Master Thesis

Autonomous Vision-based Safe Proximity Operation of a Future Mars Rotorcraft

Autumn Term 2024

Supervised by:
Roland Brockers
Robert Hewitt
Marco Hutter

Author:
Lasse Fierz

Contents

List of Acronyms

- **UAV:** Unmanned Aerial Vehicle
- **SFM:** Structure From Motion
- **LSD:** Landing Site Detection
- **LS:** Landing Site
- **BA:** Bundle Adjustment
- **DEM:** Dense Elevation Map
- **OMG:** Optimal Mixture of Gaussian
- **LOD:** Level Of Detail
- **HiRISE:** High Resolution Imaging Science Experiment (High Resolution Satellite Imagery)
- **LRF:** Laser Range Finder
- **GT:** Ground Truth
- **LSM:** Landing Site Manager

Chapter 1

Introduction

With the Ingenuity rotorcraft's lifecycle coming to an end the question about future mars rotorcrafts and their capabilities draws ever closer.

For the future NASA plans two different rotorcraft Mars missions. The first is called Sample Retrieval Helicopter (SRH). It is envisioned to be an alternative way of transporting Mars samples to the retrieval station should the Perseverance rover fail to do so.

Secondly, for future large distance missions, NASA is conceptualizing a Mars Science Helicopter (MSH) project. The aspirations for such a rotorcraft are on one hand to cover farther distances at high altitudes with accurate state estimation and on the other to land safely, autonomously and reliably in previously unknown terrain. These two feats allow a helicopter to perform much advanced science missions compared to Ingenuity.

The LOng Range NAvigation (LORNA) project that I have been involved with is working on a concept to tackle the aforementioned points while dealing with the constraints that rotorcraft missions on Mars provide us with. These are namely a limitation on the size and weight of the drone, a constraint on compuational power due to the deployment on limited embedded processors and lastly a large delay in communication which makes adaptive remote control from Earth impossible.

1.1 Objective

The endeavour in this thesis was to create a front to back landing mechanism that combines the existing vision based landing site detection procedure with the autonomous framework. In order to accomplish this, both the landing site detection mechanism as well as the autonomy had to be altered. Last but not least given that the structure from motion depth generation depends on lateral movement, which is less desirable for a drone navigating at low altitudes in unfamiliar surroundings, the utilization of a stereo camera presents a viable solution to attain real-time depth perception without necessitating lateral displacement.

The conclusive high level objective of the LORNA project is to autonomously fly high altitude long distance science missions using a map based localization enhanced state estimator. Whilst flying, point clouds are generated and processed. On that data landing sites are acquired and simultaneously ordered according to the respective quality. Initiating the autonomous landing sequence, landing sites can then be chosen and verified at low altitudes using a stereo camera. In case of successful verification the rotorcraft can land at the selected location.

1.2   My Contribution

In short my contribution in this thesis can be summarized as follows:

1.2.1   Interface Autonomy-LSD

The autonomy was altered in a way to receive the detected landing sites and order them according to an adequate heuristic to determine the best landing site at any given time.

1.2.2   Landing Site Detection Output

The landing site detection output initially only consisted of the location of the landing site. This output was enhanced to consider many more characteristics in order for the autonomy to make an informed decision with regards to what spot to select.

1.2.3   Stereo Camera Depth Alternative

Implemented a stereo camera in the simulated drone model in order to get stereo sensor images. Created a stereo camera depth node as an alternative to SFM to supply the landing site detection algorithm with a point cloud at low altitudes without the need for lateral motion.

Depth Source Switching

Implemented an automatic switch between the SFM node and the stereo camera depth node by utilizing the laser range finder sensor on board.

1.2.4   Additional

Other contributions indirectly connected to the project itself but necessary for the implementation:

Simulation Setup

As just recently the switch was made to Gazebo Garden the entire visual pipeline (SFM + LSD) had never run with this simulation environment before. Therefore I implemented the changes necessary to run the landing site detection procedure on the Gazebo sensor input. Additionally whilst implementing the stereo camera and attempting to put in place a simulated depth camera for ground truth it became apparent that the Gazebo depth camera implementation is incorrect neglecting the set intrinsic parameters. Altering Gazebo's source code the implementation could be fixed.

Deployment of LSD pipeline onto an Embedded System

The entire software stack of this project has to run on an embedded system on the future rotorcraft. Currently the used processor is modalAI's voxl2. Both the structure from motion as well as the landing site detection software did not run out of the box having an incompatible dependency handling with the voxl's AARCH architecture. Resolving these issues I was able to run the landing site detection pipeline with the structure from motion depth supply on the voxl2 using a collected rosbag of images and IMU poses.

1.3 Organisation of this Thesis

1.3.1 Related Work

As is custom, I will introduce the reader to what has been done in this area. Main focus will be placed on vision-based landing site detection procedures and previous work on autonomous landing.

1.3.2 System Overview

The entire project overview will be introduced. Emphasis lies on the methods that I have heavily interacted with in this thesis. These are mainly the structure from motion depth generation, the landing site detection mechanism and the autonomous framework.

1.3.3 Stereo Camera Depth Alternative

This chapter introduces the stereo camera depth part of the methodology. I will go into the reasoning why a stereo camera is necessary as a low altitude depth alternative. Additionally, I analyse the stereo option theoretically and describe the process of implementing it in the existing project structure. Lastly it is qualitatively compared to a depth camera based ground truth.

1.3.4 Autonomous Landing Procedure

Here I will lay out the core contribution of this project which combines the existing system with the novel contributions of this work in order to put together a front to back automous landing procedure. First I will describe the conceptual landing behavior and later on I explain the practical implementation. Lastly the working pipeline is shown in a case example of a science mission flown in simulation.

1.3.5 Evaluation

Here I introduce the test setup according to which I performed repeated randomized simulation flights. I introduce the outcome defining metrics and the results of the test flights. Lastly these results are analysed conceptually and with specific examples.

1.3.6 Conclusion

I summarize the work in this thesis and the outcome of my

1.3.7 Outlook

Chapter 2

Related Work

Autonomous safe landing is perhaps the most important part of a rotorcraft's mission. It comes therefore as no surprise, that tremendous amounts of work have been accomplished in the pursuit of achieving this crucial feat.

2.1 Artificial Landing Markers

Visual sensors are highly advantageous for navigation due to their lightweight nature and the extensive research dedicated to their development over the years. The minimal weight of these sensors makes them particularly suitable for applications where payload capacity is a critical concern, such as in the case of a rotorcraft Mars missions. Decades of intensive research have culminated in highly sophisticated algorithms and methodologies that leverage the rich data captured by visual sensors and enable the daunting task of autonomous landing. [1, 2] and [3] use artificial landing markers as indications of valid landing sites. While [1] and [3] use stationary markings, [2] enabled a rotorcraft to land on a moving target. These approaches, though useful in urban environments, are not applicable on uncharted terrain as found on Mars.

[4, 5, 6] and [7] pursue implementations based on homography assumptions. This is not possible in our setup as we cannot assume homographic conditions on Mars' rough terrain.

A very handy tool for the creation of depth maps to segment landing sites on are range sensors like Lidar as [8, 9, 10] and [11] show. As for our purposes a rotorcraft has to fly on Mars' 1% air density however, weight is a limiting constraint rendering Lidar sensor a suboptimal choice.

For the Mars Mission's lander NASA has used a vision based strategy using a predefined map of Mars' surface and a downwards facing monocular camera to orient the lander in the predefined map[12]. When compared to a lander however, rotorcrafts need to consider much smaller hazards. The available HiRISE satellite images are not sufficient in resolution to supply such prior information to the landing process of a UAV. Rover images could be used as well as Ingenuity's footage however the usage of this data would limit possible flight areas significantly.

[13] use a similar approach as the one used by LORNA. Compared to LORNA's Landing Site Detection [14, 15] however, a non-robot-centric DEM is used. The advantage of LORNA's approach is the implicit drift handling by considering the robot-centered terrain map.

Modern approaches like [16, 17] and [18] use a 2.5D terrain representation similar to the setup used in this project.

Other novel approaches use learning based methods as did [19, 20] and [21]. Though

certainly promising regarding accuracy and in the long run definitely a pathway to consider, learning based methods come with significant costs in the context of the task at hand. First of all considering the limitations present in Mars missions, the probable additional computational overhead from learning based methods can not be neglected. Furthermore, neural network based solutions give up simplicity and interpretablity for the benefit of precision. This is not to be underestimated in a tricky environment such as Mars terrain. Additionally, learning based methods require substantial training data which, in the context of autonomous UAV landing, is not available in large quantities. Lastly complex and specific missions flown on Mars pose their unique challenges. Niche information about these problems can be fused into conventional methods in the form of prior assumptions and constraints.

Chapter 3

preparation - Autonomous Landing Procedure

With the enhanced structure of the LSD output, having implemented a stereo camera as a low altitude alternative to SFM and after ensuring a correct ground truth comparison, the main contribution of this work could be faced: Bringing the visual landing site pipeline together with the autonomous framework in order to achieve reliable autonomous landing in unknown terrain.

This implementation can be split into the following parts:

- Landing Site Heuristic
- Conceptual Behavior
- Behavior Tree Implementation

3.1   Landing Site Heuristic

As mentioned in **??** the autonomy receives landing sites from LSD with the following properties:

- Position
- Roughness
- Uncertainty
- Size
- Obstacle Altitude

From these properties the characteristics that comprise the final heuristic are:

- Current Distance to Drone
- Roughness
- Uncertainty
- Size
- Verification Altitude

The final heuristic defining the quality of a landing site is in fact a square loss function:

$$L_{LS} = w_{dist}L_{dist} + w_{rough}L_{rough} + w_{var}L_{var} + w_{size}L_{size} + w_{verAlt}L_{verAlt} \quad (3.1)$$

3.1.1   Current Distance to Drone - L_{dist}

Well likely the single most important characteristic of an LSD-prefiltered[1] landing site. Each iteration the current distance to the drone's position is calculated for each retained landing site. The distance is then normalized by dividing it by the cruise altitude which is 100m. In practice there were easily enough landing sites found while moving to allow landing sites to fall off when being farther away than 100m.

3.1.2   Roughness - L_{rough}

The roughness property is the unaltered roughness value received from LSD. It is already normalized and enters the loss function as it is.

3.1.3   Uncertainty - L_{var}

The same holds for the uncertainty. It is already normalized by design and enters the loss function unaltered.

3.1.4   Size - L_{size}

Analogous to the roughness and uncertainty properties the size comes from the landing site detection directly. However unlike the two preceeding properties it is not normalized but simply denotes the metric radius of the largest circle of valid landing area that can be fit around a given landing site. This is achieved in LSD by performing a distance transform on the created landing site image.
In order to normalize this value the maximum landing site size is retained and each landing site's size is divided by it in order to achieve normalized size information.
Also as can be seen in eq. (3.1), the size contribution enters the loss function with a negative sign. This is due to the fact that compared to all other characteristics, the size defines a property that we would like to maximize.

3.1.5   Verification Altitude - L_{verAlt}

A site's verificaiton altitude is the smallest vertical distance between the drone and the landing site at which that site was (re-) detected.
The verification altitude is a useful property because of numerous reasons.

Further Indication of Certainty

First of all similar to the uncertainty metric the verification altitude indicates how certain we can be about a detected landing site as spots detected at lower flight altitudes are more likely correct due to the reduced depth error. Even though it might seem overlapping with the uncertainty property in this regard, these two characteristics are quite complementary as the uncertainty takes OMG convergence and camera specifics into consideration while the verificaiton altitude is a purely location based metric.

---

[1]Each received landing site has already undergone a threshold filtering regarding slope and roughness.

Landing Site Property Updates

As the verification altitude yields a simple and good estimation of the trustworthiness of an incoming landing site, it can be used as a flag to know, when a landing site's properties should be updated. When a landing site is redetected with a verification altitude lower than the previously stored one, the algorithm trusts it more and alters the previously stored properties to the new ones received.

Verification

Continuously updating the verification altitude upon redetection allows us to determine the lowest altitute, at which a landing site was redetected. This information can be used to verify that a given site was considered a valid landing spot even at low altitudes.

3.2   Conceptual Behavior

Bringing everything together and emphasizing landing aspects of an autonomous flight perspective, we arrive at the following procedure:

3.2.1   Takeoff

The necessary checks and initializations are performed. This includes the created mission waypoint plan, the ros-, as well as the mavros-connection setup with the initial setting of mavros parameters.
Then the drone takes off vertically until it reaches the first waypoint's target altitude.
During this phase the stereo camera feeds depth images into LSD until the laser range finder switches to SFM which results in a stop of depth supply as SFM does not detect depth during vertical motion.

3.2.2   Prerequisite - Landing Site Handling

Landing sites are constantly received by the autonomy's ROS connector which constitutes the ros interface with the landing site detector. The sites have to be processed in a separate thread. This is handled by a landing site manager (LSM) singleton class.

Landing Site Manager

The incoming landing sites are ranked according to a loss function(3.1) and stored in a max-heap buffer in order to easily switch out the worst landing site for a new one at any given time.

3.2.3   Transition into Landing

During a mission the drone flies to different waypoints. During the lateral motion periods of these flights, the drone continuously processes the incoming landing sites using the LSM.
Once the last mission waypoint has been reached, the transition to the landing behavior occurs.
At first,

3.3   Behavior Tree Implementation

As a mission is flown at 100m altitude we would like to verify it closer to the ground where we can be more certain about measured terrain.

To this end the drone moves laterally to a chosen landing site and then descends blindly to a certain verification altitude above that spot. We can do so safely as, yes, the initial landing site estimate from 100m altitude might not be a good choice, however as can be seen in eq. (3.3), given an approximate baseline of 15m at 100m altitude with a focal length of 256 and an assumed subpixel disparity error of 0.5, the structure from motion algorithm yields depth measurements with an approximate depth error of 1.3m.

$$\Delta z = \frac{z^2}{f \cdot b} \Delta d \tag{3.2}$$

$$\Delta z = \frac{100m^2}{256 \cdot 15m} 0.5 = 1.302m \tag{3.3}$$

Therefore in the very worst case szenario of a depth error of 1.3m we are still safe when we descend to a verification altitude of 2.5m above a detected landing site. Avoiding SFM verification patterns at intermediate altitudes saves a tremendous amount of time and energy.

The verification is performed using the stereo camera(**??**).

3.3.1   Action Definition

The final landing behavior is implemented in the form of a behavior tree which allows adaptive decision making.

The autonomous framework already defined the core control flow nodes as well as some of the action nodes required for the landing behavior. (**??**) Hereafter displayed is a list of additional actions needed to be defined in this work. It should be noted, that they define individual actions and should not be a description of the entire behavior. For this consider section 3.3

CheckLandingSite

Simple utility action which checks whether any landing sites have been found by querying the LSM's landing site buffer length.

ChangeAltitudeLSAction

This action was implemented with the purpose to allow us to descend to a certain fixed altitude above a chosen landing site. The creation of another action for this purpose is simply a utility which let's us avoid having to pass a function pointer in the action defition as the arguments passed in a behavior tree are always evaluated at the time of the creation.

GetClearAltitude

Once a landing site has been chosen, the failsafe mechanism to go to that landing site is the following:

1. Ascend to a safe altitude.

2. Traverse laterally to the landing site's xy-position.

3. Descend to the landing site or verification altitude.

The question remains however, what the adequate clearing altitude is. The goal is to find an altitude high enough to fly safely without the risk of collision yet low enough as to not waste energy for the increased ascent / descent distances.

This is where the aforementioned obstacle altitude from **??** comes in. The obstacle altitude gives us an indication of the height to clear around the landing site. Therefore we can take this as the start altitude for the derivation.

As the DEM created by LSD covers only a limited area, the obstacle altitude is only an indication of the highest terrain present at the chosen landing site. The worst case szenario would be the detection of the landing site at the very edge of the DEM shortly before significantly higher terrain starts. Therefore one has to anticipate this case.

In practice a safe altitude buffer is implemented by assuming a worst case 45-degree incline of the terrain starting immediately at the landing site. Thus the necessary clearing altitude can be derived by linearly interpolating the final value from the initial obstacle altitude and the distance between the drone and the landing site which needs to be covered.

In the end to not ascend to excessive heights, the clearing altitude is capped at a predetermined, terrain-based failsafe altitude

$$z_{\text{clear}} = z_{\text{obst.}} + z_{\text{buffer}} + d_{\text{drone-LS}} \tag{3.4}$$

$$z_{\text{clear}} = \min\left(z_{\text{clear}}, z_{\text{failsafe}}\right) \tag{3.5}$$

Above we can see the derivation of the clearing altitude where z_{clear} defines the clearing altitude, z_{buffer} a safety buffer on top of the obstacle altitude, $d_{\text{drone-LS}}$ the drone's current distance to the site and z_{failsafe} the failsafe altitude at which the clearing altitude is capped.

GetLandingSiteAction

Upon leaving the mission state and entering the landing state, the autonomy attempts to select the currently best landing site according to the in section 3.1 introduced loss function.

This is done by using the landing site manager in order to rank the landing sites in an ascending sorted list (as opposed to a max-heap) and selecting the first entry with the lowest loss.

The selected landing site is then stored in a blackboard interface which allows easy data sharing between all the actions within the autonomy.

In the end the GetLandingSiteAction invokes the GetClearAltitude action to also store the clear altitude on the blackboard. This is used by the NavigateToWaypointAction to move to that site on the derived safe altitude.

LandingSiteVerificationAction

Once a landing site is chosen, we have to make sure it's a good spot to land before attempting to do so.

The LandingSiteVerificationAction does this by using the verification altitude property of a given landing site:

When a landing site is detected by SFM at 100m altitude it will be overwritten when redetected at 2.5m above the ground.

This is the mechanism exploited in order to verify a landing site. The drone hovers above the landing site for a predetermined duration in place and attempts to redetect the chosen landing site. This means that the LSM continuously processing incoming landing sites and if one is close enough to an existing one, it is considered

a redetection. In that case, the landing site is verified and the landing action can be triggered.

In case of verification failure, the landing site is not only removed but actively banned in order to prevent future false positives at high altitudes.

Also as previously mentioned the verification hovering at low altitudes leads most probably to the detection of close by landing sites. This is arguably as important as the verification of a previous landing site as it yields a good candidate in close proximity. In practice, it turned out, it is equally likely to verify a landing site as it is to not verify one but detect a high quality landing site close by.

LandingSiteSearchAction

The previously described actions are core implementations of the nominal behavior in the landing sequence. However what happens when no landing sites are found, either through fault of the landing site detection setup or due to really unfavorable terrain?

The most intuitive answer seems a good idea - simply look further for a site. The technical implementation of this task at high altitudes requires the detection by SFM and therefore lateral motion. So an easy solution is to fly a pattern at a new location with the exact same landing site handling procedure as in the nominal case. The LandingSiteSearchAction implements this through a predefined rectangle of waypoints which are flown through. This sub-mission is cancelled upon detecting a single landing site. In that case the usual landing procedure is continued.

GetNextPatternCenterWPAction

In case of failure when looking for additional landing sites, the adaptive procedure is to simply move to a new location and try anew.

The drone picks a random position around the final mission waypoint, flies there and again moves laterally in a rectangle shape in the hope of detecting landing sites. Note: This procedure is repeated a fixed number of times using the retry control node described in ??. Optimally however, this would be performed as long as the battery state permits it.

Bibliography

[1] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, 2002, pp. 2799–2804.

[2] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics*, Shanghai, China, 2017.

[3] L. Mu, Q. Li, B. Wang, Y. Zhang, N. Feng, X. Xue, and W. Sun, "A vision-based autonomous landing guidance strategy for a micro-uav by the modified camera view," *Drones*, vol. 7, no. 6, p. 400, 2023.

[4] S. Bosch, S. Lacroix, and F. Caballero, "Autonomous detection of safe landing areas for an uav from monocular images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing: IEEE, 2006.

[5] R. Brockers, P. Bouffard, J. Ma, L. Matthies, and C. Tomlin, "Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision," in *SPIE Defense, Security and Sensing*, 2011.

[6] V. Desaraju, M. Humenberger, N. Michael, R. Brockers, S. Weiss, and L. Matthies, "Vision-based Landing Site Evaluation and Trajectory Generation Toward Rooftop Landing," *Autonomous Robots*, vol. 39, no. 3, pp. 445–463, 2015.

[7] R. Brockers, M. Hummenberger, S. Weiss, and L. Matthies, "Towards autonomous navigation of miniature uav," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 645–651, 2014.

[8] N. Trawny, A. Huertas, M. Luna, C. Villalpando, K. Martin, J. Carson, and A. Johnson, "Flight testing a real-time hazard detection system for safe lunar landing on the rocket-powered Morpheus vehicle," in *Proc. AIAA SciTech Conference*, 2015.

[9] M. Luna, E. Almeida, G. Spiers, C. Villalpando, A. Johnson, and N. Trawny, "Evaluation of the simple safe site selection (s4) hazard detection algorithm using helicopter field test data," in *AIAA Guidance, Navigation, and Control Conference*, 2017.

[10] A. Johnson, A. Klumpp, J. Collier, and A. Wolf, "Lidar-based hazard avoidance for safe landing on Mars," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, 2002.

[11] S. Scherer, L. Chamberlain, and S. Singh, "Autonomous landing at unprepared sites by a full-scale helicopter," *Journal of Robotics and Autonomous Systems*, 2012.

[12] A. Johnson, N. Villaume, C. Umsted, A. Kourchians, D. Sternberg, N. Trawny, Y. Cheng, E. Giepel, and J. Montgomery, "The Mars 2020 lander vision system field test," in *Proc. AAS Guidance Navigation and Control Conference (AAS-20-105)*, 2020.

[13] A. Johnson, J. Montgomery, and L. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2005.

[14] P. Schoppmann, P. F. Proenca, J. Delaune, M. Pantic, T. Hinzmann, L. Matthies, R. Siegwart, and R. Brockers, "Multi-Resolution Elevation Mapping and Safe Landing Site Detection with Applications to Planetary Rotorcraft," 2021.

[15] P. F. Proenca, J. Delaune, and R. Brockers, "Optimizing Terrain Mapping and Landing Site Detection for Autonomous UAVs," 2022.

[16] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[17] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza, "Continuous on-board monocular-vision based elevation mapping applied to autonomous landing of micro aerial vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[18] S. Daftry, M. Das, J. Delaune, C. Sorice, R. Hewitt, S. Reddy, D. Lytle, E. Gu, and L. Matthies, "Robust vision-based autonomous navigation, mapping and landing for MAVs at night," in *International Symposium on Experimental Robotics (ISER)*, 2018.

[19] F. Neves, L. Branco, M. Pereira, R. Claro, and A. Pinto, "A Multimodal Learning-based Approach for Autonomous Landing of UAV," 2024.

[20] S. Abdollahzadeh, P.-L. Proulx, M. Allili, and J.-F. Lapointe, "Safe landing zones detection for uavs using deep regression," in *2022 19th Conference on Robots and Vision (CRV)*, 2022, pp. 213–218.

[21] H. Tovanche-Picon, J. González-Trejo, A. Flores-Abad *et al.*, "Real-time safe validation of autonomous landing in populated areas: from virtual environments to Robot-In-The-Loop," *Virtual Reality*, vol. 28, no. 66, p. 66, 2024.