Towards Retail Stores Automation: 6-DOF Pose Estimation Combining Deep Learning Object Detection and Dense Depth Alignment

Virgile Foussereau, Iori Kumagai, Guillaume Caron

Towards Retail Stores Automation: 6-DOF Pose Estimation combining Deep Learning Object Detection and Dense Depth Alignment

Virgile Foussereau[1,2], Iori Kumagai[1], Guillaume Caron[1,3]

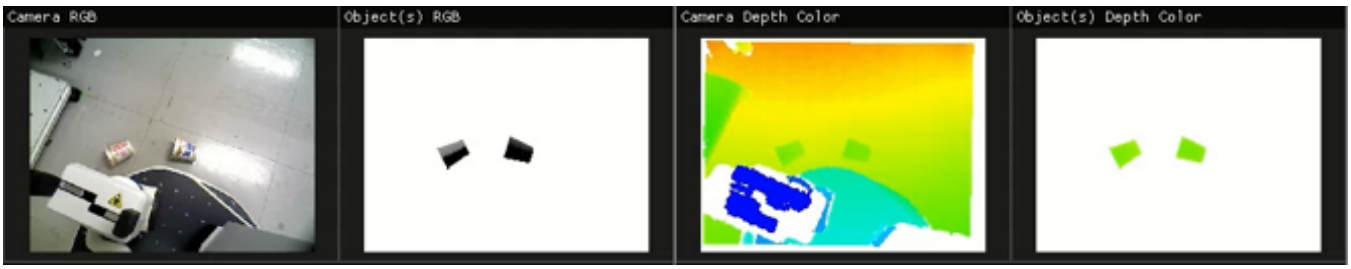


Fig. 1: Detection and pose estimation of two fallen objects on the ground. 3D models of the objects are placed on their estimated poses to compare the resulting RGB and depth images with the ones captured by the robot's camera.

***Abstract*— Automating in-store logistics processes in the retail industry poses significant challenges for robot manipulators. Contrary to warehouses, retails stores are subject to customer actions, which can imply non-standard tidying of products. This paper addresses the problem of detecting, discriminating, and accurately estimating the 6 degrees-of-freedom (6-DOF) pose of individual products, even in unexpected positions such as fallen or wrongly placed objects. The trained object detection model successfully discriminated similar-shaped objects of different brands/types commonly found in convenience stores. The detection is used to initialized the object position while several possible orientations are explored by a Fibonacci Multi-Start method. The estimated pose is then refined by a multi-scale projective Iterative Closest Point (ICP). The evaluation of the complete 6-DOF pose estimation module revealed its consistent ability to converge to the correct pose, avoiding local optima and achieving sub-millimetric precision. A working demonstration is presented, showcasing a robot rearranging a convenience store shelf. The overall system demonstrated the ability to detect fallen objects, estimate their poses, determine suitable grasping directions, and execute successful grasps. Importantly, the system's feasibility with minimal human intervention was demonstrated, allowing easy addition of new objects by convenience store employees or other stakeholders.**

I. INTRODUCTION

The utilization of robot manipulators is increasingly expanding within the retail industry, primarily for warehousing purposes [1]. However, automating in-store logistics processes using these manipulators remains a challenging endeavor. The automation of the commercial part of supermarkets not only requires to handle individual products, as opposed to boxes containing group of them, but also the capacity to cope with unexpected events caused by customers such as wrongly placed or fallen products. The challenges raised by the handling of individuals products have been exemplified during the different editions of the Amazon Picking Challenge [2]. Perception and object detection was listed as the most difficult task by the participant teams [2] and was the most common cause of failures even for the winning team [3]. The case of objects in fallen or unexpected positions is partially covered in the Future Convenience Store robotic challenge launched by the World Robotic Summit, in the Stock and Disposal Task [4]. However, participants use Augmented Reality (AR) markers for pose estimation [4] [5]. While this approach offers convenient and accurate pose estimation, it has some limitations. First, it necessitates repackaging all retail products. Second, AR markers must be present on every object face to account for potential customer-induced pose variations. Last, accurately marking non-flat surfaces could pose challenges.

Several methods can be used for 6 degrees-of-freedom (6-DOF) pose estimation of objects without markers. Using only 2D images is possible but with a precision range of 1-10 cm [6], so depth information is commonly integrated. The 2022 BOP Challenge winner [7] achieves top results on standard datasets but demands per-object network training, a time/resource-intensive process. It also necessitates challenging-to-acquire ground truth object poses for training. Other commonly used methods for pose estimation such as DenseFusion [8] and PoseCNN [9] were not considered as accurate enough by the european project REFILLS as they yield a successful pose localization if the value of the average closest point distance (ADD-S) metric is below 2 cm [10], which is not suitable for a precise grasping. To tackle this issue, [10] uses a two-step method starting with a Convolutional Neural Network (CNN) that gives an initial coarse estimation of the object's pose, which is then refined by a visual servoing module, achieving an

[1]CNRS-AIST JRL (Joint Robotics Laboratory) IRL, National Institute of Advanced Industrial Science and Technology (AIST), Japan.
[2]Ecole Polytechnique, Palaiseau, France.
[3]UPJV MIS Laboratory, Amiens, France.
Email: virgile.foussereau@polytechnique.org

impressive precision of 2 mm. However, the use of visual servoing has several drawbacks. As raised by the authors, this approach is suitable only for texture-rich objects. It also necessitates a desired grasp image for each possible grasping position. The authors worked with plain white background, but more textured ones may make the system less efficient. Finally, it requires an eye-in-hand camera and can only estimate the exact pose of one object at a time.

In this paper, we present a working demonstration of a robot rearranging a convenience store shelf using an object detection and pose estimation system with sub-millimetric precision. The perception system is able to detect objects in non-standard positions, such as objects that have fallen off the shelf. We show its capacity to detect, discriminate and estimate the pose of multiple similar objects in real-time, allowing efficient grasping and rearrangement. The overall detection and pose estimation module only requires a 3D model and a couple of pictures of the object on a plain background, from different angles. Neither depth information nor ground truth poses are required for training. Thus, a new object can be added with minimal human intervention, by a convenience store employee for example.

II. RELATED WORKS

A. YOLO architecture

The YOLO architecture is a one-stage detector based on a Convolutionnal Neural Network (CNN) [11]. In 2022, YOLOv7 achieved state-of-the-art performance and inference speed on the COCO dataset [12]. In 2023, YOLOv8 was released as the latest iteration of the YOLO family with improvements in terms of speed, accuracy and efficiency [13].

B. Multi-scale pyramidal projective ICP

The authors of [14] have developed an object tracking method based on a multi-scale pyramidal projective Iterative Closest Point (ICP). This method projects the 3D model of the object with the depth camera projection function $\mathbf{\Phi} : \mathbb{R}^3 \mapsto \mathbb{R}^3$. Unlike most previous approaches that undistort the captured depth image beforehand, this method directly performs the data association stage in the image plane with strong distortions. This difference prevents a decrease in resolution at the image center, where the targeted object is usually. The captured depth image is noted $\mathbf{D}^*$ while the image of the projected 3D model is noted $\mathbf{D}$. The method then runs an ICP algorithm [15] on three level of a multi-scale pyramids representation of the object projection and the captured depth image. The multi-scale pyramids for depth are noted $\mathbf{P_{D^*}}$ and $\mathbf{P_D}$ for the depth image and the projected object respectively. From these, the method computes multi-scale pyramids of the vertices $\mathbf{P_{V^*}}$ and $\mathbf{P_V}$, and the surface normals $\mathbf{P_{N^*}}$ and $\mathbf{P_N}$. The process runs iteratively from the coarsest to the finest pyramid level, using dense depth pixel-to-pixel correspondances $(\mathbf{u}, \hat{\mathbf{u}})$ with $\mathbf{u} \in \mathbb{R}^2$ a pixel from the predicted depth image $\mathbf{P_D}$ and $\hat{\mathbf{u}} = \mathbf{\Phi}(\mathbf{T}, \mathbf{P_V}(\mathbf{u}))$ its corresponding pixel in $\mathbf{P_{D^*}}$. As the captured depth image may suffer from depth-less pixels, only a subset Ω of pixels from the captured depth image is used. This subset contains only pixels with non-zero depth, under a certain distance $\boldsymbol{\delta_d} \in \mathbb{R}_+$ from the projected object and of similar orientation (threshold $\boldsymbol{\delta_\theta} \in [0, \pi]$. Then, the method minimizes the following energy based on point-plane metric [16]:

$$E(\mathbf{T}) = \sum_{u \in \Omega} \|(\mathbf{T}\mathbf{P_V}(\mathbf{u}) - \mathbf{P_{V^*}}(\hat{\mathbf{u}}))^\top \mathbf{P_{N^*}}(\hat{\mathbf{u}})\|_2. \quad (1)$$

The optimization process leverages the Cholesky decomposition [17].

III. APPROACH

In this section, we describe the overall approach, from the detection of an object to its manipulation by the robot. The robot uses a Red, Green, Blue - Depth (RGB-D) camera. A system overview is presented in figure 2.

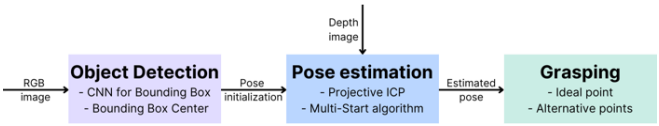


Fig. 2: System overview

A. Object Detection

Due to its performance in both accuracy and inference speed, we select the YOLOv8 architecture described in section II-A for our Object Detection module.

*1) **Synthetic dataset generation and training**:* To train the model for our targeted objects we need to create a dataset with annotations. While real images with manually made annotations are often the best way to achieve high accuracy, it can take a considerable time to create, even for a single object. In a supermarket or convenience store use-case with numerous different products, this method is therefore not practical. Consequently, we develop a method to generate a synthetic dataset from a reasonable number of pictures N_p, without any annotation.

We start by taking pictures of the targeted object on a uniform background, from different angles. From there, the dataset generation is fully automatic. First, the uniform background is removed automatically. Several random transformations are then applied to the resulting isolated object: rotation, translation, hue, saturation, lightness, contrast, brightness, blur. The transformed object is then pasted on a random background, selected among a dataset of sample images such as ImageNet for instance. Several random isolated objects are pasted as well before and after the targeted object is pasted on the background. Finally, some final transformations such as Gaussian blur are applied to the resulting image to help blend the objects with the background. Examples of synthetic training images are presented with labels in figure 3. The object is a Noodle Cup.

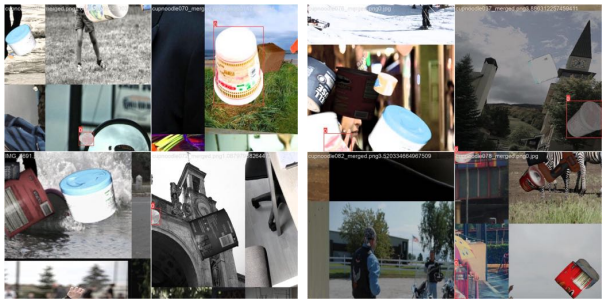


Fig. 3: Synthetic images for training (Noodle Cup)

As the targeted object is pasted on the background, its ground-truth bounding box is directly known, thus avoiding the need for manual annotation. This synthetic data generation method is therefore very efficient in terms of human intervention. However, it also raises several challenges. Indeed, there is a "gap" between synthetic data and real pictures. First, the isolated images of the object are all taken in the same conditions. This could lead to a trained model unable to recognize the object in other settings, for instance with a different light. This is where the several random transformations applied to the object prove their usefulness, avoiding the model overfitting. Another challenge is that an object pasted on a background will never achieves flawless integration with its surroundings. A neural network could then overfit to detect "pasted objects" and not the specific targeted object. This is why pasting other random objects is a very important step of the dataset generation, as it will force the network to use the distinguishing features of the object instead of relying on whether or not it seems pasted on the background.

The initial pictures used for the data generation can be taken with a studio setup and a rotation table for high resolution pictures but also in a minimal setup configuration. We tested our dataset generation with high quality pictures taken from [18] for one object, while for some other objects we used pictures taken with a standard smartphone on a basic black clothe. No significant difference was visible in the resulting performance of the model on our test cases. A studio setup with a rotation table can bring consistency and automation to the process if available, for example in large supermarkets or warehouses, while the minimal setup test shows that new objects could be added by employees without specific equipment, for instance in small convenience stores.

*2) **Pose initialization from bounding box**:* After an object has been detected, we derive a coarse pose estimation from the bounding box that will be used as initialization for the point cloud alignment phase described in section III-B. We take the center pixel of the bounding box and use the correspondence between the RGB image and the depth map to obtain its Z coordinate with respect to the camera frame. Then, the X and Y coordinates in the camera frame are determined from the camera intrinsic parameters and the pixel coordinate (x_{pixel}, y_{pixel}) using the following formulas:

$$\begin{cases} x = \frac{(x_{pixel} - c_x) * z}{f_x} \\ y = \frac{(y_{pixel} - c_y) * z}{f_y}, \end{cases} \tag{2}$$

with f_x, f_y being the focal lengths and (c_x, c_y) the principal point of the camera [19].

Using the center point of the bounding box for the position $\mathbf{P} = [X, Y, Z]^\top \in \mathbb{R}^3$ initalization seems intuitive, but it can also be justified mathematically. In addendum [20] we present a mathematical demonstration to show that given a convex shape S and its bounding box B, the center point is the only point guaranteed to be part of S.

After estimating the object's 3D position, an orientation $\theta\mathbf{w}$ is required to complete the pose, where $\theta \in [-\pi, \pi[$ and $\mathbf{w} \in \mathbb{R}^3$ with $\|\mathbf{w}\| = 1$. Initially, a rough guess is made using the bounding box height/width ratio, e.g., a standing bottle should have a ratio ¿ 1, while a fallen bottle should have a ratio ¡ 1. This serves as an initialization for the 6-DOF pose estimation module.

B. 6-DOF Pose Estimation

From the pose initialization made in the Object Detection module, we proceed with a point cloud alignement between the depth map captured by the camera, and a 3D model of the targeted object. The objective is to find the transformation matrix $\mathbf{T} \in SE(3)$ from the object frame to the camera frame. The initial $\mathbf{P}$ and $\theta\mathbf{w}$ can be combined in a single vector $\mathbf{r}$ in $se(3)$ and the exponential map from $se(3)$ to $SE(3)$ then gives the initial $\mathbf{T}$. Afterward, we use the multi-scale pyramidal projective ICP described in section II-B that we adapt for pose estimation. We defined a normalized error based on the energy from (1) that will be used to evaluate the convergence of the process:

$$e(\mathbf{T}) = \frac{\sqrt{E(\mathbf{T})}}{|\Omega|}. \tag{3}$$

While the translation initialization from the Object Detection module generally produces satisfactory results, the rotation initialization often deviates significantly from the actual orientation. This can lead to convergence to local optima or in some cases to a complete absence of convergence. In order to make the method more robust, we develop and compare two algorithms that are used if the resulting error e defined in (3) is higher than a pre-defined threshold Δ_e.

The first method is a Randomized Multi-Start algorithm [21]. In this method, orientations $\theta\mathbf{w}$ are randomly selected for the object and the projective ICP described above is re-run for each of them. The resulting pose $\hat{\mathbf{P}} \in \mathbb{R}^3$ with the lowest error e is kept as final estimated pose of the object. The number N of orientations sampled is a trade-off between exploration (higher chance of converging to the correct optimum) and computational complexity. This number can be chosen depending on the computational resources available and the application requirements.

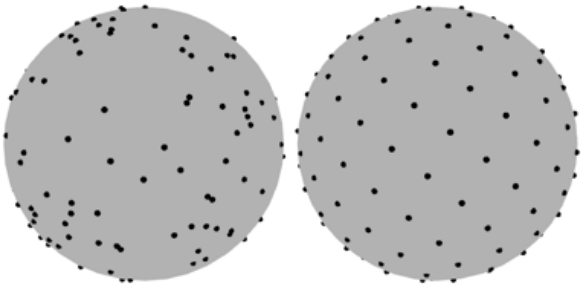


Fig. 4: Randomly selected points on a sphere (left) vs Spherical Fibonacci point set (right)

We refer to the second method as a Fibonacci Multi-Start algorithm. The process is similar to the Randomized Multi-Start but instead of using randomly selected orientations, we use a spherical Fibonacci lattice. Spherical Fibonacci point sets yield nearly uniform point distributions on the unit sphere as presented on figure 4. Furthermore, they are both simple and efficient to compute. We start from a Fibonacci lattice point set defined by:

$$\mathbf{t_i} = (\frac{i}{N}, \frac{i}{\varphi}) \quad \text{for} \;\; 0 \leq i \leq N, \tag{4}$$

where $\varphi = \dfrac{1+\sqrt{5}}{2}$ is the golden ratio.

Then, each point is mapped from the unit square $[0,1]^2$ to the sphere by the cylindrical equal area projection:

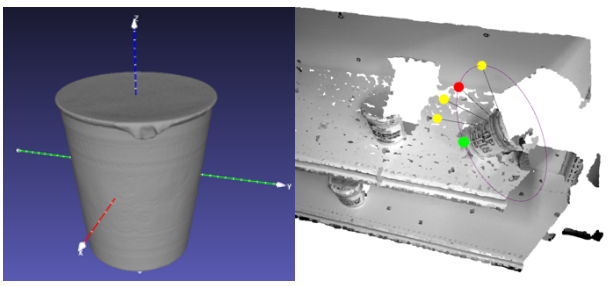
$$\begin{aligned} (x,y) \rightarrow (\theta,\phi): &\quad \left(\cos^{-1}(2x-1) - \pi/2, 2\pi y\right) \\ (\theta,\phi) \rightarrow (x,y,z): &\quad (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)\,. \end{aligned} \tag{5}$$

Each of the resulting points on the unit sphere corresponds to a directional vector $\mathbf{w}$ that is used for the object orientation. As the points are close to be evenly distributed on the sphere, it ensures maximum coverage of the possible orientations. Note that this is true for objects with an axis of symmetry, often found in retail stores. For objects with no symmetry, a final uniform rotation should be applied around $\mathbf{w}$ to define θ.

C. Grasping

After determining the object's pose, the robot can proceed with the grasping phase. We use a reachability graph-based planner from [22]. From an offline computed reachability graph containing feasible end-effector paths, this planner uses global graph search to find a feasible path between input start and goal. Therefore, we only need to provide a goal pose to the planner that will allow the robot to grasp the object. We consider the case of a 2-finger gripper as end-effector. Offline, we determine an "ideal" grasping approach based on the object shape and the gripper. To do this, we start by considering the three axes of the object frame. We select the axis along which the object is the biggest. In other words, it is the axis that accounts for the largest possible variance in the object points coordinates. If a Principal Components Analysis (PCA) was done on the object points coordinates, this axis would correspond to the first dimension selected. Therefore, we call this axis the object principal axis. We then consider its orthogonal plane and define approaching directions in this plane with $\theta \in [0, 2\pi]$. As the 2-finger gripper has a maximum extension, we define a subset $\Delta \subseteq [0, 2\pi]$ such that the object's width when approaching from this direction is smaller than the gripper extension. As it can be assumed that, at rest, the object is lying on a surface (e.g ground or shelf) due to the gravitational force, we prefer a vertical approach that will avoid collision with the supporting surface. Therefore, knowing the object pose in the base frame, we can select the angle $\theta_0 \in \Delta$ that represent the most vertical approach i.e. the vector most correlated with the gravitational acceleration $\vec{\mathbf{g}}$. This approach represents our "ideal" approach. In the particular case where the object principal axis is co-linear with $\vec{g} \in \mathbb{R}_3$ (meaning that all approach vectors defined by $\theta \in \Delta$ are horizontal in the base frame), we select $\theta_0 \in \Delta$ such that the resulting grasping approach is the most aligned with the robot to object direction.

For better understanding, we illustrate the process with an example in figure 5. The targeted object is a Noodle cup. The principal axis is found to be the z-axis of the object, represented in blue on figure 5a. It is also a symmetry axis for the object, therefore we have $\Delta = [0, 2\pi]$, represented as a purple circle on figure 5b. We are looking for $\theta_0 \in \Delta$ such that the grasping approach is the most vertical. The closed form solution can be determined and is represented by a red point on the example of figure 5b. Depending on the task space, this ideal approach may not be feasible. Particularly, an obstacle from the scene may prevent the robot to reach the object using the suggested approach. For example, the robot arm can reveal too big to use a vertical approach on an object between two shelves of a retail store. To tackle this issue, we generate several candidate grasping directions based on the ideal one introduced above. They are represented by the yellow points on figure 5b. They are generated by tilting the ideal direction by manually pre-defined angles depending on the object shape. Each candidate point is sent to the planner, which determines if it is reachable or not. When a reachable point is found, the search is stopped and the grasping can be tried by the robot. In this example, the selected point is represented in green.

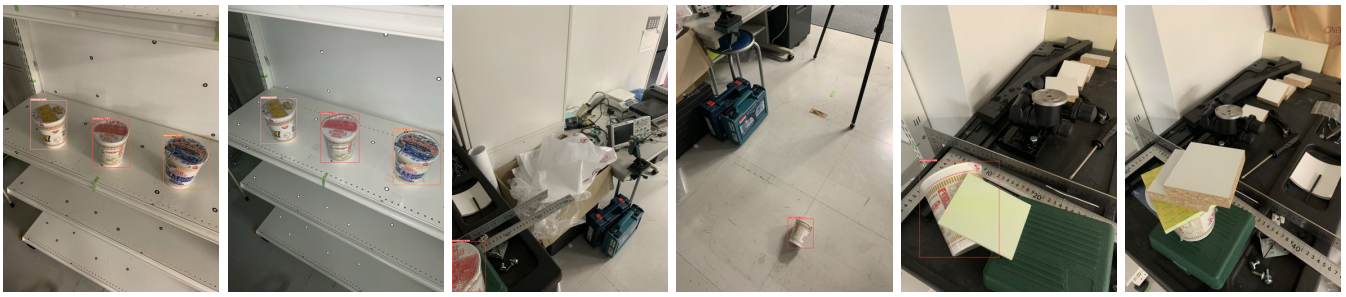


(a) Object's axes (b) Candidate grasping directions

Fig. 5: Determining grasping approach

TABLE I: Ablation study of the detection model training

Training data				Results	
Isolated objects	Random background	Random transformations	Random objects	Successful detection rate	False detection rate
✓	×	×	×	0.090	1.6
✓	✓	×	×	0.63	0.030
✓	✓	✓	×	0.84	0
✓	✓	✓	✓	0.94	0



(a) Success (b) Success (c) Success (d) Success (e) Success (f) Failure

Fig. 6: Detection module test on real pictures

IV. RESULTS

In this section, our method is tested using a Fetch Robot equipped with a Primesense Carmine 1.09 short-range RGB-D camera. Each module of our work is tested individually, before a complete experiment with the overall system.

A. Object Detection

The object detection model is trained with our synthetic data, using $N_p = 100$, on three types of noodle cup commonly found in Japanese convenience stores: Cup Noodle, Cup Noodle Curry and Cup Noodle Seafood. This will test the model ability to discriminate similar shaped objects of different brands/types as they are commonly found in supermarkets. Furthermore, as the object detection model was trained on synthetic data, testing its ability to work on real pictures is necessary to confirm the validity of our training process. For this purpose, we take 33 real pictures of a targeted object and proceed to an ablation study presented in table I

On figure 6, we see that the model can successfully discriminate the different variations of the object in different light conditions (6a, 6b) as well as handle distant views (6d). While the trained model can handle partial occlusions to a certain extent (6c, 6e), it fails to detect the object in very hard cases (6f). This can be explained by the absence of specific training for occluded scenarios in our synthetic data generation. As our experiment does not involve heavy occluded scenarios, we do not focus on this case.

B. 6-DOF Pose Estimation

For the pose estimation system described in section III-B, we set empirically $\boldsymbol{\delta_d} = 0.05m, \boldsymbol{\delta_\theta} = 45°$ and $\delta_e = 10^{-6}$. As the purpose of our demonstration is to show the robot ability to detect, estimate the pose and grasp an object in a variety of situations that could happen in a retail store, we create a benchmark for testing our 6-DOF pose estimation module on different cases. The test images are presented cropped around the targeted object in figure 7.

Using this benchmark, we compare our complete pose estimation system to the initial Multi-scale Projective ICP from [14] without modifications. Both methods are initialized using our Object Detection module (as described in III-A.2). Specifically, we test the ability of the methods to converge to the true pose versus a local optimum. For our test object, which is close to a cylinder shape despite a slight slope, a common local optimum would be to align the model to the correct position but with a backward orientation. We present the results in table II. The initial method fails because it is tailored for fast tracking of big objects but it is interesting to note that its core can be used for robust pose estimation by adding the parapet we propose in this paper.

The results of table II for the complete module ("Ours") are valid when using indifferently the Randomized Multi-Start or the Fibonacci Multi-Start algorithms described in section III-B. Therefore, we propose a more thorough comparison to evaluate the impact of choosing one method over the other. To have a benchmark with a precise ground truth, we use a RGB-D scene from the linemod dataset of the BOP challenge [7]. In figure 8, we plot for both algorithms the centroid error in millimeters for a pose estimation, depending on the number N of directions explored. As the Randomized method is not deterministic, we average its performance over 100 trials and give its maximum and minimum errors. To give an idea of the distribution, the area within one standard deviation from the mean is shown in light blue (clamped to positive values).

The randomized method has high variance when the number of directions sampled is low, which was expected. The Fibonacci method consistently outperforms the Randomized method's mean and achieves sub-milimetric precision starting from 20 directions explored. Therefore, we choose to use the Fibonnaci Multi-Start algorithm with $N = 24$.

(a) Front (b) Top (c) Side (d) Side-top (e) Side-bottom (f) Bottom

Fig. 7: Benchmark's views (cropped around the object)

TABLE II: 6-DOF pose estimation benchmark

	View						Legend	
	Front	Top	Side	Side-top	Side-bottom	Bottom	✓	Convergence to correct pose
Initial	∼	∼	×	∼	∼	×	∼	Convergence to local optimum
Ours	✓	✓	✓	✓	✓	✓	×	No convergence

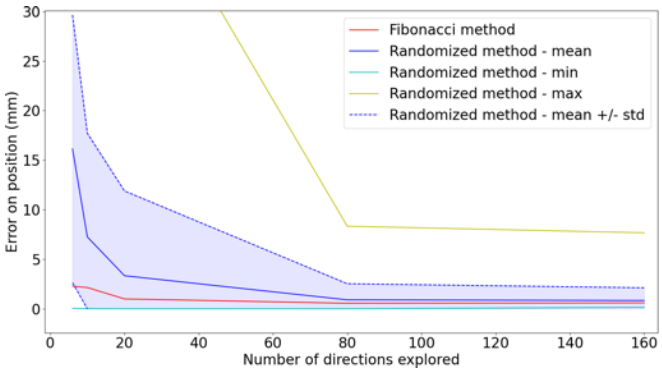


Fig. 8: Comparison of resulting error using Randomized or Fibonacci method

Fig. 9: Demonstration setup emulating a convenience store shelf, comprises a Fetch Robot, a shelf and three types of Noodle Cups. In the proposed scenario, two cups have fallen on the ground, in an unknown pose, due to a customer action.

C. Demonstration

The experimental setup shown in figure 9. The robot has to detect the two fallen cups, be able to discriminate them, estimate their pose, find a suitable grasping direction, grasp them and finally replace them on the shelf. The detection result can be seen on figure 10a while the pose estimation process is shown via the system graphical interface on figure 1. As it can be seen on the RGB image taken from the robot's camera, in the top left corner, very bright reflections of the ceiling light could make the detection difficult. Still, we see that the system correctly detected the two types of cup and estimated their 6-DOF poses. For evaluation purpose, the detection and pose estimation was repeated 70 times. The system achieved a success rate of 96%. All cases considered as failures were a ∼5 mm offset in the object principal axis, that would not have prevented grasping in this case.

After the pose estimation, a list of candidate approaches is sent to the planner as described in section III-C. Here, no obstacle prevents a vertical approach, so it is directly selected. We show on figure 10b that the robot successfully picks fallen cups to replace them on the shelf. The full demonstration can be seen in the attached video submission.

V. CONCLUSIONS

This paper addresses challenges in automating the retail industry using robot manipulators. The results confirm the success of the trained object detection model in discriminating similar-shaped objects from different brands/types commonly found in convenience stores. The evaluation of the 6D pose estimation module shows its consistent ability to converge to the true pose, avoiding local optima and achieving a position error smaller than 1mm. The experimental setup emulating a convenience store shelf demonstrates the robot's success in detecting fallen cups, distinguishing them, estimating their 6D poses, determining suitable grasping directions, and executing successful grasps.

Moreover, the system's demonstrated feasibility with minimal human intervention is noteworthy. Only a few pictures on a plain background without annotations are needed; the rest is automated and could be performed on a cloud application. This capability enables easy addition of new objects by retail store employees or other stakeholders, highlighting its practical implementation potential.

(a) Object Detection (robot's point of view) (b) Execution (external point of view)

Fig. 10: Demonstration

This research opens up perspectives for future work in automated in-store processes within the retail industry. Especially, integration with planning and control modules could be further explored to create entire autonomous behaviors such as exploring the store aisles, detecting and replacing fallen products, using the method developed in this paper.

ACKNOWLEDGMENT

The authors thank all members of JRL for the illuminating discussions that contributed to this work.

REFERENCES

[1] K. Azadeh, R. De Koster, and D. Roy, "Robotized and automated warehouse systems: Review and recent developments," *Transportation Science*, vol. 53, no. 4, pp. 917–945, 2019.

[2] N. Correll, K. E. Bekris, D. Berenson, *et al.*, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[3] C. Eppner, S. Höfer, R. Jonschkowski, *et al.*, "Lessons from the amazon picking challenge: Four aspects of building robotic systems," in *Proceedings. IJCAI*, ser. IJCAI'17. AAAI Press, 2017, p. 4831–4835.

[4] G. A. G. Ricardez, S. Okada, N. Koganti, *et al.*, "Restock and straightening system for retail automation using compliant and mobile manipulation," *Advanced Robotics*, vol. 34, pp. 235 – 249, 2020.

[5] R. Sakai, S. Katsumata, T. Miki, *et al.*, "A mobile dual-arm manipulation robot system for stocking and disposing of items in a convenience store by using universal vacuum grippers for grasping items," *Advanced Robotics*, vol. 34, no. 3-4, pp. 219–234, 2020.

[6] Y. Su, M. Saleh, T. Fetzer, *et al.*, "Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6728–6738.

[7] M. Sundermeyer, T. Hodaň, Y. Labbé, *et al.*, "Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 06 2023, pp. 2784–2793.

[8] C. Wang, D. Xu, Y. Zhu, *et al.*, "Densefusion: 6d object pose estimation by iterative dense fusion," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2019, pp. 3338–3347.

[9] Y. Xiang, T. Schmidt, V. Narayanan, *et al.*, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *CoRR*, vol. abs/1711.00199, 2017.

[10] M. Costanzo, G. De Maria, G. Lettera, *et al.*, "Can robots refill a supermarket shelf?: Motion planning and grasp control," *IEEE Robotics & Automation Magazine*, vol. 28, no. 2, pp. 61–73, 2021.

[11] J. Redmon, S. Divvala, R. Girshick, *et al.*, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2016.

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[13] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[14] K. Chappellet, M. Murooka, G. Caron, *et al.*, "Humanoid loco-manipulations using combined fast dense 3d tracking and slam with wide-angle depth-images," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2023.

[15] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[16] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images," *Proceedings. IEEE ICRA*, pp. 2724–2729 vol.3, 1991.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, *et al.*, "Kinectfusion: Real-time dense surface mapping and tracking," in *10th IEEE ISMAR*, 2011, pp. 127–136.

[18] F. Erich, B. Bourreau, C. K. Tan, *et al.*, "Neural scanning: Rendering and determining geometry of household objects using neural radiance fields," in *IEEE/SICE SII*, 2023, pp. 1–6.

[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[20] V. Foussereau, I. Kumagai, and G. Caron, "Towards retail stores automation: 6-dof pose estimation combining deep learning object detection and dense depth alignment - addendum," https://hal.science/hal-04230973, 2023.

[21] D. Aldous and U. Vazirani, ""go with the winners" algorithms," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 492–501.

[22] I. Kumagai, M. Murooka, M. Morisawa, *et al.*, "Reachability based trajectory generation combining global graph search in task space and local optimization in configuration space," in *IEEE/RSJ IROS*, 2022, pp. 4513–4520.