ISSN:2790-1688 Volume-15-(2025)

Design and Implementation of a Vision-Guided Soft Gripper Robotic System for Static and Dynamic Fruit Grasping

Yuhan Wang

The High School Attached to Northwest Normal University, Lanzhou, China

2328633089@qq.com

Abstract. This study develops a vision-guided robotic system for grasping both stationary and moving fruits using a custom-fabricated silicone soft pneumatic gripper. The design integrates monocular vision for object detection and localization, pixel-to-world calibration, and an "eye-in-hand" hand–eye transformation to enable accurate grasping. A multi-language modular architecture—Python for vision, MATLAB for calibration, and C++ for control—coordinates perception and actuation. Finite element analysis verified the gripper's predictable deformation under pressure, and real-time edge-based detection achieved consistent fruit localization in semi-structured settings. Experimental results demonstrate reliable grasp execution in static and dynamic conditions. While current limitations include fixed-height depth assumptions and open-loop control, future enhancements such as stereo vision, visual servoing, and improved gripper geometry are proposed to increase adaptability and precision. This work highlights the potential of combining soft robotics and computer vision for adaptive manipulation in agricultural and industrial automation.

Keywords: soft robotics, monocular vision, pneumatic gripper, robotic grasping, multi-language control.

1. Introduction

The increasing demand for automation in industries such as agriculture, logistics, and manufacturing has driven the development of intelligent robotic systems capable of interacting safely and effectively with a wide range of objects. Among the many tasks robotic systems are expected to perform, object grasping and manipulation remain challenging, particularly in scenarios where the target objects are deformable, irregularly shaped, fragile, or positioned dynamically in unstructured environments.

Traditional robotic grippers, typically composed of rigid materials and controlled through kinematically precise mechanisms, often struggle in such settings. Their lack of compliance can result in unstable grasps or even damage to the object. In contrast, soft robotic grippers offer a promising alternative [1]. Constructed from flexible materials such as silicone rubber and actuated pneumatically or hydraulically, these grippers can conform to object surfaces, distribute contact forces evenly, and accommodate shape variations without precise control or sensing. These characteristics make soft grippers particularly suitable for delicate tasks, such as fruits, food products, or medical instruments.

However, grasping objects effectively in real-world environments requires more than just physical compliance. Perception plays a critical role. A robotic system must be able to identify objects in its surroundings, determine their location and orientation, and plan appropriate motion trajectories for approach and grasp. Vision-based systems are often used to achieve this, particularly those based on monocular cameras due to their simplicity, low cost, and ease of integration. While monocular systems lack direct depth sensing, they can provide sufficient spatial information when paired with appropriate calibration and geometric modeling.

In this research, we designed and implemented a robotic system capable of vision-guided fruit grasping using a soft pneumatic gripper. The system handles static and dynamic scenarios—fruit placed on a stationary surface and a moving conveyor belt. Our system integrates mechanical design, visual perception, and robotic control into a cohesive pipeline. It combines a custom-fabricated soft gripper, a monocular camera mounted on the robotic arm, and a set of algorithms for image processing, coordinate transformation, and grasp execution. The mechanical subsystem includes the soft gripper,

fabricated from silicone rubber using a mold-based casting process. The gripper is pneumatically actuated and can adapt its shape to different fruit types and sizes. The vision subsystem employs a monocular camera to detect and localize the fruit using classical image processing techniques. Calibration procedures are carried out to map pixel coordinates to world coordinates, enabling accurate estimation of object positions. The control subsystem consists of a multi-language software architecture, with Python handling vision tasks, MATLAB for calibration preprocessing, and C++ managing robot control and integration.

In the following sections, we present details of our design and implementation process. It includes the mechanical design and simulation of the soft gripper, the calibration and vision algorithms used for object detection, the robotic control logic for grasp execution, and a discussion of future improvements. The work demonstrates the feasibility of integrating soft robotics and vision-based control into a functional robotic grasping system for semi-structured tasks.

2. Soft Gripper Design and Analysis

2.1 Design Principles and Fabrication

The soft gripper in this robotic system is designed to perform gentle yet stable grasping of delicate, irregularly shaped objects, such as fruits, under static and dynamic conditions. Drawing inspiration from biological structures, the gripper relies on pneumatic actuation to induce bending in its flexible silicone fingers. When pressurized air is introduced into the internal chamber of each finger, the actuator deforms asymmetrically, resulting in a predictable curling motion. This enables the gripper to adapt its shape to accommodate fruit size and contour variations, reducing the risk of slippage or damage during contact [2].

The fabrication of the soft actuator follows a mold-based casting process that ensures consistency, reliability, and scalability. First, a two-part mold is prepared using high-resolution 3D printing. A release agent is applied to the mold surface to prevent unwanted adhesion of the silicone material. The silicone elastomer is mixed in a 1:1 ratio and thoroughly degassed in a vacuum chamber to eliminate trapped air bubbles that could weaken the structure. The mixture is then injected into the mold using a syringe to allow precise filling of narrow geometries. After injection, the mold is placed in an oven and cured at elevated temperatures for several hours to accelerate polymerization and enhance mechanical integrity. Once cured, the actuator is demolded and bonded with a constraint layer on one side. This constraint, made from a stiffer elastomer or thermoplastic material, restricts expansion on one side and directs the bending motion during inflation. Additional elements, such as embedded mesh reinforcement or surface texturing, may be incorporated to improve durability and grip. Finally, the actuators are connected to flexible pneumatic tubing and mounted on a custom-designed gripper base, enabling quick assembly and replacement.

This fabrication approach results in lightweight, inexpensive actuators adaptable to different grasping tasks. More importantly, the inherent compliance of the soft fingers allows them to distribute contact forces evenly across the fruit surface, making them ideal for handling items that rigid grippers would otherwise damage.

2.2 Finite Element Analysis

To evaluate and predict the mechanical behavior of the soft actuators under varying input pressures, we conducted finite element simulations using ABAQUS. This analysis provides insight into stress distribution, deformation patterns, and the correlation between applied pressure and fingertip displacement, critical for ensuring controlled and repeatable grasping motions.

The soft silicone material used for the actuator was modeled as a Neo-Hookean hyperelastic material. It is a common choice for soft polymers because it can capture large, nonlinear deformations. The material parameters were set to a shear modulus of 184 kPa and a bulk modulus of 18.35 MPa. The constraint layer was modeled as a linear elastic material, with a Young's modulus of 19.47 MPa

and a Poisson's ratio of 0.495. The base of the actuator was fixed in all directions to simulate real-world mounting conditions.

Simulation results indicated that the actuator deformed predictably and repeatably under increasing pressure. The stress distribution was concentrated near the bending hinge. Still, it remained within the elastic limits of the material, ensuring that the actuator would not suffer permanent deformation or failure under repeated cycling. These simulations confirmed the actuator's suitability for dynamic fruit grasping tasks. Also, they guided future design iterations by highlighting regions of high strain that may benefit from reinforcement or geometric adjustment.

2.3 Design Improvements

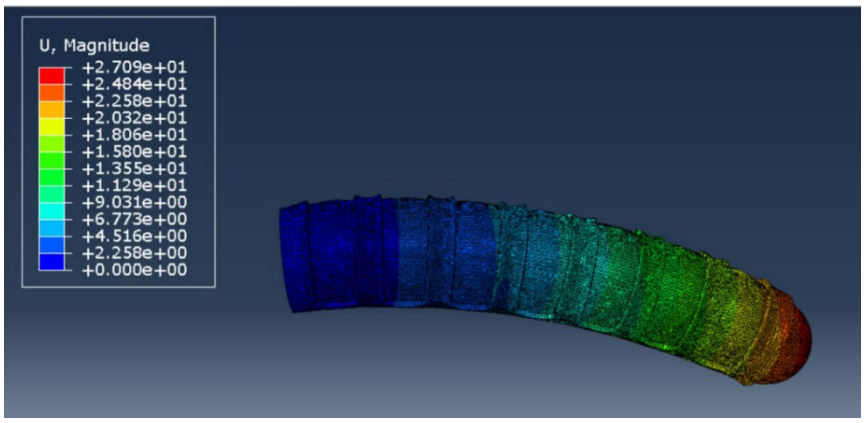

Figure 1. Simulation in ABAQUS

Although the gripper demonstrated satisfactory performance in preliminary trials, two primary limitations were identified through simulation insights and physical testing: limited contact area and structural instability during dynamic movements. The first issue concerns the relatively small contact surface between the gripper fingers and the fruit, which reduces grasp stability, especially for larger or asymmetrically shaped fruits. Currently, the gripper operates primarily in a point-contact or line-contact mode, which may lead to slippage or insufficient force distribution. To address this, future iterations will explore a shift toward surface-grasping configurations. This may include redesigning the finger geometry to wrap more fully around the object or adding soft, conformal materials, such as gel pads or suction-assist membranes, to the finger surfaces to increase friction and contact area. These enhancements are expected to reduce slippage and improve robustness, especially when handling objects with smooth or waxy exteriors.

The second challenge is related to structural precision and repeatability. During high-speed operations on a conveyor belt, the soft fingers occasionally exhibited unwanted vertical displacement or lateral torsion, which compromised grasp accuracy. This instability is partly due to the fully compliant structure of the actuator, which, while advantageous for safety and adaptability, lacks the stiffness needed for high-precision manipulation. To mitigate this issue, the integration of selectively stiffened regions is proposed. For instance, embedding rigid or semi-rigid supports at the base of the gripper can limit undesired motion without significantly sacrificing compliance at the fingertip.

3. Vision System and Algorithm Implementation

3.1 Camera Calibration

Camera calibration fundamentally involves computing the transformation from the 3D world coordinate system to the 2D pixel coordinate system. This process comprises several distinct but interconnected stages: (1) World to Camera Coordinate Transformation: This is modeled as a rigid-body transformation—i.e., a combination of rotation and translation—that maps points from the world frame to the camera frame. In the context of our robotic system, this transformation is determined by both the robot's end-effector pose and the fixed transformation from the end-effector

to the attached camera. (2) Camera to Image Coordinate Transformation: This step involves perspective projection, mapping the 3D coordinates in the camera frame onto a 2D image plane based on the pinhole camera model. (3) Image to Pixel Coordinate Transformation: Finally, a 2D affine transformation is applied, accounting for scaling and translation from image coordinates (in physical units) to pixel coordinates on the camera sensor.

Combining those transformations, we can achieve a complete transformation. Here, the first two matrices represent the intrinsic parameters, while the latter represent the extrinsic ones. The inherent parameters describe the internal optical characteristics of the camera, including the focal length, principal point, skew coefficient, and lens distortion. These parameters govern how the camera transforms points into image coordinates in its 3D coordinate frame. On the other hand, the extrinsic parameters define the position and orientation of the camera relative to the world coordinate system. Together, these components enable accurate geometric modeling of the image formation process.

$$Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dX} & -\frac{\cot\theta}{dX} & u_0 \\ 0 & \frac{1}{dY\sin\ \theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} \tag{1}$$

Meanwhile, the lens causes image distortion due to manufacturing and assembly accuracy, and it needs to be corrected by calibration methods. Here, r represents the distance between the current pixel and the center of the image.

$$\begin{aligned} x_{\text{corrected}} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{\text{corrected}} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \tag{2}$$

To perform calibration, we adopted the widely used Zhang Zhengyou method [3], which uses multiple images of a planar checkerboard captured from various orientations. In our setup, we used a checkerboard pattern with 11 × 8 inner corners and square sizes of 9.55 mm. At least ten images were taken from different viewpoints. Sub-pixel corner detection algorithms were applied to extract 2D image points with high accuracy. The homography matrix for each image was computed to relate 3D world coordinates on the planar checkerboard to 2D image coordinates.

Using these correspondences, we solved for the intrinsic matrix through linear estimation methods. Then we refined the solution using nonlinear optimization, specifically the Levenberg–Marquardt algorithm [4], to minimize the total reprojection error. Additionally, radial and tangential lens distortions were estimated and integrated into the final camera model to correct image warping effects. This entire calibration process was implemented using MATLAB's built-in computer vision toolbox, and the resulting parameters enabled reliable projection and reconstruction of object positions in world space.

3.2 Hand-Eye Calibration

In our robotic system, the camera is mounted directly onto the robot's end-effector, forming an "eye-in-hand" configuration. This setup requires an additional calibration to determine the spatial transformation between the camera frame and the robot's end-effector frame. This transformation is essential to ensure that fruit locations identified by the vision system can be accurately translated into robot movement commands.

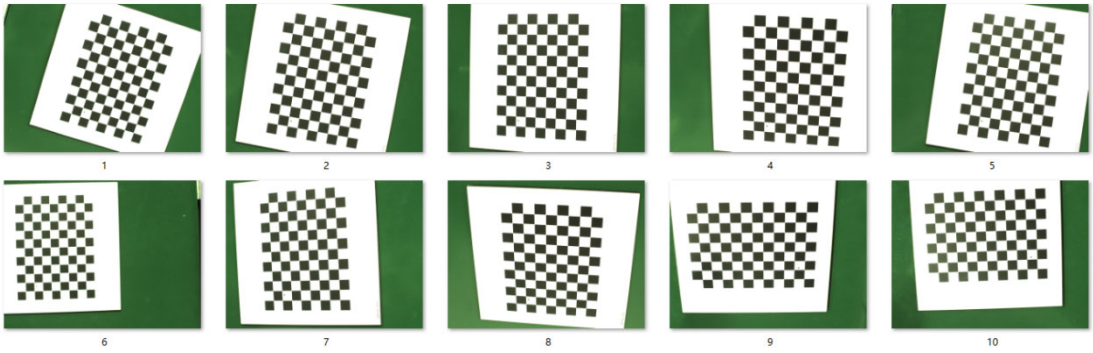

Figure 2. Images of Planar Checkerboard

We used the Tsai–Lenz algorithm [5], a well-established hand-eye calibration solution, to solve this. The calibration procedure involved moving the robot through diverse poses and capturing the transformation matrices from the robot base to the end-effector (A) and from the camera to the calibration target (B).

The algorithm was implemented using OpenCV's calibration module, which provided numerical optimization tools to solve the transformation equations. The resulting matrix was verified for consistency, and later transformed pixel-derived object locations into the robot's coordinate system. This transformation is critical for ensuring spatial coherence between perception and actuation in all subsequent grasping tasks.

3.3 Object Detection via Edge Detection

The vision module's goal is to accurately detect the position of fruits within the camera's field of view. Given the consistent appearance of the conveyor belt background (green) and the distinct coloration of the fruits, we implemented a classical edge-based image segmentation method that avoids the complexity of machine learning models while maintaining real-time performance.

The process begins by converting the RGB image to grayscale to reduce the computational burden and remove color-based noise. Gaussian blurring [6] is applied to smooth the image and suppress high-frequency variations caused by texture or lighting. Binary thresholding separates the object of interest from the background, with a fixed threshold value of 135 determined through empirical tuning. After thresholding, a morphological opening operation—consisting of erosion followed by dilation—is performed to eliminate minor noise artifacts and refine the edges of the detected blob.

The largest connected region is extracted from the cleaned binary image, and its centroid is computed using image moment calculations. The resulting pixel coordinates represent the estimated center of the fruit in the image plane. This entire pipeline was implemented using OpenCV in Python, enabling lightweight and robust performance suitable for real-time applications. The algorithm performed reliably under the experimental conditions, offering consistent detection for stationary and moving fruit objects.

3.4 Coordinate Reconstruction

Once the pixel coordinates of the fruit have been identified, the next step involves reconstructing their position in the robot's 3D workspace. This is achieved through geometric back-projection using the intrinsic matrix obtained from camera calibration and the transformation matrices obtained through hand-eye calibration.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times s \tag{3}$$

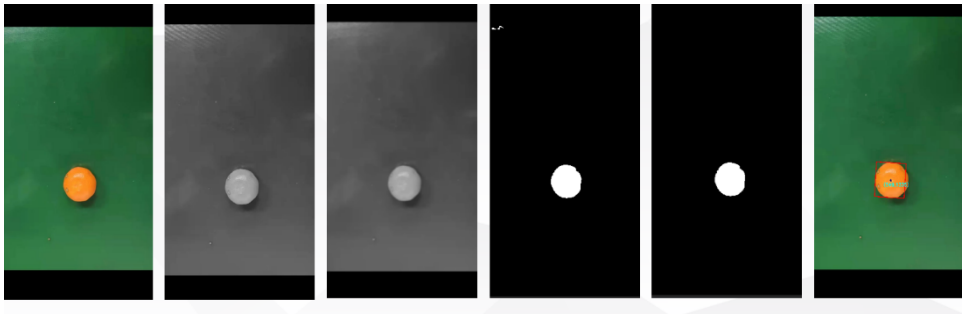

Figure 3. Process of Object Detection

Where K is the intrinsic matrix, and s is a manually adjusted scaling factor accounting for known camera-to-object height. Due to the limitations of monocular vision, direct depth information is not available. To resolve this, we introduced an approximation based on the known vertical distance between the camera and the working surface. By assuming a fixed height for the fruits on the table or conveyor belt, we could recover the X and Y coordinates using inverse projection formulas. A manually tuned scale factor was also introduced to account for slight deviations in the camera height and perspective distortion. This scale factor was empirically adjusted through test experiments to minimize spatial errors. Although this reconstruction approach is limited in depth adaptability, it is sufficient for grasping tasks in controlled environments with consistent surface heights.

3.5 Future Vision Improvements

Despite the current vision system's success in constrained environments, several limitations restrict its performance in more dynamic or unstructured conditions. The reliance on fixed-height assumptions limits the adaptability of the coordinate reconstruction process. At the same time, the simplicity of the detection algorithm restricts its robustness to background variations or changes in lighting.

To address these issues, we propose replacing the monocular vision system with a stereo vision setup in future work. A stereo configuration would provide direct depth estimation, eliminating the need for fixed-height assumptions and significantly improving localization accuracy. In addition, more advanced object recognition methods, including shape- and color-based classification, could be incorporated to enable multi-class fruit sorting or ripeness estimation.

Real-time tracking algorithms such as Kalman filters [7] could also be added to improve performance under dynamic conditions. These algorithms would allow the robot to continuously track moving fruits on the conveyor and predict their trajectories, enabling interception and grasp planning with reduced latency. These upgrades would bring the system closer to robust deployment in real-world agricultural or industrial automation scenarios.

4. Grasping Control Logic

4.1 Robotic Arm Control Strategy

The robotic control logic is at the core of enabling reliable, autonomous fruit grasping in static and dynamic scenarios. The control system was designed as a state-driven loop that integrates real-time perception with responsive actuation, allowing the robotic manipulator to execute a complete grasping cycle with minimal human intervention. The detailed procedure includes initialization, vision invocation, grasp execution, and looping.

This control logic was designed with robustness in mind. It tolerates detection errors and accounts for fruit position variations through empirical correction factors applied during grasp planning. Although the system operates open loop, its repeatability and reliability under controlled conditions have been experimentally validated. Using a soft gripper introduces an additional tolerance layer,

allowing minor positional errors to be absorbed mechanically without damaging the fruit or compromising grasp success.

4.2 Cross-Language Modular Integration

The system was implemented using a multi-language architecture to improve development efficiency and modularity. The vision module was written in Python due to its flexibility and powerful image-processing libraries. MATLAB was used for camera calibration, leveraging its robust numerical tools. C++ handled the control core, providing real-time performance and direct access to robotic hardware.

These modules communicate through structured interfaces. The C++ program calls Python scripts via the Python C API to retrieve fruit coordinates, which are then converted to robot commands. MATLAB calibration results are saved as configuration files and loaded by the C++ controller at runtime. This modular structure allowed each part of the system to be developed independently, facilitating debugging, scalability, and future integration of new functions.

4.3 Proposed Control Enhancements

The current control strategy operates in open-loop mode with empirically tuned offsets. While effective in structured environments, it lacks adaptability in dynamic or unpredictable scenarios. To address this, future improvements will include closed-loop visual servoing for real-time correction and more precise positioning [8].

In addition, implementing predictive grasping based on motion tracking could improve success rates on moving targets. Task-level intelligence, such as object classification and priority-based grasp planning, will also be explored to increase system autonomy and decision-making capacity in complex environments.

5. Conclusion

This work presents the integration of a vision-guided robotic system capable of grasping both static and dynamic fruits using a soft pneumatic gripper. The system effectively demonstrates adaptive object manipulation in semi-structured environments by combining soft actuator fabrication, monocular vision, hand-eye calibration, and multi-language modular control. The soft gripper, designed and simulated using hyperelastic material models, showed reliable bending behavior and compliant grasping of deformable targets. Monocular camera calibration and hand-eye transformation enabled accurate object localization, while real-time object detection based on classical image processing ensured effective visual feedback for grasp execution. The state-driven control logic and C++–Python–MATLAB integration allowed for reliable coordination of sensing and actuation.

Despite its success, the system has limitations, including reliance on fixed-height assumptions for depth estimation and open-loop control without real-time correction. Future work will incorporate stereo vision, visual servoing, and improved gripper structures to enhance adaptability and accuracy. Overall, this work illustrates the potential of combining soft robotics with vision-based control in practical robotic applications and provides a foundation for further exploration in adaptive, vision-guided manipulation.

References

[1] Ilievski F, Mazzeo A D, Shepherd R F, et al. Soft robotics for chemists[J]. 2011.

[2] Liu C H, Chung F M, Chen Y, et al. Optimal design of a motor-driven three-finger soft robotic gripper[J]. IEEE/ASME Transactions On Mechatronics, 2020, 25(4): 1830-1840.

[3] Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on pattern analysis and machine intelligence, 2002, 22(11): 1330-1334.

[4] Moré J J. The Levenberg-Marquardt algorithm: implementation and theory[C]//Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 105-116.

[5] Tsai R Y, Lenz R K. Real-time versatile robotics hand/eye calibration using 3D machine vision[C]//Proceedings. 1988 IEEE International Conference on Robotics and Automation. IEEE, 1988: 554-561.

[6] Gedraite E S, Hadad M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation[C]//Proceedings ELMAR-2011. IEEE, 2011: 393-396.

[7] Li Q, Li R, Ji K, et al. Kalman filter and its application[C]//2015 8th international conference on intelligent networks and intelligent systems (ICINIS). IEEE, 2015: 74-77.

[8] Hutchinson S, Hager G D, Corke P I. A tutorial on visual servo control[J]. IEEE transactions on robotics and automation, 2002, 12(5): 651-670