33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

IMAGE SEGMENTATION BASED EMERGENCY LANDING FOR AUTONOMOUS AND AUTOMATED UNMANNED AERIAL VEHICLES

Philip Hausmann[1], Henri Meeß[2] & Gordon Elger[1]

[1]Technische Hochschule Ingolstadt
[2]Fraunhofer Institute for Transportation and Infrastructure Systems IVI

Abstract

The growing application of unmanned aerial vehicles (UAVs) into diverse commercial and private applications entails serious risks and respectively requires regulations. The development towards autonomously flying drones enforces this need and justifies the development of additional safety levels. Therefore, this work presents an emergency landing operation which is able to identify safe landing spots for both the environment, as well as the UAV, using neural networks (NNs) to segment the image of a downward-facing camera. This image is projected onto the ground plane conserving metric scaling. The landing control is administered by a Behavior Tree (BT). Thereby, a fully autonomous emergency landing without any communication to the ground control, predefined safe landing spots or detailed maps with corresponding location is enabled. A software-in-the-loop (SIL) setup is implemented in which it is possible to conduct a safe emergency landing operation. Additionally, multiple real world test flights have been conducted gathering data to evaluate this method.

Keywords: unmanned aerial vehicles, semantic segmentation, emergency landing operation, vision based navigation, image processing

1. Introduction

The application of UAVs in e.g. logistics and surveillance entails serious risks, especially when the operation takes place in urban areas or critical cargo - such as medical supplies or hazardous material - is being transported. The safety requirements are even higher when the vehicles are controlled autonomously or beyond visual line of sight (BVLOS). In this case, a sudden loss of communication to the drone or electronic failure may be fatal. Because of this, it is necessary to conduct specific operational risk assessments (SORA) before UAV operations. Guerin et al. [1] note a lack of acceptable mitigation means in SORA certifications and suggest to adjust the regulations to include emergency landings operations as a mean of mitigation. The main objective has to be the avoidance of human casualties, which can also occur indirectly if the UAV is damaged while for example vital or hazardous cargo is on board. The emergency system should rely on as few computational resources - and for small UAVs with low maximum take off weight also as few sensors - as possible so the system can be transferred to arbitrary UAV models, while functional operation and redundancy (when possible) must be ensured.

Therefore, this work proposes an emergency landing function which is only dependent on a monocular camera image, the gyroscope, and height information e.g. gathered by radar.

The necessities for this emergency landing operation are identified based on a requirements analysis that has been conducted after the examination of the state of the art.

The remaining paper is structured as follows: Chapter 2 examines the state of the art in the field of emergency landing and points out the gap in research. Based on this, our method projecting semantically segmented images onto a ground map to identify suitable landing spots to close this gap is presented in chapter 3. The current status of implementation and the results are presented in chapter 4 based on a SIL setup and real world test flights and are further discussed in chapter 5.

The work is concluded in chapter 6 and the outlook in chapter 7 showcases the potential for further research on the field.

2. Autonomous Emergency Landing

The development of emergency landing operations for UAVs is an ongoing field of research. Several papers dealt with different advances to develop innovative solutions to this problem. Some approaches require a priori knowledge about the environment of the drone like the existence of landing markers. These markers are identified by Xia et al. [2] with an on-board camera. In emergency cases the UAV heads towards the markers - previously distributed around the city - using its inertial navigation system (INS) without GPS. On its way it avoids objects with an RGBD camera and Markov decision process. On the other hand, Lee et al. [3] identify a landing hub marker with a R-CNN and detect its features to decide if it is obstructed by objects. These features build up on the histogram of oriented gradients, a support vector machine (SVM), local binary pattern features and the CNN. In contrast to the previous two methods, Bodunkov and Kim [4] are not dependent on specific landing markers, but register suitable landing sites with appropriate terrain before flight. These landing sites are then recognized via on-board cameras.

For many UAV applications, e.g. rescue missions, it is not possible to rely on knowledge about the environment beforehand. Therefore, it is necessary to locate safe emergency landing spots in arbitrary locations. Accordingly, Guerin et al. [1] use a semantic segmentation based approach where the aim is to avoid busy roads with a safety architecture to monitor the Artificial Intelligence (AI) at runtime. They propose the use of emergency landing operations as a mitigation mean for SORA certifications. Considering a more comprehensive scope, Bektash et al. [5] differentiate between three phases: the detection of emergencies, finding a landing spot, and finally landing. They classify 180x180 pixel-wide image patches manually as either suitable or not suitable for landing to train a CNN running on an embedded system. The landing spot selection considers three categories: the urgency level, safety degree of the landing area and distance to the landing spot. Depending on the circumstances, a normal landing or ballistic decent is initiated. They followed their work on the decision finding of which landing spot to choose in [6]. Taking into account further data gathering methods, the work conducted by Mukadamet al. [7] processes satellite images from Google Earth. In the pre-processing stage they divide these into patches of 16x16 pixels and enhance the contrast. For feature extraction color, edge and local binary pattern features are used while training a SVM to classify the patches into suitable and not suitable for landing. In direct contrast to this, Lusk et al. [8] fly at a low altitude of 5 m and use a canny edge detector to see if the ground is clear of obstacles. The method is limited but able run on a Raspberry Pi on-board computer.

Placing a focus on the detection of humans in urban areas, Gonzalez-Trejo and Mercado-Ravell [9] developed a lightweight algorithm to identify safe emergency landing areas in crowds in real time. They used a pruned compact CNN with Bayes Loss to create a density map of crowds. Within this map they find the biggest circular free space using a polylabel algorithm. They refined their method by projecting the density map to the heads plane assuming a planar ground in [10]. Furthermore, multiple landing spots are detected using the euclidian distance, instead of the polyable algorithm, speeding up the computation. The landing areas are tracked with a Kalman filter and Hungarian algorithm to verify their stability. Lastly, they set quality criteria for landing zones to take the size, stability and risk - caused by humans walking into the zones - into account. Instead of the operation of UAVs over urban districts, Hinzmann et al. [11] specialize on emergency landings for fixed wing UAVs in open areas. This work identifies homogeneous regions using edge detection and distance transform, then classifies this region into grass or not grass. A coarse depth estimation and - for the most promising area - a fine 3D reconstruction using the UAV pose, images and feature tracks are calculated. They generate a metric with terrain texture, shape, slope, and roughness to calculate a distance-to-hazard map to identify the best landing spot. Afterwards, an approach path is calculated, taking into account the local wind estimation and the elevation map. Rather than a binary differentiation, Fraczek et al. [12] classify the ground into multiple categories with three according landing suitabilities. They define a $1m^2$ square as sufficient space to land. The metric scaling is calculated with the height of the drone and fixed camera parameters. The suitability is calculated pixel-wise by taking 21x21 image patch features, as well as a 9x9 neighbourhood feature distinction. The distance

transform is then used to find the biggest suitable landing area.
Taking the above discussed literature into account, the following requirements for the successful landing function can be identified:

- The process must be independent of pre-acquired maps.
- It must be possible to detect humans and other critical obstacles as well as incorporate terrain information to minimize the risks during and after the landing operation.
- The continuous processing frequency for the landing function must be real-time capable to enable consistent and fast decisions.
- In addition metric scaling is necessary to identify safe landing places (if available) and calculate their relative coordinates.
- A safe and explainable logic has to control the UAV to guide the emergency landing operation.

None of the discussed emergency landing operations is versatile enough to fulfill these requirements. Some are dependent on a-priori knowledge like landing hubs or search biased for e.g. roads free of cars or humans to avoid. Therefore, these approaches are vulnerable to inopportune circumstances and thus the behaviour of the UAV could be unpredictable. Our approach aims to fulfill all of the above listed requirements by combining AI-based semantic segmentation of a downward facing camera to identify obstacles, classify terrain, and identify safe landing areas while preserving metric scale to compute the relative coordinates of possible landing spots. A segmented, scaled and oriented map of the ground is generated and the results of the segmentation are stabilized to avoid uncertainties while approaching the landing spot.
Despite some similarities to our method in various of the above noted approaches, there are no methods yet that combine the inclusion of metric scaling with the versatility of semantic segmentation to generate a map and exploit the high level information to coordinate safe landing manoeuvres.

3. Method

The foundation for the emergency landing operation is the identification of suitable landing spots. The logical framework to identify safe landing areas is depicted in figure 1. After an overall introduction to our method, the details of the pipeline will be explained in the following subsections.

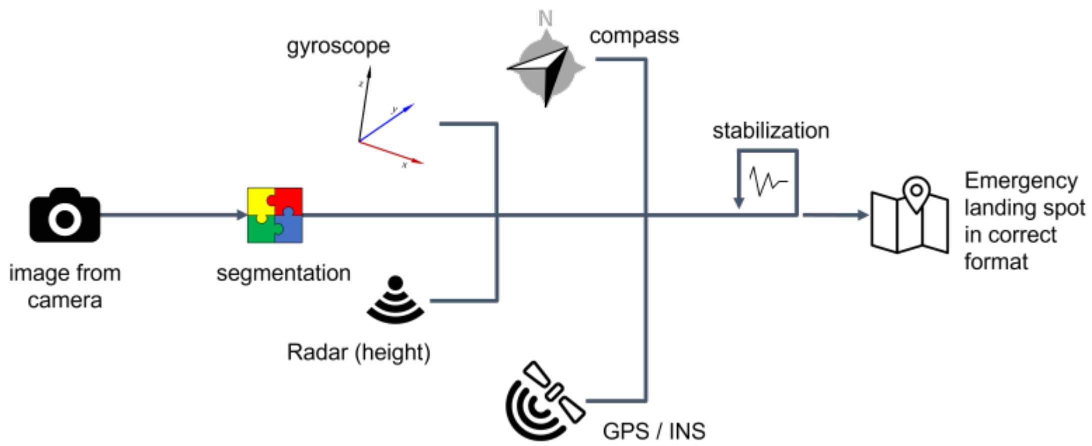

Figure 1 – Emergency landing spot detection pipeline

The framework and processing is structured as follows: First, the ground is surveyed with a downward-facing camera, the images of which are being segmented by NNs. The segmented images contain vital information about the environment by dividing the ground in different sections and assigning descriptive labels to them. This enables the differentiation between various ground surfaces e.g. roads, fields and trees, and at the same time dynamic obstacles like vehicles and humans. This high level

information enables an extensive evaluation of the possibilities and respective risks inherent in various emergency landing manoeuvres. However, the segmented images lack metric scaling, as they do not contain distance-information. By approximating the ground beneath the UAV as a flat plane and additionally using data from the gyroscope, height information and intrinsic as well as extrinsic calibration the segmented image can be projected onto the ground, where the results form a continuously generated map. The distance to the ground (i.e. the relative altitude) can be either estimated via the INS including a barometer and GPS or measured with a radar, LiDAR or any other well suited altitude measurement unit. Using the generated map, it is possible to identify suitable landing spots and save their locations. These can be relative to the UAV's position or - under utilization of GPS - absolute coordinates. Projection errors and flickering in the image segmentation disturb the mapping process and must therefore be filtered out stabilizing the map. A BT then takes control over the UAV coordinating the landing process and decision making.
The main components of this process are exemplified in the following subsections.

3.1 Segmentation

Dividing the ground into classified segments enables the evaluation of possible emergency landing spots. An example for image segmentation on real drone footage is depicted in figure 2. For example, grass fields and dirt could be prioritized as landing spots whereas flying over humans or landing in trees should be avoided.

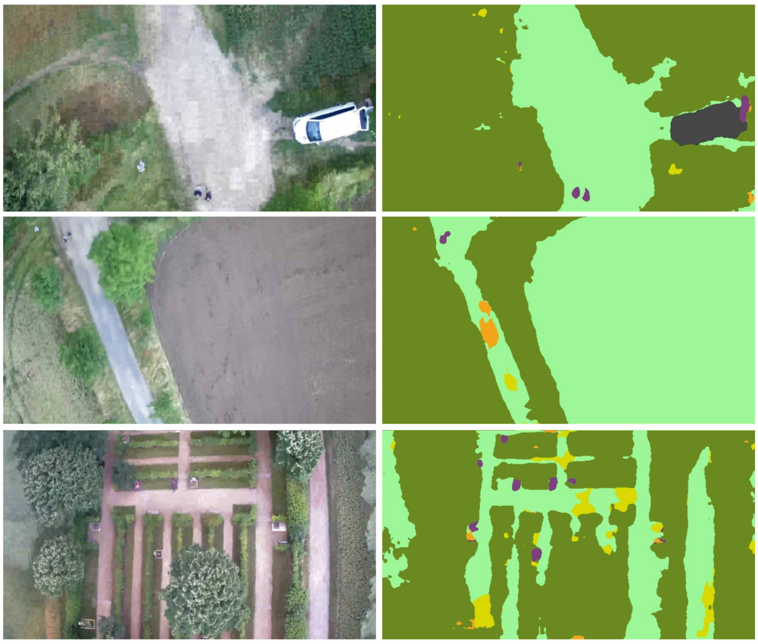

Figure 2 – Examples of image segmentation with real drone footage (differing from the training data)

The algorithms used for image segmentation are evaluated in benchmarks and widely used in the field of autonomous driving [13]. To be suitable for the application in the landing operation, the algorithm needs to excel in three criteria i.e. computational speed, accuracy, and reproducability. As of now the fast spatial feature network (FSFNet) of Kim et al. [14] has been implemented which employs FSF and Multi-Resolution-Aggregation (MRA) modules. The FSF module comprises of three downsizing subset modules where the features are recombined by the MRA to reconstruct the segmented image. Additionally we conduct tests with the RPNet of Chen et al. [15] which learns the main and residuals of segmentation by decomposing the label at different levels of residual blocks using the residual features to learn the edges and details and the identity features to learn the main part of targets. These NNs have been selected due to their achievements on the Cityscapes benchmark [13]. Using a transfer learning approach, the algorithms were trained on a heavily augmented version of the Aeroscapes data set [16], which provides 3269 labeled images, taken from an altitude of 5 to 50 m.
The availability of suitable data sets proofs to be a major issue, as a set of multiple classes to distin-

guish have been identified. This plurality may not be necessary for a basic straight forward emergency landing itself, however it is planned to use the same data set to extend the environmental perception with further functionalities. Furthermore, it is not possible to use one of the well established data sets for autonomous driving, as the perspective of UAVs and cars differ significantly. Therefore, the data of Nigam et al. [16] has been adapted. For future work it is planned to generate a hybrid data set including the Mid-Air data [17] and work with a limited amount of classes, fusing the labels of different sets for the emergency landing. The transfer from the synthetic images to the real world will be answered by e.g. transfer learning and/or domain adaptation.

3.2 Projection

The identification of suitable landing spots assumes an accurate mapping of the segmented image on the ground plane, as each pixel on this map represents metric units, for example one pixel may represent 5 cm * 5 cm at low altitude and 40 cm * 40 cm at higher altitude. Therefore, the spatial resolution of the ground map can be adjusted to the height of the drone reducing allocated memory while conserving the image resolution. Because of this, the intrinsic and extrinsic calibration of the sensors is crucial. The relations of the sensors to one another and the respective mapping is depicted in figure 3.

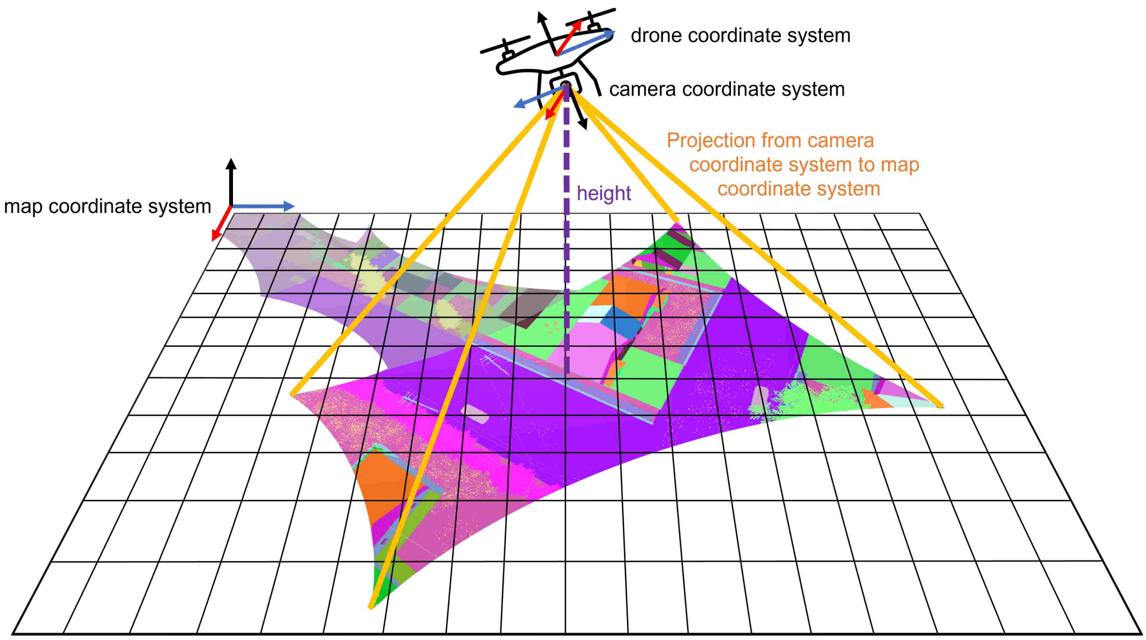

Figure 3 – Continuous projection of the image to map coordinate system (ground plane)

Assuming a pinhole-camera-model, the image can be represented as a plane projected through the focal point. A line drawn from the pin hole through a corner point intersects the ground plane as it progresses. Knowing the intrinsic and extrinsic camera parameters, the position of arbitrary ground-points p_{ground} on the image plane p_{image} can be calculated.

$$p_{image} = K * T_{ext} * p_{ground} \tag{1}$$

Where K depicts the intrinsic camera parameters and T_{ext} the extrinsics from the camera center to the ground map image. The indexes of the transformation matrices refer to the rotation and translation from one coordinate system to the next.

$$T_{ext} = T_{camera2gyroscope} * T_{gyroscope2droneCenter} * T_{droneCenter2ground} * T_{ground2groundImage} \tag{2}$$

With knowledge of four corresponding points from the image plane and the ground map it is possible to calculate the homography-matrix to project the whole image on the ground. The scaling factor is entailed in the homogeneous transformation matrix $T_{ground2groundImage}$. However, real cameras differ from the pinhole model because of distortion effects which parameters can be calculated using e.g. Zhang's [18] calibration method. Using these parameters the image can be undistorted resembling a pinhole camera model which is projected in the described manner. The resulting map contains the necessary metric scaling to identify sufficiently large areas for the UAV to land.

3.3 Stabilization

The robustness of the landing spot detection is dependent on multiple aspects. Even when the UAV is not moving, the pictures taken by the camera will still show minimal changes. Minor movements of the drone introduce motion and rotation blur on top of this. Although the semantic information in the scene does not change, slightly different data is processed by the segmentation algorithm which could consecutively lead to different results. The approach has to be insusceptible to these changes for reliable results.

Each segmented image mapped on the ground represents only one instance of the map. To avoid the loss of this information when the UAV's pose changes, the instances are Geo-tagged with absolute or relative (if no GPS would be available) GPS data. The last instances, together with the newest image are oriented to the north and overlaid with each other, where the newest instance is assumed to have the most precise information. The previous classifications for each overlaying pixel can be compared to reduce the before mentioned flickering. In GPS denied situations the Geo-tag could be substituted by INS data, accepting a possible long-term corruption of the map due to drift or reconstruct the pose of the drone by the raw image using pose estimation methods.

Assuming that e.g. grasslands are suited for an emergency landing, the presence of a single pixel in the segmented image classified as grassland does not grant a suitable emergency landing spot. The scale of a pixel is dependent on the height and orientation of the UAV. To filter the ground for large enough spaces, an erosion with a circular kernel and radius of e.g. two meters depending on the UAV's dimensions is conducted.

3.4 Landing Logic

Assuming that the necessity of performing an emergency landing is determined by certain circumstances and the scope of this work is to fulfill the emergency landing procedure, the logic for identifying the safest and most suitable emergency landing spot and safely guiding the UAV has to be covered. Therefore, a BT performs the micro logic to realize this autonomous function. BTs are very well suited for controlling autonomous agents and are thus often used in video games, simulations and robotics [19]. A major advantage in this context is the flexibility and reactivity compared to many other rule-based approaches which can be achieved, while the process is still deterministic and thus basically suitable for certification.

While the BT structure for the emergency landing is not yet fully defined and implemented, the current approach initially detects safe landing spots based on the stored metric scaled map containing the segmented and smoothed ground information. Thereby not the first possible but the safest landing spot can be selected under certain criteria and the path can be planned by a safe path planning algorithm (to be published). While approaching the spot the validity of this decision is continuously checked and testified. If any major change in the segmentation is detected or for example a human or another moving object approaches the coordinates, the landing procedure will be set on hold. Based on further rules and depending on the event triggering the emergency landing another safe landing site could be selected in this case using the stored ground map again. The general goal for future work is to adapt as close as possible to the decisions a human pilot would have made by adjusting and extending the BT continuously based on expert knowledge while ensuring consistency and safety within the emergency landing. Therefore decisions will become further dependent on the UAV's flight altitude and other environmental and situational criteria such as general ratio of segmented ground classes, weather, and mission objectives among others.

4. Results

4.1 Simulation

A SIL setup using AirSim [20], PX4 Autopilot [21], and QGroundControl [22] has been implemented in which the algorithms are tested. QGroundControl is running in the same way it is used on the field to monitor the UAV's system status. The PX4 Autopilot is the software that is running on the Pixhawk flight control unit used on the real drones and AirSim is an Unreal Engine based simulation environment used to simulate the UAV's flight physics, environmental conditions and the on-board sensors. The same toolchain that is necessary on the real UAV is resembled in this simulation. The system components communicate with each other by ROS [23] to warrant the transfer to the hardware setup. Within this virtual environment it is possible to detect landing spaces and safely guide the UAV to the desired location on ground. The simulated process is depicted in figure 4. The AirSim environment is illustrated in the upper left screen. In the upper right the footage of a straight downwards-facing camera is displayed and below the output of the respective semantic segmentation is shown, which is projected onto a ground map shown in the bottom right screen. In this example only the road class is defined as relevant landing area and the binary result of the erosion with a circular kernel is depicted next to the projection.

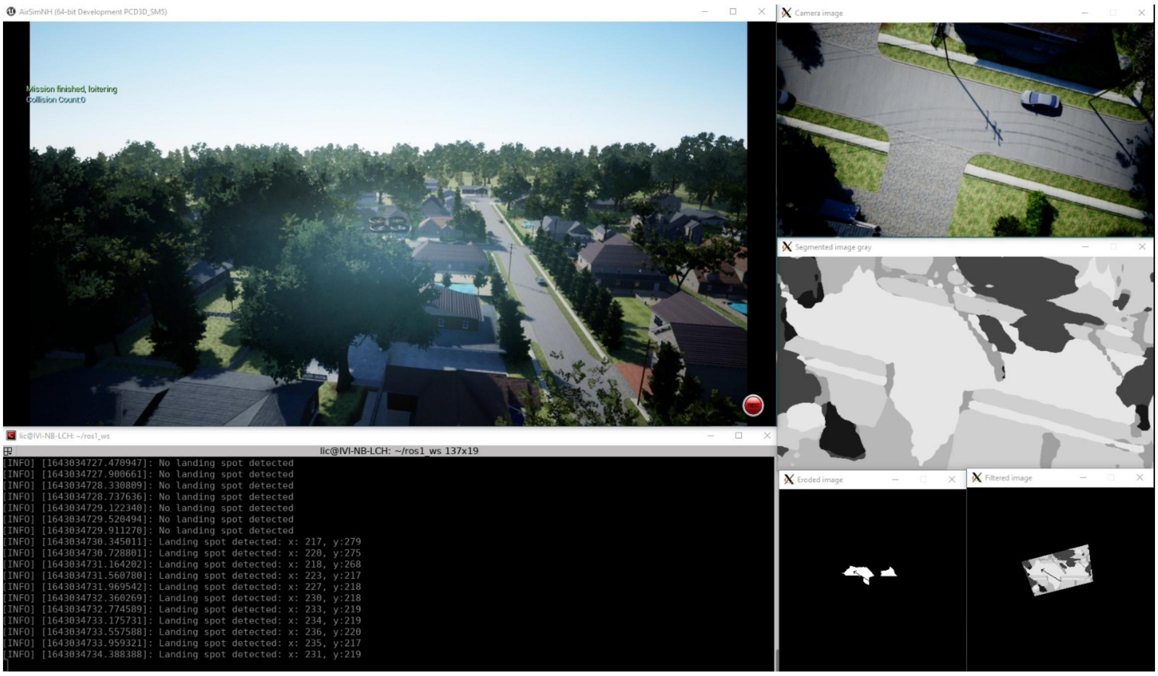

Figure 4 – Projection of the image to map coordinate system; note that the segmentation algorithms were trained on real drone footage, therefore they lack performance in the simulated setup

At the current status, the segmentation results lack stability both within the simulation as well as on real world data. It is possible to identify e.g. roads, humans, and vegetation, however the segmentation has to become more robust. Furthermore, the current algorithms do not take previous frames into account but process each image individually. Therefore, minimal changes from frame to frame can lead to different classification results which is reflected by flickering of the segmentation.
This is counteracted by stabilizing the map as described in 3.3. Further possibilities to improve the segmentation are discussed in section 5.

4.2 Real World Validation

All elements of the pipeline in figure 1 showed basic suitability in the simulation, furthermore, they have been showcased in the real word with a simplified example by identifying the nearest suitable emergency landing spot over a crowded park. Hereby, the suitability is only defined with a circular place of 6 m diameter free of obstacles and humans on grassy terrain. For this purpose, a test flight in 12 m altitude has been conducted with multiple people standing/walking beneath the UAV blocking potential landing areas.
To confirm that the identified landing spot satisfies the spatial requirements, a calibration target was placed on the ground and measured utilizing the emergency landing pipeline. For this reason, the pipeline was adjusted to map the raw camera footage instead of the segmented labels. Due to the metric scaling of the ground map the size of the calibration target corresponds directly to its length

Figure 5 – Measurement with the calibration target, the QR code has been blurred

in pixel. A square calibration target with an edge length of 2.0000 m is used for the accuracy measurement with the map resolution set to 0.01 m/pixel. The undistorted camera footage is projected on the ground using a radar to determine the UAV's flight altitude. The sub pixel dimensions of the calibration target can be measured by up scaling the image and identifying the outer most corners of the target. The cropped target is depicted in figure 5 with the measured lines drawn in red. From an altitude of approximately 10 meters the red lines are 404.20 and 403.40 pixel long, corresponding to 2.0210 m and 2.0170 m respectively, resulting in a mean error of 0.95%.
However, the segmentation of the NN is not as accurate as the projection miss-classifying patches of the image. These patches exhibit irregular shapes differing in form and location from frame to frame. Because of this, the ground map has to be stabilized using Geo-tags where ground projections of previous instances are transformed to the same reference system. This process can be visualized

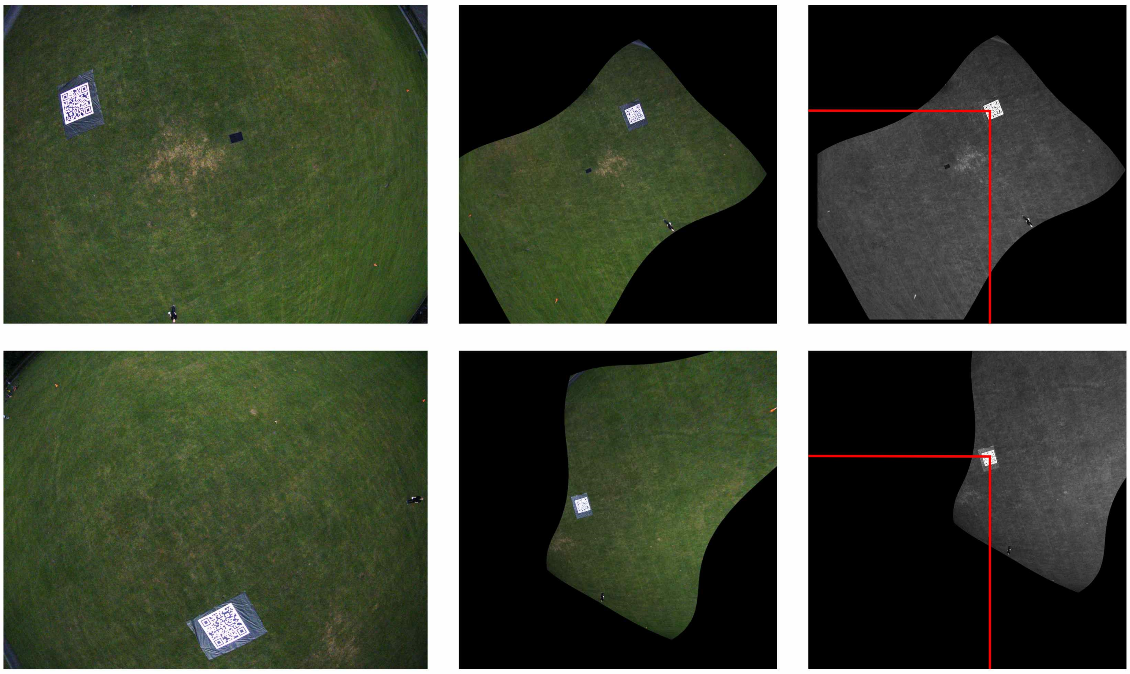

Figure 6 – Stabilization process depicted with raw image and visualized using land marker. Left: raw camera footage, middle: undistorted mapping, right: same reference frame and mark at (570, 330)

tracking a land marker over multiple frames while the UAV moves. This procedure is depicted in figure 6 using the UAV's pose estimation as Geo-tag. The raw camera footage on the left side is being undistorted and projected in the middle image. On the right side the images are transformed to the same reference system. The pixel (570,330) on the land marker is highlighted in red in both frames to visualize the accuracy. Note the orientation of the land marker in the raw footage and the projected maps are rotated to the north. In the next step a majority vote would take place on all segmented labels transformed to the same reference system for every pixel choosing the most stable class.

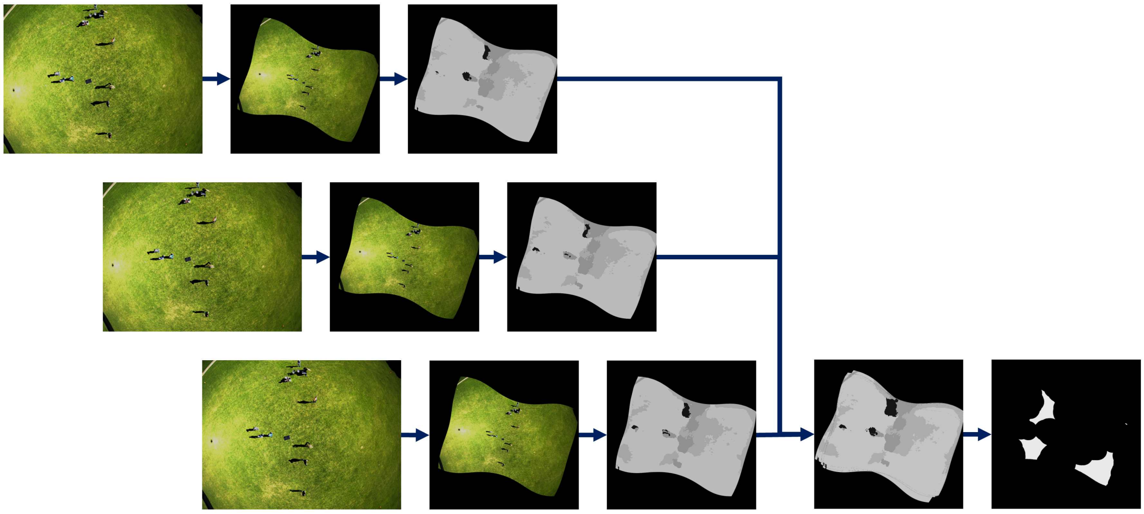

Figure 7 – Identification of suitable emergency landing spots. From left to right three consecutive camera frames are being projected, segmented, jointly filtered, and eroded

To take the aberrations - stemming from errors in the intrinsic and extrinsic sensor calibration, timely deviations in the data acquisition of the multi sensor setup as well as the performance of the NN - into account, the safety zone radius for an emergency landing spot is defined with 3 m. On the outer right side of figure 7 every possible landing spot that satisfies this requirement is represented by a white pixel. Following the pipeline from figure 1 the camera frames are segmented, undistorted and projected onto a ground map. In this example, three consecutive frames are jointly stabilized using a majority vote prioritizing the occurrence of the class *humans*. Subsequent, an erosion with the previously defined circular kernel is performed on the class *vegetation* omitting everything else. Lastly, each pixel on the resulting image represents a relative coordinate suitable for an emergency landing. At this point the commandeering logic has to take over control, tracking the chosen landing spot and reacting to the UAV's surroundings. This has been omitted in this simplified showcased example.

5. Discussion

The simulation results of the presented approach are very promising however the segmentation needs considerable improvement. The performance of the NN improves in real drone flights outside the simulation, as it has been trained on real drone footage and therefore the domain shift caused by a lack of photo realism in the simulation, results in lower performance. Nevertheless, as has been mentioned in 3.1and is clearly visible in figures 4 and 7 the image segmentation has to improve especially in terms of accuracy and stability. The first way to influence the behaviour of the segmentation is to manipulate the data set on which the network is trained. For this reason the Aeroscapes data set has already been modified by omitting all images which are not taken from an aerial perspective. Furthermore, the augmentation of the images, e.g. motion blur, various rotations and clipping, has been extended during the training. The data set could be further improved by relabeling certain classes, i.e. merging walkways and roads into one class. Although the high level information of these classes differ significantly, it is difficult to differentiate them in images. Additional hyper parameter optimization has to be performed for the segmentation algorithms and the extension by continuous information processing (e.g. image flow) has to be investigated. On top of this, it is especially important to be able to identify humans for safe emergency landing operations. Because of this it is possible to adapt the filtering process described in 3.3 and 4.2 to prioritize the class *humans* and include the actuality of the frames during the majority vote.

Although real time capabilities have been identified in 2. as a necessity for a successful emergency landing operation this is not yet possible at the current state of the implementation. However, there is a plethora of opportunities to speed up the calculations such as the parallelization of certain tasks, as well as tuning of the NN.

The accuracy of the mapping process is dependant on multiple factors, i.e. exact extrinsic and intrinsic sensor calibration of the camera and radar with regards to the flight controller as well as the choice of the altitude measurement. This has particular impact when the ground relief does not resemble a flat plane and the relative altitude above ground changes. The higher the flight altitude the stronger the error propagates.

The emergency landing functionality should be regarded as a closed system being as independent as possible. Therefore, it does not require a priori maps of the UAV's surrounding but is generating its own short lived maps in the direct vicinity of the UAV fusing the segmentation labels of the last number of acquired frames using Geo-tags as described in 3.3 In the case of GPS loss the fallback to INS is only marginally compromised by drift, as the map is short lived and the error therefore does not accumulate. However, for the integration with other map-based systems or to store possible emergency landing spaces for future flights, the absence of GPS would compromise the generated map. Methods such as the reconstruction of the pose of the drone by the raw image using pose estimation methods have to be investigated.

6. Conclusion

In this paper we described the current status of our approach to enable save and autonomous emergency landings based on image segmentation and metric transformation. The basic suitability of our pipeline has proven to perform emergency landing operations without a priori information in simulations, and its elements were validated with real world data. A radar has been integrated for advanced altitude measurements, and the functionality of the projection to conserve metric scaling has been showcased using a calibration target. In addition, the recognition of suitable emergency landing sites was demonstrated in a simplified example flying over a crowded park.

7. Outlook

Future work will enable the real-time capability on the embedded system and investigate the integration of a more advanced 3D imaging radar. This would enable an extensive investigation of the ground, registering slopes, buildings, and other objects. The simplification of a planar surface could therefore be substituted by a better approximation closing the gap to the true structure of the ground. Furthermore, the semantic segmentation will get further improved and be complemented by an instance discrimination. By tracking single objects, the trajectories of vehicles and humans can be estimated and the risk of possible collisions during and after the landing operation could be mitigated by the BT. Moreover, the information gained during the mapping process can be saved for future mission planning to consider previous routes with e.g. increased human activity or a plurality of possible emergency landing spots. The latter could be used to steer the UAV into such a direction when it can not identify any spots in the vicinity at the moment.

8. Copyright Statement

References

[1] J. Guerin, K. Delmas, and J. Guiochet, “Certifying emergency landing for safe urban uav,” in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 55–62, 2021.

[2] B. Xia, I. Mantegh, and W. Xie, “Integrated emergency self-landing method for autonomous uas in urban aerial mobility,” in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pp. 275–282, 2021.

[3] M.-F. R. Lee, A. Nugroho, T.-T. Le, Bahrudin, and S. N. Bastida, “Landing area recognition using deep learning for unammaned aerial vehicles,” in *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pp. 1–6, 2020.

[4] N. E. Bodunkov and N. V. Kim, “Autonomous landing-site selection for a small drone,” *Russian Engineering Research*, vol. 41, pp. 72–75, Jan 2021.

[5] O. Bektash, J. N. Pedersen, A. Ramirez Gomez, and A. I. Cour-Harbo, “Automated emergency landing system for drones: Safeeye project,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1056–1064, 2020.

[6] A. R. Gomez and A. la Cour-Harbo, “Emergency landing decision method for unmanned aircraft,” in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1233–1239, 2021.

[7] K. Mukadam, A. Sinh, and R. Karani, “Detection of landing areas for unmanned aerial vehicles,” in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pp. 1–5, 2016.

[8] P. C. Lusk, P. C. Glaab, L. J. Glaab, and R. W. Beard, “Safe2ditch: Emergency landing for small unmanned aircraft systems,” *Journal of Aerospace Information Systems*, vol. 16, no. 8, pp. 327–339, 2019.

[9] J. A. Gonzalez-Trejo and D. A. Mercado-Ravell, “Lightweight density map architecture for uavs safe landing in crowded areas,” *Journal of Intelligent & Robotic Systems*, vol. 102, p. 7, Apr 2021.

[10] J. Gonzalez-Trejo, D. Mercado-Ravell, I. Becerra, and R. Murrieta-Cid, “On the visual-based safe landing of uavs in populated areas: A crucial aspect for urban deployment,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 7902–7909, 10 2021.

[11] T. Hinzmann, T. Stastny, C. Cadena, R. Siegwart, and I. Gilitschenski, “Free lsd: Prior-free visual landing site detection for autonomous planes,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2545–2552, 2018.

[12] P. Fraczek, A. Mora, and T. Kryjak, “Embedded vision system for automated drone landing site detection,” in *Computer Vision and Graphics* (L. J. Chmielewski, R. Kozera, A. Orłowski, K. Wojciechowski, A. M. Bruckstein, and N. Petkov, eds.), (Cham), pp. 397–409, Springer International Publishing, 2018.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

[14] M. Kim, B. Park, and S. Chi, “Accelerator-aware fast spatial feature network for real-time semantic segmentation,” *IEEE Access*, vol. 8, pp. 226524–226537, 2020.

[15] X. Chen, X. Lou, L. Bai, and J. Han, “Residual pyramid learning for single-shot semantic segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[16] I. Nigam, C. Huang, and D. Ramanan, “Ensemble knowledge transfer for semantic segmentation,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1499–1508, 2018.

[17] M. Fonder and M. Van Droogenbroeck, “Mid-air: A multi-modal dataset for extremely low altitude drone flights,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 553–562, 2019.

[18] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[19] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, “A survey of behavior trees in robotics and ai,” 2020.

[20] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” 2017.

[21] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.

[22] “Qgroundcontrol gcs.” `http://qgroundcontrol.com/`. Accessed: 2022-02-09.

[23] “Ros - robot operating system.” `https://ros.org/`. Accessed: 2022-02-09.