Precision UAV Landing in Unstructured Environments

Kevin Pluckter and Sebastian Scherer

Abstract Autonomous landing of a drone is a necessary part of autonomous flight. One way to have high certainty of safety in landing is to return to the same location the drone took-off from. Implementations of return-to-home functionality fall short when relying solely on GPS or odometry as inaccuracies in the measurements and drift in the state estimate guides the drone to a position with a large offset from the initial position. This can be particularly dangerous if the drone took-off next to something like a body of water. Current work on precision landing relies on localizing to a known landing pattern, which requires the pilot to carry a landing pattern with them. We propose a method using a downward facing fisheye lens camera to accurately land a UAV from where it took off on an unstructured surface, without a landing pattern. Specifically, this approach uses a position estimate relative to the take-off path of the drone to guide the drone back. With the large Field-of-View provided by the fisheye lens, our algorithm can provide visual feedback starting with a large position error at the beginning of the landing, until 25cm above the ground at the end of the landing. This algorithm empirically shows it can correct the drift error in the state estimation and land with an accuracy of 40cm.

Keywords: Precision Landing · Fisheye Lens · UAV

K. Pluckter (✉) · S. Scherer
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA,
e-mail: kpluckte@cs.cmu.edu

S. Scherer
e-mail: basti@cmu.edu

1 Introduction

Landing a quadrotor autonomously is essential for the ubiquitous presence of drones. Majority of UAVs currently rely on GPS based state-estimation for landing, but drift builds up over time and these systems are unreliable if in a GPS denied environment. Given this, a drone cannot reliably land in the same spot and could potentially attempt to land in an unforgiving location, such as a nearby tree. Our proposed solutions prevents these issues by re-traversing the proven take-off path back to its starting position. We are able to precisely land the drone with this method via a monocular fisheye camera. Majority of current work on precision landing focuses on landing in a known structured environment (helipad or runway) [11]. Given a known landing pattern, these algorithms are able to identify the target and use its geometry to robustly estimate the relative state of the drone. Much work has gone into different landing pattern designs to enable accurate pose estimation [16, 18, 19]. While these methods result in accurate landings, they require the drone to land on a specific type of landing pattern. This would require drone operators to carry a landing pattern with them in order to ensure the safe return and landing of their drone. Our work focuses on a new area, landing at the UAVs starting position in an unstructured and unknown environment. This enables drones to be autonomously deployed in the field and return to their starting position while following the take-off path in reverse.

In order to accurately land a quadrotor in an unstructured environment, without prior knowledge of the take-off location several problems must be addressed: the drone must be able to localize relative to where it has taken off from and be able to guide itself to ensure a safe landing. This work addresses these problems with a method inspired by [6]. During the take-off a set of images are recorded, and during landing, the drone localizes to these images and descends along a similar path back to its initial position, as seen in Fig. 1. The approach improves the safety of landing twofold, by landing in the same starting position, there is a high likelihood of the location still being safe, and by taking a path similar to the one from take-off

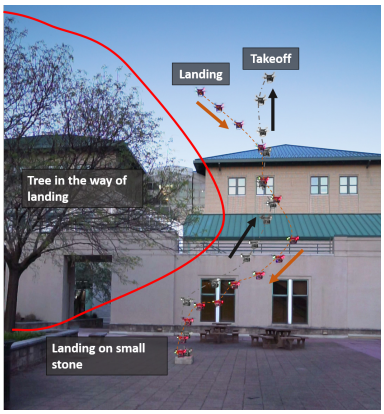


Fig. 1 Precision Landing Demonstration: the take-off path (black), and the landing path (orange). The key challenges are for the drone to safely and precisely land. This equates to the drone avoiding the tree that was avoided during take-off, and landing on the stone the drone started on. Our algorithm successfully completes both tasks.

enables the drone to avoid obstacles that were avoided during take-off. To the best of our knowledge, we present the first algorithm for precision landing of a drone in unstructured environments, such as those in Fig. 2, with an average accuracy of 40cm. In summary, the contributions of this paper are:

- An algorithm for safe precision landing of quadrotors in unstructured environments.
- Experimental results on precision landing in various environments with baseline comparisons.

2 Related Work

This section will go over work done in the area of precision landing and adjacent fields that have influenced the proposed method.

Precision Landing
Within the area of precision landing for vertical takeoff and landing (VTOL) vehicles, much work has focused on using helipad design. Specifically, helipads with concentric circles and with a H in the center are common along with other custom designed helipads [11]. Approaches using helipads with either an H or T on them use pretrained neural networks to identify the letter and then using the known geometry of either the letter or the surrounding circle to estimate 6DOF pose relative to the landing pattern [16, 18, 19]. A shortcoming of these methods is losing information on the landing pattern during the approach. Based on the size of the landing pattern and the field-of-view (FOV) of the camera, the quad-rotor will no longer be able to see the landing pattern and be unable to estimate it's pose. Approaches using more custom landing patterns use unique circular patterning with varying size to ensure good recognition and pose estimation close up and far away [2, 3, 12]. Additionally, these methods rely solely on circle detection and identifying the circles that form the correct pattern, neural networks are not used. Another approach to tackle the problem of FOV is by using two different cameras, one for approaching from a distance and another once close up [1]. As mentioned before though, these methods require there

Fig. 2: Landing locations: depicted are two of the landing locations used during testing.

to be a landing pattern of known size to be able to successfully perform an accurate landing. Without a landing pattern, other techniques and assumptions must be applied to recognize the area and land.

Visual Teach and Repeat
One particular method that has been used in similar scenarios to relatively localize and traverse long distances using a monocular camera is Visual Teach and Repeat (VTR). Primarily focusing on ground robots, this technique will "teach" a path to a robot via piloted traversal of the path, and then be able to robustly and accurately traverse the path using a monocular camera and other base level sensors for odometry [6, 7]. In order to estimate a 6DOF pose estimate, the algorithm must first recognize the image from the teach pass that it is nearest to and then estimate the pose. The pose estimation is enabled with a monocular camera with the assumption of local ground planarity. After the robot re-localizes itself, it relies upon visual odometry to take it to the next keyframe in the path.

Recent work has begun to expand this concept to quad-rotors [13, 14]. The first of these papers shows a proof of concept for VTR with a drone equipped with a downward facing camera and laser range finder. This approach builds a local 3D map is during the teach pass that the drone then localizes to during the repeat pass. The approach shows promising results, but did not look into how the technique would be affected by varying altitudes. The second paper proposes a more fully developed VTR for drones, but uses a forward facing camera and a qualitative position estimation. From these various approaches to VTR, we have developed the proposed method for precision landing, using a similar architecture, but a different method for generation of motion command and transitioning between keyframes.

Visual Servoing
The premise of visual servoing is to to directly control the robot using vision [4, 5]. One type of visual servoing is Position Based Visual Servoing (PBVS), where features are extracted in an image from an object of known size to command an input towards a goal pose. PBVS is similar in nature to the methods used landing patterns where a model is known. In the proposed work, there is no model of the position from which the drone took off, but a planarity assumption for the ground from which the quad-rotor took off can be made. From this assumption, we can estimate pose from one camera frame to another if the height at each position is known. With scale abmiguity is solved, a method similar to PBVS can be used for the controller input.

3 Approach

The approach is inspired by Visual Teach and Repeat [6], where the take-off is the teach pass and the landing is the repeat pass. In this section, we will discuss the details of this method: the teach pass (take-off), the repeat pass (landing), position

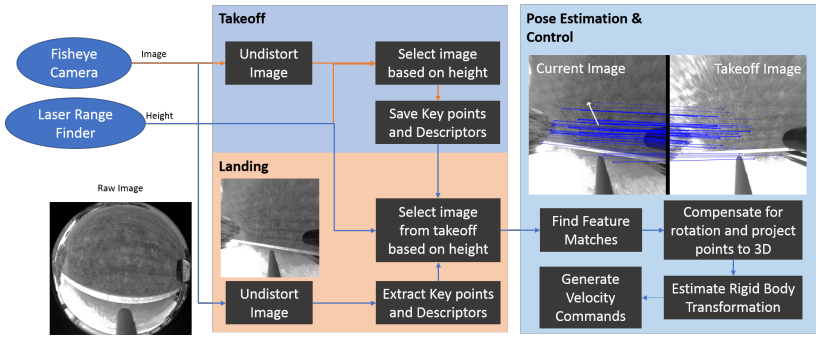


Fig. 3: Pipeline of algorithm with steps during take-off having orange arrows and steps during landing having blue arrows. Additionally, a sample of a raw image, undistorted image, and matched images are shown. In the matched image sample above the Pose Estimation and Control section shows how matches are found on the image and a resultant white arrow displays the motion required to minimize the position error.

estimation, and control will be explained. The entire pipeline can be seen in Fig. 3, which runs at 15 Hz onboard on an NVIDIA Jetson TX2.

Take-off & Landing

During take-off, a set of images is recorded in regular intervals with a fisheye lens camera. The wide FOV of the fisheye provides key information at the start of the take-off when the camera is very close to the ground and good information at higher altitudes. Images are recorded at a higher rate while the drone is close to the ground and at a slower rate as the drone ascends higher. Details on the image recording rate and other parameters used for the method can be found here. Once all of the images are recorded, key points are found and ORB descriptors are extracted and saved to be used during the landing phase. This process only takes seconds and allows for faster run-time during landing by not performing repetitive calculations.

For landing, the drone comes back to the final position from take-off at 8 m height. Then the current image is compared with the image taken at an altitude just below the current altitude of the drone. A relative position estimate to the image from take-off is estimated in the form of a 2-D rigid body transform. This estimate is then used to command the drone towards that position. While the drone approaches the position of the image from take-off, it constantly descends. Once the drone goes below the height of the current take-off image being compared to, the next closest image is used. This process commands the drone along a similar path from take-off back to it's original position on the ground.

To initialize the landing process, the drone uses its state-estimate to fly above where it started. Once the drone is in position it needs to be able to localize. In order to accomplish this task we used a downward facing fisheye camera and a set of images collected during take-off. A 180° FOV fisheye lens was selected as it provides much

more information about the ground below the drone than a pinhole lens. This is particularly beneficial at the start and end of the landing, allowing for a larger offset at the start, 8m above ground, and for more reliable motion commands until it is 0.25m above the ground. The images are projected to a flat image plane [9, 10] and ORB features are extracted from the drone's current image and matched the descriptors from the corresponding image from take-off in order to localize during the descent [15]. ORB features were selected over learned neural network features and features made for spherical images as literature shows that they are much slower [17, 20], which would hinder the speed of the control loop.

Once features are matched, the pose is estimated relative to an image from take-off. First, the feature points' position is corrected using the estimated roll and pitch. Then they are projected into 3-D space using a planar assumption, the measured height, and intrinsic camera parameters. These two steps can be accomplished by rearranging the rotational flow correction equation from [8] and use it for correcting the rotation of an individual point and project it into 3D space:

$$\mathbf{P}_z = \mathbf{Z}\cos(\theta_x)\cos(\theta_y) \tag{1}$$

$$\mathbf{P}_x = \frac{(-p_x - \sin(\theta_y)f)\mathbf{P}_z}{f} \tag{2}$$

$$\mathbf{P}_y = \frac{(-p_y + sin(\theta_x)f)\mathbf{P}_z}{f} \tag{3}$$

Where $\mathbf{Z}$ is the measured height from the laser range finder, θ is the current estimated roll and pitch of the drone, $\mathbf{P}$ is a feature point's metric coordinates in 3D with respect to the drone's current position, p is the feature point's position in the camera frame, and f is the estimated focal length of the image. Once both sets of points are projected onto the ground plane, a 2D rigid body transformation can be calculated and give a metric relative position error. Random sample consensus (RANSAC) is used to reject outliers while finding the rigid body transform. If there is a low consensus during RANSAC, the drone ascends to increase its FOV until it can again match with the image from take-off. This pipeline can be seen in the Pose Estimation & Control section in Fig. 3.

Once the position error is estimated, it is used as an input to a PI position controller using the drone's onboard state-estimate. The drone receives velocity commands from the position controller during the descent at 15Hz until it is 0.25m above the ground when it completes the landing. At 1.5m the drone aligns with the concurring image from take-off before finally descending.

4 Experiments

In this section, we discuss the experiments and results. We perform a drone landing experiment to demonstrate the performance and robustness of our precision landing algorithm, as well as present a baseline comparison to landing using only state

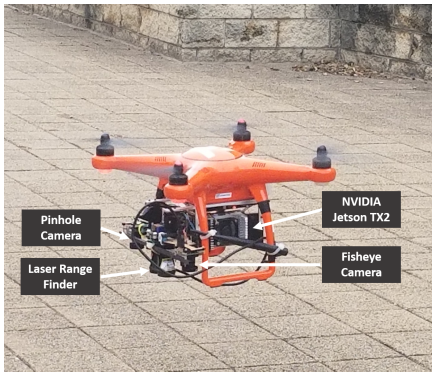


Fig. 4 Experimental Setup: experiments were conducted with the X-Star Premium Quadcopter by Autel Robotics. A NVIDA Jetson TX2, ueye camera equipped with fisheye camera lens, and SF30 laser range finder are equipped. The Jetson TX2 is connected to the X-Star Premium's embedded computing device. A second ueye camera is attached with a pinhole lens for offline comparison.

estimate. Additionally, data is collected and presented for comparison of the efficacy of both fisheye and pinhole camera lens.

A second experiment was performed to test the algorithms robustness to scene change. Landing tests were performed with a van next to the drone for takeoff, and before landing, the van was moved to a set further distance as shown in Fig. 6. The setup for our experiments can be seen in Fig. 4 and all the data collected can be found here.

Landing Experiments and Comparisons

The experimental procedure is as follows. First, an environment is selected and the drone is placed in its initial position. For measurement purposes, the drone is placed next to either a natural landmark, such as a corner of a tile, or an artificial landmark, such as a golf tee in the grass. Second, the drone is manually piloted with varying paths taken (i.e. straight, diagonally, curved, etc.). Once the takeoff is completed, it is flown in a random direction in varying amounts to simulate drift from the drone's state-estimate. Finally, the landing procedure is initiated and the drone autonomously lands. After the drone has landed, measurements are taken for the drone's displacement from the initial position. These will be compared with results using a GPS & Intertial Measurement Unit (IMU) based Extended Kalman Filter (EKF) state-estimation. This test has been performed in a wide range of environments, grass, turf, and stone tiling. Wind speeds are recorded from METAR report from the Allegheny County Airport (KAGC) to observe the algorithms robustness to high wind speeds and gusty weather. The performance for the proposed method and the GPS-IMU based EKF can be seen in Table 1.

Data is recorded from both the fisheye and pinhole cameras and analyzed offline to determine the reliability of position estimation. The algorithm is ran on the data and if RANSAC cannot determine a consensus on a rigid body transform, the image instance is considered a failed localization. The percentage of failed localizations across multiple trials can be seen in Fig. 5. This shows that the fisheye lens can localize well even when close to the ground and has a lower failure rate throughout the landing than the pinhole.

Method	Environment	Number of Trials	Accuracy (cm)	Standard Deviation (cm)	Wind Speed (m/s)	Max Gust (m/s)
Precision Landing	Patio	18	38	23	3	5
Precision Landing	Grass 1	17	44	42	3	5
Precision Landing	Turf	10	34	17	7	12
Precision Landing	Grass 2	10	45	24	8	15
State Estimate	Patio	10	183	74	2	0
State Estimate	Grass 1	10	359	205	2	0

Table 1: Precision-Landing and State-Estimate Landing Results

Scene Change Experiments

The experimental procedures for the scene change experiments are the same as the landing with the addition that a van moves from a set initial position to a new set position further away. Three sets of five experiments were ran with each set having a different starting and ending offset of the van from the drone's initial position. The results can be seen in Fig. 7.

5 Conclusions and Future Work

The results from the experiments show that our algorithm can safely and accurately land a quadrotor to its original takeoff position in various environments, and that it outperforms the IMU-GPS EKF state-estimate by a statistically significant margin. Additionally, our algorithm proves to be robust to wind as the results do not vary with the wind conditions. There have been two observed failure case, if the ground plane has a sudden large change in height, in which case the planar assumption is no longer valid and an accurate rigid body transform cannot be calculated, and when the scene has a large amount of change.

From the fisheye and pinhole camera lens comparisons, we conclude that the fisheye lens, although having high radial distortion, provides more information for the task of autonomous precision landing. This is particularly useful in two scenarios. Firstly, when a drone initially starts its descent. This can be seen in the results when there is a large offset from the end of the takeoff to the beginning of the landing. Overall, this results in a more robust landing capability as the drone relies less on the state-

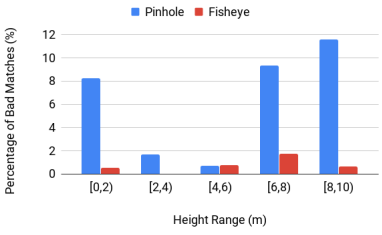


Fig. 5 Comparison of Fisheye and Pinhole Camera Lens for the task of localization. Each bar displays the percentage of bad position estimations during landing across various trials in the different environments. The fisheye camera has fewer bad matches than pinhole camera.

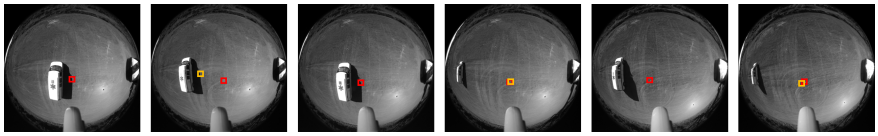


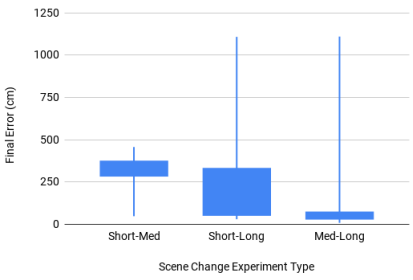
Fig. 6: Displays the three scene change experiments. From left to right is each experiment: Short-Med, Short-Long, and Med-Long, with the take-off location in red and landing in yellow.

estimate. Secondly, when the drone is closer to the ground and there are higher levels of wind. In essence, due to the fisheye lens' greater FOV, even if pushed off-course by wind, it maintains visual of the image from takeoff and can recover.

The scene change experiment demonstrated the effects of a change in the environment for the landing algorithm with three different scenarios. The first where a large object that provides many of the features in the images moves to a new position relatively close to where it started. This causes the landing algorithm to be drawn off course and land close to the area with the highest number of the original features. This short coming gives interesting insight on potential other uses for the proposed method. One, landing on a moving landing pattern. If the landing pattern is highly textured, even if the landing pad were moving in the environment, the drone would be drawn towards it. Second, if a highly textured landing pattern was used, the drone could change where is lands relatively easily. The other two scenarios where the van moved far from its initial position resulted in accurate landings. Each of these scenarios had one edge case of the drone following the van, likely because the van had not moved far enough away. From these two results it can be seen that if strong features are removed, the drone can still land accurately.

There are several interesting areas to be explored for future work. One topic to further explore is landing on a moving landing site with a highly textured surface. Another area for future work would be the final control piece for the landing when the camera can no longer match features.

Fig. 7 Results from scene change experiments. Each box-and-whisker plot shows the resulting accuracy across the three scene change tests. Short-Med means the van moved from 4ft from the drone to 16ft from the starting point. For Short-Long, the van moves from 4ft away to 41ft. For Med-Long the van moves from 16ft to 41ft.



6 Acknowledgements

This work was supported by Autel Robotics under award number A018532.

References

[1] Sankalp Arora, Sezal Jain, Sebastian Scherer, Stephen Nuske, Lyle Chamberlain, and Sanjiv Singh. Infrastructure-free shipdeck tracking for autonomous landing. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 323–330. IEEE, 2013.

[2] Wei Bai, Feng Pan, Bo Yang Xing, Chao Pan, and Meng Xin Pei. Visual landing system of uav based on adrc. In *Control And Decision Conference (CCDC), 2017 29th Chinese*, pages 7509–7514. IEEE, 2017.

[3] Alessandro Benini, Matthew J Rutherford, and Kimon P Valavanis. Real-time, gpu-based pose estimation of a uav for autonomous takeoff and landing. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3463–3470. IEEE, 2016.

[4] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[5] François Chaumette and Seth Hutchinson. Visual servo control. ii. advanced approaches [tutorial]. *IEEE Robotics & Automation Magazine*, 14(1):109–118, 2007.

[6] Lee Clement, Jonathan Kelly, and Timothy D Barfoot. Monocular visual teach and repeat aided by local ground planarity. In *Field and Service Robotics*, pages 547–561. Springer, 2016.

[7] Lee Clement, Jonathan Kelly, and Timothy D Barfoot. Robust monocular visual teach and repeat aided by local ground planarity and color-constant imagery. *Journal of Field Robotics*, 34(1):74–97, 2017.

[8] Dominik Honegger, Lorenz Meier, Petri Tanskanen, and Marc Pollefeys. An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1736–1741. IEEE, 2013.

[9] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[10] Juho Kannala, Janne Heikkilä, and Sami S Brandt. Geometric camera calibration. *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

[11] Weiwei Kong, Dianle Zhou, Daibing Zhang, and Jianwei Zhang. Vision-based autonomous landing system for unmanned aerial vehicle: A survey. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pages 1–8. IEEE, 2014.

[12] Torsten Merz, Simone Duranti, and Gianpaolo Conte. Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In *Experimental Robotics IX*, pages 343–352. Springer, 2006.

[13] Trung Nguyen, George KI Mann, Raymond G Gosine, and Andrew Vardy. Appearance-based visual-teach-and-repeat navigation technique for micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 84(1-4):217–240, 2016.

[14] Andreas Pfrunder, Angela P Schoellig, and Timothy D Barfoot. A proof-of-concept demonstration of visual teach and repeat on a quadrocopter using an altitude sensor and a monocular camera. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 238–245. IEEE, 2014.

[15] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[16] Srikanth Saripalli, James F Montgomery, and Gaurav S Sukhatme. Visually guided landing of an unmanned aerial vehicle. *IEEE transactions on robotics and automation*, 19(3):371–380, 2003.

[17] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[18] Guili Xu, Yong Zhang, Shengyu Ji, Yuehua Cheng, and Yupeng Tian. Research on computer vision-based for uav autonomous landing on a ship. *Pattern Recognition Letters*, 30(6):600–605, 2009.

[19] Shaowu Yang, Sebastian A Scherer, and Andreas Zell. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1-4):499–515, 2013.

[20] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang. Sphorb: A fast and robust binary feature on the sphere. *International Journal of Computer Vision*, 113(2):143–159, 2015. doi: 10.1007/s11263-014-0787-4.