Extending April Tag for Flexible Autonomous Drone Landing

Joshua Springer

Marcel Kyas

***Abstract*—This project introduces a flexible landing system based on the April Tag[1] fiducial system. It adds attributes to detected April Tag markers to estimate the pose of landing pads under the assumption that they are positioned level on the ground and marked with April Tags, and other attributes to enable marker tracking via a gimbal-mounted camera. It provides a lightweight April Tag family that allows for marker embedding, so that landing pads can be marked with bundles of several large and small markers, and allowing the landing pads to be continuously recognized from near and far distances. It also addresses the issue of orientation ambiguity in fiducial marker detection and its implications in the context of drone control.**

Check IEEE WG for suitable publication venue. Maybe try a workshop?

- IROS (International Conference on Intelligent Robots and Systems), conference or workshop
- ICRA Aerial Robotics Workshop
- IFAC-PapersOnLine
- *Autonomous Robots*

I. Introduction

Autonomous landing is one remaining major hurdle in fully autonomous multirotor drone flight. Many other tasks such as takeoff, waypoint-to-waypoint flight, and in-flight tasks such as picture-taking have been sufficiently automated. However, landing is difficult and risky enough that no widespread, autonomous solution is currently available. Landing with GPS alone is not reliable because target landing sites are typically small and GPS does not guarantee sufficient accuracy. Some projects have proposed autonomous landing methods (see Section II) that use some combination of different components such as RGB cameras, ultrasonic sensors, sophisticated ground control stations, and fiducial markers. These methods serve as a proof of concept for the idea of autonomous landing, but have not been widely adopted yet. Additionally, autonomous landing algorithms typically require either the addition of non-standard components (complexifying the drone system), or depend on a fixed, downward facing camera (limiting the range of behaviors during which the landing pad can be identified).

How do we get from the mentioned methods to april tag? What defines the knowledge gap of *this* paper? What are the properties of a solution to this problem? Why did we chose april tag?

This project introduces a flexible landing system which is based on the April Tag[1] fiducial system in order to keep the number of required sensors low - it requires only an RGB camera. It uses a gimbal to aim the camera and track the April Tag markers over time, increasing the range and reliability of landing pad identification. The method also requires no data describing the orientation of the gimbal, which is helpful as many widely-available gimbals do not provide this data. All computation occurs onboard the drone in order to avoid the need for extra data transmission, latency, and a sophisticated ground station. The method is implemented as a set of ROS modules and can be constrained to a single computational board, or distributed to multiple network-connected boards. The method assumes a near-level landing pad marked with nested April Tag fiducial markers for reliable, well-tested pose estimation that functions over a large range of distances. An edited April Tag ROS module adds message attributes for this application and generates position targets regardless of the orientation of the markers in the camera frame.

The ability to carry out many missions autonomously, and over long periods of time, is essential to widespread integration of drones into industry operations. Drones have already provided a cheaper, faster, and less risky alternative to humans in tasks such as monitoring ship traffic in seaports[2], geological surveys[3], and infrastructure inspection.[4] However, they still require human operators for landing, charging, and maintenance at the least. Autonomous landing is the key to further enabling fully-autonomous mission cycles requiring minimal attention from human operators.

II. Related Work

Several projects have proposed solutions to the problem of autonomous drone landing by marking landing pads with various fiducial markers (April Tag, ArUco, or custom markers).[5][6][7][8][9] Others use photographic matching between pictures of a landing site captured on takeoff.[10] The general trend in these methods is to first navigate near the anticipated landing site via GPS, and then to descend using visual clues to improve on the GPS accuracy, with the correct assumption that GPS does not provide a position estimate with sufficiently high resolution to land on a small platform (even with a clear view of the sky which provides reliable GPS reception). The drones contain one or two fixed cameras to identify markings or landmarks that provide a position estimate for the drone relative to the landing pad or landing site. One main challenge in these methods is that, since the camera is fixed to the drone body, and since the drone modulates its orientation as its main means of positional control, the movement of the drone affects the camera's field of view in way that can cause the marker to be lost. Further, the markers can eclipse the camera's field of view once the drone gets too close, meaning that the drone may have to finish the landing blind.

Another method employs a more sophisticated workflow that avoids the need for fiducial markers.[11] The drone explores an area where a landing should take place, capturing images from a fixed, downward-facing camera. The images contain tags specifying the location where they were captured, and then the drone streams them to a nearby computer for analysis. The system generates disparity maps and feature matches which imitate stereo image processing, and creates a 3D map of the terrain below. Sufficiently flat and large areas serve as possible landing sites, and the system chooses one such site for the drone to land at. The method is succesful in identifying viable landing sites, but requires a sophisticated ground control station and requires some adjustment before it can run in real time.

III. Methods

The proposed system moves away from the idea of a fixed camera in favor of a gimbal-mounted camera which is commonly available. It does not require the gimbal in order to function, but the gimbal reduces the importance of the drone's attitude and allows the drone to more reliably recognize and track the markers. Additionally, it uses a landing pad with several markers of various sizes in order to aid visibility throughout the entire landing. The system is comprised of three main parts:

1) an augmented April Tag fiducial system that recognizes a landing pad in an RGB image,
2) a tracking system that holds the landing pad in the view of the drone, and
3) a landing control policy that directs the drone safely towards the landing pad.

A. Fiducial System

A set of April Tag[1] markers attached to the landing pad allow for landing pad recognition and pose estimation. The markers are of the `Custom24h10` family which allows for marker embedding, as shown in Figure 1. The center 4 bits of each marker do not contribute to the marker ID, and can therefore serve as a location to embed a smaller, concentric April Tag marker with a different ID. This is important because the larger markers increase the distance from which the system can recognize the landing pad, and the smaller markers allow the system to continually track the landing pad once the drone has approached to the point that the larger markers are too big for the camera's field of view. Each of the individual tags are defined in the April Tag configuration files according to their size and location within a larger "bundle" structure. Since the markers are concentric, co-planar, and oriented in the same direction, they all have the same pose in reality. However, the system will most accurately determine the pose of the largest marker in the bundle, since it has the greatest pixel resolution. Therefore, the bundle takes the pose and all other attributes of the largest marker within it that is currently recognized. In this way, it is possible to define multiple landing pad bundles and to identify them separately. For example, the bundle for `landing_pad_1` may have markers with IDs 2, 1, 0 while the bundle for `landing_pad_2` may have markers 5, 4, 3. Whereas the original April Tag ROS module provides only the ID, size, and pose of the detected markers, the module used in this project adds several important attributes:

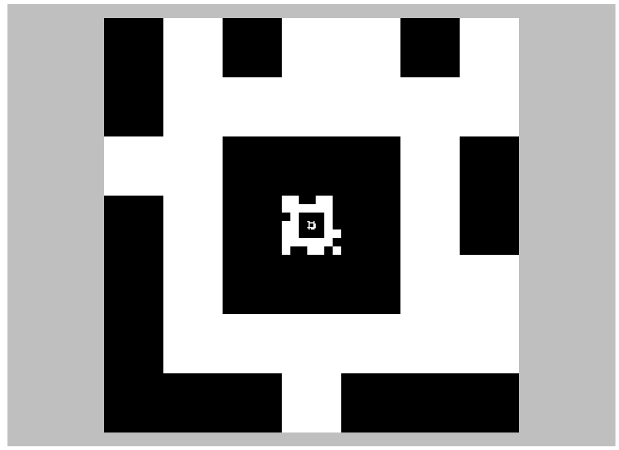

Fig. 1. The landing pad's embedded April Tag markers of the family `TagCustom24h10`. The center bit of each tag is not considered part of the tag, and can therefore be used to embed smaller tags.

1) The pixel positions (u, v) of the center of each marker. These are available in the original April Tag code but have been exposed via messages in order to allow for easy marker tracking.
2) The normalized pixel position (u_n, v_n) of the center of each marker, where $u_n, v_n \in [-1, 1]$. These are calculated from (u, v) but serve as better inputs to the PID systems that create commands in

order to track the marker.

3) A position target in “east, north, up” format which describes the relative position of the marker to the drone. This is a new parameter that is calculated by rotating the marker’s pose by the inverse of its pitch and roll, and then by the inverse of its yaw. This parameter assumes the marker is flat on the ground and facing up.
4) The name of the tag. This is also available in the original code but has been exposed via messages to allow for easy bundle tracking, and naming of specific landing pads.
5) The pitch, roll, and yaw of the landing pad in radians. The landing system uses the yaw to align itself to the landing pad before descent. The pitch and roll are currently unused.

1) April Tag 24h10 Family: A specific family of April Tag markers has been designed for the purpose of autonomously landing small drones. Since April Tag 3, it has been possible to specify squares within the definition of an April Tag marker that are not within the marker’s definition, and which therefore do not affect recognition of the marker. In these blank squares it is possible to embed smaller markers, which is important for visual landing because markers that are large enough to be recognized from far away are typically too big to be recognized once the drone has approached them. One stock April Tag family, 48h12, includes 4 blank squares in its center, which can be used for embedded markers. However, the 10x10 layout of this marker makes it prohibitively computationally expensive for a Raspberry Pi 3 B+ that is using it for visual navigation. Initial tests showed that, within the framework of this drone (which runs ArduCopter, MAVROS, a gimbal controller, a landing controller, and two PID systems, as well as a standard Raspbian OS and relevant tasks), the April Tag ROS module could detect markers in the 48h12 family at only about 1.5 hz. This is far too slow, and in-flight tests proved the resulting behavior unstable.

The 24h10 family provides an adequate computational speedup, with a final in-flight detection rate of about 10 hz. The definition of the family is provided in Table I, where

- “d” denotes a “data” square that can be either white or black depending on the marker’s ID,
- “w” denotes a “white” square,
- “b” denotes a “black” square,
- “x” denotes a bit that is unused and therefore serves as a location for embedding another marker.

The layout of this marker family attempts to optimize several features:

1) Embedded markers are concentric in order to reduce the complexity of subsequent coordinate system transforms, and to give all markers in a bundle the exact same pose. Therefore, the center square of the marker family remains unused.
2) The April Tag definition requires a back square border surrounded by a white square border, which are pushed to the center of the marker in order to take up a minimal number of squares on the grid.
3) The data squares are positioned on the outside of the marker in order to maximize the number of possible IDs in the marker family.

d	d	d	d	d	d	d
d	w	w	w	w	w	d
d	w	b	b	b	w	d
d	w	b	x	b	w	d
d	w	b	b	b	w	d
d	w	w	w	w	w	d
d	d	d	d	d	d	d

TABLE I
DEFINITION FOR THIS APRIL TAG 24H10 FAMILY.

The marker family must have a low false-positive recognition rate, which is partially tuned by a minimum Hamming distance between any two marker IDs (corresponding to the “h10” in the “24h10” marker family name). If this number is too low, the markers can be “recognized” randomly in input images, even where there is no actual marker. If this number is too high, there will be too few markers in the family, meaning that very few landmarks can be recognized. A Hamming distance of 10 provides this marker family with 16 unique markers (shown in Figure 2), which is adequate for this purpose. This marker design aims to use the minimum number of bits possible, while still allowing for a reasonable number of distinct markers, and allowing marker embedding. The reduction from a 10x10 region in family 48h12 to only a 7x7 region in family 24h10 accounts for the computational speedup. Detection of a 48h12 marker requires the identification of 96 black/white regions, while detection of a 24h10 marker requires the identification of only 48 such regions.

B. Tracking System

The drone leverages its gimbal-mounted camera in order to track the drone. A PID system controls the pitch of the gimbal in order to keep the marker in the center of the frame in the v dimension (where $v_n \approx 0$). The gimbal’s yaw stays at 0 relative to the drone during landing, and a separate PID system centers the marker in the camera frame in the u dimension (where $u_n \approx 0$), by controlling the yaw rate of the drone. The PID systems take the normalized pixel position as a state input, have a setpoint of 0, and output control efforts to change the tilt of the gimbal or the yaw rate of the drone. This

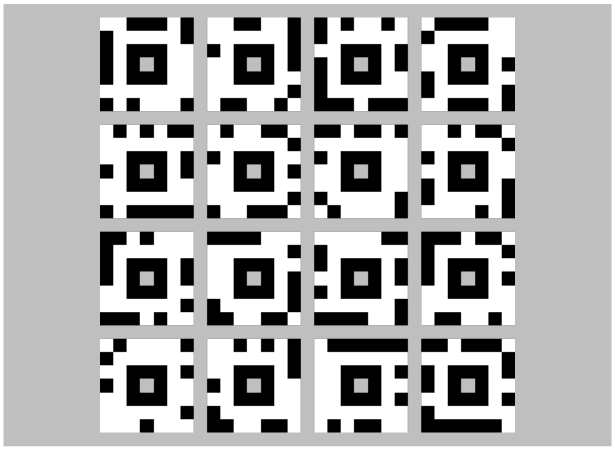

Fig. 2. All markers in this 24h10 family.

avoids the need for additional data (the yaw of the gimbal relative to the drone's yaw) and additional coordinate system transforms. The result is that the position target and tracking do not require the orientation of the gimbal in any dimension.

The overall benefit of the tracking system is that the drone can keep the marker in its view regardless of normal changes in its position that occur during approach. Since multirotor drones vector their thrust by changing their orientation, they have the potential to lose visual acquisition of the landing pad if their camera is fixed. This is because the camera's field of view changes as a direct result of the drone's thrust vectoring, which has the potential to push the marker out of view even if the drone is approaching in the correct direction. Tracking the marker allows the drone to recognize the landing pad from far away, when the camera is initially pointed forward and down (instead of just down). It also allows the drone to keep the marker in sight in the event that the drone approaches too fast and overshoots the landing pad slightly, or in the case that wind pushes the drone off course. Additionally, the drone is able to track the marker by changing only the tilt (pitch) of the gimbal, and its yaw orientation. Its capability for omnidirectional movement allows it to approach the landing pad directly, regardless of its yaw orientation.

C. Landing Control Policy

The PX4 autopilot software carries out a *precision landing* using its own method, which is informed by the location of the landing pad.

References

[1] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible Layouts for Fiducial Tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1898–1903.

[2] M. Stein, "Integrating Unmanned Vehicles in Port Security Operations: An Introductory Analysis and First Applicable Frameworks," *Ocean Yearbook Online*, vol. 32, pp. 556–583, 06 2018.

[3] "Game of drones – unmanned aerial vehicles in mineral exploration and geological mapping," Dec 2020. [Online]. Available: https://eitrawmaterials.eu/game-of-drones-unmanned-aerial-vehicles-in-mineral-exploration-and-geological-map

[4] M. Stein, "Unmanned maritime infrastructure inspection-a mixed method risk management approach from German port facilities," 04 2020.

[5] A. Borowczyk, D.-T. Nguyen, A. Phu-Van Nguyen, D. Q. Nguyen, D. Saussié, and J. L. Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 488–10 494, 2017, 20th IFAC World Congress.

[6] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 971–976.

[7] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," 10 2017.

[8] J. S. Wynn, "Visual servoing for precision shipboard landing of an autonomous multirotor aircraft system," Master's thesis, Brigham Young University, 9 2018. [Online]. Available: http://hdl.lib.byu.edu/1877/etd10385

[9] J. Wubben, F. Fabra, C. Calafate, T. Krzeszowski, J. Marquez-Barja, J.-C. Cano, and P. Manzoni, "Accurate landing of unmanned aerial vehicles using ground pattern recognition," *Electronics*, vol. 8, p. 1532, 12 2019.

[10] K. Pluckter and S. Scherer, *Precision UAV Landing in Unstructured Environments*, 01 2020, pp. 177–187.

[11] V. Desaraju, N. Michael, M. Humenberger, R. Brockers, S. Weiss, J. Nash, and L. Matthies, "Vision-based landing site evaluation and informed optimal trajectory generation toward autonomous rooftop landing," *Autonomous Robots*, vol. 39, 07 2015.