

Article

Robotic System for Post Office Package Handling

Oskars Vismanis , Janis Arents , Karlis Freivalds *, Vaibhav Ahluwalia and Kaspars Ozols 

Institute of Electronics and Computer Science, 14 Dzerbenes Street, LV-1006 Riga, Latvia;
oskars.vismanis@edi.lv (O.V.); janis.arents@edi.lv (J.A.); vaibhav.ahluwalia@edi.lv (V.A.);
kaspars.ozols@edi.lv (K.O.)

* Correspondence: karlis.freivalds@edi.lv

Abstract: Parcel sorting is becoming a significant challenge for delivery distribution centers and is mostly automated by using high-throughput sorting machinery, but manual work is still used to feed these machines by placing the parcels on the conveyor belt. In this paper, an AI-based robotic solution that automates the parcel placement task was developed. The architecture of the proposed system along with methods on how to implement it are described by using the currently available hardware and software components. The described choices lead to a well-functioning system and the gained insights will facilitate building similar systems for parcel delivery automation.

Keywords: smart industrial robots; computer vision; artificial intelligence; digitization; smart manufacturing; future factories

1. Introduction

Internet shopping is becoming more and more popular, leading to a rapid increase in postal parcels that need to be processed and shipped [1]. High-throughput parcel sorting lines have been developed and utilized in logistic centers that automate most of the work to redistribute the incoming parcels to their destinations [2]. However, manual work is still required to separate the incoming parcels from the pile and place them on the conveyor belt, forming a bottleneck in otherwise highly elaborate sorting centers. These bottlenecks are prone to get worse when demand increases, especially during holidays. In combination with an overall increase in unit labor cost, higher hourly compensation levels, and a decrease in the cost of robots, the upcoming installations of robotic systems will have decreased the required time for the return on investment [3]. From an economic point of view, these aspects highly motivate the research and development of new approaches to tackle processes that traditionally require human resources.

This seemingly easy and dull job of taking parcels from the pile and placing them on the belt poses several challenges that make its automation difficult. First, packages come in various sizes, shapes, and colors, so any computer vision algorithm, if used, must be highly flexible and not tied to any particular object type. Second, the mix of soft and hard objects, some being very thin, poses challenges to the mechanical design of their transfer mechanisms. Third, objects are piled and partially occluded, so the order in which they can be picked must be taken into account. Fourth, the resulting system has to provide high throughput and work reliably without the need for manual intervention.

The main scientific contribution of this article lies in the evaluation of an AI-based robot solution for automating the parcel-handling process. The proposed architecture tackles the challenges posed by diverse parcel characteristics, ensuring adaptability, efficiency, and reliability in parcel-sorting centers. The developed system and both the design choices and implementation in the physical setup are shown and described in detail. Various crucial aspects for the implementation of similar systems are discussed and possible shortcomings are identified. Universal Robot UR5 is employed, which is a hardware-agnostic robot control and a computer-vision system for detecting and moving parcels from an unstructured



Citation: Vismanis, O.; Arents, J.; Freivalds, K.; Ahluwalia, V.; Ozols, K. Robotic System for Post Office Package Handling. *Appl. Sci.* **2023**, *13*, 7643. <https://doi.org/10.3390/app13137643>

Academic Editor: Luigi Fortuna,
Nikolaos Papakostas and Arkadiusz
Gola

Received: 24 May 2023

Revised: 20 June 2023

Accepted: 26 June 2023

Published: 28 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

pile to the required location as illustrated in Figure 1. The solution can handle arbitrarily shaped objects, such as postal parcels, without additional training, and the achieved results show the reliability of the developed system in different scenarios.



Figure 1. Universal Robot UR5 moving around post office packages.

2. Related Work

For several decades, robotic systems have played a crucial role in automating various systems [4]. These systems are traditionally deployed in static and known environments where robotic motions can be fully preprogrammed or rely on simple input devices for an adjustment in the grasping position, typically varying only in two dimensions [5]. Robotic systems in such environments have several prerequisites that are frequently not feasible to comply with due to different limitations of the manufacturing plant. A typical limitation is the availability of space to divide and position objects on one level, which is also the case for the problem addressed in this article. In the post office parcel-handling context, the sorting process can be fully automated; however, it requires a substantially large space in the factory or even redesigning the whole process, where an industrial robot system, if used at all, could then be deployed with a rather simple 2D vision system [6,7].

In situations where space cannot be allocated to create structured conditions for the robotic system, uncertainty inevitably arises in the robot's workspace. This makes it challenging to grasp objects positioned in this region and necessitates the use of so-called smart industrial robot control [8]. While there are many application areas for smart industrial robot control [9], grasping objects in unstructured or weakly structured environments [10] remains one of the greatest challenges in modern robotics [8]. The grasping pipeline typically relies on an additional computer vision system that realizes a computer-vision-based control. This control method can be composed of several different approaches in both system architecture design level and learning strategies [11,12]. Different scenarios or combinations of conditions in the region of interest have been the main drive in the development of various approaches to tackle the grasping problem, mainly depending on the level of uncertainty in the region of interest and the type of the objects. Even though there can be a range of flexibility in the scenarios where the system can sustain the required level of performance, this can have a negative influence on the performance efficiency and maintainability aspects [13].

Firstly, the deep-learning-based approaches can be divided into two branches—working with known or unknown objects, e.g., model-based or model-free methods [14]. Even though model-based methods have become quite effective [15], they require previous knowledge about the objects of interest, which can become quite cumbersome when the training data need to be acquired. Recently, this aspect has been addressed by synthetic data generation methods [16] and sim-to-real transfer techniques [17]. Nevertheless, the goal of all the grasping techniques is to estimate the grasp pose of the object. For model-based methods, several steps before the estimated grasp pose can be present, such as object detection [18] and segmentation with respective post-processing steps depending on the

application scenario [19] or various approaches that directly estimate the 6D pose [20,21] of the object, from which the grasp pose can be derived.

On the other hand, model-free approaches search for the grasp pose directly without relying on prior knowledge of the object for training the system. These methods are highly desirable as they can generalize to previously unseen objects. They circumvent the pose estimation step and directly estimate the grasp pose by identifying a graspable spot in the region of interest [8].

One important model-free grasping method is Dex-Net [22]. To train this method, grasps across 1500 3D object models were analyzed, resulting in a dataset of 2.8 million point clouds, suction grasps, and robustness labels (Dex-Net 3.0). The dataset was then used to train a grasp quality convolutional neural network (GQ-CNN) to classify robust suction targets in single object point clouds. The trained model was able to generalize to basic (prismatic or cylindrical), typical (more complex geometry), and adversarial (with few available suction-grasp points) objects with success rates of 98%, 82%, and 58%, respectively. A more recent implementation of Dex-Net [23] (Dex-Net 4.0) containing five million synthetic depth images was used to train policies for a parallel-jaw and vacuum gripper, resulting in a 95% accuracy in gripping novel objects. In [24], a generative grasping convolutional neural network (GG-CNN) was proposed, where the quality and pose of grasps are predicted at every pixel, achieving an 83% grasp success rate on unseen objects with adversarial geometry. These methods are well suited for closed-loop control up to 50 Hz, enabling successful manipulation in dynamic environments where objects are moving. Furthermore, multiple grasp candidates are provided simultaneously, and the highest quality candidate ensures a relatively high accuracy in previously unknown environments.

A significant challenge is grasping objects in a cluttered real-world scene. In such a setting, not only should the object to be grasped be analyzed, but also the surroundings for the robot should not collide with it. GraspNet [25] addresses this challenge and learns to generate diverse grasps covering the whole graspable surface. Gripper collisions are avoided by considering them during training and by predicting grasps directly without relying on accurate segmentation.

Several object manipulation systems, applicable to post office automation, have been developed in the past. A system for package handling automation employing a robot arm with a vacuum gripper is described in [26]. Mailpiece singulation, i.e., removing items one by one from a moving conveyor, was described in [27]. They considered a bin-picking task, where a mix of objects with different shapes and weights should be grasped and removed, and designed a system comprising a range imaging camera, sparse data range imaging algorithm, grasp target selection algorithm, and a vacuum gripper. An implementation of a system for object manipulation was presented in [28] and showcased for two use cases: fish picking and postal parcel sorting. It employs model-free object localization based on Mask R-CNN for object detection and finding a grasp point for each segmented instance. A context-aware model for parcel sorting and delivery was proposed in [29]. The system includes the detection and localization of the parcel, recognizing the barcode and reading the delivery address, and performing robotic manipulation to transfer the parcel to the designated delivery box.

The literature review reveals several promising strategies for addressing this challenge. However, it is deficient in essential implementation specifics and comprehensive architectural recommendations for real-world systems. This article presents a proposed robotic system designed for post office package handling. It provides a detailed account of the key system components and offers insights into implementation aspects in physical systems.

3. Architecture

The proposed robotic system for post office package handling, as illustrated in Figure 2, builds upon the TRINITY use-case demonstration of the artificial-intelligence-based stereo vision system [30], consisting of multiple modules. The whole system is based on the robot operating system (ROS) [31] framework, which is used for communicating between

modules and controlling the system. The system includes a 6-DOF robot arm with a gripper, an RGB-Depth sensor, and two processing units for computer-vision-based and robotic-motion-based computations.

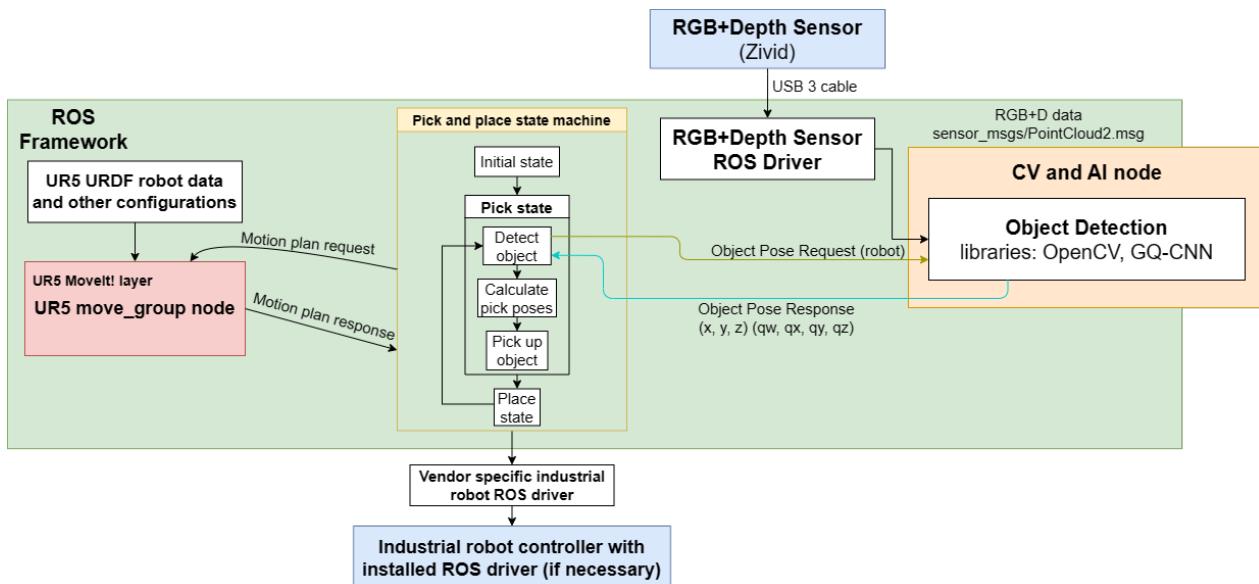


Figure 2. Architecture of the proposed system.

The workflow of the system can be described as follows: first, upon request, a 3D image of the workspace is captured with the RGB-Depth sensor and sent to a grasp quality convolutional neural network (GQ-CNN) Dex-Net 4.0 [23] for finding a grasp pose.

Second, when a pose is found, it is sent to a robot control state machine in a format of X, Y, Z coordinates and a W, X, Y, Z quaternion orientation. The robot control state machine is a set of states that describes the whole process in an overseable and defined way with predictable outcomes. It consists of multiple robot functions, such as moving to a desired pose or picking and releasing a desired object. It is implemented using the State Machine Asynchronous C++ (SMACC) [32] package.

Third, after receiving a pose, a grasp path is planned and, if successful, the state machine goes on through a full pick and place cycle. The robot control is implemented using the MoveIt! Motion Planning Framework [33], which manages the planning of paths and communication with the hardware. From the hardware side, the communication is processed using a robot-specific ROS driver by ROS-I in conjunction with a low-level controller provided by the industrial robot. After the object is placed in a predefined location, the process starts anew.

3.1. Hardware Setup

The system was built and tested with the Universal Robot UR5 6-DOF robot arm [34] with an attached Schmalz vacuum gripper [35] for gripping the objects. An image of the work environment can be seen in Figure 3, where, at the top, the depth sensor can be seen, on the right, the robot, and, in the middle, the transferable parcels.



Figure 3. Physical setup of the system.

3.1.1. Grippers

Choosing the right gripper for object manipulation is one of the main prerequisites for the successful automation of pick and place tasks. Many parameters should be considered, but usually the task itself mainly determines the gripper type. The two main gripping strategies are vacuum gripping and parallel gripping. Specifically for the robot system proposed in this article, a vacuum gripping technique enabled by Schmalz Rob-Set UR was used as it supports the system's hardware and software requirements and is well documented. Additionally, a wide variety of different vacuum caps were also included in the hardware infrastructure, which allow for rapid testing when different kinds of object material are introduced in the pick and place task, essentially also allowing us to experimentally determine the most suitable vacuum cap for parcel handling.

3.1.2. Depth Sensors

The information about the dynamically changing environment, specifically the grasping pose of randomly dropped objects, which can vary in six dimensions, was acquired using a depth sensor. Two different depth sensors (Zivid One M and Intel RealSense D415) that vary in such parameters as precision, field of view, working range, weight, and dimensions were considered.

After examining initial experiments and the nature of computer vision techniques required to handle post office packages, it was determined that the system relies on a high precision of the depth information. The common point precision of an Intel Realsense D415 is 1–3 mm, whereas, for a Zivid One M depth camera, it is around 60 μm . For this reason, the Zivid camera was chosen to be used in this robotic system. Zivid uses structured light for 3D information capture. It operates with a resolution of 1920×1200 at a frame rate of up to 12 frames per second and utilizes the USB 3.0 SuperSpeed as the connection interface. The Zivid camera is ideal for high-precision tasks when the camera can be mounted statically in a position in the workspace, where its field of view covers most of it, as the comparatively large dimensions ($226 \text{ mm} \times 86 \text{ mm} \times 165 \text{ mm}$) and weight (2 kg) of this camera would restrict some movements for the industrial robot if mounted on it.

3.1.3. Industrial Robot

One of the main hardware components of the system is the industrial robot. An industrial collaborative robot (Universal Robots UR5) was used, which is ideal for automating low/medium-weight processing tasks. It has a maximum payload of 5 kg, a maximum joint speed of 180 degrees/second, an 850 mm reach radius, which can be improved with gripper modifications, and a pose repeatability of $+/- 0.1$ mm. Nevertheless, as the system is built upon ROS, the robot control module can be adjusted for usage with different kinds of robot manufacturers [36].

3.1.4. Processing Units

As depicted in Figure 4, the proposed system is composed of two processing units, one for robot control and the other for computer vision, both of which were configured as a multi-ROS-PC setup. This was carried out to make each of the units independent from one another in terms of computing, so no unexpected interruptions would occur. To ensure that the robot control is not affected by system latencies, the robot control PC was set up with real-time capabilities by using Ubuntu 18.04 with an enabled real-time kernel. The real-time kernel has compatibility issues with NVIDIA GPU drivers, which are needed for computer vision tasks, which is why a two-unit setup was used. The robot control PC runs ROS Melodic. It has two Ethernet cards to ensure a direct connection with the robot's controller and ROS multi-master ecosystem. Apart from the two Ethernet cards, it can be an ordinary desktop computer, as the main task is to generate feasible trajectories of an industrial robot, and there are no other processes that require high-performance capabilities. In this case, it had the following specifications: Intel Core i7-8400K CPU@ 3.70 GHz and 16 GB RAM.

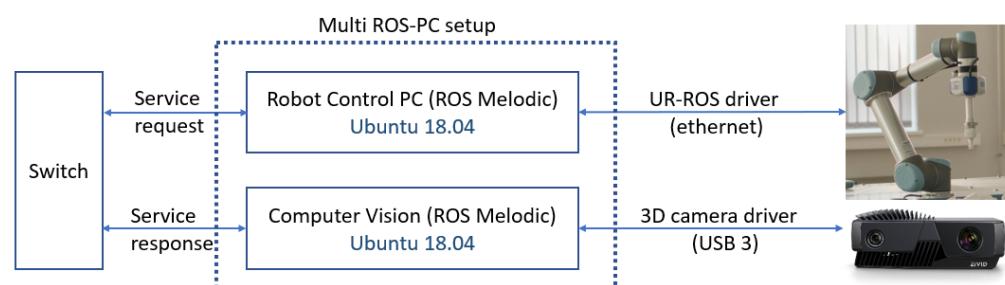


Figure 4. Hardware setup.

The processing unit for computer-vision-related computations, on the other hand, requires a graphics processing unit, both for running ZIVID and for grasp point detection. It runs on regular Ubuntu 18.04, also with ROS Melodic. In this case, the computing capabilities can have a big influence on the overall cycle times of the pick and place process. The computer vision PC has the following specifications: AMD Ryzen 5 3600@ 3.60 GHz, 32GB RAM, and NVIDIA GeForce RTX3060Ti.

3.2. Software Ecosystem

ROS. The whole system communication was conducted using ROS Melodic Morenia, as it is the ROS version compatible with Ubuntu Linux 18.04. All further described packages are also compatible with this ROS version.

State Machine Asynchronous C++. The state machine control of the system was carried out using the SMACC library for ROS. Conceptually, SMACC defines each individual state machine component as Orthogonal, which also includes a Client. Clients define connections outside of the state machine. In the state machine, transitions between the states are triggered by Events [32].

As possible alternatives for controlling the system in such a manner, there exist such ROS packages as SMACH [37] and behavior trees [38]. Behavior trees offer similar

functionalities as state machines. With behavior trees, the development of more complex systems, such as those with uncertainties, is possible, but such possibilities can also make the regular development process more complex than necessary, especially when debugging. For this reason, it was chosen to make the system using state machines.

As a different option for implementing state machines, the SMACH package was explored. The previously mentioned SMACC package is based on the SMACH package, the only real difference being that SMACC can be developed in C++ but SMACH is developed in the Python programming language. Since the overall system is all in C++, the SMACC package was chosen for easier development.

Data acquisition. An important factor for choosing any hardware that will need to work with an open-source framework, such as ROS, is said hardware's support for this framework. In Section 3.1.2, sensors such as Intel Realsense and Zivid are mentioned, and both of them also support being used with ROS by providing special ROS drivers. These drivers include functions for easy use such as capturing an image or a video stream, changing capture settings, and so on.

It is also important to note that, when using additional sensors in such systems, a special calibration process called hand–eye calibration is required in order to have all the physical system units operate in a combined coordinate space. This requires acquiring data with both the sensor and the actuator at the same time and then calculating coordinate transformations between them.

MoveIt! The path planning and execution were carried out using the MoveIt! Motion Planning Framework. It is a collection of tools for robot motion planning, visualization, and execution, which can be used in C++ and Python. It employs such tools as the Open Motion Planning Library (OMPL) [39], which provides a set of sampling-based motion planners to choose from for a particular task, and it comes with a set of functions to use for robot control. In addition, it provides the option to visualize a digital twin of the system for easy offline development and can also provide useful information about the planned paths in both offline and online scenarios.

UR driver. Similar to choosing sensors, the choice of the robot itself is also strongly influenced by its support for the framework used. Universal Robot supports work with their robots in the ROS framework by providing the UR ROS driver. This includes all the necessary tools for connecting, communicating, and controlling the robot easily, which makes this robot an easy choice for the development of such systems.

3.3. Communication and Synchronization

Since the whole system is controlled by a state machine, the system functions by transitioning from one definite state to the next. As mentioned in Section 3.1.4, the robot control and workspace sensing are processed on two different units. This provides an extra layer of complexity to the system in the form of communication and synchronization.

The signal to start detection can be sent once the robot is no longer blocking the workspace, but this happens in multiple states before the robot is ready to begin a new cycle. Since the grasp point detection takes an unpredictable amount of time, the robot system can both run forward or fall behind during the process in terms of synchronizing when the new grasp point is sent. To make the process more time efficient, the communication between processing units happens in parallel with the robot's motion. When a new grasp point is detected, the point is not sent as a message but, rather, it is set as a new value on the ROS parameter server with the tag `new`, describing that the robot has not yet used this point. Once the robot has reached a state where it can act upon a new point, it checks this new tag, which can be set to either true or false. Depending on when both of these parts of the system reach this point, the robot either waits until a new pose with this tag set to true appears, or reaches the new cycle starting pose and instantly starts working on the new cycle. In this way, the two parts of the system are both co- and independent in their functionality.

3.4. Grasp Point Detection

As the postal parcels have irregular shapes—see Figure 5a—for grasp point calculation, the Dex-Net 4.0 [23] library was selected, which delivers robust performance in the proposed application. It was configured to utilize the grasp quality convolutional neural networks (GQ-CNNs) method for grasp detection and delivers information for a system with a suction gripper, for which it is well suited. The camera intrinsics were obtained directly from the camera using its ROS driver and the calibration was performed beforehand. The chosen Zivid camera naturally produces images in 1920×1200 resolution and such images take approximately 10 s to process with Dex-Net. To increase the performance, the images were firstly cropped to show only the region of interest (ROI) where the packages reside. Afterward, they were downsampled to 640×480 .

The output quality of the Dex-Net is greatly enhanced if one provides a segmentation mask in its input showing precisely the region occupied with objects (the objects do not need to be segmented individually). To produce such a mask, the depth image of a clean table was taken during the calibration phase. When the system is operating, the active workspace data were compared with the empty table depth image to show only pixels with a depth value of fewer than 2 millimeters (minimal parcel height) above it. Such a setup allows Dex-Net to produce reliable grasps; see Figure 5.



(a) A pile of postal packages

(b) The detected grasp point

Figure 5. Dex-Net input and output. It takes an RGBD image of the object pile as input and produces a grasp point.

3.5. Robot Control

The control system consists of multiple states that manage different robot actions. The whole robot movement process is divided into several states, such as moving to a predefined start position, moving to an ever-changing grasp position, activating the gripper, and so on.

Trajectory planning for movement in these states was performed using MoveIt!, which is connected to the robot and gripper drivers with the movegroup interface. In the MoveIt! setup of the system, the planners best suited for the job can be chosen from the ones provided in the OMPL. The physical setup of the system is also represented digitally in the MoveIt! setup. Both the table and the robot have a 3D model, which represents the system in real-time. This provides the necessary information for the planner to plan safe, collision-free trajectories. This also allows the system operator to see both the planned trajectories and found object coordinates before they are acted upon.

4. Results and Discussion

The described system was implemented in the lab environment and validated for post office parcel handling in multiple scenarios, illustrated in Figure 6. The experiments were conducted in five different scenarios, each consisting of a different amount of packages, and the system was tested for its ability to clear the workspace. The physical setup of the robot is depicted in Figure 3. The robot is located near the table holding a pile of realistic parcels—see Figure 5a—and they should be moved to a container standing nearby. The camera is mounted approximately 1 m above the table.



Figure 6. Testing scenario examples. From left to right, 1, 5, 10, 15, 20 packages in the workspace.

Five scenarios, consisting of 1, 5, 10, 15 and 20 parcels in the workspace, were used for testing. Each scenario was tested 10 times, and the packages were randomly distributed in the workplace. The packages were piled on top of each other when possible to replicate real-life setups as much as possible. The attempts for each scenario were counted and compared to the expected amount, which was set to one attempt per package in the scenario. Afterward, the average deviation from the expected attempt amount was calculated and, lastly, an overall average deviation was calculated to see the mean error of the system. The results can be seen in Table 1.

Table 1. Experimental results.

Scenario, Packages	Grasp Attempts, Average	Additional Grasps	Success Rate
1	1	0.00%	100%
5	5.3	5.66%	100%
10	10.4	3.85%	100%
15	16	6.25%	100%
20	21.3	6.10%	100%
Overall additional grasps		4.37%	

As per results, the 1-package scenario showed a perfect score, 5 packages resulted in an average of 5.3 attempts per 5 packages, for 10 packages, it took an average of 10.4 attempts, for 15 packages, an average of 16 attempts were needed, and, for 20 packages, an average of 21.3 attempts was required. Overall, the system achieved a 100% success rate for emptying the workspace, meaning that all the packages were successfully handled; however, an average of a 4.37% increase in additional grasps can be observed when compared to the perfect one attempt per package. By measuring only one attempt per package, the system succeeded with 94.1% successful grasps. Even though some grasps did not achieve success, the system could recover and grasp the object on the second try.

While testing, it was observed that most of the failed attempts can be attributed to a grasp pose located on an uneven part where the vacuum gripper cannot achieve the required vacuum to grasp the desired package. This issue could be addressed by a more flexible vacuum cap or increasing the vacuum, whereas the available hardware in this experimental setup was already adjusted as much as possible for these experiments. A customized vacuum system for a specific use-case and package materials could increase the average grasping success rates.

To the best of the author's knowledge, there is no relevant literature available on the post office parcel-handling process; however, the available similar systems working with unknown and arbitrarily placed objects show similar precision ratings [22–24]. Furthermore, the developed system experimentally proves the viability of the robotic system for deployments in complex scenarios, where the objects can be very similar but also differ in terms of size and materials.

In a real use case, there would be a conveyor belt in place of the container. The system was tested with different pile configurations and parcel types, and, in all cases, the system

worked reliably, as can be seen in the video demonstration <https://www.youtube.com/watch?v=djxrbQZtiKk>, accessed on 20 June 2023.

During the initial stages of development, several issues, which were later fixed, were encountered:

1. Very thin parcels such as envelopes were undetectable with low-precision depth cameras. Therefore the high-precision Zivid camera was chosen.
2. The distance calibration from the camera to the table has to be precise; otherwise, the pickup point may be seemingly inside the table and the robot driver refuses to grasp at that location as it might see it as a potential collision.
3. The package could have wrinkles or bends at the grasp point and the suction gripper would therefore be unable to hold the object. In addition, the detected grasp point and the gripper approach direction could turn out to be physically impossible for the robot, of which Dex-Net was not aware (this case mostly occurs with large, bulky parcels near the edges of the pickup zone). To deal with these cases, recovery actions were implemented. In such cases, the system tries to grasp again at a newly calculated point. Due to randomized grasp point selection, the system can eventually achieve a successful grasp, although some time might be wasted.

After these fixes, the system always recovered from failures in the experiments and was able to transfer the packages.

5. Conclusions

In this paper, an AI-based robotic solution for automating the task of parcel placement was presented, and a series of choices and solutions for implementing such a system with currently available hardware and software components were proposed. The architecture of the proposed system, which comprises a Universal Robot UR5 robotic arm, a Zivid RGB-D camera, multiple processing units, Dex-Net grasp pose estimation software, and the robot control system components, all successfully integrated under the ROS framework, is described in detail in this paper.

The architecture details of the proposed solution and insights that will facilitate building similar systems have been given. The choices presented in this paper lead to a well-functioning system. It was found that Dex-Net grasp detection provides a high precision of the grasp point. The solution can handle arbitrarily shaped objects, such as postal parcels, without additional training. It easily integrates within any workflow under the ROS framework and successfully addresses the identified challenges. With this solution, installed at parcel delivery centers, human workers can be freed to perform more pleasant work.

Author Contributions: Conceptualization, J.A., O.V. and K.F.; methodology, J.A., O.V. and K.F.; validation, K.F.; formal analysis, K.F.; investigation, J.A., O.V., V.A. and K.F.; resources, J.A.; data curation, J.A., O.V., V.A. and K.F.; writing—original draft preparation, J.A., O.V., V.A. and K.F.; writing—review and editing, J.A., O.V., V.A., K.F. and K.O.; visualization, J.A., O.V., V.A. and K.F.; supervision, J.A. and K.O.; project administration, K.O.; funding acquisition, K.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research is the result of activities within the “Digital Technologies, Advanced Robotics and increased Cyber-security for Agile Production in Future European Manufacturing Ecosystems” (TRINITY) project, which has received funding from the European Union’s Horizon 2020 Research and Innovation programme under grant agreement No 825196.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. EUROPEAN E-COMMERCE REPORT 2022. Available online: https://ecommerce-europe.eu/wp-content/uploads/2022/06/CMI2022_FullVersion_LIGHT_v2.pdf (accessed on 22 May 2023).
2. Xu, X.; Xue, Z.; Zhao, Y. Research on an Algorithm of Express Parcel Sorting Based on Deeper Learning and Multi-Information Recognition. *Sensors* **2022**, *22*, 6705. [CrossRef]
3. Jung, J.H.; Lim, D.G. Industrial robots, employment growth, and labor cost: A simultaneous equation analysis. *Technol. Forecast. Soc. Chang.* **2020**, *159*, 120202. [CrossRef]
4. Zamalloa, I.; Kojcev, R.; Hernandez, A.; Muguruza, I.; Usategui, L.; Bilbao, A.; Mayoral, V. Dissecting robotics-historical overview and future perspectives. *arXiv* **2017**, arXiv:1704.08617.
5. Tan, K.S.; Ayob, M.N.; Hassrizal, H.B.; Ismail, A.H.; Azmi, M.S.B.M.; Hashim, M.S.M.; Othman, S.M.; Shahriman, A.B.; Low, Y.H. Integrating Vision System to a Pick and Place Cartesian Robot. *J. Phys. Conf. Ser.* **2021**, *2107*, 012037. [CrossRef]
6. Andhare, P.; Rawat, S. Pick and place industrial robot controller with computer vision. In Proceedings of the 2016 International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 12–13 August 2016; pp. 1–4. [CrossRef]
7. Alghamdi, B.; Lee, D.; Schaeffer, P.; Stuart, J. An Integrated Robotic System: 2D-Vision Based Inspection Robot with Automated PLC Conveyor System. In Proceedings of the 4th International Conference of Control, Dynamic Systems, and Robotics (CDSR'17), Toronto, ON, Canada, 21–23 August 2017.
8. Arents, J.; Greitans, M. Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.* **2022**, *12*, 937. [CrossRef]
9. Racinskis, P.; Arents, J.; Greitans, M. A Motion Capture and Imitation Learning Based Approach to Robot Control. *Appl. Sci.* **2022**, *12*, 7186. [CrossRef]
10. Torres, P.; Arents, J.; Marques, H.; Marques, P. Bin-Picking Solution for Randomly Placed Automotive Connectors Based on Machine Learning Techniques. *Electronics* **2022**, *11*, 476. [CrossRef]
11. Mohammed, M.Q.; Kwek, L.C.; Chua, S.C.; Al-Dhaqm, A.; Nahavandi, S.; Eisa, T.A.E.; Miskon, M.F.; Al-Mhiqani, M.N.; Ali, A.; Abaker, M.; et al. Review of Learning-Based Robotic Manipulation in Cluttered Environments. *Sensors* **2022**, *22*, 7938. [CrossRef]
12. Yin, Z.; Li, Y. Overview of robotic grasp detection from 2D to 3D. *Cogn. Robot.* **2022**, *2*, 73–82. [CrossRef]
13. Shahria, M.T.; Sunny, M.S.H.; Zarif, M.I.I.; Ghommam, J.; Ahamed, S.I.; Rahman, M.H. A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions. *Robotics* **2022**, *11*, 139. [CrossRef]
14. Zhang, H.; Tang, J.; Sun, S.; Lan, X. Robotic grasping from classical to modern: A survey. *arXiv* **2022**, arXiv:2202.03631.
15. Kleeberger, K.; Bormann, R.; Kraus, W.; Huber, M.F. A survey on learning-based robotic grasping. *Curr. Robot. Rep.* **2020**, *1*, 239–249. [CrossRef]
16. Arents, J.; Lesser, B.; Bizuns, A.; Kadikis, R.; Buls, E.; Greitans, M. Synthetic Data of Randomly Piled, Similar Objects for Deep Learning-Based Object Detection. In Proceedings of the Image Analysis and Processing—ICIAP 2022, Lecce, Italy, 23–27 May 2022; Springer International Publishing: Cham, Switzerland, 2022; pp. 706–717.
17. Duplevska, D.; Ivanovs, M.; Arents, J.; Kadikis, R. Sim2Real image translation to improve a synthetic dataset for a bin picking task. In Proceedings of the 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 6–9 September 2022; pp. 1–7.
18. Shashank, T.; Hitesh, N.; Gururaja, H. Application of few-shot object detection in robotic perception. *Glob. Transit. Proc.* **2022**, *3*, 114–118. [CrossRef]
19. Sun, G.J.; Lin, H.Y. Robotic Grasping Using Semantic Segmentation and Primitive Geometric Model Based 3D Pose Estimation. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 337–342. [CrossRef]
20. Zhuang, C.; Li, S.; Ding, H. Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking. *Robot. Comput.-Integr. Manuf.* **2023**, *82*, 102541. [CrossRef]
21. Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv* **2018**, arXiv:1809.10790.
22. Mahler, J.; Matl, M.; Liu, X.; Li, A.; Gealy, D.; Goldberg, K. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5620–5627.
23. Mahler, J.; Matl, M.; Satish, V.; Danielczuk, M.; DeRose, B.; McKinley, S.; Goldberg, K. Learning ambidextrous robot grasping policies. *Sci. Robot.* **2019**, *4*, eaau4984. [CrossRef] [PubMed]
24. Morrison, D.; Corke, P.; Leitner, J. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv* **2018**, arXiv:1804.05172.

25. Sundermeyer, M.; Mousavian, A.; Triebel, R.; Fox, D. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13438–13444.
26. Lee, J.; Bastuscheck, M. Intelligent sensing and machine reasoning for robotic manipulation of irregular moving objects. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 1562–1563.
27. Frederick, B.; Carrell, R.M.; Alexander, E.M.; Van Sant, G.J. Handling of irregular-sized mailpieces by adaptive robotics. *IEEE Control Syst. Mag.* **1989**, *9*, 3–8. [[CrossRef](#)]
28. Castaman, N.; Cenzato, A.; Tonello, S.; Menegatti, E.; Ghidoni, S.; Pretto, A. Model-free Bin-Picking: Food Processing and Parcel Processing Use Cases. In Proceedings of the 2nd Italian Conference on Robotics and Intelligent Machines, Italian Institute of Robotics and Intelligent Machines, Rome, Italy, 10–12 December 2020.
29. Shin, M.S.; Ko, M.; Ju, Y.; Jung, Y.; Lee, B. Implementation of context-aware based robot control system for automatic postal logistics. *Stud. Inform. Control* **2013**, *22*, 71–80. [[CrossRef](#)]
30. Artificial Intelligence Based Stereo Vision System. Available online: <https://trinityrobotics.eu/use-cases/artificial-intelligence-based-stereo-vision-system/> (accessed on 17 April 2023).
31. Quigley, M. ROS: An open-source Robot Operating System. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.
32. SMACC—State Machine Asynchronous C++. Available online: <https://smacc.dev/> (accessed on 17 April 2023).
33. Coleman, D.; Sucan, I.A.; Chitta, S.; Correll, N. Reducing the Barrier to Entry of Complex Robotic Software: A MoveIt! Case Study. *arXiv* **2014**, arXiv:1404.3785.
34. UR5 Collaborative Robot Arm. Available online: <https://www.universal-robots.com/products/ur5-robot/> (accessed on 17 April 2023).
35. Schmalz. Available online: <https://www.schmalz.com/en/vacuum-technology-for-automation/vacuum-components/vacuum-generators/vacuum-generators-end-of-arm/handling-sets-ecbpi-308297/10.03.01.00504/> (accessed on 17 April 2023).
36. UR5 Data Sheet. Available online: https://www.universal-robots.com/media/1801303/eng_199901_ur5_tech_spec_web_a4.pdf/ (accessed on 17 April 2023).
37. GitHub—Ros/Executive_SMACH: A procedural Python-Based Task Execution Framework with ROS Integration. Available online: https://github.com/ros/executive_smach (accessed on 2 May 2023).
38. Colledanchise, M.; Ögren, P. How Behavior Trees Modularize Hybrid Control Systems and Generalize Sequential Behavior Compositions, the Subsumption Architecture, and Decision Trees. *IEEE Trans. Robot.* **2017**, *33*, 372–389. [[CrossRef](#)]
39. Sucan, I.A.; Moll, M.; Kavraki, L.E. The Open Motion Planning Library. *IEEE Robot. Autom. Mag.* **2012**, *19*, 72–82. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.