Course Code: AR514
Course Title: Vision and Learning Based Control
IBVS system design for drone operations

Jagadeesh Rachapudi, Satwik Jaiswal, Yogesh namdeo Dasgaonkar, Ankit Mehra

***Abstract*—This paper presents the design and implementation of a new IBVS algorithm for drone operations. The system is engineered to execute crucial tasks including search, hover, landing, and tracking autonomously. The primary aim is to perform some basic drone operations using IBVS technology. The proposed IBVS system utilizes visual feedback from drone cameras to navigate in the given environment. Our main objectives are, firstly, to enable the drone to autonomously search for a moving vehicle and hover over it, and secondly, to follow the ground vehicle. The system is demonstrated through a simulation environment.**

***Index Terms*—Image-Based Visual Servoing, Drone Operations, Autonomous Navigation, Target Tracking, Landing Control, Simulations and Modeling**

I. Introduction

Drones have become important tools across multiple industries, offering a range of applications from spying to delivering products. However, for drones to perform tasks like search, hovering, landing, and tracking, they need to be smart. Traditional methods often struggle in dynamic environments, where real-time adjustments are very much needed. Enter Image-Based Visual Servoing (IBVS), a solution that uses camera feedback to navigate drones.

This paper introduces an IBVS system made specifically for drone operations. Our goal is to enable drones to autonomously perform the above tasks. By introducing IBVS and control algorithms, we aim to improve drones' ability to navigate and perform crucial tasks in new environments. Through simulations, we'll show how our IBVS system improves drone performance.

II. Literature Review

TABLE I: Summary of the state-of-the-art

Existing state-of-the-art approach	***Limitations***
IBVS	The primary challenges with Image-Based Visual Servoing (IBVS) include sensitivity to different environmental changes, low performance in the presence of image noise, complex algorithms, and issues with low light conditions.
PBVS	The challenges with Position-Based Visual Servoing (PBVS) arise when geometrically complex objects are present in the environment and fail to perform well due to its complex algorithms.

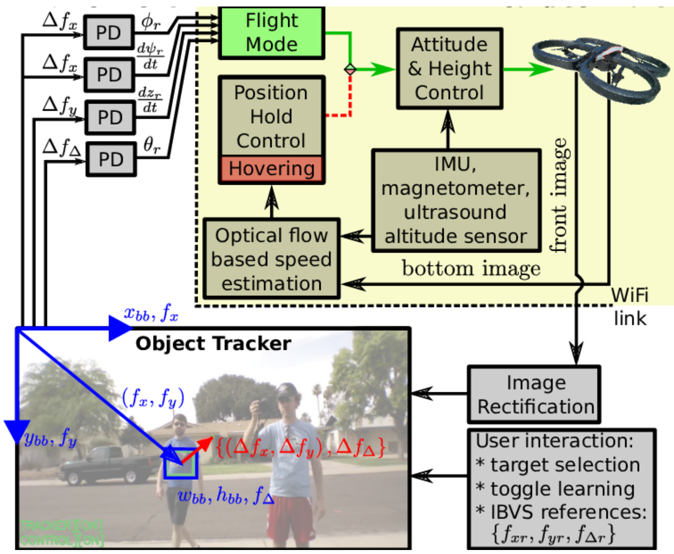

Fig. 1: This figure shows how Image-based visual servoing works.

A. Problem Statement

To design an IBVS system for drones that can perform the operations: search, hover, and tracking.

Assumptions:

1) We assume that every object has an ArUco marker attached to it.
2) We assume that the drone will know the target ArUco ID prior to starting the search operation.
3) We assume that there are no obstacles in the environment other than the drone and the object.

Constraint - This system has the following constraints:

1) This system only functions effectively when each object in the environment is equipped with an ArUco marker for searching and tracking purposes.
2) The search operation of our system requires prior knowledge of the ArUco ID associated with the target.
3) Our system's performance is optimized when the environment contains no extraneous objects other than the drone and the target object.

B. The Proposed Approach

The proposed solution aims to address the challenges associated with designing an Image-Based Visual Servoing (IBVS) system designed for drone operations. By making IBVS technology, the system enables the autonomy and operational capabilities of drones, enabling them to perform tasks such as search, hover, landing, and tracking.

Visual feedback processing is a important component of the proposed solution. This involves the detection and tracking of the ground vehicle in the environment. By accurately localizing the ground vehicle, the system obtains information for navigation and interaction with the surroundings.

The heart of the proposed solution lies in the development and implementation of a sophisticated control algorithm. This algorithm interprets the visual feedback from the onboard cameras and generates control commands to guide the drone's movements. By integrating with the drone's navigation system, the control algorithm enables autonomous navigation based on the detected markers.

Validation of the proposed solution is conducted through simulation experiments. These experiments assess the performance of this new IBVS system in various scenarios and environmental conditions.

The proposed solution offers several expected benefits and contributions. These include enhanced operational efficiency and autonomy for drones, improved accuracy and less time complexity as compared to traditional IBVS technology, and advancements in the field of IBVS for drone operations. The proposed solution contributes to addressing real-world challenges and expanding the capabilities of autonomous aerial robotics.

C. *Key Contributions*

- Contribution 1: Established an environment with a ground vehicle and drone, facilitating autonomous drone following and precise hovering capabilities.
- Contribution 2: Developed an algorithm with lesser time complexity than normal IBVS for hovering and tracking the ground vehicle.

III. Methodology of the Proposed Approach

The methodology of the proposed approach involves several key steps to enable the drone to autonomously track and interact with the target marker. The process includes mathematical modeling, algorithmic steps, and the utilization of flowcharts and block diagrams for visualization.

Mathematical Modeling: The methodology begins with obtaining the coordinates of the target marker, providing essential positional information for the tracking process. Subsequently, the camera center is calculated, serving as a reference point for the drone's visual perception of the environment.

Algorithm

The goal of the algorithm is to enable a drone to autonomously follow a ground vehicle using visual feedback from a downward-facing camera. The vehicle is identified based on distinct color features: the vehicle body is blue, the left wheel is red, and the right wheel is green. By processing the captured images, the drone can calculate the necessary adjustments to its position to maintain alignment with the ground vehicle.

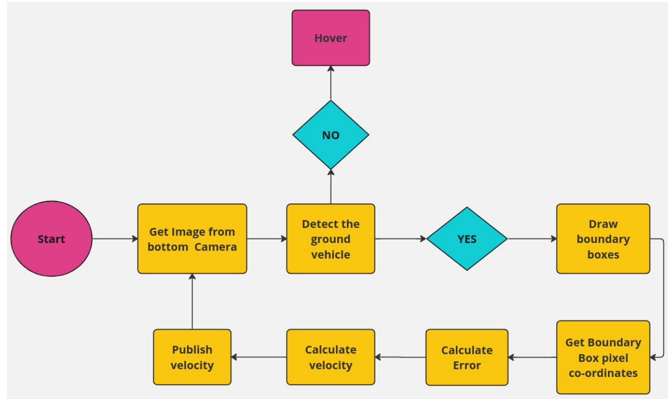

Fig. 2: The figure shows the flow chart of the algorithm.

Components of the Algorithm

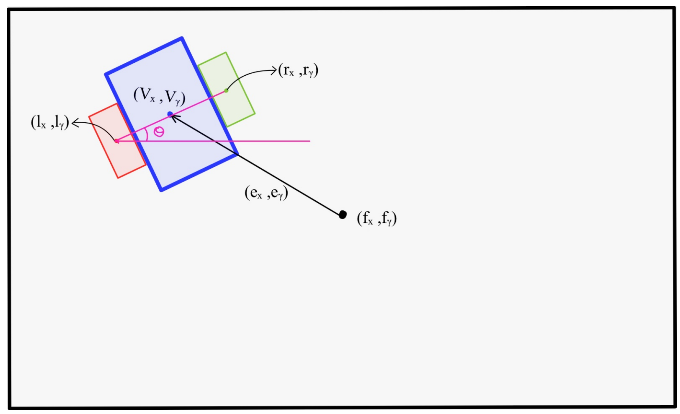

Fig. 3: The figure shows a ground vehicle in the camera frame. θ represents the error in angular position of the vehicle; (f_x, f_y), (v_x, v_y), (l_x, l_y), and (r_x, r_y) represent the center of the camera frame, the vehicle body, the left wheel, and the right wheel, respectively.

Image Acquisition and Processing

- The drone captures images using a downward-facing camera.
- These images are processed to identify and locate the blue vehicle body, red left wheel, and green right wheel.

Color Segmentation

- The image is converted to the HSV color space to facilitate color-based segmentation.
- Masks are created to isolate the blue, red, and green regions corresponding to the vehicle body and wheels.
- Contours are detected within these masks to determine the positions of the vehicle body and wheels.

Bounding Box Detection

- For each color segment (blue, red, green), the algorithm finds the bounding box around the detected contours.
- The corners of these bounding boxes are used to approximate the locations of the vehicle body and wheels.

Center Calculation

The centers of the vehicle body and wheels are calculated as the average of the bounding box corner coordinates.

For a box defined by corners $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ in pixel co-ordinates, the center is calculated as:

$$\text{center} = \left(\frac{x_1 + x_2 + x_3 + x_4}{4}, \frac{y_1 + y_2 + y_3 + y_4}{4} \right)$$

Angular Error Calculation

The angular error of the drone relative to the vehicle is estimated using the centers of the wheels. The tangent of the angle θ is computed as:

$$\tan(\theta) = \frac{r_y - l_y}{r_x - l_x}$$

$$\theta = \arctan\left(\frac{r_y - l_y}{r_x - l_x} \right)$$

where θ is the angular error and (r_x, r_y) and (l_x, l_y) are the centers of the right and left wheels, respectively.

Linear Error Calculation

The positional error between the drone's camera center (f_x, f_y) and the vehicle center (v_x, v_y) is calculated:

$$\begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} v_x - f_x \\ v_y - f_y \end{bmatrix}$$

P Control

Proportional(P) control is used to compute the required linear and angular velocities for the drone.

The linear velocity components v_{linear} are proportional to the positional error:

$$v_{\text{linear}} = \text{kp} \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$

The angular velocity component v_{angular} is proportional to the angle:

$$v_{\text{angular}} = \text{-kp} \times \tan(\theta) \qquad (Meathod1)$$

$$v_{\text{angular}} = \text{-kp} \times \theta \qquad (Meathod2)$$

The constant kp is the proportional gain.

Velocity Command Generation

The computed velocities are used to generate the command for the drone's movement. The velocity command is formulated as:

$$\text{vel.linear.x} = v_{\text{linear}}[0]$$

$$\text{vel.linear.y} = v_{\text{linear}}[1]$$

$$\text{vel.angular.z} = v_{\text{angular}}$$

These commands are published to the drone's control system to adjust its position accordingly.

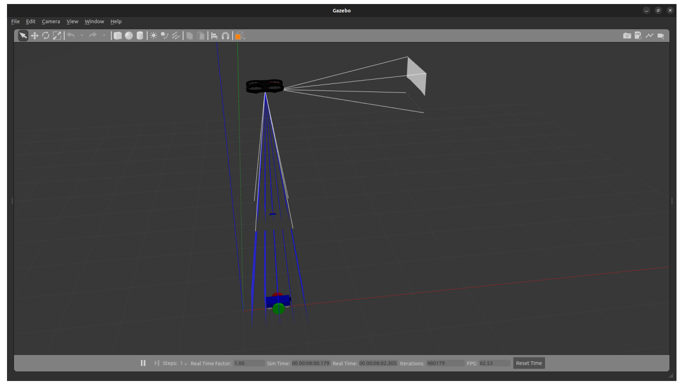

Fig. 4: The figure shows the simulation environment.

A. Implementation Details

- **Software Requirements:** List the required software components to run the system are: ROS2, Gazebo, Python etc.
- **Simulation setup:** There is ground vehicle along with Parrot Drone in the simulation environment. Drone has a bottom facing camera from which images are being captured and published on the topic '/drone/bottom/img_raw' which is subscribed by '/image/˙subscriber' and processes it to generate the velocity command getting published on the topic '/drone/cmd/˙vel. The ground vehicle can be controlled by Teleop Twist Keyboard. As soon as Ground vehicle starts moving drone starts following it.

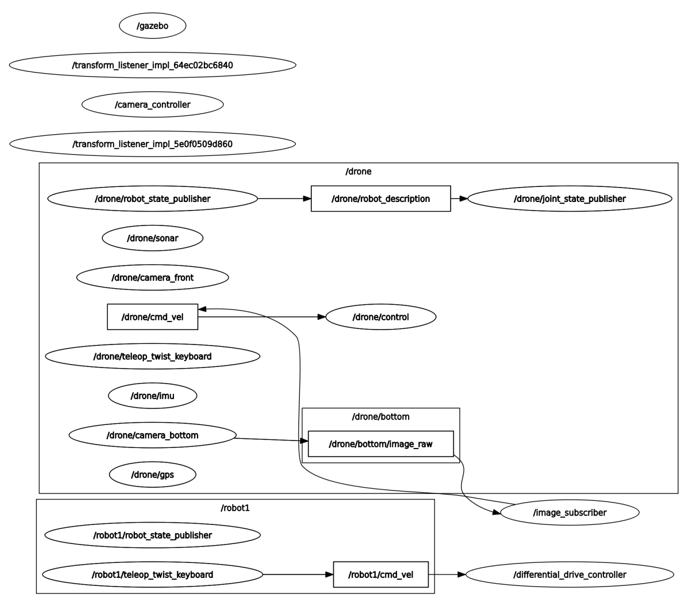

Fig. 5: The figure shows active topics and nodes

IV. Results and Analysis

In this section, we present the simulation results of our IBVS system for drone operations. We provide an analysis of the performance metrics and discuss the implications of the results.

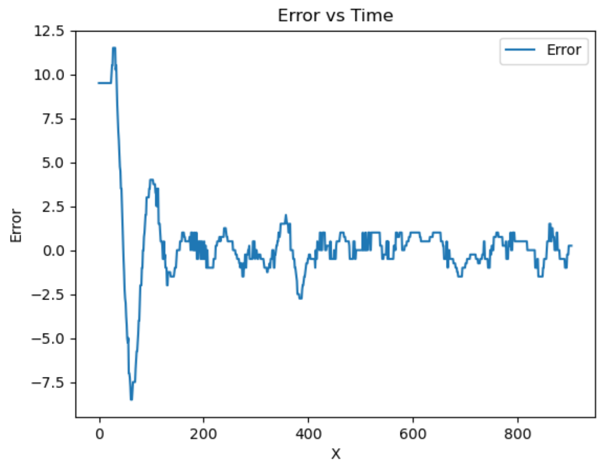

Fig. 6: Error vs time

A. Simulation Results

The simulation results demonstrate the effectiveness of our IBVS system in performing tasks such as search, hover, landing, and tracking. We present the following key findings:

- **Search and Hover:** The drone successfully locates and hovers over the target marker with high accuracy and stability.
- **Tracking:** The drone demonstrates precise tracking of the specified target vehicle while it is moving.

B. Analysis

We analyze the performance of the IBVS system based on various metrics, including accuracy, efficiency, and robustness. Our analysis reveals the following insights:

- **Accuracy:** The IBVS system achieves high accuracy in positioning and tracking the target marker, resulting in precise drone movements.
- **Efficiency:** The system demonstrates efficient operation, with minimal computational overhead and rapid response to environmental changes.

Conclusion

The algorithm integrates computer vision techniques with control theory to enable a drone to autonomously follow a ground vehicle. By detecting and processing color-segmented regions, calculating positional and angular errors, and applying PID control, the drone can maintain a consistent position relative to the moving vehicle. This method demonstrates a practical application of image feedback for autonomous navigation in robotics.