AIAA Infotech@Aerospace 20-22 April 2010, Altanta, Georgia

Online Assessment of Landing Sites

Sebastian Scherer* and Lyle Chamberlain† and Sanjiv Singh‡

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Assessing a landing zone (LZ) reliably is essential for safe operation of vertical takeoff and landing (VTOL) aerial vehicles that land at unimproved locations. Currently an operator has to rely on visual assessment to make an approach decision; however. visual information from afar is insufficient to judge slope and detect small obstacles. Prior work has modeled LZ quality based on plane fitting, which only partly represents the interaction between vehicle and ground.

Our approach consists of a coarse evaluation based on slope and roughness criteria, a fine evaluation for skid contact, and body clearance of a location. We investigated whether the evaluation is correct for using terrain maps collected from a helicopter. This paper defines the problem of evaluation, describes our incremental real-time algorithm, and discusses the effectiveness of our approach.

In results from urban and natural environments, we were able to successfully classify LZs from point cloud maps collected on a helicopter. The presented method enables detailed assessment of LZs without an landing approach, thereby improving safety. Still, the method assumes low-noise point cloud data. We intend to increase robustness to outliers while still detecting small obstacles in future work.

I. Introduction

Landing safely is an essential capability for all aerial vehicles; however, the majority of landings on unimproved terrain are performed by helicopters and other vertical takeoff and landing (VTOL) vehicles, increasing the risk of accidents. The situational awareness for teleoperation or onboard pilots is limited, so it is difficult to see small obstacles. A pilot can have difficulty judging the slope of terrain and identifying obstacles at the landing site from a distance. Helicopters tend to tip over easily since a large amount of mass is at the top. Even very small obstacles such as railroad tracks can make an otherwise good landing site unsuitable for landing.

A NASA study by Harris et al. analyzed rotorcraft accidents from 1963-97 and found that 36.19% of them were related to collision with objects, roll over, and hard landings.[1] A partial cause for accidents is a lack of situational awareness of the pilot regarding the suitability of landing sites. Reliable detection of landing sites will increase the safety of operation by presenting vital information to the pilot before a landing approach.

Our algorithm evaluates the terrain using a combination of in a coarse and then a fine evaluation. The coarse evaluation find candidate areas applying a fast plane fit to measure the slope, roughness, and other statistics. The fine evaluation then evaluates promising areas by fitting a 3D model of the helicopter and landing gear to a triangulated surface of the environment.

The main contributions of this paper are

- an analysis of the problem of landing site evaluation.
- an incremental model-based method to calculate evaluate landing sites for VTOL vehicles.
- results based on ladar sensor data that show the landing sites found in real environments.

*Graduate student, Field Robotics Center, basti@cmu.edu

†Research Engineer, Field Robotics Center, lylesjazz@gmail.com

‡Research Professor, Field Robotics Center, ssingh@cmu.edu

This paper first reviews related work, details our approach, and then presents results for the landing site evaluation algorithm.

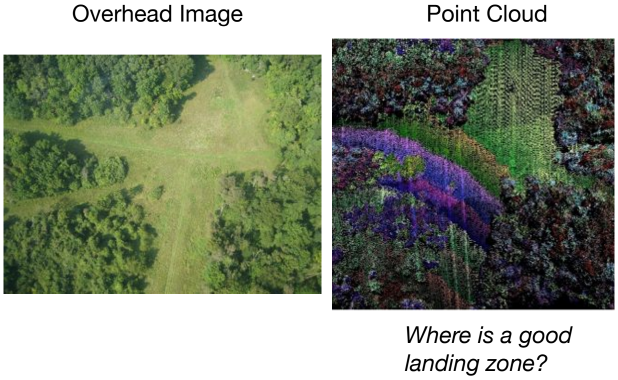


Figure 1: The problem addressed in this paper is to find good landing zones for rotary wing vehicles. On the left is an overhead image for reference. The input to the algorithm is an incrementally updated point cloud that is evaluated to find suitable landing zones.

II. Related Work

There has been some prior work on landing and landing site selection. From a control perspective the problem has been studied by Sprinkle[2] to determine if a trajectory is still feasible to land. Saripalli et al.[3,4] have landed on a moving target that was known to be a good landing site. Barber et al.[5] used optical flow based control to control a fixed wing vehicle to land.

Vision has been a more popular sensor modality for landing site evaluation because the sensor is lightweight. De Wagter and Mulder[6] describe a system that uses vision for control, terrain reconstruction, and tracking. A camera is used to estimate the height above ground for landing in Yu et al.,[7] and similarly in Meijas et al.[8] a single camera is used to detect and avoid powerlines during landing.

Using a stereo camera, Hintze[9] developed an algorithm to land in unknown environments. Bosch et al. propose an algorithm for monocular images that is based on detecting nonplanar regions to distinguish landing sites from non-landing sites.[10] Another popular approach is to use structure from motion to reconstruct the motion of the camera and to track sparse points that allow reconstruction of the environment and plane fitting. This method as been applied to spacecraft by Johnson et al.[11] and to rotor craft by Templeton et al.[12]

Ladar based landing site evaluation has not been studied very deeply except as suggested in Serrano et al.[13] where a framework for multiple sources like ladar, radar, and cameras is proposed to infer the suitability of a landing site. For ground vehicles ladar based classification is popular because of its reliability and direct range measurement. It has been used in conjunction with aerial vehicle data for ground vehicles by Sofman et al.[14] and by Hebert & Vandapel.[15]

Our work in landing site evaluation uses ladar data and some of the proposed techniques in related work such as plane fitting and roughness. However, our method goes beyond pure plane fitting to actually fit a model of the helicopter geometry to a triangulation of the environment.

III. Problem

The suitability of a landing zone depends to a large degree on the characteristics of the aircraft that will be landing on it. We consider larger vehicles such as manned helicopters, however the problem scales to smaller aircraft. Landing of a helicopter can roughly be separated into two independent phases: an approach, and a ground contact. During the approach, the helicopter ideally keeps a steady and relatively shallow descent angle in a straight line to come to a hover at some altitude above the ground. Then it orients with respect to the ideal location on the ground and touches down.

Assessment of a LZ needs to include two main factors: the ground conditions and the approach conditions. The ground conditions are all factors that are relevant when the aircraft is contact with the ground, while the approach conditions are related to moving to the LZ.

The ground conditions that need to be considered for assessing an LZ are

- minimum size of the site
- skid contact
- static stability on ground based on the center of gravity of the aircraft
- load bearing capability of the contact surface
- loose foreign objects (FOD) and vegetation surrounding the site
- clearance of aircraft/rotor with surrounding terrain and objects

To ensure a reliable and successful landing it is necessary to consider all these factors in a landing site evaluation algorithm. However some factors such as the load bearing capability, and foreign objects are difficult to classify purely based on the geometric approach that we use. In our approach we assume that a geometrically good site will be able to hold the load of the helicopter and FOD objects will be classified as sufficient roughness to cause a rejection. Future work will also incorporate approach and abort paths for flying a landing pattern.

IV. Approach

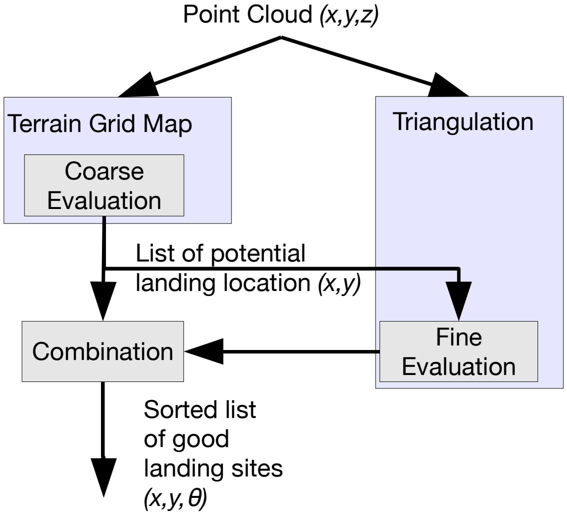


Figure 2: The steps in our landing site evaluation algorithm. The input is a list of globally registered points and the output is a sorted list of good landing sites that include location as well as landing heading. Two representations of the environment are created. One representation is based on a terrain grid map and the other is based on a triangulation of the environment and incorporates the constraints of the helicopter.

The input to our algorithm is a set of 3D points $\{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^3$ that are registered in a global coordinate system. These points could be generated by a ladar range scanner and registered using an inertial navigation system with GPS, for example. Figure 2 shows the data flow in our approach where the raw measurements are then processed to create a regular grid for the coarse evaluation of potential landing areas, and a separate triangulation used for the fine evaluation. Any updated grid cell is reevaluated to reflect changes in the available landing sites. A list of the available landing sites can be displayed to an operator or can be transmitted to the guidance system of an autonomous vehicle. A touchdown point is defined by a 3D position and orientation.

A. Coarse Evaluation

The first step in our algorithm is a coarse evaluation of a regular terrain grid map. Each cell in the map, roughly the size of the landing gear, tests how the vehicle would come into contact with the ground. The map keeps a set of statistics in a two dimensional hash table of cells. The hash key is calculated based on the point coordinate at a fixed resolution. This provides a naturally scrolling implementation and cells can be set as data are measured. As the vehicles moves and the map scrolls, old cells will be overwritten. This data structure allows the algorithm to quickly add new data to the map. Each cell in the grid keeps a set of statistics: The mean, minimum, and maximum z-value, the number of points, and the moments necessary to calculate the plane fit.

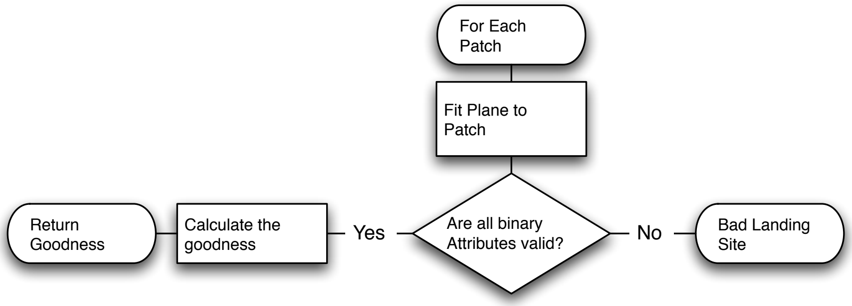


Figure 3: A flow chart showing the control flow of evaluating a patch in the terrain grid map. Only if all binary attributes shown in table 1 are valid a landing zone will be considered.

#	Description	Symbol	Operator	Value
1	Do we have more than x points for a plane fit?	c	$>$	15
2	Is the spread of the points less than x cm?	σ_n	$<$	50
3	Was the plane fit successful?		$=$	True
4	Is the residual for the plane fit less than x?	r	$<$	4
5	Is the slope of the plane less than x degrees?	α	$<$	5

Table 1: The binary attributes considered in the rough evaluation for landing site evaluation. Only if all operations evaluate to true a landing site can be available.

The statistics accumulated in each terrain cell are evaluated to assess a particular location. Based on these tests a series of binary attributes and goodness values can be computed. The statistics can be accumulated with neighboring cells which allows us to compute the suitability of larger regions. The binary tests are designed to reject places where the ground is too rough or too step to be valid; furthermore, the algorithm cannot make a reliable estimate of a landing site, if not enough data is available.

The binary attributes are computed as follows: First, we check if enough points are available to perform a plane fit. We require at least 15 points per cell, which is an average density of one point every 20 cm for a 3 m cell. This average density is the required density for the smallest obstacle we consider; however, we make an optimistic assumption since points are not necessarily distributed evenly. Next, we consider the standard deviation of the z-value. The algorithm rejects cells if the vertical standard deviation is larger than 50 cm, rejecting any area with large altitude changes. The value has to be large enough to avoid interfering with the results of the plane fitting since a large change in altitude also characterizes a large slope. Finally, the slope and residual of a plane fit are checked for a planar surface. The residual will be a good indicator for the roughness of the terrain since it measures deviation from the plane. Currently the slope threshold is set at 5 degrees and the residual threshold is set at 4.

The mean altitude and standard deviation of the plane is calculated based on Welford's algorithm by keeping the mean and the corrected sum of squares.[16] After receiving a new altitude measurement in a patch the mean altitude μ_c and sum of squares S_c are updated:

$$\mu_{new} \quad = \quad \mu_{c-1} + (z - \mu_{c-1})/n \tag{1}$$

$$S_c = S_{c-1} + (z - \mu_{c-1}) \cdot (z - \mu_{new}) \tag{2}$$
$$\mu_c = \mu_{new}$$

The standard deviation σ can be calculated from S as follows:

$$\sigma_c = \sqrt{\frac{S_c}{c}} \tag{3}$$

Additionally we want to evaluate the slope and need to fit a plane with respect to the measurements. A fast algorithm to calculate a least squares plane fit is a moment based calculation. The fit is a least squares fit along z and therefore assumes no errors in x and y exist (Also see Templeton et al.[17]). We only have to keep track of the moments

$$M = \begin{pmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{xy} & M_{yy} & M_{yz} \\ M_{xz} & M_{yz} & M_{zz} \end{pmatrix} \tag{4}$$

and the offset vector

$$M_o = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} \tag{5}$$

and the number of total points c. The moments can be updated for a new measurement (x, y, z) by updating the moments starting from 0 for all initial values.

$$M_x = M_x + x,\; M_y = M_y + y,\; M_z = M_z + z \tag{6}$$

$$M_{xx} = M_{xx} + x^2,\; M_{yy} = M_{yy} + y^2,\; M_{zz} = M_{zz} + z^2 \tag{7}$$
$$M_{xy} = M_{xy} + x \cdot y,\; M_{xz} = M_{xz} + x \cdot z,\; M_{yz} = M_{yz} + y \cdot z \tag{8}$$

$$c = c + 1 \tag{9}$$

Consequently, an update is the constant time required to update the 10 values and the plane fit is calculated in closed form. Another advantage is that we can accumulate the moments from the neighboring cells to fit a larger region in the terrain that represents the whole area of the helicopter, not just the landing skids. See Ahn[18] for more details on moment based fitting. The plane normal $n = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$ and residual r are calculate as such:

$$M_n = (M_{xy}^2 c - 2 \cdot M_x M_{xy} M_y, +M_x^2 M_{yy} + (M_y^2 - cM_{yy})M_{xx} \tag{10}$$

$$A = (M_y^2 M_{xz} - (M_x M_{yz} + M_{xy} M_z) M_y + c(-M_{xz} M_{yy} + M_{xy} M_{yz}) + X_x M_{YY} M_z)/M_n \tag{11}$$
$$B = (-(M_{xy} M_z + M_{xz} M_y) M_x + c M_{xy} M_{xz} + M_x^2 M_{yz} + M_{xx}(-c M_{yz} + M_y M_z)/M_n \tag{12}$$

$$C = (-(M_x M_{yz} + M_{xz} M_y) M_{xy} + M_x M_{xz} M_{yy} + M_{xy}^2 M_z + M_{xx}(M_Y M_{yz} - M_{yy} M_z))/M_n \tag{13}$$

$$r = \sqrt{\left| C^2 c + 2ACM_x + A^2 M_{xx} + 2BCM_x + 2ABM_{xy} + B^2 M_{YY} - 2CM_z - 2AM_{xz} - 2BM_{yz} + M_{zz} \right| / c} \tag{14}$$

We can infer the slope and roughness based on the normal and residual. The slope can be calculated as follows

$$\alpha = acos(v_{up}^T n) \tag{15}$$

where $v_{up} = (0,\, 0,\, -1)$ is the up vector and n is the plane normal.

If all the binary attributes shown in table 1 pass, the algorithm calculates the goodness attributes used in the final evaluation. All potentially good landing sites are now evaluated with the fine evaluation to decide if touching down is feasible.

B. Fine Evaluation

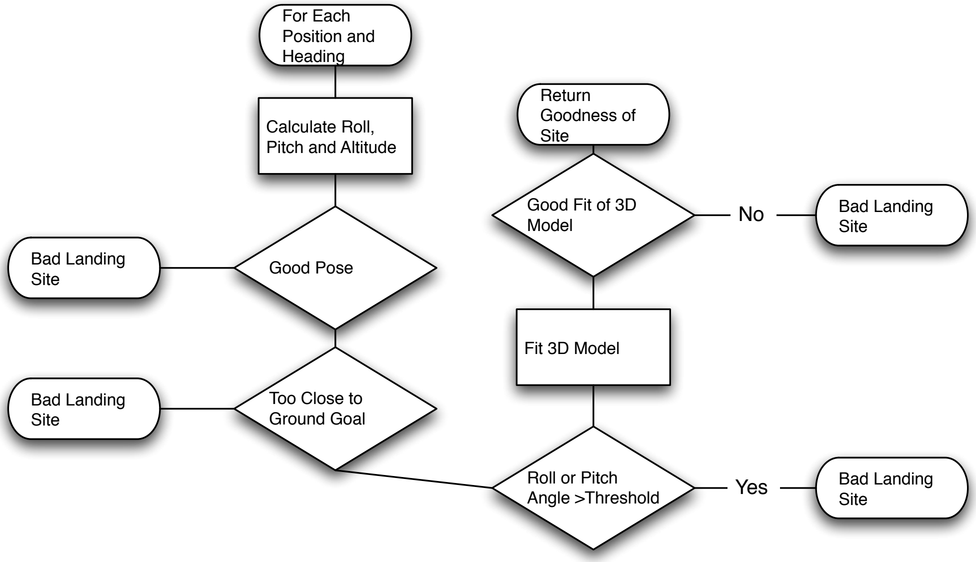


Figure 4: The flow chart of the fine evaluation algorithm. The pose of the helicopter is calculated based on a (x, y, θ) position and heading. If the pose is calculated successfully a surface model of the aerial vehicle is evaluated.

Once we have determined a list of general locations, we finely evaluate each potentially good site with respect to a 2D Delaunay triangulation of all available data. A Delaunay triangulation is a triangular network of points that can be constructed efficiently (See Preparate & Shamos for more details[19]). The reason for this staged processing is that it is easier to know when to incrementally update a cell-based representation, and the incremental calculation for the rough grid is much faster than for the triangulation.

While the rough evaluation considers the plane geometry and roughness of a landing site, the triangulation actually tries to simulate the helicopter coming in contact with the surface. As a first step we incrementally update a hierarchical triangulation with a maximum number of points per patch area. For an implementation see Fabri et al.[20] If the point density is exceeded, we randomly remove points from the triangulation to keep below a maximum amount of memory used. Currently, the patch size is set to 3 meters and the maximum point density is set to 200 points/cell, which corresponds to an even point spacing of approximately 20cm.

Once the coarse evaluation calculates a list of potential (x, y) locations, the fine evaluation tries to find the best orientation and determines if the chosen location is feasible. We test a series of possible orientations θ of the helicopter (8 in our experiments) by intersecting the virtual landing gear with the measured terrain to determine what the resulting pose of the helicopter would be (Also see figure 4). Currently our algorithm only works for two landing skids, which is a very common configuration of landing gear.

First we determine the intersections of the skids with the triangulation by walking along the two lines representing the landing skids, and extract the height above ground as shown in figure 5. Now we assume that the helicopter's center of gravity is on top of the landing skids and calculate the two highest points of the intersection. These two points will be the points that determine the position of the skid in the static case. We repeat this process for all eight potential landing gear positions.

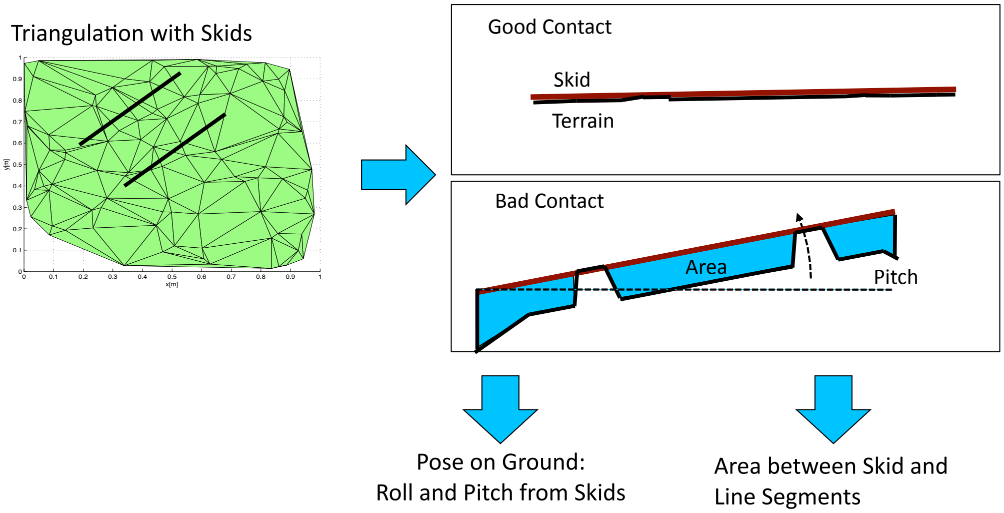


Figure 5: Illustration of skid - triangulation contact. The skids in contact with the triangulation determine the roll and pitch of the vehicle. A good contact is characterized by a small pitch angle and small area between the skid and triangulation.

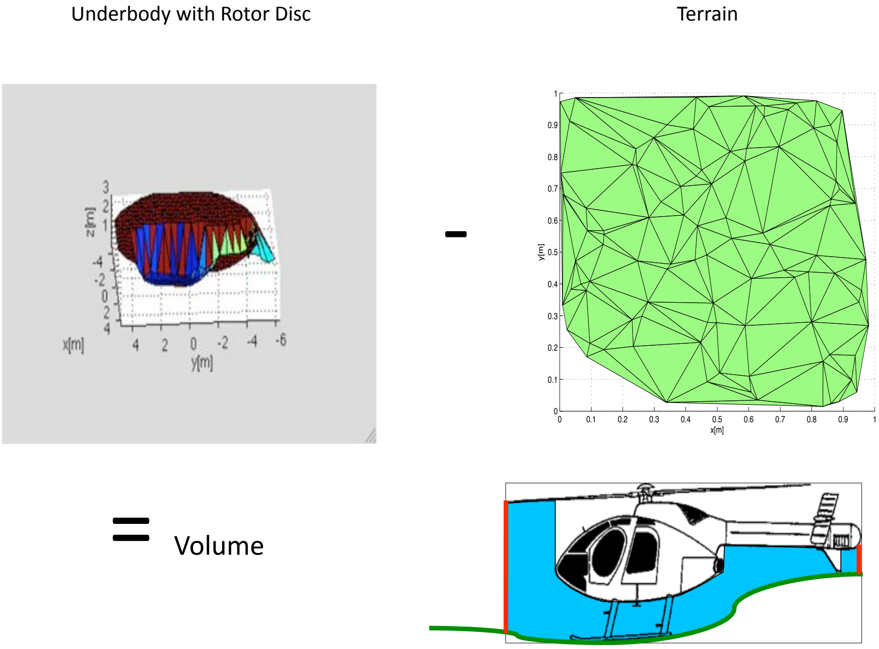


Figure 6: Underbody volume between model and triangulation. The underbody model is positioned with the pose calculated in figure 5. The volume between the two is evaluated to check for collision and calculate a goodness measure of a landing site.

The algorithm to position the landing gear is as follows: First we determine all intersection points

$$p_i^l = (x_i^l, y_i^l, z_i^l) \tag{16}$$

$$p_i^l \in \mathbb{R}^3 where\, i \in \{1, \ldots, n\} \tag{17}$$

between the landing gear l and the triangulation and partition the points into two sets: Points aft a_i^l and in front f_i^l of a distance along the landing gear. Currently we set the distance to be half the length of the landing gear because we assume that the center of mass is centered on top of the landing gear; however. this position can be changed depending on the mass distribution.

Next we look at the z-values and find the maximum z-value in the first and in the second half of the landing skid for each side.

$$p_{aft}^l = argmax(a_i^l) \tag{18}$$

$$p_{front}^l = argmax(f_i^l) \tag{19}$$

A minimum distance between p_{aft} and p_{front} is set to ensure that the slope will fit to a stable value. Now that we know how the landing gear will be positioned, we calculate the area between the line of the skid and the triangulation to get a measure of contact area. For example, if we were to land centered on railroad tracks at an 90 degree angle our skids would rest on the rails because they are the highest points above the ground. In this case even though the roll, pitch, and roughness will be small (rails are very small deviations from the ground and usually level) the integral of the area under the landing skids will have a large value because most of the landing skid is elevated above the ground. For skids in full contact with the ground the integral would be zero.

The next step is to determine the roll and pitch of the helicopter when it is in contact with the ground. We use the center of the two landing skid lines to determine the roll angle. Finally we use the lower landing gear line to calculate pitch because the center of gravity will shift towards that line. Using this algorithm we are now able to predict the position and orientation of the helicopter when it is in contact with the ground. The roll angle α is calculated from the interpolated centers p_c^l of the skids

$$p_c^l = \frac{p_n^l - p_1^l}{2} \tag{20}$$

$$\alpha = \arctan 2(z_c^2 - z_c^1, \sqrt{(x_c^2 - x_c^1)^2 + (y_c^2 - y_c^1)^2}) \tag{21}$$

The pitch of the helicopter depends on the roll angle since more weight will shift towards the lower skid. The pitch angle β is calculated as

$$\lambda \quad = \quad \begin{cases} 1 & if\, \theta_{roll} > 0 \\ 2 & otherwise \end{cases} \tag{22}$$

$$\beta = \arctan 2(z_{front}^\lambda - z_{aft}^\lambda, \sqrt{(x_{front}^\lambda - x_{aft}^\lambda)^2 + (y_{front}^\lambda - y_{aft}^\lambda)^2}) \tag{23}$$

With the orientation and position, we now calculate the volume between the undersurface of a 3D model of the helicopter with the triangulation of the terrain as shown in figure 6. This volume allows us to predict if a tail strike or similar bad contact with the ground would occur given the predicted position and orientation of the helicopter. It also gives us a measure of goodness that will maximize the volume under the helicopter. The calculation of the volume is simplified by taking a regular sample of points on the surface of the helicopter at an interval e. Even though some errors can occur we can overapproximate this regular sampling when we create the model. Given the sampled surface as a set of points on the underbody of the helicopter p_s, we first have to transform the points given the pose of the helicopter p_c^h and the orientation (α, β, θ). The transformed points $p_s^{'}$ are then compared to the intersected point p_t of the triangulation. The volume is defined as

$$v_j = \begin{cases} e^2(z_s^{'} - z_t) & if\, z_s^{'} - z_t > 0 \\ -\infty & otherwise \end{cases} \tag{24}$$

$$V_{underbody} = \sum_{j=0}^{m} v_j \tag{25}$$

If the underbody surface intersects with the triangulation, the volume becomes $-\infty$ because we have a collision with the terrain.

C. Goodness Assessment

Description	Function	Weight
Goodness from Rough Evaluation	$w_s\alpha + w_r r$	ω_1
Area under the landing gears	$\frac{1}{A_{gear}}$	ω_2
Volume under the Helicopter	V_{heli}	ω_3
Distance from the closest bad landing site	d_l	ω_4
Distance to the ground goal point	d_g	ω_5

Table 2: A linear combination of factors determines the goodness of a landing site. The weights needs to be set by the user to determine which of the acceptable sites is the best.

Landing site selection is a problem where many factors need to be considered to evaluate the cost of a site. Depending on the weights for the goodness calculation, different sites will be chosen as best sites. A linear combination of the factors described in table 2 is considered to calculate the goodness of a landing site. Currently we manually set the weights $\omega_1, \ldots, \omega_5$ to determine a goodness.

Since some of the goodness values are expensive to compute the map is only updated when necessary. We therefore keep track of the changes to rough terrain cells and only process areas with significant changes. Because the triangulation intersections are expensive to compute, we also cache the results of the fine evaluations.

V. Experiments

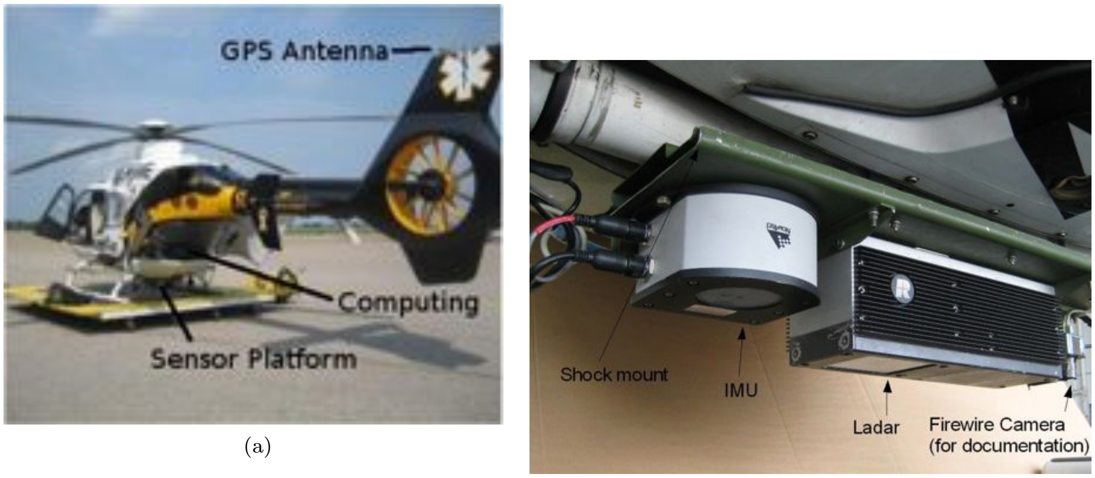
In this section we show results for the coarse and fine evaluation based on 3D pointcloud datasets we collected. In these results we assume a straight line path from the landing site to a ground goal location. The results are shown for natural and man-made environments.

A. Setup

For our first experiment we collected point cloud datasets at various locations around Pittsburgh, PA, USA. We overflew the areas in approximately straight lines with speeds between 10-20 knots. On the bottom of the helicopter we mounted a laser scanner, inertial navigation system, and a documentation camera (See figure 7). The Riegl LMS-Q140i-60 scans a line perpendicular to the flight direction at 8000 points per second. The scan frequency of the ladar is 60 Hz. The data are fused by a Novatel SPAN INS/GPS to calculate a globally registered point cloud. The inertial measurement unit has a drift rate of 10 degrees/hour and the GPS is corrected with a basestation to a reported 1 cm accuracy. In total we collected data on 9 sites with a flight time of 8 hours. We discuss results from three representative sites.

B. Results

The first site was scanned at an altitude of about 100 meters (figure 8). It is a field, and our ground goal point is located next to the slope on the hill (see figure 8d for exact goal location). The result shows the



(a)

(b)

Figure 7: The experimental setup for data collection for our experiments in Pittsburgh, PA, USA. We mounted a ladar, and inertial navigation system on a helicopter as shown in figure (a). The sensor platform is shown in figure (b). The inertial measurement unit was located next to the ladar on a shock mount to minimize errors in the registration of the data due to vibration.

coarse and fine evaluation of landing sites in the environment. Several possibly good landing sites in the coarse evaluation are rejected in the fine evaluation phase. Finally, the best landing site is picked in a spot with small slope and a large clearance from obstacles.

While the previous dataset was mostly in vegetated terrain we also collected a dataset at a coal power plant (figure 9) at 200m elevation above ground. The geometry of obstacles at the power plant is very different from vegetation, but the algorithm is still able to determine good landing sites. The ground goal location is close to a car, and one can see how the fine evaluation reject the car and the red region around the ground goal. The red region is a safety buffer that prevents landing directly on top of the ground goal.

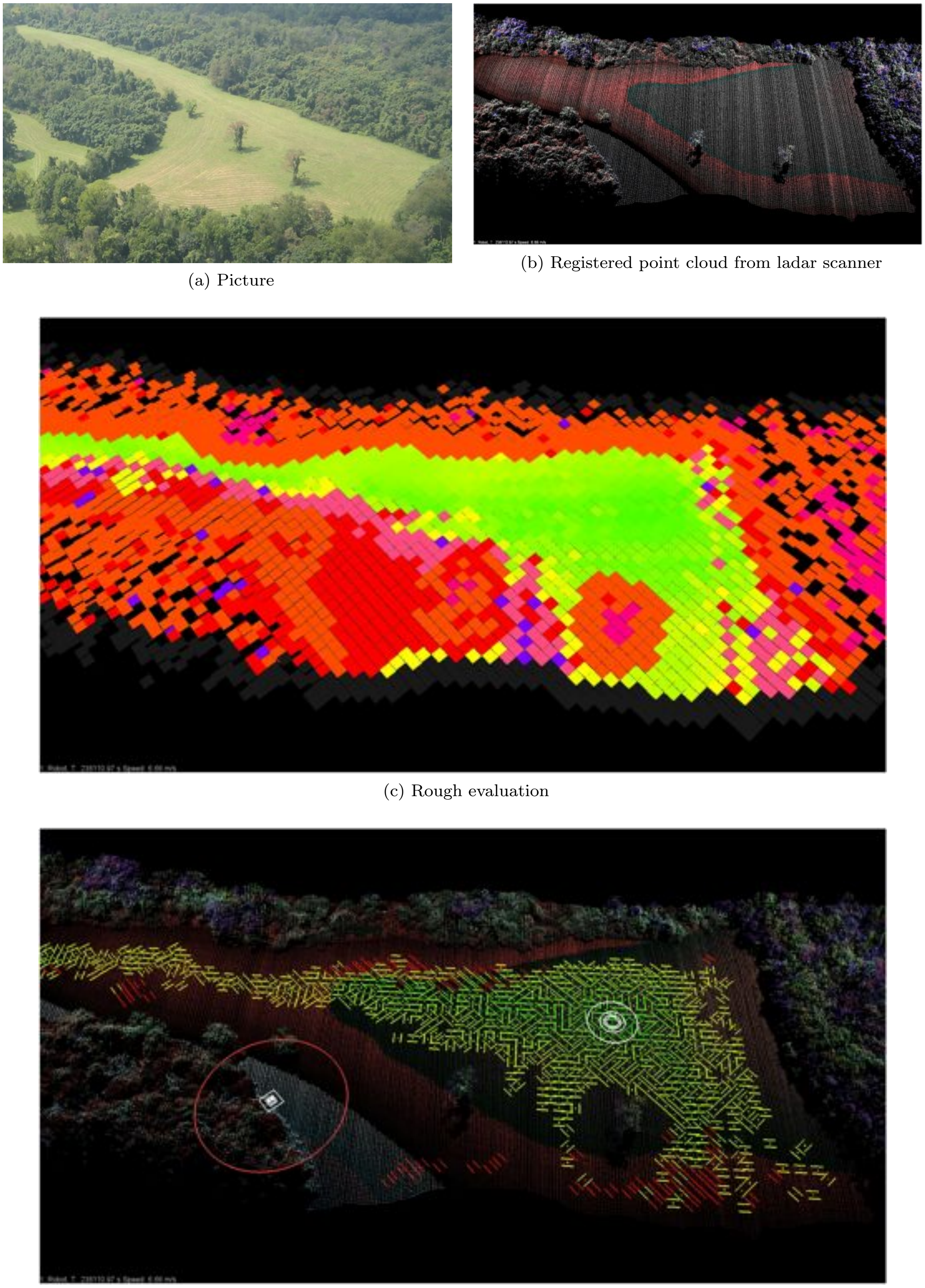
As a final experiment, we varied the weights for our evaluation function for four different parameters in the same terrain (figure 10). First we changed the weight for slope and roughness (figure 10a). Compared to the nominal evaluation, one can see that we now choose a landing site further away from the ground goal point because it is in smoother and less sloped terrain. A similar location is chosen in figure 10b; however, since we try to emphasize the volume of the underbody, the shape of the LZ is actually more convex than the evaluation in Fig 10a.

If we give a large weight to the distances to the casualty, we expect the distance to be small. We can see in the evaluation for figure 10c that far points have low weight and close points have the lowest weight. In figure 10d, the result is almost the same as figure 10c, except that now points close to the boundary also have a low value. Varying the weights, one notices that the weight vector has a large influence on the resulting chosen landing site.

	Time(ms)	Units
Adding Points	16	/100 points
Coarse Evaluation	49	/1000 cells
Fine Evaluation	26	/cell

Table 3: Computation time for the three parts of the algorithm.

Table 3 shows the computation time required for the different steps in the algorithm on a 2.5 GHz Intel Core 2 Duo. Adding new data is relatively fast and takes currently about 16 ms/100 points which is about realtime for the laser used in the experiments. The data input is designed in such a way that data is dropped if the processing cannot keep up with the ladar data. The coarse evaluation is very fast at $49ms/1000$ cells.



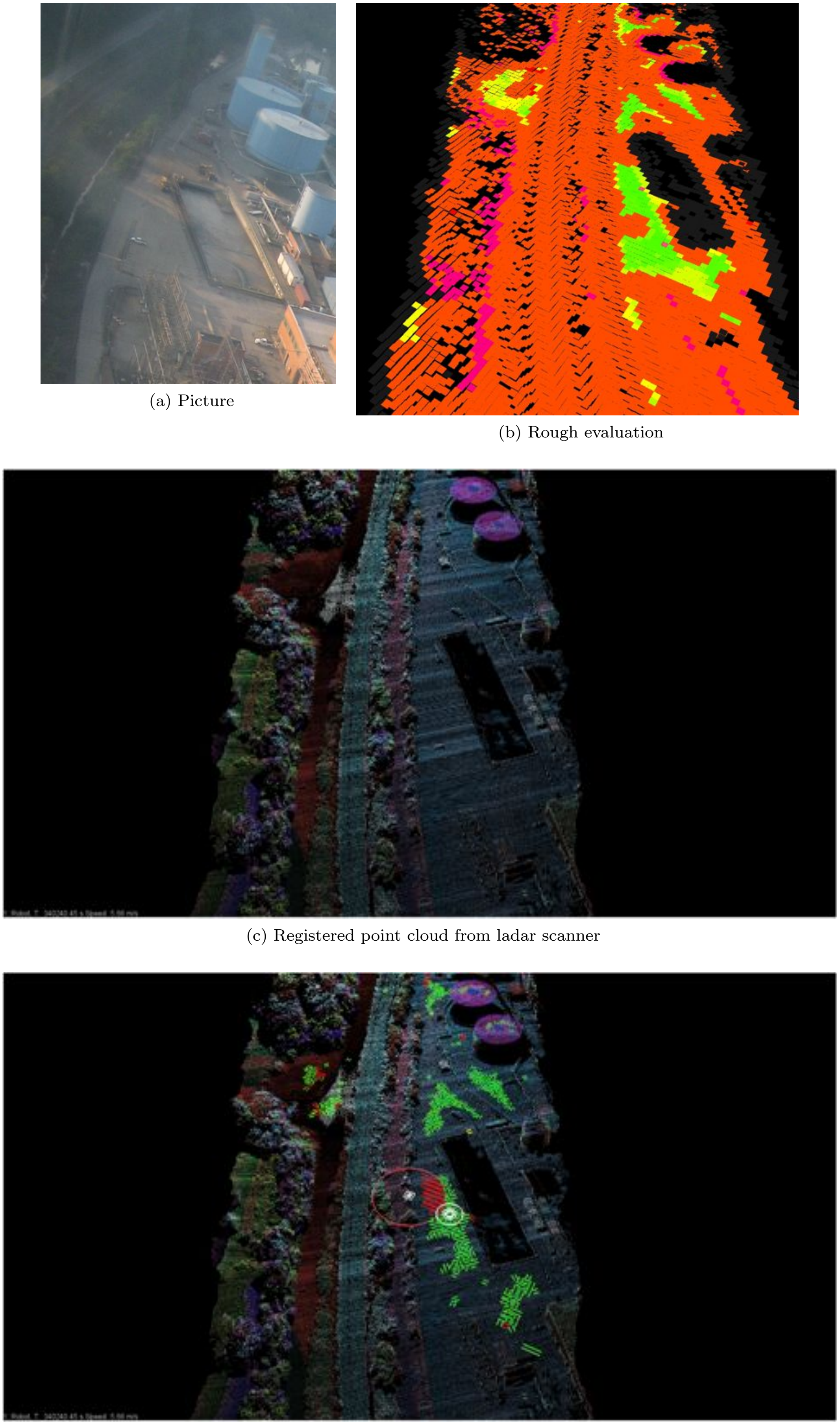
(a) Picture

(b) Registered point cloud from ladar scanner

(c) Rough evaluation

(d) Fine evaluation

Figure 8: Results from the “three trees” area. Figure (a) shows an aerial view of the scanned area and (b) shows the point cloud at the end of the scan. The legend for the coarse evaluation (c) is as follows: Red = slope & roughness exceeded, orange = roughness exceeded, pink = slope exceeded, light orange = large spread, purple = large residual, dark purple = bad fit. The legend for the fine evaluation (d) is as follows: Red = bad site, yellow to green = goodness of site, white circles = best site, red circle = no land zone, white box = goal location.



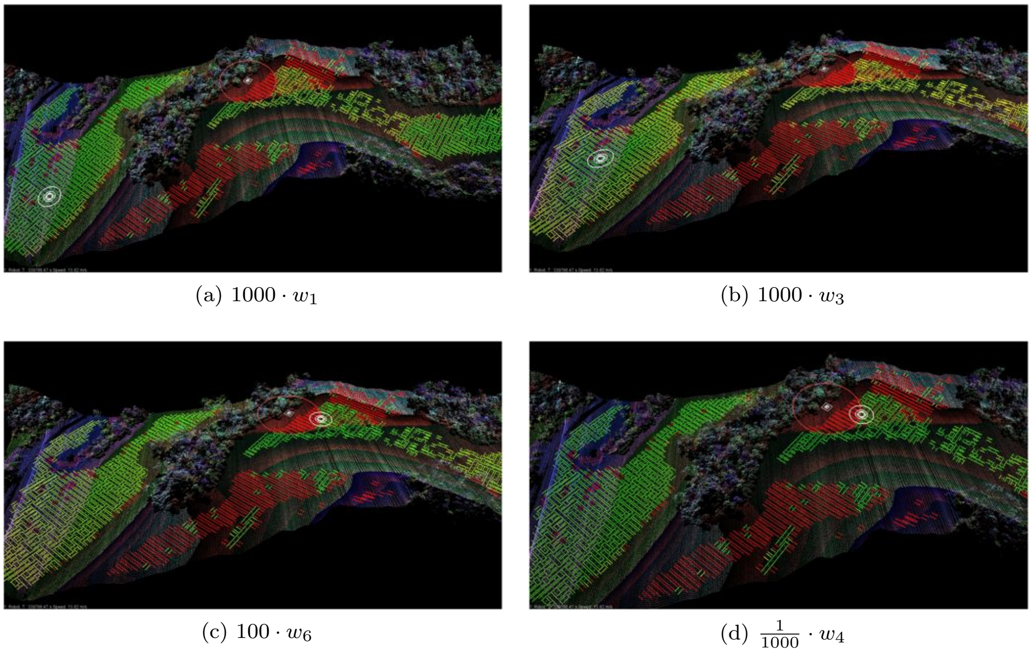
(a) Picture

(b) Rough evaluation

(c) Registered point cloud from ladar scanner

(d) Fine evaluation

Figure 9: Results from the “power plant” area. For a legend of the results see figure 8.



(a) $1000 \cdot w_1$

(b) $1000 \cdot w_3$

(c) $100 \cdot w_6$

(d) $\frac{1}{1000} \cdot w_4$

Figure 10: This figure shows the result of varying the weights for the final decision on selecting the best landing site. For a legend of the results see figure 8. In this plot we keep the function constant and vary the weight for some of the factors described in table 2. In (a) we increase the weight of $1000 \cdot w_1$ - the slope and roughness, for (b) we increase $1000 \cdot w_3$ to emphasize the clearance of the helicopter above the terrain, for (c) we try to minimize the distance to the ground goal by increasing $100 \cdot w_5$, and in (d) we decrease the weight of the distance to the closet obstacles by changing $\frac{1}{1000} \cdot w_4$. One can see that depending on the cost function the distribution of good sites changes, as well as the best picked landing site. The weighting does not influence the binary decision of acceptable sites because it is based on hard thresholds of the helicopters constraints.

This rate corresponds to about $184\frac{m^2}{ms}$ of area covered. The fine evaluation is much slower which justifies the coarse evaluation to limit the number of fine evaluations performed.

VI. Conclusion

We presented, to our knowledge, the first geometric model-based algorithm that gives estimates of available landing sites for VTOL aerial vehicles. In our algorithm we are able to incorporate many constraints that have to be fulfilled for a helicopter to be able to land. While the current state of the art is similar to our coarse evaluation algorithm, the fine evaluation allows us to consider more aerial-vehicle-relevant constraints and reject false positives in the coarse evaluation. Furthermore, since the two paths of evaluation are based on two different representations of the terrain, we conjecture that the combined system will be more robust to failure cases of a pure cell-based representation.

We also presented results based on real sensor data for landing site evaluation at vegetated and urban sites. All of the hard constraints on accepting or rejecting a landing site are based on the actual constraints of the vehicle geometry and weight distribution. However, which of the good landing sites is preferable is an open problem because combining the different metrics is non-trivial. In future work we propose to learn the weight vector of the landing site evaluation based on human preferences or self-supervised learning in simulation.

Even though the inertial navigation system we are using has a high accuracy (10 degree/hour drift IMU, 1cm GPS) it is not sufficient to register the points with high enough accuracy to evaluate landing sites in front of the helicopter at the glide slope, because the smallest potentially lethal obstacle we consider is the height of a rail (20cm). One could apply scan matching techniques such as the one proposed by Thrun et al.[21] to smooth the matching of adjacent scans, however, in the process one might smooth over important obstacles.

Currently our algorithm will accept water as a potentially good landing site since it is flat and smooth. We have observed two behaviors of water with our ladar scanner: In some water the beam is reflected and we get a smooth surface, and in other situations we do not get a reflection and therefore no measurement.

No measurement in this case is good because we will reject such areas. On the other hand, we might keep searching the water areas because we are not able to gather data on them. Some ways to directly sense water would be with other sensor modalities such as a camera of a suitable spectrum, a radar sensor, or a ladar that can give more than one target for a pulse.

While water is a false positive, we will also reject potentially good landing sites (false negatives) because we currently classify vegetation as roughness. We have no information that allows us to distinguish grass from steel wire, for instance, and therefore we take a pessimistic approach.

The only landing gear we model in the fine evaluation are two skids. In future work we would like to make the algorithm more flexible to be applicable to different kinds of landing gear.

Acknowledgments

The authors gratefully acknowledge Ben Grocholsky for executing the experiments and insightful comments.

References

[1] Harris, F., Kasper, E., and Iseler, L., "US Civil Rotorcraft Accidents, 1963 through 1997," *Technical Report (NASA/TM–2000-209597)*, Jan 2000.

[2] Sprinkle, J., Eklund, J., and Sastry, S., "Deciding to land a UAV safely in real time," *Proceedings of the American Control Conference*, May 2005, pp. 3506 – 3511 vol. 5.

[3] Saripalli, S. and Sukhatme, G., "Landing on a moving target using an autonomous helicopter," *Proceedings of the International Conference on Field and Service Robotics*, 2003.

[4] Saripalli, S. and Sukhatme, G., "Landing a Helicopter on a Moving Target," *Proceedings IEEE International Conference on Robotics and Automation*, Mar 2007, pp. 2030 – 2035.

[5] Barber, D., Griffiths, S., McLain, T., and Beard, R. W., "Autonomous Landing of Miniature Aerial Vehicles," *Proceedings of the AIAA Infotech@Aerospace Conference*, 2005.

[6] Wagter, C. D. and Mulder, J., "Towards Vision-Based UAV Situation Awareness," *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, 2005.

[7] Yu, Z., Nonami, K., Shin, J., and Celestino, D., "3D Vision Based Landing Control of a Small Scale Autonomous Helicopter," *International Journal of Advanced Robotic Systems*, Vol. 4, No. 1, 2007, pp. 51–56.

[8] Mejias, L., Campoy, P., Usher, K., Roberts, J., and Corke, P., "Two Seconds to Touchdown - Vision-Based Controlled Forced Landing," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 3527 – 3532.

[9] Hintze, J., "Autonomous Landing of a Rotary Unmanned Aerial Vehicle in a Non-Cooperative Environment using Machine Vision," *Master's Thesis (Brigham Young University)*, Jan 2004.

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

[11] Johnson, A., Montgomery, J., and Matthies, L., "Vision Guided Landing of an Autonomous Helicopter in Hazardous Terrain," *Proceedings IEEE International Conference on Robotics and Automation*, Mar 2005, pp. 3966 – 3971.

[12] Templeton, T., Shim, D., Geyer, C., and Sastry, S., "Autonomous Vision-based Landing and Terrain Mapping Using an MPC-controlled Unmanned Rotorcraft," *Proceedings IEEE International Conference on Robotics and Automation*, Mar 2007, pp. 1349 – 1356.

[13] Serrano, N., "A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 5112 – 5117.

[14] Sofman, B., Bagnell, J., Stentz, A., and Vandapel, N., "Terrain Classification from Aerial Data to Support Ground Vehicle Navigation," *Technical Report (CMU-RI-TR-05-39) Robotics Institute, Carnegie Mellon University*, 2006.

[15] Hebert, M. and Vandapel, N., "Terrain classification techniques from ladar data for autonomous navigation," *Proceedings of the Collaborative Technology Alliances Conference*, Jan 2003.

[16] Welford, B., "Note on a method for calculating corrected sums of squares and products," *Technometrics*, Vol. 4, No. 3, Jan 1962, pp. 419–420.

[17] Templeton, T., "Autonomous Vision-based Rotorcraft Landing and Accurate Aerial Terrain Mapping in an Unknown Environment," *Technical Report University of California at Berkeley (UCB/EECS-2007-18)*, 2007.

[18] Ahn, S. J., "Least squares orthogonal distance fitting of curves and surfaces in space," *Lecture Notes in Computer Science*, Jan 2004.

[19] Preparata, F. P. and Shamos, M. I., "Compuational geometry: an introduction," Jan 1985.

[20] Fabri, A., Giezeman, G., Kettner, L., and Schirra, S., "The CGAL kernel: A basis for geometric computation," *Lecture Notes in Computer Science*, Jan 1996.

[21] Thrun, S., Diel, M., and Hahnel, D., "Scan Alignment and 3-D Surface Modeling with a Helicopter Platform," *Proceedings of the International Conference on Field and Service Robotics*, Jan 2003.