2016-1

Landing Zone Determination for Autonomous Rotorcraft in Surveillance Applications

Timothy McLain
Department of Mechanical Engineering, Brigham Young University, mclain@byu.edu

Gary J. Ellingson
Department of Mechanical Engineering, Brigham Young University, gellings13@gmail.com

Justin Mackay
MIT Lincoln Labratories

Original Publication Citation

Mackay, J., Ellingson, G., and McLain, T. Landing Zone Determination for Autonomous Rotorcraft in Surveillance Applications, Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA 2016-1137, January 2016, San Diego, California.

BYU ScholarsArchive Citation

McLain, Timothy; Ellingson, Gary J.; and Mackay, Justin, "Landing Zone Determination for Autonomous Rotorcraft in Surveillance Applications" (2016). *Faculty Publications*. 1893.
https://scholarsarchive.byu.edu/facpub/1893

Landing Zone Determination for Autonomous Rotorcraft in Surveillance Applications

Justin Mackay[*] Gary Ellingson[†] Timothy W. McLain[‡]

Brigham Young University, Provo, UT, USA

This paper presents an approach for finding possible landing sites for a rotorcraft from an inertially referenced point-cloud model of the environment. To identify potential landing sites that are suitably flat and level, a grid-based random sample consensus algorithm separates the terrain map into discrete areas for plane-fitting analysis. Landing sites are selected that satisfy constraints on flatness and levelness while optimizing the surveillance target's visibility. Flight test results are presented from a small multirotor aircraft flying over a scale-model cityscape. Results from real-time landing-site experiments are presented and discussed.

I. Introduction

Unmanned air vehicles (UAVs), especially rotorcraft, have the capability to land at any flat location devoid of hazardous obstacles. Automated landing removes the need for a pilot to control the aircraft, allowing personnel to focus on other mission critical objectives. Although automated landing of rotorcraft UAVs has been performed at prepared sites,[1–4] the ability to land at unprepared sites offers significant advantages by fully exploiting a rotorcraft's versatility to land anywhere that is suitably flat and level.

One example where landing at an unprepared site would be helpful is for delivering military cargo with a UAV. To deposit the cargo near troops, the UAV needs to recognize landing hazards and navigate to avoid harming individuals and the cargo. The idea of delivery using UAVs extends to commercial applications as well. Amazon, DHL, Google, and other companies have publicly expressed interest in using UAVs for rapid package delivery purposes (e.g., Ref. 5). The ability to land safely and reliably in unprepared locations is a critical capability to fulfill this vision of UAV-based package delivery.

Landing can also conserve onboard energy and extend the duration of missions. Some applications, such as surveillance and monitoring (e.g., wildlife or infrastructure), may allow the mission to be performed from a fixed vantage point. Given that the largest use of energy on small UAVs is for sustaining flight, perching on a raised structure and conserving energy offers potentially significant advantages for missions of longer duration.

When a surveillance target remains stationary for a period of time, it is sensible for the UAV to land in a position where it can continue surveillance without exerting the energy to hover or fly. Perching at a landing location allows it to conserve energy while continuing to fulfill the surveillance objectives. Thus, the vehicle can remain in the area for a longer period of time without the need to refuel or recharge. The best vantage point to continue surveying an area would be from an elevated height above the surveillance target. It would be ideal for the vehicle, therefore, to land on a raised structure near the area of interest. One area where this strategy could be utilized is in urban environments. Typically, urban environments have multiple vantage points due to the variety of structures. Some structures give a better view of the area of interest than others. This creates the need to choose between multiple landing location possibilities.

This paper presents a method for finding suitable landing sites for small UAVs that take into account surveillance objectives. A grid-based RANSAC approach is used to identify flat and level areas based on point-cloud data produced from onboard sensor measurements. Metrics for surveillance target visibility are used to select the landing locations for optimal target surveillance. The landing site optimization algorithms

---

[*] Associate Staff, MIT Lincoln Laboratory

[†] Graduate Student, Department of Mechanical Engineering, Student Member, AIAA

[‡] Professor, Department of Mechanical Engineering, Associate Fellow, AIAA.

presented are implemented on a small hexacopter aircraft. Flight results over a model cityscape are presented demonstrating the ability to identify suitable landing sites in real time.

II. Related Work

Automated landing has been used for years in commercial and military aircraft.[6] Typical automated landing with piloted aircraft has been done at prepared sites, such as helicopter landing pads or aircraft runways. The rotorcraft capability of being able to land at unprepared sites provides more possible landing locations for a UAV. Landing at unprepared sites requires a three-dimensional (3D) representation of the environment that is then evaluated for possible landing sites. In Ref. 7, monocular images are used to identify planar areas for landing. Refs. 8–11 use a stereo camera system to create a point cloud that is evaluated locally for slope and roughness. Both Refs. 12 and 13 use structure from motion to generate a point cloud used to evaluate the local slope and roughness of the environment. In Refs. 14, 15 lidar is used with precise position information on a full-size helicopter to generate a point cloud. Once again, a local slope and roughness technique is used to find possible landing locations. The work in Ref. 14 further analyzes the point cloud data by evaluating ground contact, center-of-gravity, wind direction, and helicopter clearance.

Excluding Ref. 7, all of these attempt to create a point cloud of the environment and evaluate the point cloud based on local slope and roughness. This work follows a similar approach. Two major differences in this work are the grid-based analysis accounting for noise and gross errors in sensor measurements and independent analysis of point clouds removing the need to align multiple scans of the environment.

III. System Architecture

One objective of this work was to demonstrate the ability to select landing zones in flight in real time. This was done using a small multirotor aircraft flying over a model cityscape in an indoor lab environment. The following sections describe the aircraft hardware and sensors used and the software architecture that was developed to meet this objective.

A. Hardware and Sensors

The aircraft used for testing is the MikroKopter Hexacopter XL. The onboard processor is an Intel Core i7-2710QE. The weight including the sensors is 3.9 kilograms. The aircraft has a flight time of 15 minutes. When landed, the motherboard can run for 7 hours with the same batteries. The discrepancy between battery life flying and landed highlights the possible energy savings when perching during surveillance missions. The range sensor used was the ASUS Xtion PRO Live. The ASUS Xtion PRO Live gives RGB and range images at 30 frames per second at a resolution of 640 × 480 pixels, resulting in a maximum of 307,200 depth measurements for each image. The precision and accuracy of the Xtion sensor decreases as the distance to an object increases. In Ref. 16, the precision is shown to change from between 2 mm and 6 mm for an object 1 meter away to between 4 mm and 12 mm for an object 2 m away. Similarly, the accuracy of the sensor changes from between 5 mm and -15 mm for an object 1 meter away to between 5 mm and -25 mm for an object 2 m away. One other important discovery from Ref. 16 is that the accuracy and precision degradation is proportional to the distance squared. This precision enables laboratory testing with the miniature cityscape.

One other limitation is the Xtion camera does not work in direct sunlight. For outdoor flights, a different sensor, such as lidar, needs to be used. Lidar can produce a dense point cloud similarly to the Xtion camera, except with better precision at greater distances. Lidar can have a maximum range above 1 km with an accuracy of 10 mm and precision of 5 mm.[17]

B. Landing Architecture

The landing zone detection process, displayed in Figure 1, is broken up into four components: data collection, point cloud modeling, safe landing zone analysis, and landing. Data collection consists of gathering camera and sensor information about the landing environment. The point cloud modeling components consists of using the sensor information to calculate 3D point cloud of the environment. This involves processing the imagery, calculating 3D points relative to the aircraft, and putting the 3D points into an inertial reference frame. The point cloud modeling process provides the necessary terrain information for finding areas that are adequate for landing in safe landing zone analysis. The safe landing zone analysis component fits multiple planes to the point cloud and determines the slope and roughness of each plane. It can then categorize areas as possible landing locations. The landing component then selects the optimal landing location for surveillance based on calculated distance and grazing angle to the target and commands the aircraft to maneuver to that location.

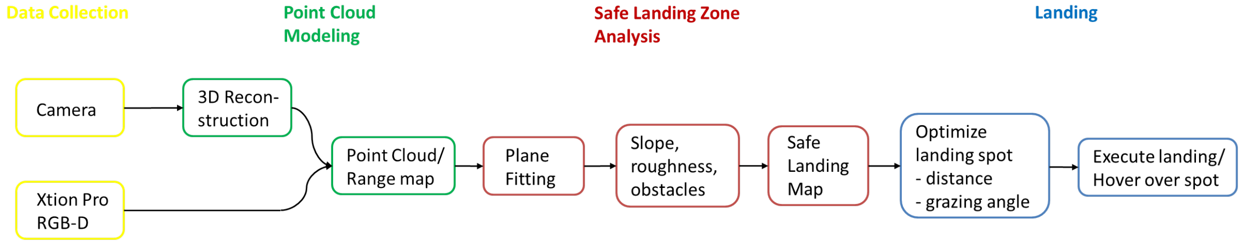
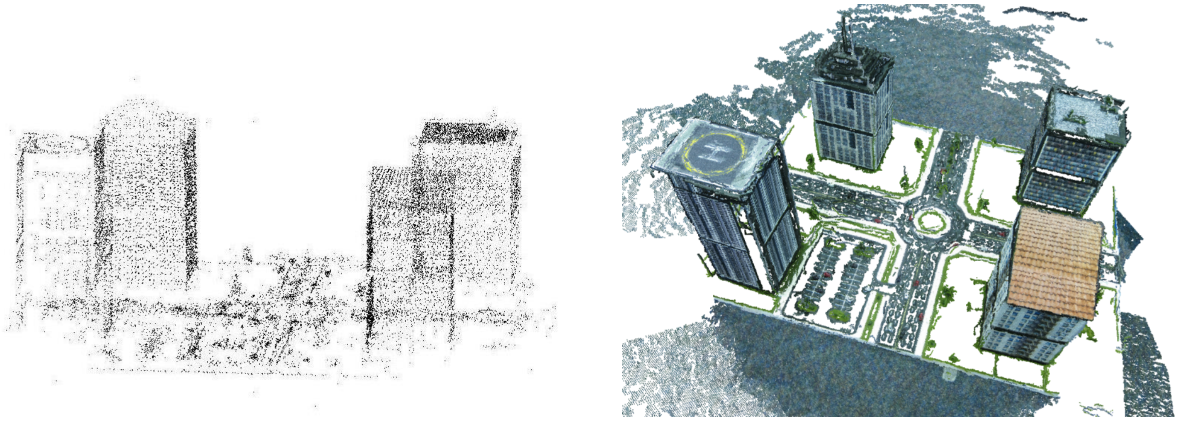


Figure 1: Safe Landing Process

C. Point Cloud Generation

The cityscape environment was modeled using a 3-dimensional (3D) point cloud. The surfaces of the environment were represented using points in Euclidean space. Point-cloud models can be created through a variety of means. Two approaches used in this work are structure from motion using camera imagery and direct measurement of range. Both methods create a suitable point-cloud model for safe landing area determination.

Structure from motion utilizes multiple images of a scene from different perspectives to obtain 3D information that is used to create a 3D model of the environment. This reconstruction of the model allows for the capability to find the optimum landing location over the entire area. Using only the most distinct features in the image allows for the generation of a sparse point cloud model seen in Figure 2a. A dense model can also be created, as seen in Figure 2b. The sparse model was created in 13 sec, and the dense model was created with an additional 92 sec from 20 images with 1600 × 1200 resolution using the onboard i7 computer.

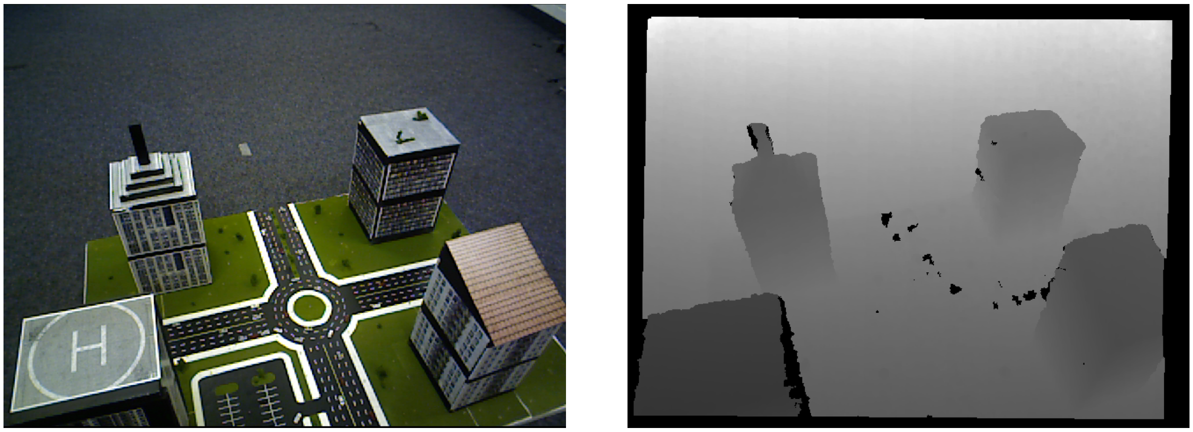


(a) Sparse point cloud.

(b) Dense point cloud.

Figure 2: Structure from motion point clouds.

Alternatively, a range or depth sensor, such as the ASUS Xtion, can also be used to directly generate the point cloud. The depth sensor provides information at a faster rate than the stereo options, requiring a faster processing time if all of the depth image information is to be utilized. Outputs from the Xtion camera are shown in Figure 3, with the color image shown in Figure 3a and the depth image shown in Figure 3b. In the depth image, darker shades of gray correspond to shorter depth measurements. The black portions of the image are areas where the sensor could not determine the distance.



(a) Xtion color image.

(b) Xtion depth image.

Figure 3: Images from the Xtion camera.

IV. Safe Landing Area Determination

One method to find possible landing locations is to analyze small portions of the point cloud and determine whether each portion is suitable for landing. Separating the area with a grid map has been used in Refs. 14, 18, 19 to find potential landing sites. These papers used the vertical standard deviation of points and a least squares fit to estimate horizontal planes. A major assumption of using least-squares and the standard deviation approach is the range measurements have little noise and no spurious errors. The least-squares approach assumes that the deviation of the points is proportional to the size of the point cloud, ignoring uncompensated gross errors.

A. Planar Modeling

The objective of planar modeling is to determine the flat and level portions of the 3D point cloud that are suitable for landing. A grid-based RANSAC approach is used to divide the depth image into grid elements. RANSAC is used to fit planes to the data points in each grid element. The result is a discretized map where each grid element is designated as a possible landing location or not a possible landing location based on the goodness of fit with the planar model. To form the grid, the horizontal span of the point cloud in the x and y dimensions is found from the minimum and maximum x and y values of the points in the point cloud. The grid element width in the x and y dimenstions are chosen to equal the size of the aircraft. This ensures the landing areas are large enough for the aircraft to land. The point cloud data is then iteratively analyzed until all grid elements have been evaluated with RANSAC. This gridding method is described in Algorithm 1.

Algorithm 1 Grid-based RANSAC

```
Require: Point cloud (X,Y,Z points)
Determine the minimum and maximum for the $x$ ($x_{\min}$, $x_{\max}$) and $y$ ($y_{\min}$, $y_{\max}$) values of the points
Determine increments for values of $X$ ($x_{\text{inc}}$) and $Y$ ($y_{\text{inc}}$) based on the aircraft size
for $i = x_{\min}$ to $x_{\max}$, incremented by $x_{\text{inc}}$ do
    for $j = y_{\min}$ to $y_{\max}$, incremented by $y_{\text{inc}}$ do
        Select all points with $X$ values between $i$ to $i + x_{\text{inc}}$ and $Y$ values between $j$ to $j + x_{\text{inc}}$
        Pass these points into RANSAC for plane fitting
    end for
end for
```

The RANSAC algorithm is used to find the equation of the plane that best models the set of 3D points. Regular least-squares attempts to minimize the distance from all data points to the planar model. RANSAC assists by finding the best model for a subset of the points (called the consensus set) and designating the remaining points as outliers to the model. The approach for fitting a plane to a set of points using RANSAC is shown by Algorithm 2. The threshold τ is based on the measurement noise from the sensor. RANSAC estimates the parameters of the plane that best fit the consensus set, removing the effect of outliers and gross measurement error from the estimation. This creates a more accurate estimate of the consensus set. The parameters of the planes fit to the point cloud data for each grid element provide the slope and roughness of the terrain. Outliers to the consensus set are classified as hazardous landing locations.

Algorithm 2 RANSAC Plane Fitting

```
Require: Three or more points ($\mathbf{p}$)
for $i = 1$ to $\ell$ iterations do
    Select three different random points ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$)
    Fit a plane to these points
    for $j = 1$ to number of points do
        Calculate orthogonal distance $\varepsilon$ from $\mathbf{p}_j$ to the plane
        if Threshold $\tau > \varepsilon$ then
            Add $\mathbf{p}_j$ to temporary consensus set $\chi_i$
        end if
    end for
    if Size $\chi_i > \chi$ then
        $\chi \leftarrow \chi_i$
    end if
end for
Use orthogonal regression with $\chi$ to find the best-fit plane
```

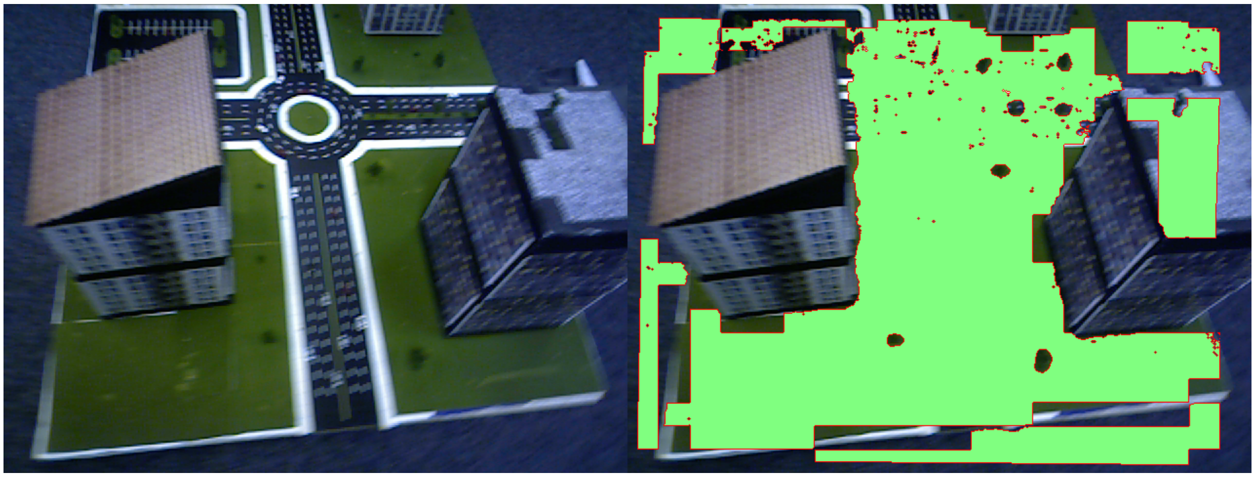


Figure 4: Grid-based RANSAC result. The left portion of the image is the original color image. The right portion shows possible landing locations in green. The block-like landing locations can clearly been seen.

Separating the points in this manner is a fast and efficient process. Unfortunately, this approach also results in block-like evaluations of good or bad landing locations. This can be seen in Figure 4. This issue is documented in other literature using grid maps to determine possible landing locations. This loss of resolution is critical when trying to land on the edge of a building for surveillance. This loss of resolution can be solved by changing the distance between grid elements.

Grid-based RANSAC can be modified to ensure that each grid element overlaps with the adjacent sections. Overlapping adjacent grid elements ensures each point is evaluated more than once with a different portion of the point cloud. Having overlapping grid elements reduces block-like issues and recognizes more possible landing locations. The improved result comes at the cost of increased computation time. Results are shown in Figures 5 and 6. These figures show a significant improvement in finding possible landing sites over the original grid-based approach. The block-like issues are resolved allowing for a better knowledge of possible landing sites. This improves the capability to land on the edge of a building for surveillance purposes.

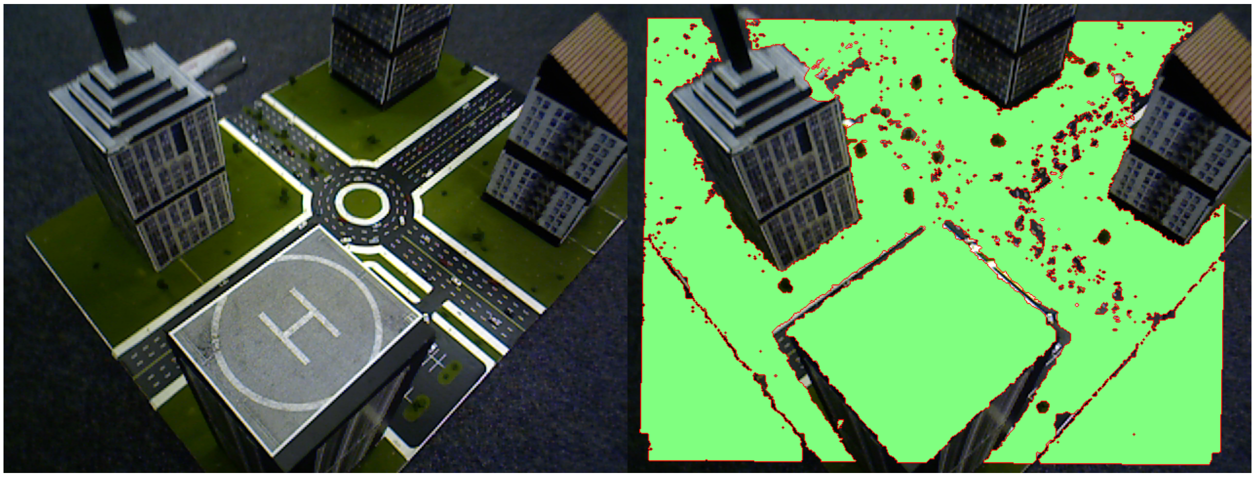


Figure 5: Modified grid-based RANSAC to reduce block-like results.

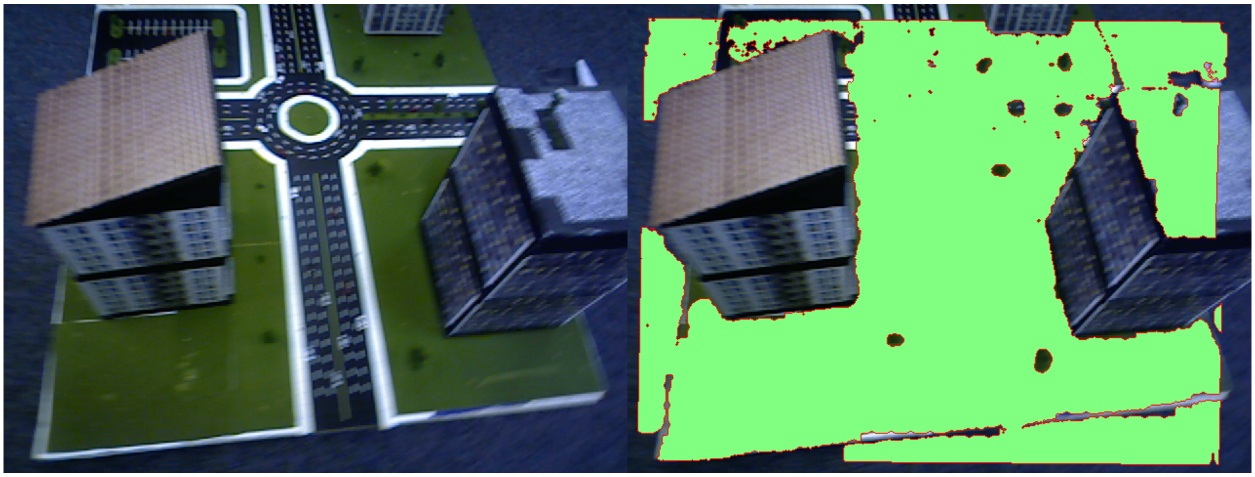


Figure 6: Second result of modified grid-based RANSAC.

B. Surveillance Landing Selection

Plane fitting results in many potential landing planes, but each position within the plane must be further evaluated to be classified as a possible landing location to produce a safe landing map. The map is determined by first selecting the planes that have an appropriate slope for landing. Points within the plane are then given boundary areas around landing hazards to classify points as possible landing locations. The size of the boundary area depends on the size of the aircraft. Having boundary areas prevents the aircraft from landing too close to an obstacle. The boundary areas are computed by iterating through the bad landing selection points and checking for points within the distance to the aircraft.

In our approach, the aircraft must also determine the optimal landing location to perform surveillance by evaluating the landing locations. This optimization is based on metrics describing the distance to the desired surveillance area, the height above the target, the grazing angle to the target, and the absence of occlusions. Some of these characteristics are more critical than others. For example, an occlusion would completely prevent viewing the object of interest, while grazing angle, distance to the area, and height above the target all affect how well the target can be seen. The height above ground affects how much additional area around the object is in the viewing field. This is helpful when observing a large area. The distance to the area also affects how much additional area around the object is in the viewing field. Being too close to the object of interest causes the surveillance to be more overt. It also increases the amount of detail in the image.

The first step after finding possible landing locations is to evaluate the grazing angle to the target. This is done by calculating the angle ϕ from each possible landing site to the target using the expression

$$\phi = \tan^{-1}\left(\frac{p_z - t_z}{\sqrt{(p_x - t_x)^2 + (p_y - t_y)^2}}\right),$$

where p_x, p_y and p_z are the position of the point and t_x, t_y, and t_z are the position of the surveillance target. The grazing angle is limited by 20 deg for the lower bound and 80 deg for the upper bound. The lower bound removes the majority of ground points since they have a grazing angle close to 0 deg. The upper bound removes possible landing sites on structures that are too close to permit viewing the target. The sites outside the grazing angle window are still safe landing sites and can be used during landings when surveillance is not an objective.

The landing sites left after boundary checking and between the grazing threshold are those that can provide surveillance of the target. To choose between these sites, the distance to the target is used. The site with the shortest distance to the target is then chosen as the optimal landing location.

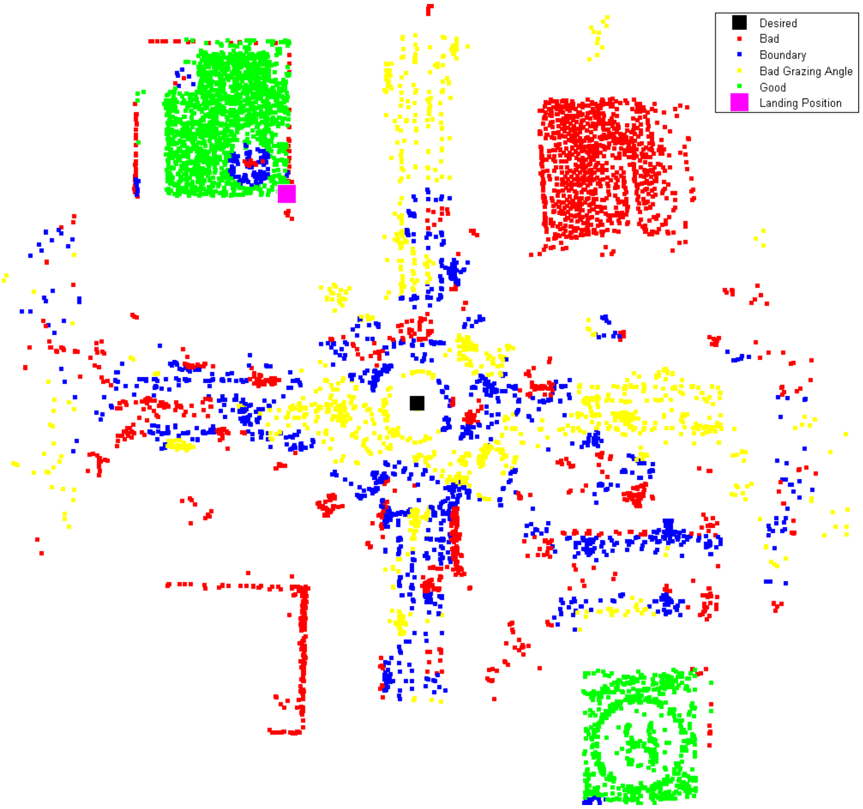


Figure 7: Surveillance selection showing different classifications of points based on their suitability for landing.

A result from surveillance landing selection is shown in Figure 7 where points are classified by color. The green dots are possible landing locations. Red dots show poor or unfeasible landing locations. Blue dots are boundary areas around poor landing locations. Yellow dots show possible landing locations with a poor grazing angle, in this case, along the ground. The black square is the surveillance target location, and the magenta square shows the selected landing location.

V. Testing

Safe landing determination was performed indoors using the hexacopter and Xtion camera. The hexacopter initially starts flying a spiral pattern around the target area. This allows the Xtion camera to obtain depth images from multiple perspectives. The depth images are used to create a point cloud model of the environment. Suitable landing sites are calculated and analyzed to determine the optimal landing site for surveillance of the target area. After a predetermined amount of time, the hexacopter stops spiraling and hovers over the selected landing site.

The safe landing zone determination execution time for each Xtion depth image is 9.5 msec in C++. This allows for real-time landing zone analysis for each depth image. Performing landing zone analysis on each image removes the necessity to correlate each depth image to align the point cloud. Instead, the cost value of the landing site selection from each depth image is compared to find the optimal landing choice from all acquired data.

A flight path of the hexacopter performing real-time landing analysis is shown in Figure 8. The black points are the 3D reconstruction result of the city model. The red points show the spiral path of the hexacopter as it searches for a landing location. The blue dots show the position of the hexacopter hovering over the selected landing location. In this scenario, the hexacopter chose to land on the corner of the flat roof without trees. Multiple flight tests were performed finding the optimal landing location for different target positions. The UAV successfully selected flat-portions of rooftops that were clear of obstacles for landing.

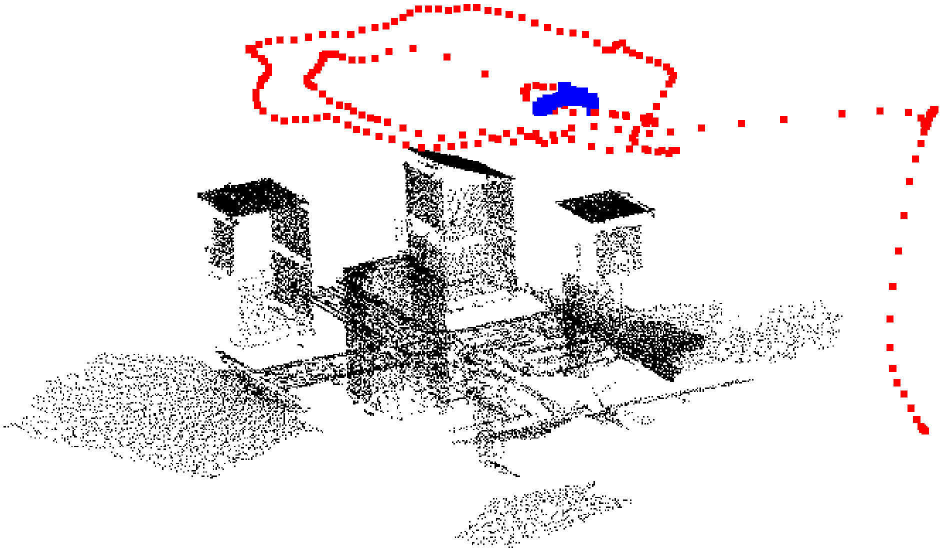


Figure 8: Flight pattern. Red dots show the position of the aircraft. Blue dots are where it hovered for performing a simulated landing.

VI. Conclusion

Grid-based RANSAC rapidly finds portions of an urban environment point cloud model that represent suitable landing sites using a RANSAC plane-fitting technique. The low computational requirements enable this approach to be used on embedded systems on board UAVs. Tests on a simulated city environment were performed demonstrating optimal landing site selection for a surveillance task. The results were obtained from point clouds generated using structure from motion and an ASUS Xtion depth sensor. This landing zone determination approach can be extended to other range sensors and algorithms that generate point-cloud models.

Acknowledgments

This research was performed in collaboration with Scientific Systems Company, Inc. in support of OSD SBIR contract number FA8651-13-M-0094.

References

[1] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Visually-Guided Landing of an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 3, 2003, pp. 371–381.

[2] Shakernia, O., Ma, Y., Koo, T. J., and Sastry, S., "Landing an Unmanned Air Vehicle: Vision Based Motion Estimation and Nonlinear Control," *Asian Journal of Control*, Vol. 1, No. 3, Oct. 2008, pp. 128–145.

[3] Wenzel, K. E., Masselli, A., and Zell, A., "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle," *Journal of Intelligent & Robotic Systems*, Vol. 61, No. 1-4, Oct. 2010, pp. 221–238.

[4] Arora, S., Jain, S., Scherer, S., Nuske, S., Chamberlain, L., and Singh, S., "Infrastructure-free Shipdeck Tracking for Autonomous Landing," *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 323–330.

[5] Amazon.com, "Amazon Prime Air," `http://www.amazon.com/b?node=8037720011`, 2014.

[6] Charnley, W. J., "Blind Landing," *Journal of Navigation*, Vol. 12, 1959, pp. 115–140.

[7] Bosch, S., Lacroix, S., and Caballero, F., "Autonomous Detection of Safe Landing Areas for an UAV from Monocular Images," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 5522–5527.

[8] Hintze, J., Christian, D., Theodore, C., Tischler, M., McLain, T., and Montgomery, J., "Automated Landing of a Rotorcraft UAV in a Non-cooperative Environment," *Proceedings of the AHS International 60th Annual Forum*, June 2004, pp. 1448–1458.

[9] Theodore, C., Sheldon, S., Rowley, D., McLain, T., Dai, W., and Takahashi, M., "Full Mission Simulation of a Rotorcraft Unmanned Aerial Vehicle for Landing in a Non-Cooperative Environment," *Proceedings of the 61st Annual Forum of the American Helicopter Society*, June 2005.

[10] Theodore, C. and Tischler, M., "Precision Autonomous Landing Adaptive Control Experiment (PALACE)," *25th Army Science Conference*, Orlando, 2006.

[11] Hubbard, D., Morse, B., Theodore, C., Tischler, M., and McLain, T., "Performance Evaluation of Vision-Based Navigation and Landing on a Rotorcraft Unmanned Aerial Vehicle," *IEEE Workshop on Applications of Computer Vision*, February 2007.

[12] Johnson, A., Montgomery, J., and Matthies, L., "Vision Guided Landing of an Autonomous Helicopter in Hazardous Terrain," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Vol. 19, 2005, pp. 371–381.

[13] Templeton, T., Shim, D., Geyer, C., and Sastry, S., "Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft," *IEEE International Conference on Robotics and Automation*, No. April, 2007, pp. 1349–1356.

[14] Chamberlain, L., Scherer, S., and Singh, S., "Self-aware helicopters: Full-scale Automated Landing and Obstacle Avoidance in Unmapped Environments," *Proceedings of AHS Forum*, Vol. 15213, 2011.

[15] Johnson, A., Klumpp, A., Collier, J., and Wolf, A., "Lidar-based Hazard Avoidance for Safe Landing on Mars," *AAS/AIAA Space Flight Mechanics Meeting*, Jet Propulsion Laboratory, California Institute of Technology, 2001, pp. 1091–1099.

[16] Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Martínez-Sánchez, J., and Arias, P., "Metrological evaluation of Microsoft Kinect and Asus Xtion sensors," *Measurement*, Vol. 46, No. 6, July 2013, pp. 1800–1806.

[17] RIEGL Laser Measurement Systems, "RIEGL VUX-1," 2014.

[18] Scherer, S., Chamberlain, L., and Singh, S., "Autonomous landing at unprepared sites by a full-scale helicopter," *Robotics and Autonomous Systems*, Vol. 60, No. 12, Dec. 2012, pp. 1545–1562.

[19] Scherer, S., Chamberlain, L., and Singh, S., "First results in Autonomous Landing and Obstacle Avoidance by a Full-Scale Helicopter," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 951–956.