

Article

Visual Servoing Approach to Autonomous UAV Landing on a Moving Vehicle

Azarakhsh Keipour^{1,*}, Guilherme A. S. Pereira², Rogerio Bonatti³, Rohit Garg⁴, Puru Rastogi⁵, Geetesh Dubey⁴ and Sebastian Scherer⁶¹ Robotics AI, Amazon, Arlington, VA 22202, USA² Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506, USA³ Autonomous Systems Group, Microsoft, Redmond, WA 98052, USA⁴ Gather AI, Pittsburgh, PA 15217, USA⁵ Mowito, Bengaluru 560091, India⁶ Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

* Correspondence: keipour@gmail.com; Tel.: +1-412-273-2712

† The publication was written prior to A. Keipour joining Amazon. During the realization of this work, he was affiliated with the Robotics Institute at Carnegie Mellon University.

Abstract: Many aerial robotic applications require the ability to land on moving platforms, such as delivery trucks and marine research boats. We present a method to autonomously land an Unmanned Aerial Vehicle on a moving vehicle. A visual servoing controller approaches the ground vehicle using velocity commands calculated directly in image space. The control laws generate velocity commands in all three dimensions, eliminating the need for a separate height controller. The method has shown the ability to approach and land on the moving deck in simulation, indoor and outdoor environments, and compared to the other available methods, it has provided the fastest landing approach. Unlike many existing methods for landing on fast-moving platforms, this method does not rely on additional external setups, such as RTK, motion capture system, ground station, offboard processing, or communication with the vehicle, and it requires only the minimal set of hardware and localization sensors. The videos and source codes are also provided.

Keywords: visual servoing; autonomous landing; monocular vision; aerial robotics



Citation: Keipour, A.; Pereira, G.A.S.; Bonatti, R.; Garg, R.; Rastogi, P.; Dubey, G.; Scherer, S. Visual Servoing Approach to Autonomous UAV Landing on a Moving Vehicle. *Sensors* **2022**, *22*, 6549. <https://doi.org/10.3390/s22176549>

Academic Editors: Gregor Klancar, Marija Seder and Sašo Blažič

Received: 17 July 2022

Accepted: 25 August 2022

Published: 30 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/>).

1. Introduction

The recent advances in Unmanned Aerial Vehicles (UAVs) have allowed innovative applications ranging from package delivery to infrastructure inspection, early fire detection, and cinematography [1–3]. While many of these applications require landing on the ground and static platforms, the ability to land on dynamic platforms is essential for some other applications. A typical example of a real-world scenario is an autonomous landing on a ship deck [4] or maritime Search and Rescue operations [5].

The autonomous landing of Unmanned Aerial Vehicles (UAVs) on known patterns has been an active area of research for several years [6–12]. Some of the key challenges of the problem include dealing with environmental conditions, such as changes in light and wind, and robust detection of the landing zone. The subsequent maneuver in trying to land also needs to take care of the potential ground effects at the proximity of the landing surface.

The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) is a set of real-world robotics challenges happening every few years [13]. Challenge 1 took place in March 2017, focusing on landing UAVs on a moving platform. In this challenge, there was a ground vehicle (truck) moving on an 8-shaped road in a 90 × 60 m arena with a predefined speed of 15 km/h (4.17 m/s). On top of the truck, at 1.5 m height, there was a flat horizontal ferromagnetic deck with a predefined 1.5 × 1.5 m pattern (Figure 1) printed on top of it [14].

The goal was to land the UAV on this deck autonomously. To make the challenge realistic, no communication between the UAV and the ground vehicle was allowed, no precise state estimation sensors such as RTK or Motion Capture were provided, and finally, the location of the vehicle was not given to the UAV, requiring a visual detection of the pattern, which could result in false and imperfect detections among other issues.



Figure 1. Landing zone pattern utilized at the MBZIRC competition [14].

This paper describes our approach to landing on moving platforms, which uses only a single monocular camera, a point Lidar, and an onboard computer. The method is based on a visual-servoing controller and has been successfully tested for landing on moving ground vehicles at speeds of up to 15 km/h (4.17 m/s).

Our contributions include proposing a method that does not depend on a special environment setup (e.g., IR markers and communication channels), can work with the minimal UAV sensor setup, only requires a monocular camera, has minimum processing requirements (e.g., a lightweight onboard computer), and approaches the landing platform with high certainty. We also provide our source codes and simulation environment to assist with future developments. The videos of our experiments and all the codes are available at <http://theairlab.org/landing-on-vehicle>.

The paper is organized as follows: Section 2 reviews the relevant work and state of the art; Section 3 explains our method for autonomous landing; Section 4 describes the architecture and the experiments performed and discusses the results from simulation, indoor and outdoor tests.

2. Background and Relevant Research

A review of the existing research reveals several approaches to solving this problem. One of the earliest vision-based approaches introduced an image moment-based method to land a scale helicopter on a static landing pad that has a distinct geometric shape [15]. This work further extends to the helicopter landing on a moving target [6]. However, a limitation of the approach is that it tracks the target in a single dimension, and the image data are processed offboard.

A more recent work tries to land a quadcopter on top of a ground vehicle traveling at high speed using an AprilTag fiducial marker for landing pad detection [16]. They devise a combination of Proportional Navigation (PN) guidance and Proportional-Derivative (PD) control to execute the task. The UAV has landed on a target moving at up to 50 km/h (13.89 m/s). However, the PN guidance strategy used to approach the target has to rely on the wireless transmission of GPS and IMU measurements from the ground vehicle to the UAV, which is impractical in most applications. In addition, two cameras are used for the final approach to detect the AprilTag from far and near.

Visual servoing control is another approach that can be applied to this problem. In [17], it is used to generate a velocity reference command to the lower level controller of a quadrotor, guiding the vehicle to a target landing pad moving at 0.07 m/s. It is computationally cheaper to compute the control signals in image space than in 3D space. However, this work relies on offboard computing as well as a VICON motion capture system for accurate position feedback during its patrol to search for the target. Another

closed-loop approach for landing was proposed in [18]. In this paper, the authors presented a landing vector field. Unlike visual servoing, the method's main advantage is that the vector field can enforce the shape of the vehicle's trajectory during landing. In addition, different from visual servoing, the 3D localization of the UAV is necessary.

Some other works emphasize the design of a landing pad that is robust to detection, which simplifies the task by eliminating the detection uncertainty [19–21]. In [7], a Wii (IR) camera is used to track a T-shaped 3D pattern of infrared lights mounted on the vehicle, and, in [22], an IR beacon is placed on the landing pad. Reliable detection enables a ground-based system to estimate the position of the UAV relative to the landing target. Xing et al. [21] propose a notched ring with a square landmark inside to enable robust detection of the landing zone, then it uses a minimum-jerk trajectory to land on the detected pattern. Finally, the method presented in [23] localizes the vehicle using two nested ArUco markers [24] in an illuminated landing pad, allowing landing during the night using visual servoing.

Optical flow is another technique used to provide visual feedback for guiding the UAV. Ruffier and Franceschini [25] developed an autopilot that uses a ventral optic flow regulator in one of its feedback loops for controlling lift (or altitude). The UAV can land on a platform that moves along two axes. The method in [26] also uses the optical flow information obtained from a textured landing target but does not attempt to reconstruct velocity or distance to the goal. Instead, the approach chooses to control the UAV within the image-based paradigm.

The method in [27] transfers all the computing power to the ground station eliminating the need for an onboard computer. They only report results for very slow vehicle speeds (10–15 cm/s), and the method cannot extend to higher speeds due to communication delays.

Several approaches were introduced for the MBZIRC challenge. The method used by [10] tries to follow the platform until the pattern detection rate is high enough for landing. Then, it continues to follow the platform on the horizontal plane while slowly decreasing the altitude of the UAV. A Lidar (Light Detection and Ranging sensor) is used to determine if the UAV should land on the platform or not. The landing is performed by a fast descent followed by switching off the motors. The state estimation uses monocular VI-Sensor data fused with GPS, IMU, and RTK data.

Researchers from the University of Catania implemented a system that detects and tracks the target pattern using a Tracking-Learning-Detection-based method integrated with the Circle Hough Transform to find the precise location of the landing zone. Then, a Kalman filter is used to estimate the vehicle's trajectory for the UAV to follow and approach it for landing [28,29].

Baca et al. [30] took advantage of a SuperFisheye high-resolution monocular camera with a high FPS rate for pattern detection. They used adaptive thresholding to improve the method's robustness to the light intensity, followed by undistorting the image. Then, the circle and the cross are detected to find the landing pattern in the frame. After applying a Kalman filter on the detected coordinates to estimate and predict the location of the ground vehicle, a model-predictive controller (MPC) is devised to generate the reference trajectory for the UAV in real-time, which is tracked by a nonlinear feedback controller to approach and land on the ground vehicle.

Researchers at the University of Bonn have achieved the fastest approach for landing among the other successful runs in the MBZIRC competition (measured from the time of detection to the successful landing). They used two cameras for the landing pattern detection, a Nonlinear Model Predictive Control (MPC) for time-optimal trajectory generation and control, and a separate proportional controller for yaw [11].

Tzoumanikas et al. [31] used an RGB-D camera for visual-inertial state estimation. After the initial detection, the UAV flies in the vehicle's direction until it reaches the velocity and position close to the ground vehicle. Then, the UAV starts descending toward the target until a certain altitude, when it will continue descending in an open-loop manner. During

the landing, the UAV uses pattern detection feedback only to decide between aborting the mission or not and not for correcting controller commands.

University of Zurich researchers have introduced a system in [32] that first finds the quadrangle of the pattern and then searches for the ellipse or the cross, validating the detection using RANSAC. Then, the platform's position is estimated from its relative position to the UAV, and an optimal landing trajectory is generated to land the UAV on the vehicle.

An ultimate goal in robotics' real-world applications is to achieve good results while reducing the robot's costs and the amount of the prior setup. Dependence on the additional hardware or external hardware (e.g., motion capture systems, RTK, multiple cameras) is costly and not always feasible in real-world applications. In contrast with other available methods, our approach uses only a single monocular camera (with comparatively low resolution and low frame rate), a point Lidar, and an onboard computer to land on the moving ground vehicle at high speeds (tested at 15 km/h (4.17 m/s)) using a visual-servoing controller. The approach does not depend on accurate localization sensors (e.g., RTK or motion capture systems) and can work with the state estimation provided by a commercial UAV platform (obtained only using GPS, IMU, and barometer data). Our real-time elliptic pattern detection and tracking method can track the landing deck in challenging environment conditions (e.g., changes in lighting) with a high frequency [33].

3. Materials and Methods

This section explains our approach to sensing the vehicle passing below the UAV and then landing the quadrotor on the vehicle. Our system was developed in the MBZIRC Challenge 1 context, discussed in Section 1. Therefore, our primary goal was to land on a known platform (see Figure 1) fixed on top of a vehicle moving at a fixed speed.

3.1. General Strategy

In our approach, we assume that the drone is hovering above the point where the moving vehicle is expected to pass (which can be any point on the road if the vehicle is in a loop). In addition, we assume that the direction and the speed of the moving vehicle are known, which can be previously obtained, for example, from some consecutive deck pattern detections. We further assume that the vehicle is almost moving in a straight line during the few seconds when the UAV attempts to land.

A time-optimal trajectory to landing is theoretically possible, while in practice, there is a variation in the speed of the moving vehicle (which is driven by a human driver), and there are delays and errors in state estimation. Therefore, an approach with constant feedback to correct the trajectory would work better than an open-loop landing using an optimal trajectory. The method we chose is to continually measure the target's position, size, and orientation directly in the image space and translate it to velocity commands for the UAV using visual servoing. This approach is described in Section 3.3.

Given the assumptions mentioned above, the defined strategy is as follows:

1. The quadrotor hovers at the point at the height of h meters until it senses the passing ground vehicle.
2. As soon as the ground vehicle is sensed, the UAV starts flying in the direction of movement with a predefined speed slightly higher than the vehicle speed to compensate for the distance between the UAV and the vehicle. This gap results from the delays in processing the sensors and dynamics of the UAV in getting up to the vehicle speed.
3. After the initial acceleration, the UAV flies with a feed-forwarding speed equal to the ground vehicle speed. Then, the visual servoing controller tries to decrease the remaining gap between the UAV and the target.
4. After successfully landing, magnets at the bottom of the UAV legs stick to the metal platform, and the propellers are shut down.

Figure 2 depicts the steps of the described strategy. The following subsections explain the main components of the landing strategy in greater detail.

We used a path tracker similar to the one proposed by [34] to fly to the hover point and to fly the straight lines.

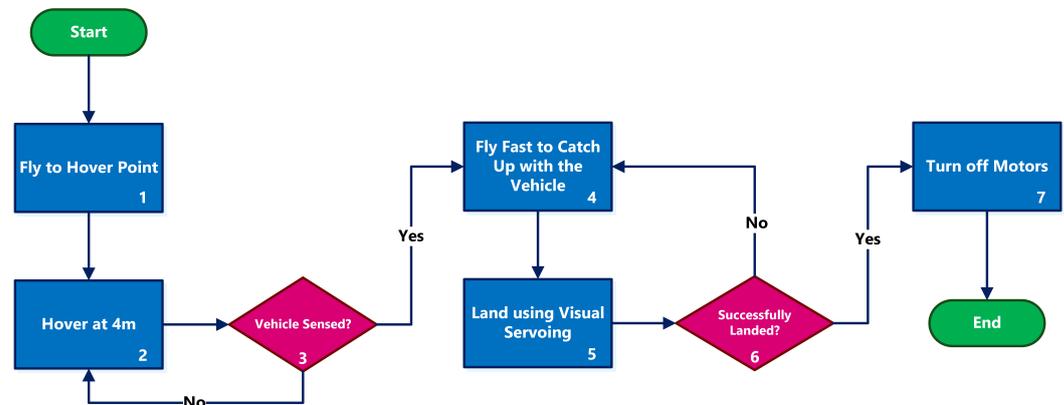


Figure 2. Our strategy for landing on a moving vehicle. The vehicle hovers above the road until it senses the passing vehicle. Then, it accelerates to catch up with the vehicle and tries to land using the visual servoing method. As soon as the landing is detected, all the motors are turned off.

3.2. Sensing the Passing Vehicle

We considered two strategies for detecting the passing vehicle below the UAV when it is hovering above the road on the vehicle's path: the camera or using laser sensors. Due to the reliability of our pattern detection method, using the camera gives more reliable results. The chosen hovering altitude can range from a few to tens of meters, where the pattern detection is reliable. The lower altitude allows for a faster approach to landing, while the higher altitude results in more flight time required to catch up with the vehicle (box 4 in Figure 2).

The strategy of sensing the vehicle using a camera worked well in our tests; however, for higher vehicle speeds, we developed a laser-based solution to avoid the delays introduced by the image processing and to get an even lower reaction time. We used three point-laser sensors on the bottom of the UAV (as described in Section 4.1) to detect the vehicle passing below it. If the distance measured by any of the three lasers has a sudden drop of more than a certain threshold within a specified period, we assume that the vehicle is passing below the quadrotor, and the landing system is triggered.

The choice of the number of the lasers and the ideal height of the hovering depend on the road's width and the vehicle's width and height. For our tests, the road was 3 m wide, the vehicle was 1.5 m wide, and its height was 1.5 m. In order to capture this vehicle, the center laser points directly downwards, and the other two lasers are angled at approximately 30 degrees on each side. With this configuration, at the altitude of 4 m, two additional laser rays will hit the ground at a distance of 1.44 m from the center laser. When the moving vehicle (with a width and height of 1.5 m) passes on the road below the quadrotor, at least one of the three lasers will sense the change in the measured height. Algorithm 1 illustrates the method.

Algorithm 1: Approach for sensing the passing vehicle using three point lasers.

```

1: function DETECTPASSINGVEHICLE
2:   ▷ Keep reading the sensors until the vehicle is sensed
3:   while true do
4:     ▷ Read the laser measurements
5:      $dist_l, dist_c, dist_r \leftarrow \text{READLASERS}()$ 
6:     ▷ Check if we have a sudden decrease in the measured distance
7:     ▷  $thr$  is set to a large number smaller than the vehicle's height
8:     if  $(last_l - dist_l) > thr$  or  $(last_c - dist_c) > thr$  or  $(last_r - dist_r) > thr$  then
9:       return true
10:    else
11:       $last_l \leftarrow dist_l, last_c \leftarrow dist_c, last_r \leftarrow dist_r$ 
12:    end if
13:  end while
14: end function

```

Figure 3 shows examples of a laser triggering after sudden measurement changes.

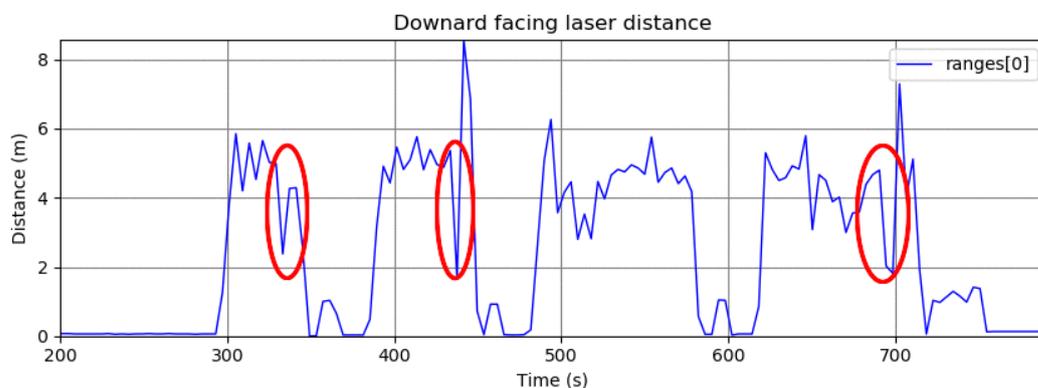


Figure 3. Plot of laser distance measurements during a flight in our trials. The passing vehicle is detected in several moments, indicated by the red ellipses. Detection by this laser fails in the third trial due to vehicle misalignment with the laser direction.

3.3. Landing Using Visual Servoing

After the vehicle is detected (using the lasers or the visual pattern detection), the next task is trying to land on the defined pattern on the deck of the moving ground vehicle. In our strategy, the UAV follows and lands the moving deck using visual servoing. Visual servoing consists of techniques that use information extracted from visual data to control the robot's motion. Assuming that the camera is attached to the drone through a gimbal, the vision configuration is called an eye-in-hand configuration. We used an Image-Based Visual Servoing scheme (IBVS), in which the error signal is estimated directly based on the 2D features of the target, which we considered to be the center of the target ellipse and the corners of its circumscribed rectangle. The error is then computed as the difference (in pixels) between the features' positions when the UAV is right above the target, at the height of about 50 cm, and the current features' positions given by the deck detection algorithm (discussed in Section 3.6). Since the robot operates in task space coordinates, there must be a mapping between the changes in image feature parameters and the robot's position. This mapping is applied using the image Jacobian, also known as the Interaction Matrix [35].

3.3.1. Interaction Matrix

If τ represents the task space and F represents the feature parameter space, then the image Jacobian L_v is the transformation from the tangent space of τ at r to the tangent space of F at f , where r and f are vectors in task space and feature parameter space, respectively. This relationship could be represented as:

$$\dot{f} = L_v(r)\dot{r}, \quad (1)$$

which implies

$$L_v(r) = \left[\frac{\partial f}{\partial r} \right]. \quad (2)$$

In practice, \dot{r} needs to be calculated given the value of \dot{f} and the interaction matrix at r . Depending on the size and rank of the interaction matrix, different approaches can be used to calculate the inverse or pseudo-inverse of the interaction matrix, which gives the required \dot{r} [36].

3.3.2. Calculating Quadrotor's Velocity

The developed visual servoing controller for this project assumes the presence of five 2D image features: the center of the detected ellipse \mathbf{s}_c and the four corners of the rectangle that circumscribes the detected elliptical target. Since the gimbal has faster dynamics than the quadrotor, two separate controllers were developed. The first controller is a simple proportional controller for the gimbal pitch angle to keep the deck in the center of the image:

$$\delta = k\|\mathbf{s}_c^* - \mathbf{s}_c\|, \quad (3)$$

where k is a positive gain, \mathbf{s}_c^* is the desired location of the deck center in the image, and $\|\cdot\|$ stands for the Euclidean distance. Since the gimbal is not mounted at the center of the quadrotor, position \mathbf{s}_c^* has an offset to compensate for this.

The second control problem, which relies on image features $\mathbf{s} = [x, y]^T$, is a traditional visual servoing problem, which is solved using the following control law:

$$\mathbf{v} = -\lambda \left(\mathbf{L}_s^c \mathbf{V}_b^c \mathbf{J}_b^b \right)^+ (\mathbf{s}^* - \mathbf{s}) + \mathbf{v}_{ff} \quad (4)$$

where \mathbf{v} is the quadrotor velocity vector in the body control frame, λ is a positive gain, \mathbf{L}_s is the interaction matrix, ${}^c\mathbf{V}_b$ is a matrix that transforms velocities from the camera frame to the body frame, ${}^b\mathbf{J}_b$ is the robot Jacobian, $()^+$ is the pseudo-inverse operator, $(\mathbf{s}^* - \mathbf{s})$ is the error computed in the feature space and \mathbf{v}_{ff} is a feed-forward term obtained by transforming the truck velocity to the body reference frame [37]. In our case, the error $(\mathbf{s}^* - \mathbf{s})$ is minimized using only linear velocities and yaw rate. Therefore, $\mathbf{v} = [v_x, v_y, v_z, \omega]^T$, which forces the Jacobian matrix to be written as:

$${}^b\mathbf{J}_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

This matrix is responsible for the transformation from the general six-dimensional velocity space to a four-dimensional space that the UAV can indeed follow.

The interaction matrix $\mathbf{L}_s(\mathbf{s}^*, Z^*)$ in our method is constant, and is computed at the target location using image features ($\mathbf{s}^* = [x^*, y^*]^T$) and the distance of the camera to the deck (Z^*) as [35]:

$$\mathbf{L}_s = \begin{bmatrix} -1/Z^* & 0 & x^*/Z^* & x^*y^* & -(1+x^{*2}) & y^* \\ 0 & -1/Z^* & y^*/Z^* & 1-y^{*2} & -x^*y^* & -x^* \end{bmatrix}. \quad (6)$$

It is essential to mention that, ideally, the interaction matrix should be computed online using the current features and height information. However, in our approach, we use a constant matrix to be more robust regarding the errors related to the UAV position

estimation and information necessary to compute Z . For the same reason, the distance from the deck is not used as part of the feature, as is done in some visual servoing approaches [37]. In addition, a constant matrix causes the system to be less sensitive to eventual spurious detection of the target. On the other hand, since the camera is mounted on a gimbal, matrix ${}^c\mathbf{V}_b$ must be computed online as:

$${}^c\mathbf{V}_b = \begin{bmatrix} {}^c\mathbf{R}_b & [{}^c\mathbf{t}_b]_{\times} {}^c\mathbf{R}_b \\ \mathbf{0}_{3 \times 3} & {}^c\mathbf{R}_b \end{bmatrix} \quad (7)$$

where ${}^c\mathbf{R}_b$ is the rotation matrix between the image and the robot body, which is a function of the gimbal angles, ${}^c\mathbf{t}_b$ is the corresponding constant translation vector, and $[\mathbf{t}]_{\times}$ is the skew-symmetric matrix related to \mathbf{t} .

Since we are dealing with a moving deck, the feed-forward term \mathbf{v}_{ff} in Equation (4) is necessary to reduce the error $(\mathbf{s}^* - \mathbf{s})$ to zero [38]. In our case, \mathbf{v}_{ff} originates from the ground vehicle velocity vector, which is a vector tangent to the moving vehicle's path. Thus, the precise estimation of this vector would require the localization of the vehicle with respect to the track, which is a challenging task. To make such an estimation more robust to sensor noise, we relaxed this problem to compute \mathbf{v}_{ff} only on the straight line segments of the track, where we assume the ground vehicle speed vector to be constant for several seconds. This restriction causes our UAV to land only on such segments.

All the visual servoing control laws in Equations (1)–(7) are implemented using the Visual Servoing Platform (ViSP) [37] version 3.0.1, which provides a suitable data structure for the problem and efficiently computes the required pseudo-inverse matrix.

3.4. Initial Quadrotor Acceleration

As described in Section 3.1, the quadrotor will hover until it senses the ground vehicle passing below it. Due to the relatively slow dynamics of the quadrotor controller, it takes some time to accelerate to the ground vehicle's speed. By this time, the deck is already out of the camera's sight, and the visual servoing procedure cannot be used for landing.

To compensate for the delay in reaching the ground vehicle's speed and to gain sight of the deck again, an initial period of high acceleration is set right after sensing the passing ground vehicle. In this period, the desired speed is set to a value higher than the vehicle's speed until the quadrotor compensates for the created gap with the vehicle. After this time, the quadrotor switches to the visual servoing procedure described in Section 3.3 with a feed-forward velocity lowered to the current ground vehicle's speed.

We tested two different criteria for switching from the higher-speed flight to the visual servoing (box 5 in Figure 2) procedure: vision-based and timing-based. In the vision-based approach, the switch to visual servoing happens when the pattern on the moving vehicle is detected again. In the timing-based approach, depending on the processing and dynamics delays of the sensing and considering the known speed of the vehicle and the set speed for the UAV, it is possible to calculate a fixed time t_a that is enough for catching up with the vehicle. After this time, the UAV can start the visual servoing landing.

3.5. Final Steps and Landing

When the UAV gets too close to the ground vehicle, the camera can no longer see the pattern. To prevent the UAV from stopping when the target detector (described in Section 3.6) is unable to detect the deck during landing, the UAV is still commanded to move with the known vehicle's speed for a few sampling periods. If using one of its point lasers, the UAV detects that it has the correct height to land, it increases its downward vertical velocity and activates the drone landing procedure, which shuts down the propellers.

Note that the "blind" approach can only work due to the low delays between losing the pattern and landing (generally 30–70 ms). This approach may fail if the ground vehicle aggressively changes its direction or velocity. However, for many applications, the vehicle's motion (e.g., a ship or a delivery truck) can be reasonably assumed stable for such a short time.

3.6. Detection and Tracking of the Deck

Figure 1 shows the target pattern on top of the deck. There are some challenges in the detection of this pattern from the camera frames, including:

- This algorithm is used for landing the quadrotor on a moving vehicle. Therefore, it should work online (with a frequency greater than 10 Hz) on a resource-limited onboard computer.
- The shape details cannot be seen in the video frames when the quadrotor is flying far from the deck.
- The shape of the target is transformed by a projective distortion, which occurs when the shape is seen from different points of view.
- There is a wide range of illumination conditions (e.g., cloudy, sunny, morning, evening).
- Due to the reflection of the light (e.g., from sources like the sun or bulbs), the target may not always be seen in all the frames, even when the camera is close to the vehicle.
- In some frames, there may be shadows on the target shape (e.g., the shadow of the quadrotor or trees).
- In some frames, only a part of the target shape may be seen.

Considering these challenges and the shape of the pattern, different participants of the MBZIRC challenge chose different approaches for detection. For example, in [10], the authors implemented a quadrilateral detector for detecting the pattern from far distances and a cross detector for close-range situations. The method in [11] uses the line and circular Hough-transform algorithms to calculate a confidence score for the pattern for the initial detection and then uses the rectangular area around the pattern for the tracking. In [39], a Convolutional Neural Network is developed to detect the elliptic pattern, which was trained with over 5000 images collected from the pattern moving at a maximum of 15 km/h (4.17 m/s) at various heights. In [40] the cross and the circle are detected for far images, and only cross detection is used for the closer frames. The method uses the [41] method for ellipse detection. In [31], the outer square is detected first, and then the detection is verified by a template matching algorithm.

In our work, we developed a novel method to overcome the problem challenges mentioned above, which detects and tracks the pattern by exploiting the structural properties of the shape without the need for any training [33]. More specifically, the deck detector system detects and tracks the circular shape (seen as an ellipse due to the projective transformation) on top of the deck, while ignoring the cross in the middle and then verifying that the detected ellipse belongs to the deck pattern. The developed real-time ellipse detection method can also detect the ellipses with partial occlusion or the ellipses that are exceeding the image boundaries.

Figure 4 shows results for the detection of the deck in some sample frames. A more detailed description of the methods, database, and results are provided in [33].



Figure 4. Cont.

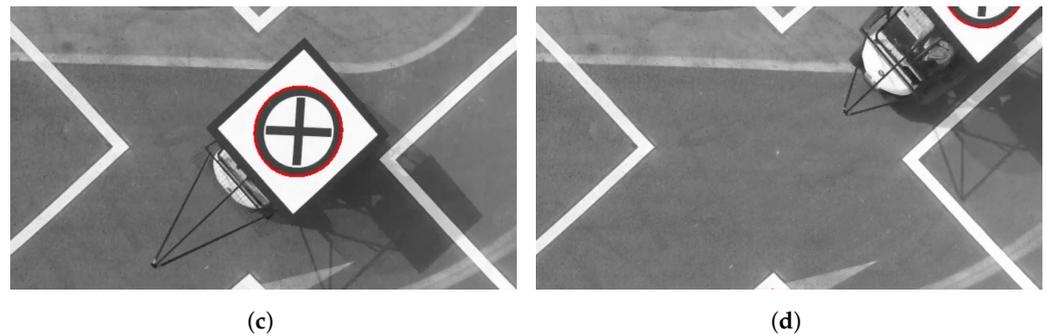


Figure 4. (a–d) Result of the deck detection and tracking algorithm on sample frames. The red ellipse indicates the detected pattern on the deck of the moving vehicle.

4. Experiments and Results

This section outlines the architecture of our proposed landing method from both hardware and software perspectives and presents the results obtained from the tests in different conditions: simulation, indoor testing, and outdoor testing. Some videos of experiments and sequences shown in this section are available at <http://theairlab.org/landing-on-vehicle>.

4.1. Hardware

The aerial platform chosen for this project is a DJI Matrice 100 quadrotor. This UAV is programmable, can carry additional sensors, is fast, provides velocity commands, and is accurate enough for this problem. Using an off-the-shelf vs. home-built platform vastly increased the speed of the development, reducing the time needed for dealing with hardware bugs. The selected platform has a GPS module for state estimation and is additionally equipped with an ARM-based DJI Manifold onboard computer (based on NVIDIA Jetson TK1 computer), a DJI Zenmuse X3 Gimbal and Camera system, a set of three SF30 altimeters, and four permanent magnets at the bottom of legs. One altimeter is pointed straight down to measure the current height of the flight with respect to the ground. The other two altimeters are mounted on the two sides of the quadrotor, pointing down with approximately 30 degrees outward skew to measure the sudden changes in the height on the sides of the quadrotor for the detection of the truck passing next to the quadrotor. The camera is mounted on a three-axis gimbal and outputs grayscale images at a frequency of 30 Hz with a resolution of 640×360 pixels.

A wireless adapter allows communication between the robot computer and the ground station for development and monitoring. For indoor tests, a set of propeller guards and a DJI Guidance System, a vision-based system able to provide indoor localization, velocity estimation, and obstacle avoidance, are used to provide velocity estimation and improve the safety of the testing. The guards and the Guidance system are removed for outdoor tests to reduce the weight and increase the robustness to the wind. Figure 5 shows the picture of the robot.



Figure 5. The robot developed for our project. The figure shows our DJI Matrice 100 quadrotor equipped with the indoor testing parts (DJI Guidance System and the propeller guards), as well as the DJI Zenmuse X3 Gimbal and Camera system, the SF30 altimeter, and the DJI Manifold onboard computer.

The main characteristics of the robot are summarized in Table 1.

Table 1. The main characteristics and parameters of the UAV.

Parameter	Value
Maximum time of flight	19 min
Maximum horizontal speed	17 m/s (61 km/h)
Maximum vertical speed	4 m/s (14.4 km/h)
External diameter (with propeller guards)	1.2 m
Height	0.46 m
Processor	Quad-core ARM CORTEX-A15
RAM Memory	2 GB
Maximum robot speed	8.33 m/s (30 km/h)

4.2. Software

The robot's software is developed using the Robot Operating System (ROS) to achieve a modular structure. The modularity allowed the team members to work on different parts of the software independent of each other and reduced the debugging time. DJI's Onboard SDK is used to interact with the quadrotor's controller. The software is constructed in a way that it can control both the simulator and the actual robot with just a few modifications. Figure 6 shows a general view of the system architecture.

In Figure 6, the main block of our architecture (implemented as a ROS node), called "Mission Control", dictates the robot's behavior, informing the other blocks of the current task (mission status) using a ROS Parameter. The Mission Control block also generates trajectories for the robot when the current mission mode requires the robot to fly to a different position. To dictate the robot's behavior, Mission Control relies on the robot's odometry information and the robot's distance to the ground or target.

The other blocks in Figure 6 are:

- Deck tracking—detects the deck target and provides its position and orientation in the image reference frame;
- Visual Servoing—controls the UAV to track and approach the deck;
- Trajectory controller—provides velocity commands to the robot so it can follow the trajectories generated by the Mission Control node;
- Mux—selects the velocities to be sent to the quadrotor, depending on the mission status.

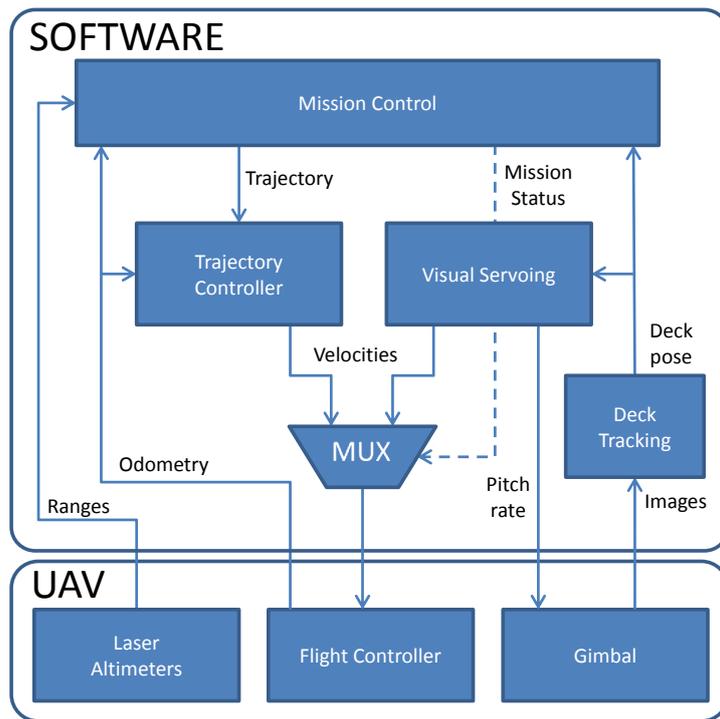


Figure 6. System architecture used in the development of the system.

The implementation of the project and the datasets used for our tests are provided as open-source for public use on our website.

4.3. Test Results

Before testing the landing outdoors, we performed several indoor experiments to develop and validate our visual servoing approach. We performed experiments both with a static and moving deck using this setup at speeds up to 2 m/s. Figure 7 shows snapshots of one of our experiments. A video of the continuous sequence of experiments can be accessed from our website.

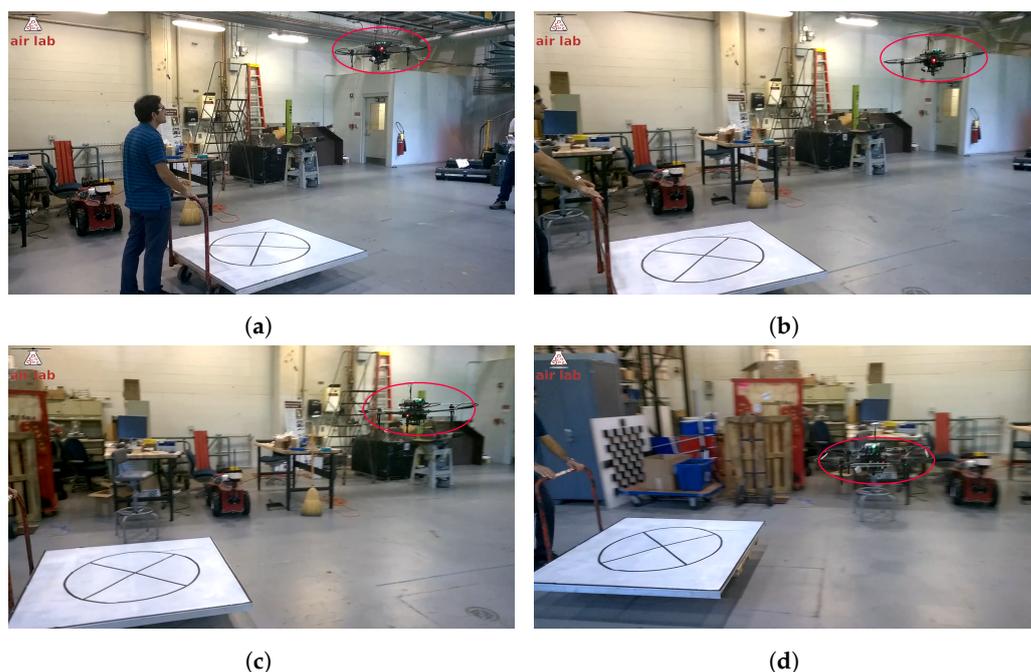


Figure 7. Cont.



Figure 7. (a–f) Screenshots from a video sequence showing our quadrotor landing on a moving platform. In this experiment, the quadrotor is additionally equipped with a DJI Guidance sensor for safe operation in the indoor environment.

Out of 19 recorded trials to land indoors on a platform moving at a speed of 5–10 km/h (1.39–2.78 m/s), there were 17 successful landings on the platform. One failure was due to loss of detection of the pattern, which happened midway to landing and the other failure was due to loss of the pattern at the last stage, which resulted in trying to land behind the deck.

We developed a simulation environment using Gazebo [42]. It has a ground vehicle (truck) of 1.5 m in height with the target pattern on top of it. The truck’s motion is controlled by a ROS node that is parameterized by the truck speed and the direction of movement.

Our UAV is simulated using the Hector Quadrotor [43], which provides a velocity-based controller similar to the one provided by DJI’s ROS SDK. The simulated quadrotor is also equipped with a (non-gimbaled) camera and a height sensor. Since the simulator has no gimbal, we fixed the camera on 45° . The odometry of the drone provides position estimates in the same message type provided by DJI. This similarity allows testing the entire software in simulation before actually flying the UAV. Therefore, the same software could be used both in simulation and on real hardware, provided that some parameters related to target detection were changed. Due to the kinematic nature of our controller, it is possible to run the trajectory controller with unchanged parameters.

Figure 8 shows a snapshot of the simulator when the quadrotor is about to land on the moving deck. The upper-left corner of the figure shows the UAV’s view of the deck.

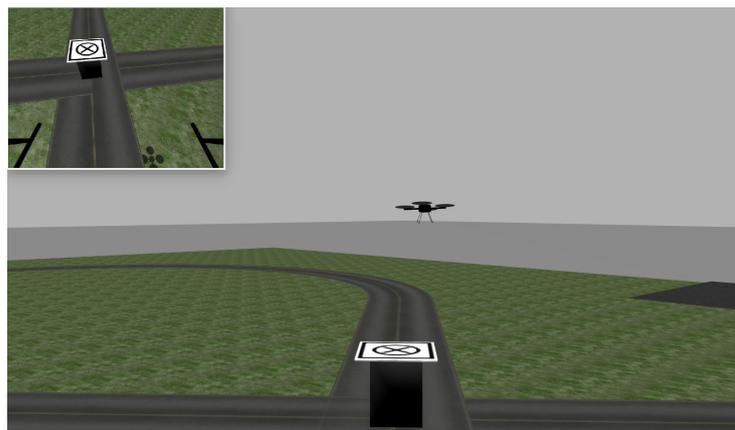


Figure 8. A snapshot of the Gazebo simulation. The quadrotor’s camera view is shown in the top left corner.

To test the system in the real scenarios, a Toro MDE eWorkman electric vehicle was modified to support the pattern at the height of 1.5 m (Figure 9). A ferromagnetic deck with a painted landing pattern was attached to the top as the landing zone for the quadrotor.



Figure 9. The Toro Workman ground vehicle used for our tests.

The ground vehicle was manually driven with an approximate speed of 15 km/h (4.17 m/s) measured by a GPS device. Visual servoing gains were empirically calibrated, and the robustness of the approach to landing was evaluated.

Although we did not collect detailed statistics on the landing procedure, there were several successful landings on the vehicle at the speed of 15 km/h (4.17 m/s). It is hard to compare our method's success rate with the other methods, as none of the publications from the MBZIRC challenge have reported their success rates and have only reported limited results from the attempts during the challenge trials.

Figure 10 shows a successful autonomous landing of the quadrotor on the moving vehicle used in our experiments at 15 km/h (4.17 m/s) speed.

Out of 22 recorded trials, visual servoing was successful in bringing the UAV to the deck in 19 cases. All three failure cases were due to the extreme sun reflection where the pattern was no longer seen in the frame; therefore, the visual servoing lost track of the pattern, and the mission was aborted. The cases where the switching from initial acceleration to visual servoing happened too soon or too late (which results in the loss of the target even at the beginning) were excluded from the analysis, as they were irrelevant to the study of the visual servoing performance. The results show the successful development of our visual servoing approach in bringing the UAV to the vicinity of the moving vehicle.

The time from the first detection of the platform to the stable landing varied between different runs from 5.8 to 6.5 s, which is less than the reported times of approaches claiming to be the fastest methods (7.65 s reported by [31] and 7.8 s reported by [11]). This timing can also be seen in the accompanying videos on our website.



Figure 10. *Cont.*

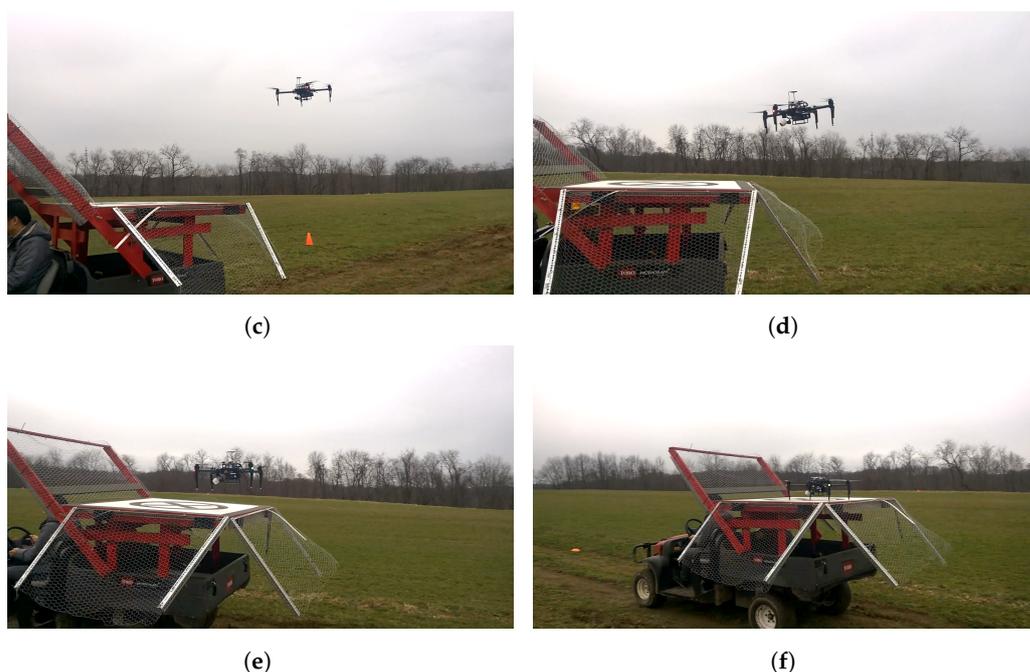


Figure 10. (a–f) Screenshots from a video sequence showing our quadrotor landing on the moving vehicle at 15 km/h (4.17 m/s) speed. The video is available on our website.

5. Conclusions

This paper presented our approach to landing an autonomous UAV on a moving vehicle. With a few modifications, such as close-range landing zone detection and vehicle turn estimation, the proposed methods can work for a generalized autonomous landing scenario in a real-world application.

The visual servoing controller showed the ability to approach and land on the moving deck in simulation, indoor environments, and outdoor environments. The controller can work with a different set of localization sensors like GPS and DJI Guidance and does not rely on very precise sensors (e.g., motion capture system).

The visual servoing controller showed promising results in reliably approaching the moving vehicle. A potential way to improve the landing efficiency further and tackle the loss of visual target problem is to divide the landing task into two subtasks: approaching the vehicle and landing from a short distance. After reaching the vehicle, the visual servoing approach can switch to a short-range landing algorithm. We believe that, by extending this approach, it can be used in a wide range of applications.

The robustness of the approach to occasional false positives in tracking the target and nondeterministic target detection rate has allowed us to use monocular vision with a real-time general-purpose ellipse detection method developed for this work (see [33]), further reducing the overall cost and weight of the system. However, as discussed in Section 4.3, this has caused the UAV to stop the landing procedure due to the sun reflection in 13.6% of our outdoor tests. If this compromise cannot be accepted in an application, a more reliable target tracking method (such as infrared or radio markers) can be devised, or another attempt should be made at finding and approaching the vehicle.

While the approach has been tested for a range of vehicle velocities up to 15 km/h, using the estimated vehicle speed as a feed-forwarding velocity means that the maximum ground vehicle's speed for UAV landing is mainly limited by the maximum speed of the UAV platform. However, the other potentially limiting factor can be the target detection rate and the gust factor, which at higher speeds can result in higher errors between two detections with insufficient time to correct the course.

Author Contributions: Conceptualization, A.K., G.A.S.P. and R.B.; methodology, A.K. and G.A.S.P.; software, A.K., G.A.S.P., R.B., R.G., P.R. and G.D.; validation, A.K., G.A.S.P. and S.S.; data curation, A.K., R.B., R.G., P.R. and G.D.; writing—original draft preparation, A.K., G.A.S.P. and R.B.; writing—review and editing, A.K., G.A.S.P., R.G., G.D., P.R. and S.S.; supervision, G.A.S.P. and S.S.; project administration, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: The project was sponsored by Carnegie Mellon University Robotics Institute and Mohamed Bin Zayed International Robotics Challenge. During the realization of this work, Guilherme A.S. Pereira was supported by UFMG and CNPq/Brazil.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available at <http://theairlab.org/landing-on-vehicle>.

Acknowledgments: The authors want to thank Nikhil Baheti, Miaolei He, Zihan (Atlas) Yu, Koushil Sreenath, and Near Earth Autonomy for their support and help in this project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
IMU	Inertial Measurement Unit
MBZIRC	Mohamed Bin Zayed International Robotics Challenge
RTK	Real-Time kinematic positioning
MPC	Model Predictive Control
RANSAC	Random Sample Consensus
RGB-D	Red, Green, Blue and Depth
MUX	Multiplexer

References

- Bonatti, R.; Ho, C.; Wang, W.; Choudhury, S.; Scherer, S. Towards a Robust Aerial Cinematography Platform: Localizing and Tracking Moving Targets in Unstructured Environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 229–236. [\[CrossRef\]](#)
- Keipour, A.; Mousaei, M.; Ashley, A.T.; Scherer, S. Integration of Fully-Actuated Multirotors into Real-World Applications. *arXiv* **2020**, arXiv:2011.06666.
- Keipour, A. Physical Interaction and Manipulation of the Environment Using Aerial Robots. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2022.
- Arora, S.; Jain, S.; Scherer, S.; Nuske, S.; Chamberlain, L.; Singh, S. Infrastructure-free shipdeck tracking for autonomous landing. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 323–330. [\[CrossRef\]](#)
- Almeshal, A.; Alenezi, M. A Vision-Based Neural Network Controller for the Autonomous Landing of a Quadrotor on Moving Targets. *Robotics* **2018**, *7*, 71. [\[CrossRef\]](#)
- Saripalli, S.; Sukhatme, G.S. Landing on a Moving Target Using an Autonomous Helicopter. In *Field and Service Robotics: Recent Advances in Research and Applications*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 277–286. [\[CrossRef\]](#)
- Wenzel, K.E.; Masselli, A.; Zell, A. Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle. *J. Intell. Robot. Syst.* **2011**, *61*, 221–238. [\[CrossRef\]](#)
- Gautam, A.; Sujit, P.B.; Saripalli, S. A survey of autonomous landing techniques for UAVs. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218. [\[CrossRef\]](#)
- Kim, J.; Jung, Y.; Lee, D.; Shim, D.H. Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1243–1252. [\[CrossRef\]](#)
- Bahnmann, R.; Pantic, M.; Popović, M.; Schindler, D.; Tranzatto, M.; Kamel, M.; Grimm, M.; Widauer, J.; Siegwart, R.; Nieto, J. The ETH-MAV Team in the MBZ International Robotics Challenge. *J. Field Robot.* **2019**, *36*, 78–103. [\[CrossRef\]](#)

11. Beul, M.; Nieuwenhuisen, M.; Quenzel, J.; Rosu, R.A.; Horn, J.; Pavlichenko, D.; Houben, S.; Behnke, S. Team NimbRo at MBZIRC 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles. *J. Field Robot.* **2019**, *36*, 204–229. [[CrossRef](#)]
12. Alarcón, F.; García, M.; Maza, I.; Viguria, A.; Ollero, A. A Precise and GNSS-Free Landing System on Moving Platforms for Rotary-Wing UAVs. *Sensors* **2019**, *19*, 886. [[CrossRef](#)]
13. Bhattacharya, A.; Gandhi, A.; Merkle, L.; Tiwari, R.; Warrior, K.; Winata, S.; Saba, A.; Zhang, K.; Kroemer, O.; Scherer, S. Mission-level Robustness with Rapidly-deployed, Autonomous Aerial Vehicles by Carnegie Mellon Team Tartan at MBZIRC 2020. *Field Robot.* **2022**, *2*, 172–200. [[CrossRef](#)]
14. MBZIRC. *MBZIRC Challenge Description*, 2nd ed.; Khalifa University: Abu Dhabi, United Arab Emirates, 2015. Available online: www.mbzirc.com (accessed on 24 August 2022).
15. Saripalli, S.; Sukhatme, G.S.; Montgomery, J.F. An Experimental Study of the Autonomous Helicopter Landing Problem. In *Experimental Robotics VIII*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 466–475. [[CrossRef](#)]
16. Borowczyk, A.; Nguyen, D.T.; Phu-Van Nguyen, A.; Nguyen, D.Q.; Saussié, D.; Le Ny, J. Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle. *J. Guid. Control Dyn.* **2017**, *40*, 1–8. [[CrossRef](#)]
17. Lee, D.; Ryan, T.; Kim, H.J. Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 971–976. [[CrossRef](#)]
18. Gonçalves, V.M.; McLaughlin, R.; Pereira, G.A.S. Precise Landing of Autonomous Aerial Vehicles Using Vector Fields. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4337–4344. [[CrossRef](#)]
19. Lange, S.; Sunderhauf, N.; Protzel, P. A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments. In Proceedings of the 2009 International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009; pp. 1–6.
20. Merz, T.; Duranti, S.; Conte, G. Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing. In *Experimental Robotics IX: The 9th International Symposium on Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 343–352. [[CrossRef](#)]
21. Xing, B.Y.; Pan, F.; Feng, X.X.; Li, W.X.; Gao, Q. Autonomous Landing of a Micro Aerial Vehicle on a Moving Platform Using a Composite Landmark. *Int. J. Aerosp. Eng.* **2019**, *2019*, 4723869. [[CrossRef](#)]
22. Xuan-Mung, N.; Hong, S.K.; Nguyen, N.P.; Ha, L.N.N.T.; Le, T.L. Autonomous Quadcopter Precision Landing Onto a Heaving Platform: New Method and Experiment. *IEEE Access* **2020**, *8*, 167192–167202. [[CrossRef](#)]
23. Wynn, J.S.; McLain, T.W. Visual Servoing for Multirotor Precision Landing in Daylight and After-Dark Conditions. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 1242–1248.
24. Romero-Ramirez, F.; Muñoz-Salinas, R.; Medina-Carnicer, R. Speeded Up Detection of Squared Fiducial Markers. *Image Vis. Comput.* **2018**, *76*, 38–47. [[CrossRef](#)]
25. Ruffier, F.; Franceschini, N. Optic Flow Regulation in Unsteady Environments: A Tethered MAV Achieves Terrain Following and Targeted Landing Over a Moving Platform. *J. Intell. Robot. Syst.* **2015**, *79*, 275–293. [[CrossRef](#)]
26. Herissé, B.; Hamel, T.; Mahony, R.; Russotto, F.X. Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow. *IEEE Trans. Robot.* **2012**, *28*, 77–89. [[CrossRef](#)]
27. Chang, C.W.; Lo, L.Y.; Cheung, H.C.; Feng, Y.; Yang, A.S.; Wen, C.Y.; Zhou, W. Proactive Guidance for Accurate UAV Landing on a Dynamic Platform: A Visual-Inertial Approach. *Sensors* **2022**, *22*, 404. [[CrossRef](#)] [[PubMed](#)]
28. Cantelli, L.; Guastella, D.; Melita, C.D.; Muscato, G.; Battiato, S.; D’Urso, F.; Farinella, G.M.; Ortis, A.; Santoro, C. Autonomous Landing of a UAV on a Moving Vehicle for the MBZIRC. In *Human-Centric Robotics, Proceedings of the CLAWAR 2017: 20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Porto, Portugal, 11–13 September 2017*; World Scientific: Singapore, 2017; pp. 197–204. [[CrossRef](#)]
29. Battiato, S.; Cantelli, L.; D’Urso, F.; Farinella, G.M.; Guarnera, L.; Guastella, D.; Melita, C.D.; Muscato, G.; Ortis, A.; Ragusa, F.; et al. A System for Autonomous Landing of a UAV on a Moving Vehicle. In *Image Analysis and Processing, Proceedings of the International Conference on Image Analysis and Processing, Catania, Italy, 11–15 September 2017*; Battiato, S., Gallo, G., Schettini, R., Stanco, F., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 129–139.
30. Baca, T.; Stepan, P.; Spurny, V.; Hert, D.; Penicka, R.; Saska, M.; Thomas, J.; Loianno, G.; Kumar, V. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J. Field Robot.* **2019**, *36*, 874–891. [[CrossRef](#)]
31. Tzoumanikas, D.; Li, W.; Grimm, M.; Zhang, K.; Kovac, M.; Leutenegger, S. Fully autonomous micro air vehicle flight and landing on a moving target using visual-inertial estimation and model-predictive control. *J. Field Robot.* **2019**, *36*, 49–77. [[CrossRef](#)]
32. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-based autonomous quadrotor landing on a moving platform. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 200–207. [[CrossRef](#)]
33. Keipour, A.; Pereira, G.A.; Scherer, S. Real-Time Ellipse Detection for Robotics Applications. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7009–7016. [[CrossRef](#)]
34. Hoffmann, G.M.; Waslander, S.L.; Tomlin, C.J. Quadrotor helicopter trajectory tracking control. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008; pp. 1–14.

35. Espiau, B.; Chaumette, F.; Rives, P. A new approach to visual servoing in robotics. *IEEE Trans. Robot. Autom.* **1992**, *8*, 313–326. [[CrossRef](#)]
36. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [[CrossRef](#)]
37. Marchand, É.; Spindler, F.; Chaumette, F. ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robot. Autom. Mag.* **2005**, *12*, 40–52. [[CrossRef](#)]
38. Corke, P.I.; Good, M.C. Dynamic effects in visual closed-loop systems. *IEEE Trans. Robot. Autom.* **1996**, *12*, 671–683. [[CrossRef](#)]
39. Jin, R.; Owais, H.M.; Lin, D.; Song, T.; Yuan, Y. Ellipse proposal and convolutional neural network discriminant for autonomous landing marker detection. *J. Field Robot.* **2019**, *36*, 6–16. [[CrossRef](#)]
40. Li, Z.; Meng, C.; Zhou, F.; Ding, X.; Wang, X.; Zhang, H.; Guo, P.; Meng, X. Fast vision-based autonomous detection of moving cooperative target for unmanned aerial vehicle landing. *J. Field Robot.* **2019**, *36*, 34–48. [[CrossRef](#)]
41. Fornaciari, M.; Prati, A.; Cucchiara, R. A fast and effective ellipse detector for embedded vision applications. *Pattern Recognit.* **2014**, *47*, 3693–3708. [[CrossRef](#)]
42. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154.
43. Meyer, J.; Sendobry, A.; Kohlbrecher, S.; Klingauf, U.; von Stryk, O. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots, Proceedings of the Third International Conference, SIMPAR 2012, Tsukuba, Japan, 5–8 November 2012*; Noda, I., Ando, N., Brugali, D., Kuffner, J.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 400–411.