VISION GUIDED 6-AXIS ROBOTIC ARM FOR INSPECTION ON A CONVEYOR LINE

Universidad de Oviedo

Bibek Gupta

bibek.gupta@eu4m.eu

Supervisors:
Juan Carlos Álvarez Álvarez, UNIOVI, juan@uniovi.es
Miguel Ángel José Prieto, UNIOVI, mike@uniovi.es

Erasmus Mundus Master in Mechatronic Engineering

Abstract

Vision-guided robotic systems enhance industrial automation by integrating advanced imaging systems with robotic precision, allowing robots to perform tasks like assembly, inspection, and material handling with high accuracy. This work presents an industrial setup for inspection and rejection of objects on a conveyor line. It explains the stepwise calibration of an Intel RealSense L515 RGBD camera and Hand to Eye Calibration of a UR3E robot and accuracy validation based on practical tests. It includes the design and integration of a conveyor system with the robot and camera-gripper assembly to perform pick and place operation based on Image (2D) to World (3D) coordinate transformations and various image processing techniques. Three different conveyor inspection scenarios are replicated to detect the objects on the basis of size, color and deficient bolts and integrated on a single program with online visualization of results.

Resumen

Los sistemas robóticos guiados por visión mejoran la automatización industrial al integrar sistemas de imagen avanzados con precisión robótica, permitiendo tareas como ensamblaje, inspección y manipulación de materiales con alta exactitud. Este trabajo presenta una configuración industrial para la inspección y rechazo de objetos en una cinta transportadora. Se explica la calibración paso a paso de una cámara RGBD Intel RealSense L515 y la calibración mano-ojo de un robot UR3E, validando la precisión con pruebas prácticas. Se incluye el diseño e integración de un sistema de transporte con un ensamblaje de robot y cámara-pinza para operaciones de recogida y colocación, basadas en transformaciones de coordenadas de imagen (2D) a mundo (3D) y técnicas de procesamiento de imágenes. Tres escenarios de inspección se replican para detectar objetos según tamaño, color y tornillos defectuosos, integrados en un único programa con visualización en línea de resultados..

Keywords: Universal Robot, RoboDK, Python, Object Detection, Computer Vision, Quality Control

1. Background and Objectives

Without robotic vision, robots operate as blind machines that follow their programming. They rigidly adhere to the predefined code that dictates their functions, making them ideal for repetitive tasks. With the arrival of Industry 4.0, robots are evolving too, and vision is a key feature of this evolution, introducing new levels of precision and accuracy in smart automated process.

This work is conducted in the "Grupo de Investigación SiMuR" laboratory of Universidad de Oviedo. This work aims to upgrade the previous work done on pick and place applications on the UR3E robot. In a Vision Guided Robotic System, at least one camera will be mounted on the robotic arm, literally serving as the eye of the machine. The robot camera will take 2D or 3D scans of the object (Fig.1). The image will then be stored in the robot database and programmed to trigger the machine to move and perform specific tasks.

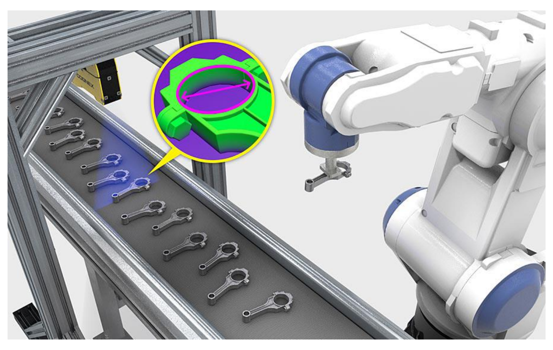

Fig. 1. Vision Robot used for Object Inspection

The main objectives of the work was to develop and integrate a vision-guided robotic system for the purpose of inspection and classification. To achieve it, a conveyor prototype is designed and integrated with the robot and image processing algorithms were applied to detect the defective object and reject it from the moving conveyor line.

2. Camera Intrinsic and Extrinsic Calibration

Accurate calibration ensures precise measurements and reliable analysis by correcting distortions and estimating intrinsic and extrinsic camera parameters.

Camera Intrinsics:

The process of estimating the parameters of the camera model. This includes the focal length coordinates (fx, fy), principal axis offset (Cx, Cy), distortion coefficients and skewness.

The calibration is done using a 9X6 chessboard pattern and taking images of the chessboard in various angles keeping the camera at a fixed position. The calibration accuracy is found by reprojecting the pattern on the sample images Fig.2 and computing the Reprojection error which was found to be 0.12.

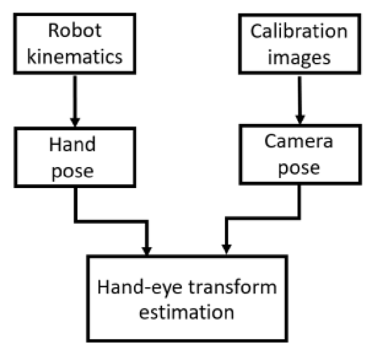

Fig. 3. Hand-Eye Calibration Flow

Camera Extrinsics:

It represents the pose of camera eye from the robot tool flange. The task of computing the relative 3D position and orientation between the camera and the robot hand in an eye on-hand configuration, where the camera is rigidly attached to the robot hand, is known as hand-eye calibration.

Robot kinematics provide the hand pose, while calibration images determine the camera pose. The hand-eye transform estimation combines these poses Fig. 3 to create a transformation matrix that aligns the coordinate system of the robot's hand with that of the camera.

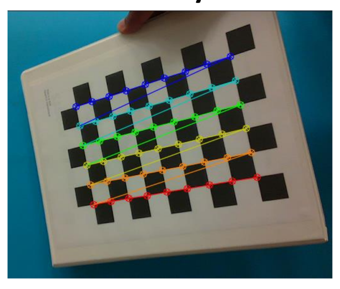

Fig. 2. Sample Image (Drawn Chessboard pattern)

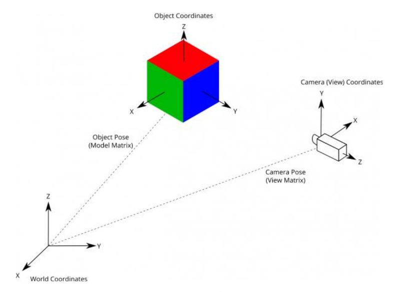

Fig. 4. Coordinate System

2. Methodology and Implementation

System Integration:

The Fig. 5 depicts the hardware integration of the entire setup. The UR3e robot, equipped with a gripper via a Lumberg RKMV 8-354 connection and interfacing with a PC over TCP/IP, controls the gripper's movements and the relay that powers the 220V AC conveyor. The PC runs a main program in the SPYDER Python IDE, integrating with RoboDK for simulation and the Intel RealSense L515 camera for real-time visual feedback via USB 3.2. The UR3e driver facilitates communication between the robot and the main program, managing the robot's pose and digital I/O signals to operate the gripper and conveyor.

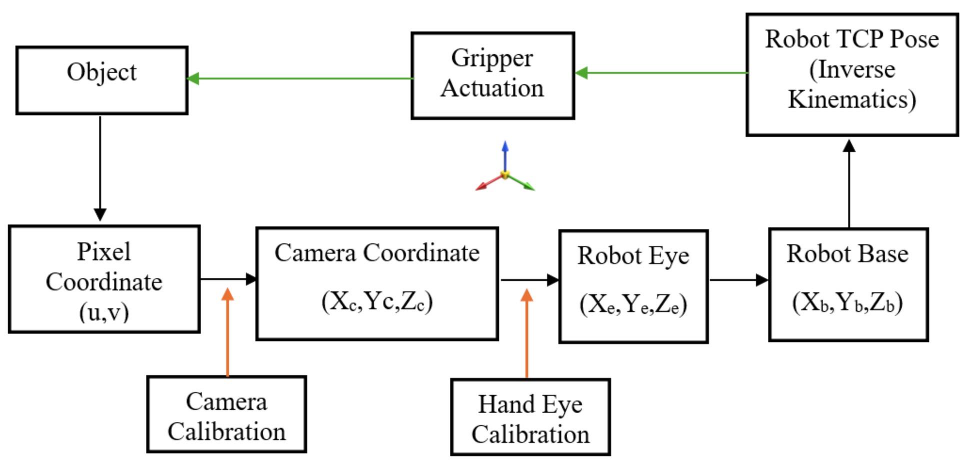

Fig. 5. System Working Architecture

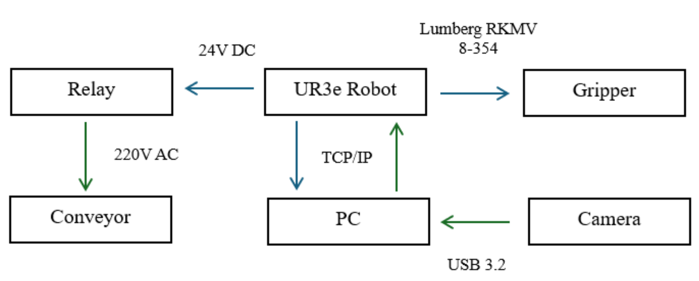

Fig. 6. System Integration

System Working Architecture:

The flowchart outlines the system design for a vision-guided robotic setup, detailing the process from object detection to robotic actuation. Initially, the object is detected, and its position is identified in pixel coordinates (u,v). Through camera calibration, these pixel coordinates are converted to camera coordinates (Xc, Yc, Zc). Hand-eye calibration then transforms the camera coordinates into robot eye coordinates (Xe, Ye, Ze). These coordinates are subsequently mapped to the robot base coordinates (Xb, Yb, Zb). Using inverse kinematics, the robot base coordinates are utilized to calculate the robot's tool center point (TCP) pose. Finally, the robot moves to the calculated TCP pose and actuates the gripper to interact with the object, completing the task.

System Implementation:

The Fig. x shows the digital twin model of the setup assembly developed in RoboDK station. The different targets are defined, and corresponding robot movement programs are created which are accessed by the RoboDK API via Python IDE. The conveyor and gripper actuation programs are controlled via the Digital I/O terminals from the robot controller panel. Fig, x shows the actual setup of the Inspection system with different rejection points for different scenarios.

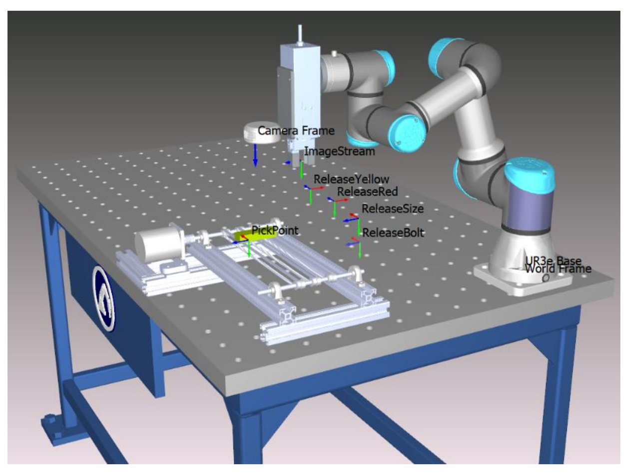

Fig. 7. Digital Twin Setup

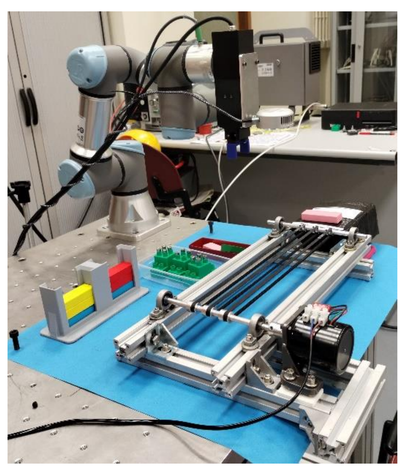

Fig. 8. Actual Setup

3. Results and Discussion

Scenario 1: Detection Based on Colored Objects

In this case Fig.8, the algorithm is developed to detect red/yellow-colored objects and perform pick–place operation to the rejection bin, using different Image processing techniques. The Camera streams the images at 30 fps and a detection window is created on the conveyor line capturing only relevant information for efficient computation.

Color Masking is used to extract the required objects profile from the background. Next Thresholding is applied to the image and filtered to prepare it for the edge detection step. The Contours are filtered out and centroids are calculated the color definition criteria as well which is defined based on average RGB values.

The centroids are pixels coordinates which are passed to the coordinate transformation calculations, and the robot TCP is moved to the object centroid and grasped to remove it from the conveyor.

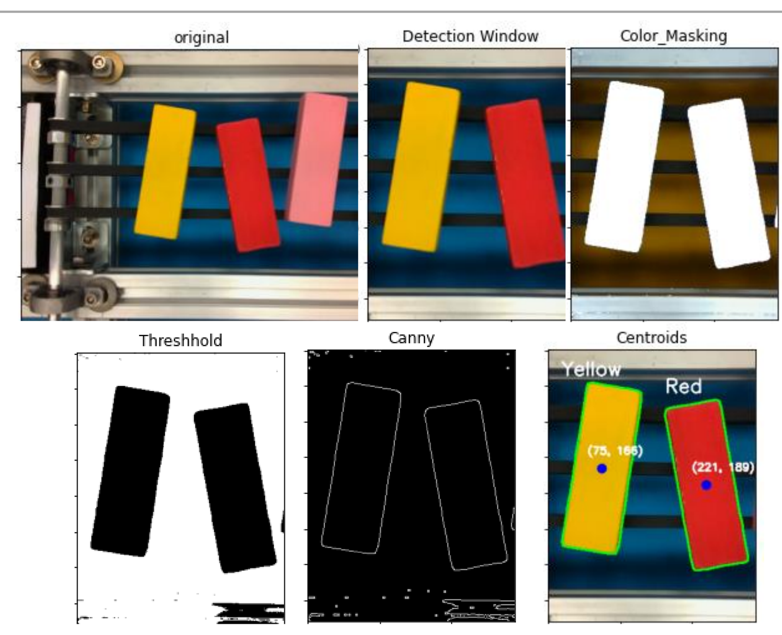

Fig. 9. : Detection Based on Colored Objects

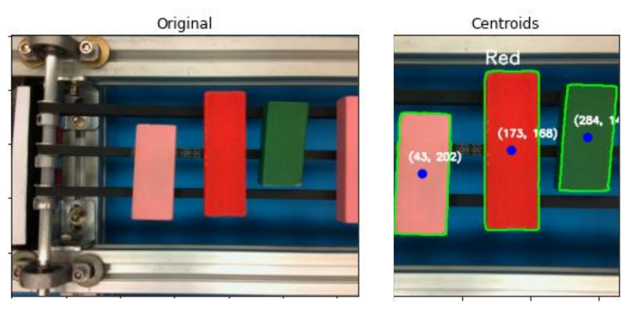

Fig. 10. : Detection Based on Object Sizes

Scenario 2: Detection Based on Object Sizes

Fig. 9 shows the detection of the objects and its centroids. The program rejects the objects which are less than the standard size based on the extracted contour area.

Scenario 2: Detection Based on Deficient Bolts

In this scenario, an algorithm is developed to detect the object and the bolts fitted to the object. The standard object contains four bolts, in case of deficient bolts or no bolts found in the object the robot shall reject the defective object from the conveyor line.

The object is detected with the same algorithm as previously explained. After the object profile is extracted, bounding line is drawn and centroid values found Fig. 10. Next, the object passes to additional processing to detect the bolts.

Here the Region of Interest (ROI) is extended, and a color mask is applied to extract only the silver bolts from the object. Next the bolt contours are approximated based on the radius of bolts and drawn over the image and rejection criteria defined. Histogram Equalization and Median Blurring techniques were implied to preserve the edges and remove noise for accurate detection.

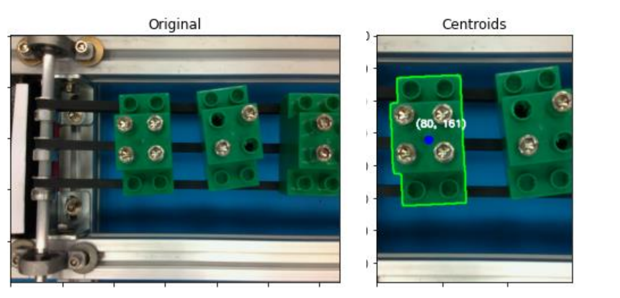

Fig. 11. Detection of the Bolt-fitted Objects

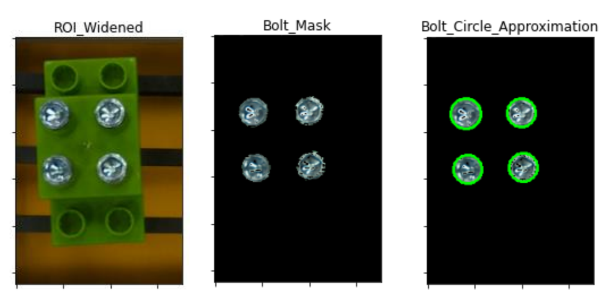

Fig. 12. Detection of the Bolts on the Object

4. Conclusions & Future Directions

1. Successful integration of Robot, Gripper-Camera assembly and Conveyor with Vision algorithm; Camera Intrinsic and Extrinsic Calibration accuracy in Pick and place operation analyzed.
2. Successful Implementation of advanced image processing to accurately detect objects (based on color, size and deficient bolts) and extract centroids to perform pick and place operation.
3. To Implement of Eye-Hand Camera setup with different gripper design (soft, vacuum, pneumatic) to explore pick and place operation with wide variety of objects.
4. To use Machine Learning to enhance the adaptability to perform complex object detection with conveyor tracking (using encoder) which shall improve system flexibility and efficiency.

References

1. R. API, “"RoboDK API for Python".” https://robodk.com/doc/en/PythonAPI/index.html, 2024.
2. I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, “A comparative review of hand-eye calibration techniques for vision guided robots,” IEEE Access, vol. 9, pp. 113143–113155, 2021.
3. Peter Corke, “Robotics, Vision and Control: Fundamental Algorithms in Python”, Springer, 2023
4. U. Robots, “"Universal Robots e-Series User Manual Version 5.0.2.“, User_Manual_en_Global.pdf, 2023

Acknowledgements

I would like to thank Prof . Juan Carlos Álvarez Álvarez Grupo de Investigación SiMuR, at the Universidad de Oviedo, for granting the opportunity to carry out the Thesis of my interest and access the lab and equipments freely. Would also like to extend my thanks to Prof. Miguel Ángel José Prieto and Prof. José Manuel Sierra Velasco whose contributions were significant in design and fabrication of conveyor.