Autonomous Precision Drone Landing with Fiducial Markers and a Gimbal-Mounted Camera for Active Tracking

Joshua Springer, Marcel Kyas

Abstract—Precision landing is a remaining challenge in autonomous drone flight, with no widespread solution. Fiducial markers provide a computationally cheap way for a drone to locate a landing pad and autonomously execute precision landings. However, most work in this field has depended on fixed, downward-facing cameras which restrict the ability of the drone to detect the marker. We present a method of autonomous landing that uses a gimbal-mounted camera to quickly search for the landing pad by simply spinning in place while tilting the camera up and down, and to continually aim the camera at the landing pad during approach and landing. This method demonstrates successful search, tracking, and landing with 4 of 5 tested fiducial systems on a physical drone with no human intervention. Per fiducial system, we present the number of successful and unsuccessful landings, and the distributions of the distances from the drone to the center of the landing pad after each successful landing, with a statistical comparison among the systems. We also show representative examples of flight trajectories, marker tracking performance, and control outputs for each channel during the landing. Finally, we discuss qualitative strengths and weaknesses underlying the performance of each system.

I. INTRODUCTION

Landing is a remaining task of autonomous drone flight that does not yet have a widespread solution. Finding some reliable method for autonomous landing is a necessary step in enabling fully autonomous mission cycles, and therefore several projects offer potential solutions (see Section III). Although GPS is the main navigational tool of drones in general, it does not provide enough positioning accuracy for landing in many situations (e.g. bad weather, urban canyons, even certain areas of the world). One of the main methods for increasing positioning accuracy in drone landing is to mark a known landing site with fiducial markers (see Section II) that provide a computationally cheap way for the drone to estimate its position relative to the landing pad using just a monocular camera – arguably the most common drone peripheral sensor. However, to this end, most projects to date have used a fixed, downward-facing camera, which makes it easy for the drone to lose sight of the landing pad in the event of antagonistic events such as wind gusts. It also makes it more difficult for the drone to find a landing pad because it must move a large distance in order to see new areas.

In this paper, we evaluate a method of landing with fiducial markers and a gimbal-mounted camera that allows the drone to track the landing pad during approach and descent, and also allows the drone to search for the landing pad by merely spinning in place and tilting the camera up and down. The gimbal-mounted camera also increases the complexity of the system in multiple ways. First, the tracking system requires some extra, relatively simple components in order to aim the camera at the gimbal correctly – typically the pixel position of the marker in the camera frame as input, and some controller to generate an output signal for the gimbal. Second, whereas systems with downward-facing cameras can use only the position of the detected marker to direct the drone during a landing, systems with gimbal-mounted cameras require both the position of the detected marker, and a coordinate system transform that takes into account the orientation of either the camera/gimbal or the marker itself. The problem here is that many commercially-available gimbals do not provide their orientation data as an output, and the orientations of fiducial markers are subject to ambiguity in many cases (see Section II). It is also possible to add more sensors, e.g. an IMU, to the camera to extract its orientation, but we prefer to minimize requirements of physical components in order to make the system more generalizable. Instead, we assume that the landing pad is relatively level – a warranted assumption in any reasonable case – and carry out the required pose transforms using the somewhat unreliable orientation of the detected fiducial marker. As a precursor to this study, we first conducted tests of 5 fiducial systems to determine their prevalence of orientation ambiguity [8]. We have also conducted tests of the landing method itself in simulation with success [14]. We use these 2 precursor tests to inform our real world drone tests in this paper.

II. BACKGROUND

Fiducial markers – such as April Tag [9], WhyCode [11], ARTag [5], ArUco [6], etc. – are 2D patterns whose pose (position + orientation) can be determined computationally cheaply using monocular images. While most fiducial systems provide accurate estimates of position (3D translation from the camera to the marker), their orientation estimates are often ambiguous as a result of the limitations of embedding 2D patterns into 3D space. The ambiguity manifests as sign flips in the components of the orientation of the recognized pose, and propagates through subsequent calculations, e.g. coordinate system transforms. This can cause erratic behavior if a drone control signal derives from a transformed pose.

In a precursor study [8], we have evaluated 2 existing fiducial systems – April Tag 48h12 (Figure 1d) and WhyCode (Figure 1c) – and 3 custom fiducial systems that we have made through modifications of the existing systems – April Tag 24h10 (Figure 1b), WhyCode Ellipse (Figure 1c), and WhyCode Multi (Figure 1a). We evaluate them in terms of orientation ambiguity and runtime framerate when executing on a Raspberry Pi 4. Our modifications are as follows: WhyCode Ellipse changes the sampling locations for the decision problem that determines the orientation of WhyCode markers, WhyCode Multi uses the positions of multiple coplanar markers to determine the orientation of the plane connecting
Some projects have accomplished precision drone landing with fiducial markers and a fixed camera. In some cases \cite{18}, \cite{17}, they use multiple ArUco markers of different sizes to allow a drone to maintain detection of the landing pad as it approaches very close. In the case of \cite{4}, the drone detects a single, custom X-shaped marker with 2 fixed cameras – one pointing down and the other pointing forward and down. In \cite{13}, the drone has a front-facing camera and a downward facing camera, and detects a single AR Tag for landing on a moving boat. The same is also done in \cite{10} but with an April Tag marker. It is also possible to use GPS for an initial approach, with a final approach guided by a fiducial marker – in one case, April Tag \cite{2}. One method \cite{12} uses a DJI Phantom 4, which has a gimbal-mounted camera, but keeps the gimbal pointed 90° down (at the ground) during the landing, when detecting fiducial markers. The drone in \cite{11} uses a fixed, downward facing camera, but does address the issue of orientation ambiguity using many bundled April Tag markers, each of which estimates the location of the drone relative to the landing pad, and uses a voting scheme to choose which estimate it believes. In \cite{3}, the drone has a single camera facing forwards and down, and detects April Tag markers on its landing pad. Finally, one method \cite{15} uses a closed-source, commercialized marker system called Lentimark \cite{16}, which mitigates the planar pose ambiguity problem using Moiré patterns on the outside of otherwise conventional AR Tags. They embed a single Lentimark marker inside of an AR Tag, mount them on a post, and land a drone autonomously while tracking the marker during landing with a gimbal-mounted camera. The drone lands on the ground in front of the post.

Some of these projects report visual loss of the marker during landing, as a result of the lack of camera tracking. We contribute a method that goes beyond these projects to actively track the marked landing pad with a gimbal-mounted camera independently of the drone’s movement, enabling both autonomous precision landing in unfriendly environments (e.g. wind) and the ability to easily and safely search for the marker by simply spinning in place and tilting the camera up and down. The drone lands on top of the marked landing pad. We also test new fiducial systems that have not yet appeared in such autonomous landing scenarios, and we focus only on widespread, open source marker systems that have a large community using, supporting, and developing them.

\section{Related Work}

Some projects have accomplished precision drone landing with fiducial markers and a fixed camera. In some cases \cite{18}, \cite{17}, they use multiple ArUco markers of different sizes to allow a drone to maintain detection of the landing pad as it approaches very close. In the case of \cite{4}, the drone detects a single, custom X-shaped marker with 2 fixed cameras – one pointing down and the other pointing forward and down. In \cite{13}, the drone has a front-facing camera and a downward facing camera, and detects a single AR Tag for landing on a moving boat. The same is also done in \cite{10} but with an April Tag marker. It is also possible to use GPS for an initial approach, with a final approach guided by a fiducial marker – in one case, April Tag \cite{2}. One method \cite{12} uses a DJI Phantom 4, which has a gimbal-mounted camera, but keeps the gimbal pointed 90° down (at the ground) during the landing, when detecting fiducial markers. The drone in \cite{11} uses a fixed, downward facing camera, but does address the issue of orientation ambiguity using many bundled April Tag markers, each of which estimates the location of the drone relative to the landing pad, and uses a voting scheme to choose which estimate it believes. In \cite{3}, the drone has a single camera facing forwards and down, and detects April Tag markers on its landing pad. Finally, one method \cite{15} uses a closed-source, commercialized marker system called Lentimark \cite{16}, which mitigates the planar pose ambiguity problem using Moiré patterns on the outside of otherwise conventional AR Tags. They embed a single Lentimark marker inside of an AR Tag, mount them on a post, and land a drone autonomously while tracking the marker during landing with a gimbal-mounted camera. The drone lands on the ground in front of the post.

Some of these projects report visual loss of the marker during landing, as a result of the lack of camera tracking. We contribute a method that goes beyond these projects to actively track the marked landing pad with a gimbal-mounted camera independently of the drone’s movement, enabling both autonomous precision landing in unfriendly environments (e.g. wind) and the ability to easily and safely search for the marker by simply spinning in place and tilting the camera up and down. The drone lands on top of the marked landing pad. We also test new fiducial systems that have not yet appeared in such autonomous landing scenarios, and we focus only on widespread, open source marker systems that have a large community using, supporting, and developing them.

\section{Methods}

\subsection{System Overview}

Our testing platform is a DJI Spark, which provides stable performance, the ability to test indoors (reducing logistical considerations), little risk in the case of crashes or malfunctions, and access to the DJI Mobile SDK for autonomous control. We have created a custom app \cite{7} based on the DJI Mobile SDK code samples, which decodes video frames from the Spark via its controller and offloads them to an external companion board for analysis. The companion board – a Raspberry Pi 4 – runs software to detect fiducial markers in the images, then generates control commands and sends them to the drone via the app and controller. While the control system stays the same, the landing pad and fiducial software change, to test each of the 5 fiducial systems in the precursor study (see Section \ref{sec:methods}). Each of the fiducial systems provides exactly the same attributes to the control system: the relative position from the drone to the landing pad (a position target), the pixel position \(u, v\) of the landing pad in the camera frame (normalized so \(u, v \in [-1, 1]\) for ease of use, where \((0, 0)\) is the center), and the yaw of the landing pad (for alignment in the later stages of landing). The control system constrains these attributes to specific intervals depending on the landing phase, and then passes to them DJI Mobile SDK as VirtualStick inputs, such that they appear to the system as input from the controller. The autopilot software uses these as velocity setpoints. The position targets control the drone’s translational velocity, and the normalized pixel positions \(u, v\) control the tilt of the gimbal and the yaw of the drone during approach.

The control policy has several phases. During \textit{takeoff}, the drone ascends in place to an altitude of 1.2 meters (fully automated within DJI Mobile SDK). It then transitions to \textit{search}, where it spins in place while tilting the camera up and down. After it finds the marker, it can transition to

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{The 4 landing pads in this paper.}
\end{figure}
approach, where it moves toward the landing pad quickly, without changing altitude. This phase also has a “deadzone” parameter that disables planar movement if the drone is within a small planar distance to the marker. Then it enters the yaw align phase, where it spins in order to align with the yaw of the landing pad. Once it has aligned, it enters descent and decreases altitude until a specific minimum altitude (different for each marker, see Section V), while correcting its horizontal position above the landing pad. It then enters a landing commit phase, where the DJI Mobile SDK controls its descent and touchdown detection, and disables the motors. Finally, the drone has landed.

At the start of each landing attempt, the drone and landing pad are always placed a constant distance \( d = 2.5 \text{m} \) from each other, with the drone facing directly away from the landing pad so that it must search for the landing pad after takeoff. At the first landing attempt, the landing pad faces away from the drone, and it is rotated clockwise 18° after each of the 20 landing attempts, in order to simulate approaches from all directions. We consider a landing attempt successful if it requires no human intervention except starting the landing process, and if the drone touches down fully on the landing pad (not partially touching the ground).

**B. Basis of Comparison**

We compare the landing performance of the system using each of the 5 fiducial systems, keeping all other factors the same. The markers on each landing pad have the same width at the widest point, although they have different areas as a result of being different shapes. We compare the systems on the number of successful landings they produce, and on the accuracy of the landing - the ability of the system to minimize the distance from the camera to the center of the marker after touchdown. This distance should be minimized. A Kruskal-Wallis test determines any statistically significant differences in the accuracies of the landings produced by each marker system, where a p-value of \( p_{>0.05} < 0.05 \) implies that we should reject the null hypothesis that there is no statistical difference among the groups. Pairwise tests rank the systems against each other, where the result of a two-tailed Wilcoxon test \( p_{<0.05} \) implies that the two systems show some statistical difference, and the result of a two-tailed Wilcoxon test \( p_{>0.05} < 0.05 \) implies that the first system results in statistically more accurate landings than the second. If \( p_{>0.05} \geq 0.05 \), we do not calculate \( p_{>0.05} \), because we cannot determine that the two systems are statistically different.

**V. RESULTS**

Table [I] shows an overview of the performances of the system using each fiducial system. All systems except WhyCode Multi achieved 20 successful landings, with varying numbers of failed landing attempts. WhyCode Multi achieved no successful landings. Figure [2] shows the distribution of the distances from the camera to the center of the landing pad, which is ideally 0. Lower values imply more accurate landings. Table [II] shows the result of a Kruskal-Wallis test on the distances from the camera to the center of the landing pad, concluding a statistical difference among the groups. Table [III] shows a result of pairwise Wilcoxon tests on the same data, concluding that April Tag 48h12 outperforms all other systems, and all the other systems are not statistically different from each other.

Figure [3] shows the perceived positions of the drone during its approach during one landing that is representative of the whole. Figure [4] shows the position of the landing pad in the camera frame during the same landing. Figure [5] shows the control output of the drone during that landing.

**VI. DISCUSSION**

In the given landing scenario, 4 out of 5 systems demonstrated successful landings, with April Tag 48h12 providing the closest-to-ideal landings. April Tag 24h10, WhyCode Ellipse, and WhyCode Orig have statistically similar levels of accuracy, as a result of their limitations. April Tag 24h10’s high orientation ambiguity means it cannot guarantee proper

| System          | Successes | Failures |
|-----------------|-----------|----------|
| April Tag 48h12 | 20        | 3        |
| April Tag 24h10 | 20        | 5        |
| WhyCode Ellipse | 20        | 2        |
| WhyCode Orig    | 20        | 4        |
| WhyCode Multi   | 0         | 10       |

**TABLE II: A Kruskal-Wallis test to determine if there are statistical differences in the systems’ abilities to accurately land the drone. It finds a statistical difference.**
| System 1          | System 2          | $p_{eq}$ | $p_{2}$ |
|------------------|-------------------|---------|---------|
| April Tag 48h12  | April Tag 24h10   | 0.014   | 0.007   |
| April Tag 48h12  | WhyCode Ellipse   | 0.001   | 0.000   |
| April Tag 48h12  | WhyCode Orig      | 0.005   | 0.003   |
| April Tag 24h10  | April Tag 48h12   | 0.014   | 0.993   |
| April Tag 24h10  | WhyCode Ellipse   | 0.701   | 1.000   |
| WhyCode Ellipse  | April Tag 48h12   | 0.001   | 1.000   |
| WhyCode Ellipse  | April Tag 24h10   | 0.701   | 0.643   |
| WhyCode Orig     | WhyCode Ellipse   | 0.643   | 0.997   |
| WhyCode Orig     | April Tag 24h10   | 0.985   |         |
| WhyCode Orig     | WhyCode Ellipse   | 0.643   |         |

TABLE III: A pairwise Wilcoxon test to determine a possible ranking of the landing systems by lowest distance from the ideal landing location. April Tag 48h12 outperforms the other systems, and the other systems do not have significant differences.

Fig. 3: An example landing trajectory, representative of the set of landing attempts. The East (left/right) position target reduces quickly as the drone turns to face the landing pad. The North (forward/backward) position target decreases slowly as the drone approaches the landing pad. Once both the North and East position targets are adequately small, the drone rotates to align with the landing pad, reducing its yaw displacement. Finally, the Up position target gets closer to 0 at the end of the landing, during the descent phase. The position targets are no longer generated once the drone reaches the landing commit phase, as the DJI Mobile SDK then forces the camera to point directly forward such that it can no longer see the landing pad.

Fig. 4: The normalized pixel positions of the landing pad during an example landing. The main takeaway from this is that the drone was able to keep the marker relatively centered (i.e. near $(u_n, v_n) = (0, 0)$) in the camera frame even as it was moving towards the landing pad and changing its orientation. The minor variations in the landing pad’s pixel positions do not negatively affect the landing process.

positioning over the landing pad during approach, however it does allow the drone to track the landing pad above a height of 0.4 meters. WhyCode Ellipse and WhyCode Orig, while exhibiting less orientation ambiguity, require higher landing commit heights, below which the rest of their landing is blind, which could be problematic in windy conditions.

All systems experienced occasional visual loss of the landing pad, contributing to the small number of landing failures. In the case of the April Tag systems, this visual loss typically occurred upon initial acquisition of the landing pad, probably because of the intricate structure of April Tag markers, as well as the fact that initial acquisition happens from a relatively long distance and under motion. Once the April Tag markers are identified, however, they provide reliable detection even at very close distances, so that the drone can commit to the landing later, thereby relinquishing control at a safer altitude. On the other hand, the WhyCode markers exhibit reliable long-distance detection even under motion, and present problems at close range, as they begin to eclipse the camera frame completely. They therefore require a higher landing commit altitude, which means they execute more of the landing blindly than April Tag markers allow. These results exactly echo findings from our original precursor study [14], where we tested this algorithm in simulation, finding that April Tag markers are very sensitive to obstructions, while WhyCode markers are robust to some obstructions. Finally, some visual loss of the landing pad occurs simply as a result of intense sun glares or shadows. This is an inherent problem in identifying markers in the visible spectrum and must be considered when implementing such a landing system.

Perhaps the most interesting finding of this study is that the orientation ambiguity problem does not prohibit autonomous
frames can reach the app relatively quickly, the tablet is not an app for e.g. authentication with DJI’s servers. While video architecture imposed by the DJI Mobile SDK, which requires outputs, however.

Ultimately, it would be better to use some filtering or outlier detection to attempt to entirely remove these erroneous control signals as a result of orientation ambiguity should be explored. However, these discontinuities resulted only in minor disturbances in the behavior of the drone, rather than fully destructive interference. Importantly, most of the discontinuities occur when the drone is almost directly above the landing pad, since the orientation is most ambiguous when the camera is normal to the fiducial marker. In these cases, the drone’s planar radius to the landing pad is small, and so it is likely to be inside the deadzone, where the control outputs are reduced to 0. Ultimately, it would be better to use some filtering or outlier detection to attempt to entirely remove these erroneous control outputs, however.

A main obstacle in this setup is the restrictions on the system architecture imposed by the DJI Mobile SDK, which requires an app for e.g. authentication with DJI’s servers. While video frames can reach the app relatively quickly, the tablet is not representative of the hardware that will eventually be embedded onboard a drone, and it is difficult to run the necessary fiducial software within the Android environment. Therefore, the video frames were offloaded via a WiFi connection to a Raspberry Pi 4. The task of compressing and transmitting the images reduced the maximum processing framerate to about 6-7 Hz, with an inconsistent delay of between 0.5 and 2 seconds from acquisition of an image to a corresponding command from the companion board. The low framerate and inconsistent processing times caused the bulk of the problems in these experiments. This latency can be solved by using a different drone platform with an onboard processing unit and using only wired connections, thereby avoiding image compression and transfer. In such a context, the control policy could be less conservative, with higher limits on the control outputs, more tightly tuned gimbal tracking, etc.

WhyCode Multi unfortunately did not demonstrate any successful landings, although it did provide good performance in the search phase and the initial part of the approach phase. However, it invariably overshot the landing pad every time. While it does appear to output correct position targets in lab experiments, it seems that its position targets are highly dependent on the angle from which the marker is viewed. The camera for the spark can only point between 0° (straight forward) and −85° down. It is possible that the drone could recover from an overshoot if the gimbal were able to point backwards at the marker, but we pass this test to a future study.

VII. Conclusion

We have shown that autonomous precision landing is possible with fiducial markers and a gimbal-mounted camera for active tracking. Of our tested fiducial systems, 4 out of 5 have demonstrated success in the real world, even with high processing latency - with the only non-successful system being WhyCode Multi. Our platform is a DJI Spark which does not have extra onboard processing hardware, but all essential image processing and control signal generation is possible onboard a Raspberry Pi, meaning that all hardware and processing can be embedded into a single drone system in the future. We have also demonstrated that the gimbal-mounted camera setup allows the drone to search for the landing pad by simply spinning in place and sweeping its gimbal up and down, thereby quickly scanning a large search area with little effort.

VIII. Future Work

The natural next step is to embed the companion board into a drone and partially re-conduct this proof of concept with a larger drone outdoors. This will significantly reducing the processing latency and likely improve performance. It would be helpful to re-test the WhyCode Multi setup using a gimbal that is able to point backwards, in order to potentially carry out some successful autonomous landings. Finally, methods for filtering any erroneous control signals as a result of orientation ambiguity should be explored.
REFERENCES

[1] Araar, Oualid and Aouf, Nabil and Vitanov, Ivan. Vision Based Autonomous Landing of Multirotor UAV on Moving Platform. *Journal of Intelligent & Robotic Systems*, 85, 02 2017.

[2] Alexandre Borowczyk, Duc-Tien Nguyen, André Phu-Van Nguyen, Dang Quang Nguyen, David Saussié, and Jerome Le Ny. Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle. *IFAC-PapersOnLine*, 50(1):10488–10494, 2017. 20th IFAC World Congress.

[3] Chaves, Stephen and Wolcott, Ryan and Eustice, Ryan. NEEC Research: Toward GPS-denied Landing of Unmanned Aerial Vehicles on Ships at Sea. *Naval Engineers Journal*, 127:23–35, 03 2015.

[4] Davide Falanga, Alessio Zanchettin, Alessandro Simovic, Jeffrey Delmerico, and Davide Scaramuzza. Vision-based Autonomous Quadrotor Landing on a Moving Platform. 10 2017.

[5] Mark Fiola and Mark Fiola. ARTag, a Fiducial Marker System Using Digital Techniques. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)* - Volume 2 - Volume 02, CVPR ’05, pages 590–596, Washington, DC, USA, 2005. IEEE Computer Society.

[6] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Manuel Martín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47:2280–2292, 06 2014.

[7] Joshua Springer. Edited Video Decoding Sample (for Autonomous Landing). https://github.com/uzgit/Android-VideoStreamDecodingSample, 2021.

[8] Joshua Springer, Marcel Kyas. Evaluation of April Tag and WhyCode Fiducial Systems for Autonomous Precision Drone Landing with a Gimbal-Mounted Camera. Unpublished, March 2022.

[9] M. Krogius, A. Haggenmiller, and E. Olson. Flexible layouts for fiducial tags. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1898–1903, 2019.

[10] Tianpei Liao, Amaldev Haridevan, Yibo Liu, and Jinjun Shan. Autonomous Vision-based UAV Landing with Collision Avoidance using Deep Learning. 2021.

[11] Peter Lightbody, Tomáš Krajník, and Marc Hanheide. A Versatile High-performance Visual Fiducial Marker Detection System with Scalable Identity Encoding. In *Proceedings of the Symposium on Applied Computing, SAC ’17*, pages 276–282, New York, NY, USA, 2017. ACM.

[12] Phong Ha Nguyen, Ki Wan Kim, Young Won Lee, and Kang Ryong Park. Remote Marker-Based Tracking for UAV Landing Using Visible-Light Camera Sensor. *Sensors*, 17(9), 2017.

[13] Polvar, Riccardo and Sharma, Sanjay and Wan, Jian and Manning, Andrew and Sutton, Robert. Vision-Based Autonomous Landing of a Quadrotor on the Perturbed Deck of an Unmanned Surface Vehicle. *Drones*, 2(2), 2018.

[14] Joshua Springer. Autonomous Landing of a Multicopter Using Computer Vision. Master’s thesis, Reykjavík University, 2020.

[15] Hideyuki Tanaka and Yoshio Matsumoto. Autonomous drone guidance and landing system using ar/high-accuracy hybrid markers. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pages 598–599, 2019.

[16] Hideyuki Tanaka, Kunihiro Ogata, and Yoshio Matsumoto. Solving pose ambiguity of planar visual marker by wavelike two-tone patterns. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 568–573, 2017.

[17] Jamie Wubben, Francisco Fabra, Carlos Calafate, Tomasz Krzeszowski, Johann Marquez-Barja, Juan-Carlos Cano, and Pietro Manzoni. Accurate Landing of Unmanned Aerial Vehicles Using Ground Pattern Recognition. *Electronics*, 8:1532, 12 2019.

[18] Jesse S. Wynn. Visual Servoing for Precision Shipboard Landing of an Autonomous Multirotor Aircraft System. Master’s thesis, Brigham Young University, 9 2018.