RL Project - Applied case studies of Machine Learning

Sommaruga Tommaso, Rava Miro, Lucchina Luca, Nazifi Endrit
SUPSI

Abstract
This report presents the development of a vision-guided pick-and-place system integrating computer vision and robotic control. The system detects, localizes, and manipulates objects using advanced techniques like PCA and ICP for pose estimation and alignment. Calibration methods ensure accuracy in mapping between the camera and robot frames. Experiments validate the system's performance on circular tapes, highlighting challenges and potential improvements for broader applications. The proposed pipeline offers a scalable approach to automating object handling tasks in industrial settings.

Contents

1 Problem Definition

1.1 Task Overview

Our project focuses on the vision-guided picking of tapes placed randomly on the worktable. These objects, identifiable by their circular shape and distinct inner holes, were selected due to their consistent structure, making them suitable for both segmentation and robotic manipulation. The tapes are placed on a black working surface to facilitate object detection through contrast-based methods, ensuring clear isolation of the parts from the background.

The task involves detecting the precise position and orientation of the tapes in the image using computer vision techniques. Once localized, the robot calculates an appropriate picking pose to successfully grasp and place the tapes in an organized manner. This approach allows for a robust and accurate pick-and-place operation, demonstrating the integration of image processing, pose estimation, and robotic control.

1.2 Objectives of the Project

The primary goal is to design a system that detects and localizes objects in images using computer vision. More complex objects provide an additional challenge and help demonstrate the robustness of the approach. Once detected, the system calculates the 3D coordinates of these objects, enabling precise robotic manipulation. The project also seeks to establish a seamless pipeline that integrates image processing, calibration, and robotic control into an end-to-end workflow. Ultimately, the system should accurately command a FANUC robot to pick up and place objects in predefined locations.

2 Provided Setup

Codebase structure and pipeline overview

2.1 Codebase and Structure

The project is built on a modular codebase comprising calibration scripts, vision and detection algorithms, robot control modules, and utility libraries. The individual scripts play important roles in enabling the vision-guided pick-and-place system to operate efficiently.

- **extrinsic_calibration**: Focuses on aligning the camera and robot frames by solving the extrinsic calibration problem. It begins by detecting chessboard corners in undistorted images, refining their positions, and using the PnP algorithm to compute rotation and translation matrices. The computed transformation matrix is saved for future use, and visualization tools are included to verify calibration.
- **image_support**: Provides utility functions for rigid registration of 3D points, constructing transformation matrices,

and 3D visualization. It also includes functions for interactive scatter plots of point clouds and drawing coordinate axes on images, aiding in debugging and calibration validation.

- **intrinsic_calibration**: Calibrates the camera's intrinsic parameters by analyzing chessboard images. It estimates the camera matrix, lens distortion coefficients, and refines them using subpixel corner detection, ensuring precise camera calibration.

- **robot_control**: Provides a comprehensive set of functions for controlling FANUC robots via Ethernet/IP communication. It supports both joint and Cartesian movements, reads and writes positions, and controls the gripper's actions, among other features.

- **vrobot_control**: Extends the functionality of `robot_control` by adding image acquisition capabilities. It automates the process of capturing images through the robot's onboard camera, triggering TP programs, retrieving compressed image files, and converting them for processing.

- **main file**: Demonstrates an end-to-end example of robot control and image acquisition. It initializes communication with the robot, retrieves its Cartesian position, and controls the gripper while capturing images.

- **fanuc_driver**: The low-level driver, enabling Ethernet/IP communication with the robot. It ensures reliable execution of commands, such as reading and writing positions, and monitoring robot states.

2.2 Pipeline Overview

The vision-guided pick-and-place system follows a robust and structured pipeline that integrates computer vision and robotic control to accurately detect, localize, and manipulate objects. The pipeline consists of several steps:

2.2.1 Camera and Robot Initialization

- **Camera Calibration:** The camera's intrinsic parameters (camera matrix K and distortion coefficients dist) and the extrinsic transformation matrix (cam2robot) between the camera and robot are loaded from precomputed files (`intrinsics.npz` and `cam2rob.pkl`).
- **Robot Setup:** A connection to the FANUC robot is established via its IP address (`192.168.1.12`). A `go_home` function ensures that the robot starts from a predefined home position.

2.2.2 Image Acquisition and Preprocessing

- The robot captures an image using its onboard camera.
- The captured image is undistorted using the camera parameters to eliminate lens distortions and provide accurate spatial information.
- The image is converted to grayscale and thresholded to create a binary image that isolates objects against the black background.

2.2.3 Object Detection and Shape Matching

- **Connected Components Analysis:** A preprocessing function (`preprocess`) filters the binary image by identifying connected components, removing noise, and selecting valid objects based on size and aspect ratio (e.g., area between 1000 and 8000 pixels, aspect ratio between

0.5 and 2.0).

- **Template Matching:** The pipeline matches the contours of detected objects against a predefined template using contour matching (`matchShapes`), which computes similarity based on Hu moments values. A threshold for valid matches is set at n (where n depends on the template's complexity and the distinctiveness of the object, and lower values indicate closer matches).

2.2.4 Centroid Calculation and Principal Component Analysis (PCA)

- For each matched contour, the centroid (center of mass) is calculated using image moments.
- **Principal Component Analysis (PCA):** PCA is applied to extract the principal directions (eigenvectors) of the detected objects, which helps estimate their orientation.

2.2.5 Coordinate Transformation

- **Pixel to Camera Coordinates:** The centroid coordinates (in pixels) are transformed into camera coordinates, assuming a fixed working distance ($Z_c =$ 835 mm).
- **Camera to Robot Coordinates:** The camera coordinates are then converted into robot coordinates using the pre-calculated transformation matrix (cam2robot). This ensures the robot can move accurately to the object's position in its workspace.

2.2.6 Robot Movement to Detected Objects

- The robot is commanded to move to the detected object positions by calculating the new coordinates with necessary offsets.
- The robot's gripper is controlled to pick up the object, and once the task is completed, the robot moves back to its home position.

This pipeline enables a robust vision-guided pick-and-place system, where the robot accurately detects, localizes, and manipulates objects based on real-time image analysis.

---

3 Calibration

3.1 Intrinsic Calibration

The intrinsic calibration process involves analyzing chessboard images to compute the camera matrix and lens distortion coefficients. By using subpixel corner detection, the calibration achieves high precision, ensuring accurate measurements. Distortion correction further refines these parameters, enabling reliable object localization in the robot's workspace.

This step is critical in ensuring that the robot can accurately interpret images taken from the camera. Correct calibration is essential for object localization and precise robotic manipulation.

3.2 Extrinsic Calibration

Extrinsic calibration aligns the camera frame with the robot frame using the PnP (Perspective-n-Point) algorithm. The process calculates rotation and translation matrices, which are validated through visualization tools. The final output is a transformation matrix that facilitates mapping coordinates between the two frames, ensuring seamless integration of vision and robotic systems.

This calibration step ensures that the robot and camera are operating in sync, with the correct coordinate system transformations allowing for precise object manipulation in the robot's workspace.

4 Experiments and Methodology

This section provides an overview of the experimental setup, methodology, and the hyperparameter tuning process.

4.1 Object Picking

The object-picking experiments were designed to explore two key methods: the Iterative Closest Point (ICP) algorithm for circular tapes and Principal Component Analysis (PCA) for markers and pipes. The ICP algorithm was particularly effective in aligning the reference shapes with the target shapes, facilitating accurate registration. This alignment process is crucial for the robotic manipulation of circular tapes, allowing the robot to consistently pick and place the objects in a controlled manner. The ICP technique proved to be efficient in handling slight misalignments or deformations in the circular shape, ensuring the stability of the picking process.

PCA, in contrast, was employed to calculate the orientation and barycenters of markers and pipes. The use of PCA for these objects enabled an understanding of their spatial positioning, which is important for identifying the objects' centers and aligning them for picking. The PCA method exhibited promising results in detecting the overall orientation of markers and pipes, but it faced challenges in reliably determining object location when the markers or pipes were tilted or obscured. While PCA's results were generally accurate in controlled settings, further testing is needed to evaluate its performance in real-time robotic applications, especially in the context of varying orientations and environments.

4.2 Threshold and Template Matching Tuning

Thresholding and template matching were key components of the object-detection pipeline. To improve the detection of objects, adaptive thresholding was implemented to isolate the objects from the black background, ensuring that the foreground was clearly separated from the surrounding environment. This separation was vital for minimizing false positives during object recognition. However, thresholding alone was insufficient for detecting objects in environments with varying lighting conditions, so it was coupled with an optimized template-matching technique.

Template matching was further refined to improve the recognition of valid object contours while excluding noise and other extraneous elements that might interfere with detection. The matching algorithm was tuned to recognize specific object features more accurately, reducing the chance of incorrectly identifying objects in cluttered scenes. By adjusting various parameters within the template-matching algorithm, such as the matching threshold and the shape parameters, the accuracy of the detection system was significantly enhanced. This refinement led to better identification of complex shapes, particularly those that were difficult to distinguish from the background or other objects.

The improved thresholding and template-matching processes resulted in a more robust system for object detection, especially for more intricate objects with similar visual features. These improvements also reduced the system's susceptibility to errors caused by background noise, leading to higher overall detection accuracy and reliability.

4.3 Generalizing the Codebase

To ensure that the object-detection pipeline could adapt to a wide range of environments and object types, significant work was done to generalize the underlying codebase. One of the key objectives was to enhance the system's ability to operate under various lighting conditions. Initially, we attempted to integrate adaptive thresholding into the codebase, aiming to automatically adjust detection parameters based on changes in light intensity or environmental conditions. However, despite several trials, we were unable to achieve consistent results with this automatic adjustment, especially under fluctuating lighting conditions.

As a result, we opted to create different configuration files for manual thresholding adjustments, which were refined after multiple iterations. In these files, thresholding values were manually adjusted to improve the detection performance across different lighting environments, ensuring better consistency with slight changes in light intensity. Although this approach required additional effort and lacked the flexibility of automatic adjustments, it helped to improve the system's robustness when the lighting conditions varied, particularly in indoor or controlled environments.

We also focused on refining the codebase to handle multiple objects, but we did not introduce multi-object template matching at this stage. Instead, we made optimizations to improve the detection of individual objects in cluttered scenes. Although the system could detect single objects reliably, handling multiple objects in the same scene remains an area for future development. Further work will be needed to implement a robust solution for multi-object detection, which will be critical for tasks that involve the manipulation of several objects simultaneously.

By making these adjustments, the codebase has become more adaptable to dynamic environments, but further work is required to enhance the system's robustness in real-world scenarios. These ongoing efforts aim to extend the pipeline's utility, making it a more reliable tool for object-picking tasks, even in environments with unpredictable lighting or varying object orientations.

5 Results and Observations

5.1 Performance and Behavior

The system successfully demonstrated its ability to detect, localize, and manipulate circular tapes with high accuracy. The Iterative Closest Point (ICP) algorithm for circular tapes enabled the precise alignment of reference and target shapes, ensuring that the robotic manipulator could reliably pick up and handle objects. The system's ability to consistently perform, in a closed and controlled environment, object-picking tasks without significant errors was a key achievement, showcasing its potential for real-world applications. The system was particularly effective when dealing with simple, well-defined shapes such as circular tapes, where the ICP method excelled in minimizing misalignment and handling slight deformations.

In addition to circular tapes, the Principal Component Analysis PCA-based approach for markers and pipes showed promising results. PCA was able to compute the orientation and barycenters of these objects with good precision, which is crucial for correctly identifying the spatial position and orientation of non-circular objects. However, the PCA method's performance was more sensitive to variations in object orientation and occlusion. Despite these challenges, the results indicated that PCA could be a valuable tool for handling more complex shapes, but it requires further validation and refinement to improve its reliability and robustness, particularly in dynamic or cluttered environments.

The combination of ICP and PCA methods proved to be beneficial, as they leveraged each other's strengths. While ICP provided precise alignment for circular objects, PCA contributed to accurate orientation and position estimation for markers and pipes. Integrating these approaches into a unified system could significantly enhance the overall robustness and flexibility of the object-picking pipeline, allowing the system to handle a wider range of object types and complex scenarios.

5.2 Challenges

Several challenges emerged throughout the project, highlighting areas for further development. One of the primary challenges was the calibration process, which required multiple iterations to address alignment errors.

5.2.1 Repeated Calibration

Frequent recalibration was necessary due to the sensitivity of the system to camera displacements and changes in intrinsic parameters. One of the primary challenges was the calibration process, which required multiple iterations to address alignment errors. These errors were primarily caused by slight changes in the camera's focus and aperture, resulting in misalignment between the reference and target shapes. While the system performed well when the camera was stable, even minor shifts in the camera position led to noticeable deviations in object detection and manipulation. This iterative calibration process, although time-consuming, was essential for maintaining alignment accuracy, as small misalignments could significantly affect the system's overall performance. This challenge underscored the importance of more robust calibration techniques that could account for minor positional shifts and eliminate the need for frequent manual adjustments.

5.2.2 Lighting Sensitivity

Another significant challenge was the effect of varying lighting conditions on segmentation and object detection. The system's reliance on thresholding for object isolation was heavily influenced by changes in lighting intensity and shadowing. In low-light conditions or environments with uneven lighting, the object detection process became less reliable, leading to misclassifications and missed detections. The sensitivity of the thresholding algorithm to lighting variations pointed to the need for more adaptive and robust algorithms that can adjust to different lighting conditions automatically. Additionally, environments with strong contrasts or fluctuating light levels could cause significant difficulties in object segmentation. These challenges emphasized the necessity for algorithms that can handle real-time variations in ambient lighting, which is critical for deploying the system in dynamic, unpredictable environments without requiring manual recalibration or fine-tuning.

5.2.3 Object Orientation and Occlusion

The integration of PCA for markers and pipes revealed its limitations when dealing with tilted or partially obscured objects. Although PCA performed well under ideal conditions, its accuracy sometimes decreased when objects were rotated or blocked by other items. In such cases, the algorithm struggled to detect and calculate the correct orientation, leading to errors in manipulation. This limitation highlighted the need for further improvements in object orientation detection. Future work could incorporate more advanced algorithms that can handle partial occlusion or more complex orientations, potentially integrating additional sensors or utilizing more sophisticated image processing techniques to improve detection robustness. Developing solutions that can address the challenges posed by occluded or partially obscured objects will be crucial for expanding the system's ability to handle a wider range of scenarios.

5.3 Enhanced Observations

The combination of ICP and PCA methods provided complementary capabilities that enhanced the overall performance of the object-picking system. ICP's precision in aligning circular objects was paired with PCA's ability to determine the orientation and position of markers and pipes. This synergy allowed the system to address a broader range of object types, but further refinement is needed to ensure that the system can seamlessly switch between the two methods depending on the object type or scenario.

Visualization tools played a critical role throughout the project by enabling the debugging and validation of the calibration processes. These tools allowed the team to monitor real-time feedback on the alignment of the reference and target shapes, making it easier to identify errors and adjust parameters accordingly. The ability to visualize object detection, alignment, and orientation estimates in real-time streamlined the workflow, allowing for quicker adjustments and improved accuracy during testing.

The development of these visualization tools also contributed to the efficient validation of the overall system performance. By providing visual representations of the objects, their positions, and orientations, these tools helped identify inconsistencies in the detection and manipulation pipeline. They proved invaluable in verifying the results of the ICP and PCA algorithms, ensuring that the system was functioning as intended and that any issues could be addressed promptly.

Overall, the observations from these experiments emphasized the importance of integrating different algorithms and tools to tackle the challenges inherent in object detection and manipulation. The combination of ICP and PCA methods, coupled with the use of effective visualization tools, laid the foundation for a robust and adaptable system. However, ongoing work will be required to further optimize the system's performance, particularly in terms of adaptive thresholding, multi-object detection, and handling dynamic lighting conditions.

6 Conclusions and Acknowledgments

6.1 Broader Implications

The results of this project highlight the transformative potential of vision-guided robotic systems in automating repetitive tasks across various industries. These systems offer a scalable solution to tasks that demand high pre-

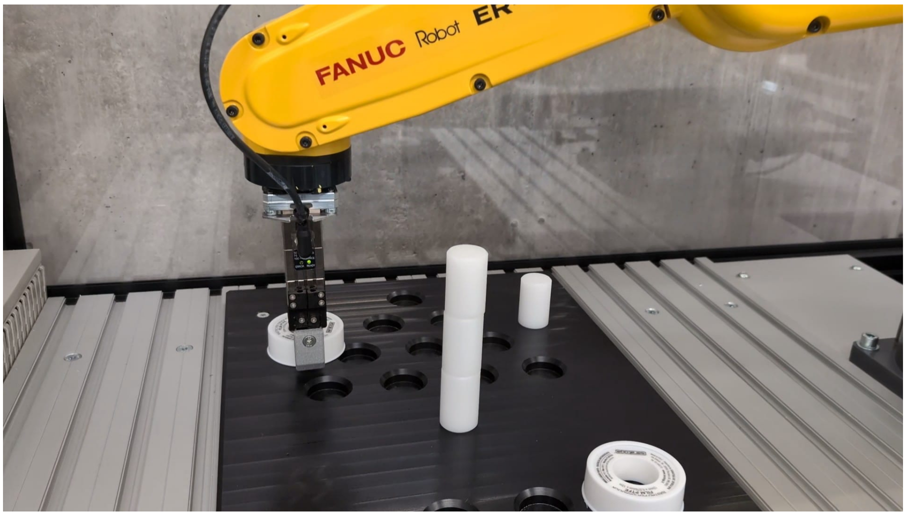

Figure 1: Tape picking

cision and consistency, such as assembly lines, quality control in manufacturing, and sorting operations in logistics.

6.1.1 Industrial Relevance

The integration of robotic control with computer vision enhances productivity by reducing human intervention in monotonous and error-prone tasks. This advancement is particularly beneficial in environments where safety is critical, such as hazardous material handling or medical device assembly. Additionally, the system's ability to adapt to different objects and conditions paves the way for its application in dynamic industries, such as e-commerce fulfillment centers and healthcare robotics.

6.1.2 Challenges to Scalability

Despite the promising results, scaling such systems to real-world applications demands addressing persistent challenges. Precision calibration and robust detection algorithms are prerequisites for reliable performance. Moreover, sensitivity to environmental factors such as lighting variations and occlusions requires advanced adaptability, both algorithmically and in terms of hardware design. These limitations highlight areas that need focused research and development to ensure the scalability and dependability of such systems in diverse operational settings.

6.2 Recommendations for Future Iterations

The insights gained during this project lay a strong foundation for future advancements. Several recommendations for further development are outlined below:

6.2.1 Extending Object Variety

Future work should test the ICP and PCA approaches on a broader range of objects, including those with irregular shapes, varying textures, and complex orientations. This evaluation will provide deeper insights into the generalization capabilities of the pipeline and help refine its adaptability to real-world applications. In industrial applications, techniques like YOLO, PointNet/PointNet++ and CNNs are frequently employed to ensure robust object detection and manipulation under varying conditions.

6.2.2 Real-Time Error Correction

Incorporating real-time error correction mechanisms into the robotic control system could significantly enhance its performance. For instance, dynamic recalibration techniques and feedback-driven adjustments during object manipulation would mitigate the effects of alignment errors and environmental changes. Machine learning models trained on error patterns could be integrated to predict and correct inaccuracies in object detection and manipulation on the fly.

6.2.3 Improving Algorithm Robustness

Adaptive algorithms capable of handling lighting variations and partial occlusions should be a key focus for future iterations. Techniques such as adaptive thresholding, deep learning-based object detection, and multi-modal sensing (e.g., combining vision with depth sensors) can bolster the system's robustness. Additionally, exploring sensor fusion strategies could improve reliability in challenging environments, such as those with dynamic lighting or complex backgrounds.

6.3 Conclusion

This project successfully integrated computer vision and robotic control to develop a vision-guided pick-and-place system. The pipeline demonstrated the feasibility of combining ICP-based alignment, PCA-based pose estimation, and precise calibration techniques to achieve accurate manipulation tasks. The complementary strengths of ICP and PCA were effectively leveraged to handle circular tapes and markers or pipes, respectively, laying the groundwork for further exploration of combined methods.

While the system's capabilities were validated in controlled environments, further work is required to enhance its adaptability and robustness. Future iterations should focus on extending object diversity, implementing real-time error correction mechanisms, and improving algorithmic resilience to environmental variations. These advancements will enable the system to handle dynamic and unpredictable scenarios, making it a viable solution for a wide range of industrial and research applications.

6.4 Acknowledgments

We extend our sincere gratitude to our mentor and instructor, Dr. Stefano Baraldo, Senior Researcher at the Institute for Systems and Technologies for Sustainable Production, for his invaluable guidance, insights, and support. His expertise was instrumental in helping us overcome the challenges encountered during this project and in refining our approach. We also acknowledge the resources and facilities provided by our institution, which greatly contributed to the successful execution of this work.

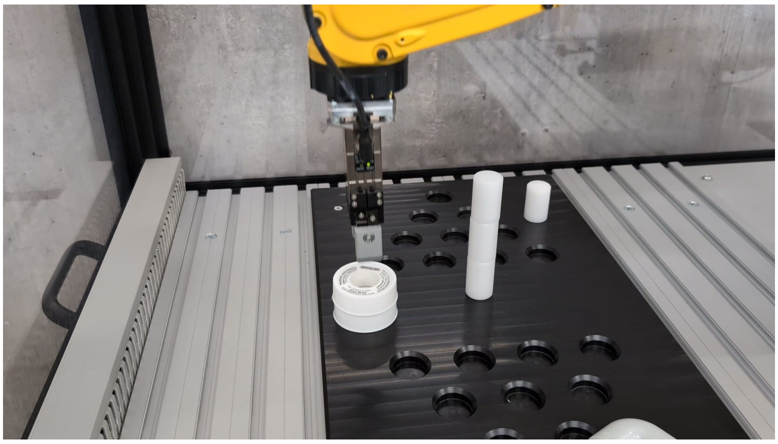

Figure 2: Tape stack