evaluation of the box detection algorithm has been performed in order to verify accuracy, precision, and also computational cost of the box layout detection algorithm presented in the previous sections. The dataset consists of different box pallet layouts that simulate common configurations used in industrial practice. A single box format having upper face size of $315 \times 232\ mm$ has been used to compose all configurations. Said configurations consist of two pallet layers, although we focus only on the top layer. The configurations include both complete and partially complete layouts, as it was intended to simulate the state of the pallet at different stages of the depalletization process, which generally happens by removing one box after the other through a robotic device which picks and places them. The position of each box has been measured using an OptiTrack motion capture system, by placing four markers on the vertices of a box, as described in [8]. Given the centroids of the boxes in OptiTrack reference frame, we have used their pairwise distances as ground truth values to assess the accuracy of the proposed detection algorithm.

Figure 3 shows the pictures of the box layouts and the configurations estimated using the proposed method. The combinatorial layout detection algorithm has correctly estimated most of the configurations with a notable exception of layout C. However, the computed configuration for layout C is compatible with the given contour due to symmetry between the real and estimated box placements. The disambiguation for such cases could be obtained either by improving accuracy of contour detection or by extracting the edges in raw intensity image acquired by the camera IFM O3D303. While the premise of this work is the limited reliability of edges detection and standard computer vision with these type of sensors, edge information could complement the geometric score adopted in section IV.

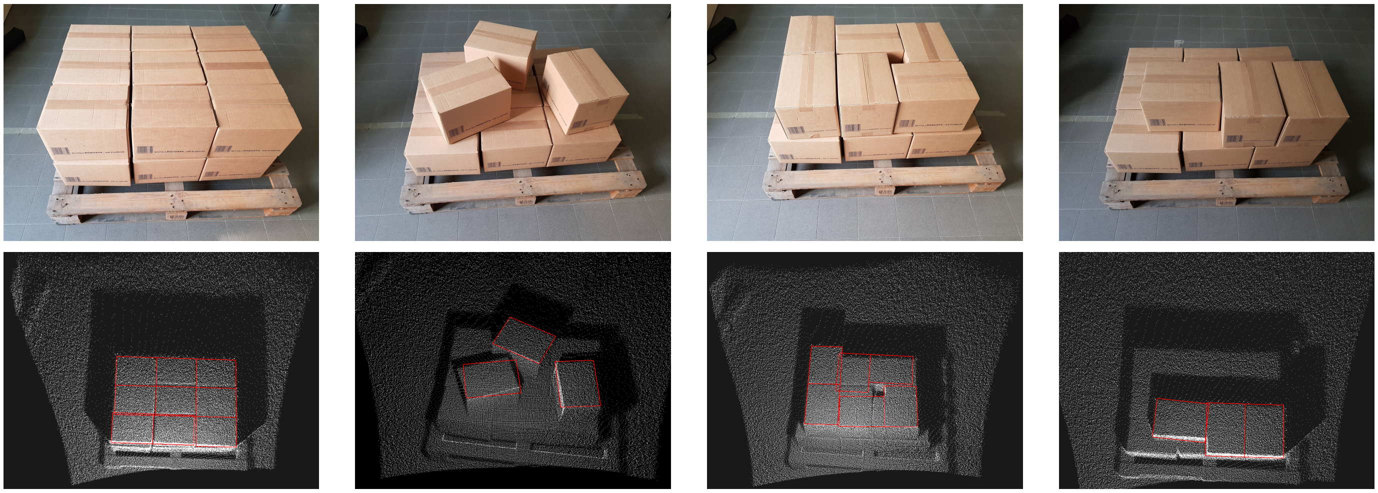
TABLE I
Position accuracy of the box detection algorithm with respect to ground truth

config	num_boxes	num_clusters	avg_pw_error [mm]	avg_stdev [mm]
A	9	1	13.86	6.2
B	3	3	13.46	5.2
C	6	1	21.52	20.1
D	3	1	11.34	4.4

The metric accuracy has been assessed by matching and comparing the pairwise distances between the centroids of the detected boxes and the ground truth distances. The association has been computed by comparing the pairwise distances of the boxes and then calculating the mean error across all measurements for every single configuration. In the case of configuration C, the matching is partial due to the symmetry of the estimated configuration. This test measures the accuracy metric, although it does not directly capture the correctness of the estimated box layout. Table I presents the results. This *centroids pairwise distances* metric is a useful test for establishing the accuracy of the working algorithm. However, pallet layout detection would need additional testing on other characteristics of the pallet layer (such as the area overlap of boxes) in order to better establish the validity of the resulting configuration. The average distance error is about 12 mm, except for configuration C that doubles the error, likely for the discussed mismatch. The estimated accuracy is adequate for most industrial manipulation tasks and could be further refined through registration.

VI. Conclusion

This paper has presented a new algorithm for detection, layout inference and pose estimation of cardboard parcel boxes on the top layer of a pallet. The core contribution is the procedure for the enumeration and estimation of the layout of boxes, as feasibility of the proposed approach has been assessed. The system was successfully tested on real datasets acquired using a depth camera, albeit with possibility of ambiguous outcomes when used on specific configurations. In our future work we plan to improve the algorithm by scoring the box layout hypotheses according to their match with the raw image, and by refining the final result with registration. We expect to run additional tests on heterogeneous configurations with larger numbers of boxes. Finally, an improved version of the algorithm will be integrated in a palletizing station as part of an industrial plant.



A B C D

Fig. 3. Pictures of four box configurations (from A to D) used in the experiments (top row) and the corresponding estimated layouts overlapped with their point cloud (bottom row).

ACKNOWLEDGMENT

This work is supported by OCME S.r.l. and project "MAF Macchine Autonome & Flessibili", MISE_FCS_DM 5 marzo 2018.

REFERENCES

[1] 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.

[2] T. Dey, K. Mehlhorn, and E. Ramos, "Curve reconstruction: Connecting dots with good reason," *Computational Geometry*, vol. 15, no. 4, pp. 229–244, 2000.

[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, and M. Müller, H. andTen 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, M. 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] T. Stoyanov, R. Mojtahedzadeh, H. Andreasson, and A. Lilienthal, "Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1094 – 1105, 2013.

[8] R. Monica, J. Aleotti, and D. Lodi Rizzini, "Detection of Parcel Boxes for Pallet Unloading Using a 3D Time-of-Flight Industrial Sensor," in *IEEE Int. Conf. on on Robotic Computing (IRC)*, Nov. 2020, pp. 314–318, iSBN: , DOI.

[9] D. Chiaravalli, G. Palli, R. Monica, J. Aleotti, and D. Lodi Rizzini, "Integration of a Multi-Camera Vision System and Admittance Control for Robotic Industrial Depalletizing," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sept. 2020, pp. 667–674, iSBN: 978-1-7281-8956-7, DOI 10.1109/ETFA46521.2020.9212020.

[10] J. Aleotti, A. Baldassarri, M. Bonfè, M. Carricato, D. Chiaravalli, R. Di Leva, C. Fantuzzi, S. Farsoni, G. Innero, D. Lodi Rizzini, C. Melchiorri, R. Monica, G. Palli, J. Rizzi, L. Sabattini, G. Sampietro, and F. Zaccaria, "Toward future automatic warehouses: An autonomous depalletizing system based on mobile manipulation and 3d perception," *Applied Sciences*, vol. 11, no. 13, 2021.

[11] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2017.

[12] Y. Xiang, S. Tanner, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[13] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep Iterative Matching for 6D Pose Estimation," *Int. Journal of Computer Vision*, vol. 128, pp. 657–678, 2020.

[14] T. Stoyanov and M. Magnusson, "Safe feature-based navigation for industrial agvs," in *ILIAD (Intra-Logistics with Integrated Automatic Deployment: Safe and Scalable Fleets in Shared Spaces) Deliverable 6.4: Perception system for detecting boxes and wrapping*, 2018, pp. 1–29.

[15] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Object-RPE: Dense 3d reconstruction and pose estimation with convolutional neural networks," *Robotics and Autonomous Systems*, p. 103632, 2020.

[16] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European journal of operational research*, vol. 141, no. 2, pp. 241–252, 2002.

[17] A. Lodi, S. Martello, and D. Vigo, "Recent advances on two-dimensional bin packing problems," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 379–396, 2002.

[18] Y.-L. Wu, W. Huang, S.-C. Lau, C. Wong, and G. H. Young, "An effective quasi-human based heuristic for solving the rectangle packing problem," *European Journal of Operational Research*, vol. 141, no. 2, pp. 341–358, 2002.

[19] W. Huang, D. Chen, and R. Xu, "A new heuristic algorithm for rectangle packing," *Computers & Operations Research*, vol. 34, no. 11, pp. 3270–3280, 2007.

[20] A. Bortfeldt, "A reduction approach for solving the rectangle packing area minimization problem," *European Journal of Operational Research*, vol. 224, no. 3, pp. 486–496, 2013.

[21] ——, "A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces," *European Journal of Operational Research*, vol. 172, no. 3, pp. 814–837, 2006.

[22] K. Sim, E. Hart, and B. Paechter, "A lifelong learning hyper-heuristic method for bin packing," *Evolutionary Computation*, vol. 23, no. 1, pp. 37–67, 2015.

[23] H. Hu, X. Zhang, X. Yan, L. Wang, and Y. Xu, "Solving a new 3d bin packing problem with deep reinforcement learning method," *arXiv preprint arXiv:1708.05930*, 2017.

[24] L. Duan, H. Hu, Y. Qian, Y. Gong, X. Zhang, Y. Xu, and J. Wei, "A multi-task selected learning approach for solving 3d flexible bin packing problem," *arXiv preprint arXiv:1804.06896*, 2018.

[25] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, 1985.

[26] N. Amenta, M. Bern, and D. Eppstein, "The crust and the β-skeleton: Combinatorial curve reconstruction," *Graphical models and image processing*, vol. 60, no. 2, pp. 125–135, 1998.