Vision Guided Landing of an Autonomous Helicopter in Hazardous Terrain*

Andrew Johnson and Jim Montgomery

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
aej@jpl.nasa.gov, monty@helios.jpl.nasa.gov

***Abstract* – Future robotic space missions will employ a precision soft-landing capability that will enable exploration of previously inaccessible sites that have strong scientific significance. To enable this capability, a fully autonomous onboard system that identifies and avoids hazardous features such as steep slopes and large rocks is required. Such a system will also provide greater functionality in unstructured terrain to unmanned aerial vehicles. This paper describes an algorithm for landing hazard avoidance based on images from a single moving camera. The core of the algorithm is an efficient application of structure from motion to generate a dense elevation map of the landing area. Hazards are then detected in this map and a safe landing site is selected. The algorithm has been implemented on an autonomous helicopter testbed and demonstrated four times resulting in the first autonomous landing of an unmanned helicopter in unknown and hazardous terrain.**

Index Terms* – *autonomous landing, hazard detection, structure from motion, UAV.

I. INTRODUCTION

This work has been conducted in the context of providing autonomous image-based navigation algorithms to space science missions. Autonomous spacecraft systems have the potential to reduce costs while enhancing existing systems and enabling new capabilities for future deep space missions. In particular, landing on planets, moons, comets, and asteroids will benefit tremendously from on-board systems that autonomously and accurately determine spacecraft velocity and position relative to a landing site. In addition, autonomous detection of hazards during descent will enhance safety and enable missions to landing sites that are scientifically interesting but hazardous.

To date, no space science mission has employed hazard detection and avoidance during landing and this has had an impact on landing site selection. For example, the Mars Exploration Rovers mission selected Gusev Crater and Meridiani Planum for two reasons: they are flat plains that are relatively free of landing hazards and they are potentially scientifically interesting. Given a hazard avoidance capability, future missions will be able to pick landing sites with a greater emphasis on science return and less on engineering safety criteria.

Proposed sensors for hazard detection and avoidance are typically based on active range imaging. These active sensors are expensive, massive, power hungry, large and complicated. In contrast, cameras are cheap, small, low power and relatively simple. If an efficient and robust algorithm were developed for processing imagery to produce a terrain map then hazard detection using a single moving camera would be possible and the cost and accommodation savings to missions would be large.

This paper describes a novel algorithm for hazard detection and avoidance from imagery taken by a single moving camera. The specific novel components of the algorithm are as follows. Unlike in binocular stereo vision, this algorithm uses images from a single camera. Consequently, it must compute the motion between images and use this estimate when triangulating to establish the structure of the scene. Since the motion between images is generally unconstrained the algorithm uses 2D feature tracking (instead of searching along the scan line) to establish correspondences; this approach is more general than binocular stereo-vision. When compared to other structure from motion algorithms this algorithm is novel in that it generates a dense terrain map and does this in a computationally efficient and robust fashion. The final novel component of the algorithm is its use of an altimetry measurement to establish the overall scale of the scene.

Autonomous testbeds (e.g., rovers, aerobots, and helicopters) are commonly used by NASA to demonstrate technology on earth under mission relevant conditions. At the Jet Propulsion Laboratory an autonomous small-scale helicopter is used to demonstrate algorithms for planetary landing and small body exploration. Image-based hazard detection and avoidance has been implemented on the JPL Autonomous Helicopter Testbed which has resulted in the first autonomous landing of an unmanned helicopter in unknown and hazardous terrain.

A. Related Work

Vision-based control of autonomous aerial vehicles has been an area of active research for a number of years. In [8], image-based motion estimates are combined in an Extended Kalman filter along with IMU, GPS and sonar altimeter measurements to provide a navigation solution for an autonomous helicopter. Amidi et al. [1] present a visual odometer to estimate the position and velocity of a helicopter by visually locking on to and tracking ground features. Attitude information is provided by a set of gyroscopes while position and velocity is estimated based upon template matching from sequences of stereo vision data. [4][13][12] extend vision-based control to the autonomous landing problem. In [4], no autonomous landing is attempted, however a vision-based approach for safe-landing site detection in unknown, unstructured terrain is described. Both [13] and [12] describe a vision-based approach for locating a known target and then tracking it while navigating to and landing on the target. However, in

* The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract from the National Aeronautics and Space Administration.

these two approaches, the target area is known *a priori* to be flat and safe.

Recently there have been flight missions that use terrain imaging for spacecraft control. The Mars Exploration Rover Descent Image Motion Estimation System (MER-DIMES) used images to estimate velocity, but had no capability to generate terrain maps. MDRobotics/Optech have developed a scanning lidar system for the XSS-11 Mission that can generate terrain maps, but like all scanning lidar systems it consumes many more resources (~10Kg, 75W) than a camera-based system (<1kg, <5W for MER DIMES). The Near Earth Asteroid Rendezvous Mission used imagery to touchdown on the surface of Eros, but all operations were manual. MUSES-C will attempt to return a sample from an asteroid. The terminal control for this mission is performed by placing a known marker on the surface of the asteroid; no landing hazard detection is employed. The purpose of the Deep Impact mission is to impact a comet at high velocity with a penetrator spacecraft while another spacecraft images the impact site as it passes by. The targeting requires closed loop image-based control using autonomous centroiding, but no terrain reconstruction or hazard avoidance is needed.

II. TERRAIN MAP GENERATION

The inputs into the hazard detection and avoidance (HDA) algorithm are two overlapping images of the surface and a measurement of the distance between the camera and the surface along the camera optical axis (i.e., a slant range from a narrow beam altimeter). for the first image. The outputs from the algorithm are: the change in position and attitude between images, a dense terrain map of the imaged surface and a safe landing site on the surface. The algorithm has multiple stages. First a sparse set point features are selected and tracked between the images. These features are then used as inputs to a motion estimation routine that solves for the change in pose (attitude and position) of the camera between image acquisitions and the depth to each of the sparse features. Next, a dense grid of features are selected and tracked between the images using the motion and depth estimates to bound the search for feature tracks. Triangulation using these dense feature tracks results in a cloud of 3D points which are projected into a 2D grid to create a terrain map. Local operators are applied to the terrain map to estimate slope and roughness. A safe site for landing is then selected that is farthest from all slope and roughness hazards. The details of each stage of the algorithm, with an emphasis on computational efficiency, are described below. Run times and important parameters for each stage are described in TABLE I.

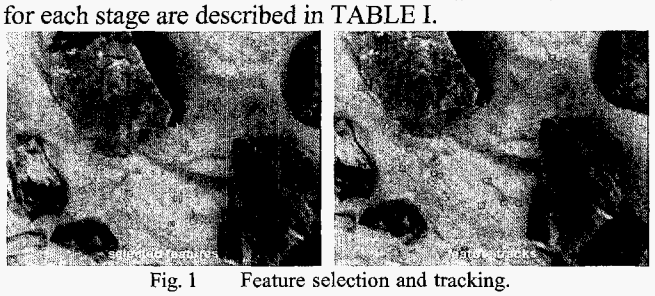


Fig. 1 Feature selection and tracking.

A. *Initial feature selection and tracking*

The first stage in the algorithm finds locations in the first image that will be good for tracking and then searches for their corresponding location in the second image using image correlation.

Feature selection is done using the efficient implementation of the Shi, Tomasi and Kanade feature detector described in [2]. First image gradients $I_r(r,c), I_c(r,c)$ are computed using finite differences over the entire first image. Next the autocorrelation matrix *A(r,c)* for a small window *T* around each pixel (hereafter called the template) is computed.

(1) $$A(r,c)=\begin{bmatrix} a & b \\ b & c \end{bmatrix}=\begin{bmatrix} \sum_T I_r^2(r,c) & \sum_T I_r(r,c)I_c(r,c) \\ \sum_T I_r(r,c)I_c(r,c) & \sum_T I_r^2(r,c) \end{bmatrix}$$

For efficiency, the elements of *A* are computed using a sliding sum; each time the template is shifted by a pixel, the gradients that leave the template are subtracted from the sum and the gradients that appear in the window are added. Pixels are better for tracking when *A* has two large eigenvalues. As described in [2] the check for large eigenvalues can be replaced by the check against a minimum allowable eigenvalue λ_t.

(2) $$P=(a-\lambda_t)(c-\lambda_t)-b^2>0$$
$$a>\lambda_t$$

Motion estimation is more likely to be well conditioned if the selected features are evenly spread over the image. To enforce an even distribution, the image is broken into blocks of pixels and the feature that meets the conditions in (2) and maximizes *P* over the block is selected as the best pixel in the block. As shown in Fig. 1, this approach, spreads the feature evenly across the image.

Once features are selected they are tracked into the second image using a 2D correlation-based feature tracker. No knowledge of the motion between frames is assumed, so the correlation window is typically square and large enough to handle all expected feature displacements. To increase efficiency a sliding sums implementation of pseudo-normalized correlation *C(r,c)* is used [6].

(3) $$C(r,c)=2\sum_T \tilde{I}_1(r,c)*\tilde{I}_2(r,c)/(\sum_T \tilde{I}_1^2(r,c)+\sum_T \tilde{I}_2^2(r,c))$$

where $\tilde{I}$ corresponds to the *I* with the mean subtracted

Correlation is applied in a coarse to fine fashion as follows. First, block averaging is used to construct an image pyramid for both images. The number of image pyramid levels n_l depends on the size *w* of the window *W* (hereafter called the window) over which the feature is correlated.

(4) $$n_l=\log_2(w)-2$$

The template half-width t_l and window half-width w_l at each level are scaled depending on the level in the pyramid according to the following rules.

(5) $$t_l=\max(2,t/2^l+0.5) \quad l\le n_l$$
$$w_l=\begin{cases} \max(2,w/2^l+0.5) & l=n_l \\ 2 & l<n_l \end{cases}$$

Feature tracking starts at the coarsest level of the pyramid with a template and a window size scaled to match the coarse resolution. The pixel of highest correlation is used to seed the correlation at the next finer level. As given in (5), after the coarse level, the template size increases as the pyramid level increases while window size is fixed. At the finest scale, the original image data is correlated, albeit with a small window size, and the feature track is accepted if the correlation value is higher than a threshold. Sub-pixel

tracking is obtained by fitting a biquadratic to the correlation peak and selecting the track location as the peak of the biquadratic.

The coarse to fine nature of this feature tracker makes it efficient even for large translations between images. However, since a 2D correlation is used to track features, it is susceptible to rotations between images and large changes in scale. In practice we have found it is possible to track features when the change in attitude between frames is less than 10° in roll about the optical axis, less than 20° in pitch and yaw and the change in altitude between images is less than a 20%.

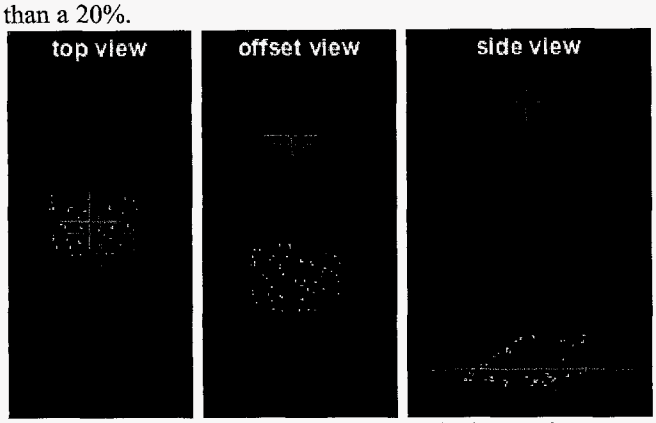


Fig. 2 Motion estimation and coarse depth estimation.

B. Structure from motion

The next stage in the algorithm, is a structure from motion estimation that uses feature tracks to solve for the change in position and attitude (e.g., the motion) of the camera between the images and the depth to the selected features in the first image (e.g., the structure). Structure from motion has been studied for decades, and there are numerous structure from motion algorithms in existence (see [8][9] for the state of the art).

This stage uses a previously reported [10] robust non-linear least squares optimization that minimize the distance between feature pixels by projecting the features from the first image into the second image based on the current estimate of the scene structure and camera motion. In this approach the motion between two camera views is described by a rigid transformation *(R, t)* where the rotation *R*, represented as a unit quaternion *q*, encodes the rotation between views and *t* encodes the translation between views. The altimetry measurement is used to set the initial depths to the features in the scene. This altimetry augmentation to our structure from motion algorithm eliminates the scene scale ambiguity present in structure from motion algorithms based solely on camera images. The output of this stage of the algorithm is the 6 DOF motion between images and the depth to the features selected in the first image. Fig. 2 shows three views of the computed motion and structure for the images shown in Fig. 1. The two positions of the camera are shown as red and green coordinate axes. The fields of view of the images are shown as red and green rectangles and the 3D position of the feature tracks are shown as white dots.

C. Dense structure recovery

The final stage of the algorithm uses the motion between images and the coarse structure provided by the depths to the feature tracks to efficiently generate a dense terrain map. Unlike in stereo vision where the images are separated by a know baseline aligned with the image rows, when using a single camera to recover scene structure, the motion between images is arbitrary. Consequently, standard scan-line rectification algorithms cannot be applied to make surface reconstruction efficient; other approaches need to be developed.

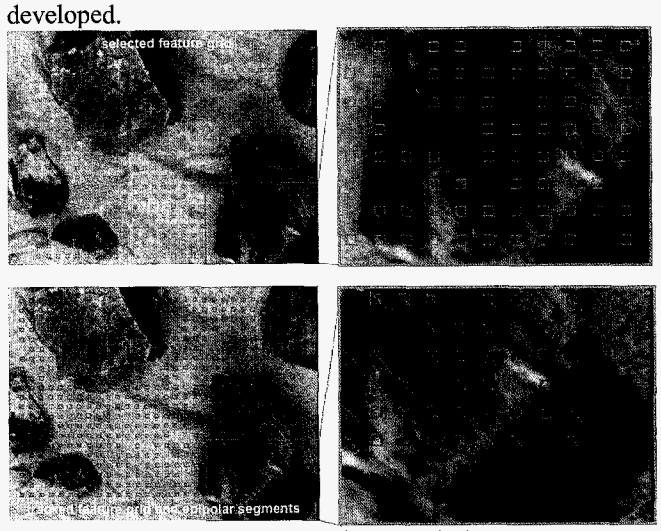


Fig. 3 Dense feature tracking on epipolar segments.

For a pinhole camera, the projection of a pixel in the first image must lie on a line in the second image that is determined by the motion between images (the epipolar line). The depth to the pixel determines the location of the pixel along the line. If the depth to the pixel is unknown, but bounded, then the pixel will lie along a segment of the line (an epipolar segment). By applying image correlation along this segment, the depth to the pixel can be determined exactly with minimal search. Using these observations an efficient algorithm for terrain map generation that can operate with images under arbitrary motion has been developed.

First the maximum and minimum scene depths are established. Because the features are spread over the entire image, the depth to features estimated in the structure from motion stage of the algorithm are used to indicate how much depth variability there is in the entire scene. However, there may be some parts of the scene closer or farther than the feature depths. To deal with this uncertainty, the range of allowable scene depths is increased by a small fraction from that estimated during structure from motion.

To generate a dense set of scene depths, a grid of pixels are selected in the first image. The spacing of the grid is an important parameter; a coarse grid may miss landing hazards while a fine grid will have an increased processing time. At the moment grid spacing is a user defined parameter, but it could be set automatically based on the size of the helicopter (or lander) and the pixel resolution.

Next, the epipolar segment is determined for each pixel in the grid. Let the minimum and maximum scene depths be α_{min} and α_{max}, and let the unit focal length homogenous coordinates of the pixel p in the first image be $h = [h_o, h_1]$. The 3D coordinate of pixel h at minimum scene depth is

$$(6) \quad X_{\min} = [h_o \alpha_{\min}, h_1 \alpha_{\min}, \alpha_{\min}]^T.$$

Its 3D coordinate in the second image is

$$(7) \quad X'_{\min} = [x'_{\min 0}, x'_{\min 1}, x'_{\min 2}]^T = R(q) X_{\min} + t$$

and the projection of h into the second image is

(8) $h'_{\min} = [x'_{\min 0} / x'_{\min 2}, x'_{\min 1} / x'_{\min 2}]^T$.

An analogous procedure is used to compute $h'_{\max}$ and the camera model is then used to convert $h'_{\min}$ and $h'_{\max}$ into pixel locations $p'_{\min}$ and $p'_{\max}$ that define the epipolar segment in the second image. The CAHVOR camera model is used [16]. If $p'_{\min}$ or $p'_{\max}$ are outside the image then the pixel is removed from consideration. This process is repeated for each pixel in the grid. In Fig. 3 the green segments in the bottom image correspond to the epipolar segments for pixels (red squares) shown in the top image.

Next the matching location of pixel p along the epipolar segment is determined. First a window around p in the first image is compared to a window around $p'_{\min}$ in the second image using sum-of-absolute differences (SAD).

(9) $S(r,c) = \sum_T I_1(r,c) - I_2(r,c)$

The window in the second image is then incremented by one pixel along the epipolar segment and the SAD is recomputed. This process repeats until $p'_{\max}$ is reached. Let p' be the location in the second image of the maximum SAD value along the segment. In a final clean up procedure, correlation values (3) at the eight pixel locations bordering p' are computed and a biquadratic is fit to them. As with the correlation tracker, a sub-pixel correlation peak is obtained from the bi-quadratic and p' is assigned to its location. If the correlation value is less than a threshold, the pixel is eliminated from consideration.

Notice that in contrast to the search for feature tracks over a large window done in the initial stage of the algorithm, the search for dense depth is done along a small one dimensional segment. This increased the efficiency of feature tracking for dense depth recovery and makes it possible to use the efficient SAD tracker. Correlation is more accurate than SAD, but it is less efficient to compute. However, because the search space is constrained, experiments have shown that he SAD tracker rarely tracks incorrectly. Fig. 3 shows the result of SAD tracking the red boxes shown in the top image along the green epipolar segments with matching locations shown as blue boxes.

Once the grid of feature tracks is established, triangulation, using the method described in [15], is applied to establish the depth to each feature. Next, the homogenous coordinates of each feature are scaled by the correspond depths to produce a cloud of 3D points in the coordinate frame of the first image.

D. *Terrain map generation*

For hazard detection, the terrain data should be represented in a surface fixed frame, (i.e., a frame aligned with gravity that is fixed to surface independent of the camera motion) so that (1) local slope relative to gravity can be computed and (2) the helicopter can use surface fixed pose information to navigate to the safe landing site. Furthermore, for efficiency, the terrain data should be evenly sampled so that local operators of fixed size can be applied to detect hazards. The point cloud generated from the dense feature tracks does not meet these criteria. The points are in the coordinate frame of the moving camera, and the points are unevenly sampled in Cartesian space due to the even sampling in image space of a perspective camera that is likely pointed off nadir. To satisfy the hazard detection criteria, a the point cloud is projected into a digital elevation map (DEM).

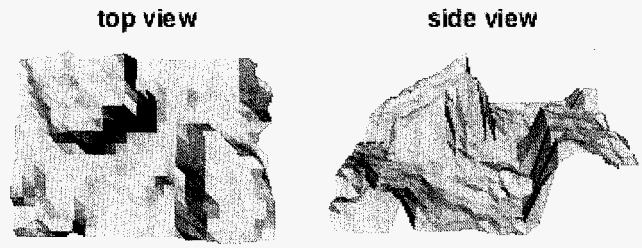


Fig. 4 Digital elevation map.

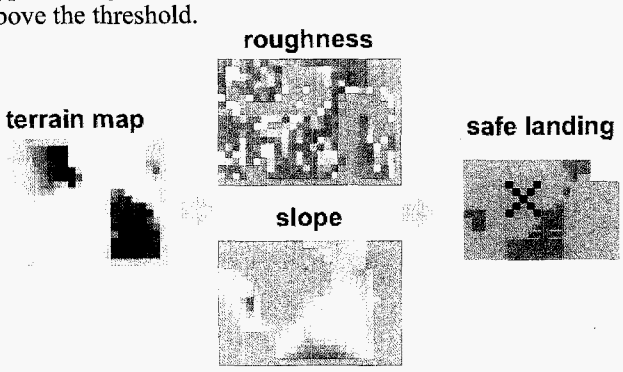
To generate the DEM, a transformation from the camera frame to a surface fixed frame is needed. This transformation can come from an onboard filter that estimates position and attitude in the surface fixed frame or it can be constructed on the fly using the height of the camera above the ground and the surface relative roll and pitch angles of the camera (yaw or azimuth is not needed). Roll and pitch can be measured using an inclinometer, or, if the terrain is assumed to be relatively flat, they can be estimated by fitting a plane (robustly if necessary) to the point cloud. The roll and pitch of the camera are the two angles the describe the relationship between the camera optical axis and the surface normal of the plane. Height above the surface can come from a direct altimetry measurement or it can be computed from the camera roll and pitch and a slant range to the surface.

The DEM is generated as follows. The 3D points in the point cloud are transformed to the surface fixed frame. Next, the horizontal bounding box that contains all of the points is determined and its area A is computed. If there are N points, the size s of the bins in the digital terrain map is set such that $s = \sqrt{A/N}$. With these settings, the DEM will cover roughly the same extent as the point cloud data and each grid cell will contain approximately one sample. Once the bounds and bin size of the elevation map are determined and the points are in the surface fixed frame, the DEM is generated using the same procedure as described in [5]. Stated briefly, for each point, the bin in the DEM that the point falls in is determined and then bilinear interpolation of point elevation is used to deal with the uneven sampling of the surface by the point cloud data. Fig. 4 shows two views of the DEM generated by this process for the feature tracks shown in Fig. 3.

III. HAZARD DETECTION AND AVOIDANCE

Steep slopes, rocks, cliffs and gullies are all hazards for landing. By computing the local slope and roughness, all of these hazards can be detected. We use the algorithm described in [5] to measure slope and roughness hazards. The algorithm proceeds as follows. First the DEM is partitioned into square regions the size of the lander footprint. In each region a robust plane is fit to the DEM using least median squares. A smooth underlying elevation map is generated by bi-linearly interpolating the elevation of the robust planes at the center of each region. A local *roughness map* is then computed as the absolute difference between DEM elevation and this smooth underlying terrain map. Slope is defined as the angle between the local surface normal and vertical; each robust plane has a single slope. A *slope map* is generated by bi-linearly interpolating the robust plane slope from the center of each region.

The lander will have constraints on the maximum slope and maximum roughness that can be handled by the mechanical landing system. These thresholds are set by the used. At the top of Fig. 5 the elevation map, roughness map and slope map are shown for the terrain shown in Fig. 4. For the elevation map, dark corresponds to high terrain and bright corresponds to low terrain. For the slope and roughness maps, green corresponds to regions that are well below the hazard threshold, yellow is for regions that are approaching the threshold and red is for regions that are above the threshold.



Red: unsafe, Green: safe, : unknown
Black X: a-priori landing site, Purple +: selected safe site

Fig. 5 Hazard detection and avoidance maps.

Selection of the safe site starts by generating binary images from the slope and roughness maps; parts of the maps that are above the threshold (hazards) are positive while parts that are below are negative (not a hazard). The roughness and slope hazards are grown by the diameter of the lander using a grassfire transform applied to each map. The logical-OR the grown slope and roughness hazard maps creates the safe landing map. A *safe landing map* is shown in Fig. 5 where safe areas are in green, hazardous areas are in red. Near the border and near holes in the map where there is no elevation data, it is unknown if a hazard exists. These regions are considered hazards, but are marked yellow in the safe landing map.

A grassfire transform is applied to the safe landing map and the bin that is farthest from all hazards is selected as the landing site. If there are multiple bins with the same distance from hazards then the one closest to the a-priori landing site is selected. An a-priori landing site is the site that the lander will land at if no other information is available (i.e., if hazard detection fails to converge). On the safe landing map in Fig. 5 the a-priori landing site is marked as a black X and the selected safe site is shown as a purple +. At the bottom of Fig. 5 the safe landing map is shown texture mapped onto the terrain data from Fig. 4. In this figure it is obvious that the safe site was selected in a low slope and low roughness region.

TABLE I EXAMPLE ALGORITHM RUN TIMES (FOR GIVEN PARAMETSRS) ON A 400 MHZ R12000 PROCESSOR

Algorithm Stage	Run Time	Parameters
Feature selection & tracking	0.21 s	11x11 pixel templates 81x81 pixel windows
Structure from motion	0.10 s	59 feature tracks
Terrain map generation	0.41 s	600 structure pixels
Hazard detection & avoidance	0.05 s	19x27 terrain map

The run time for each stage of the algorithm on an SGI O2 with a 400 MHz R12000 are shown in TABLE I. As can bee seen from the table, the total processing time is less than one second.

IV. JPL AUTONOMOUS HELICOPTER TESTBED

The JPL Autonomous Helicopter Testbed (AHT) is a twin-cylinder, gas powered radio-controlled model helicopter approximately 2 meters in length and capable of lifting approximately 9 kg of payload. Onboard avionics include a PC/104-based computer stack running the QNX RTOS, (700 MHz PIII CPU with 128Mb DRAM and 128 Mb flash disk), NovAtel RT2 GPS receiver, Inertial Sciences ISIS IMU, Precision Navigation TCM2 compass & roll/pitch inclinometers, and downward-pointing MDL ILM200B laser altimeter and a 640-480 Sony XC-55 progressive scan grayscale CCD camera. A Dell Inspiron 8200 laptop functions as a ground station used to send high-level control commands to, and display telemetry from, the JPL AHT as well as being a conduit for differential corrections from a NovAtel RT2 GPS basestation receiver to the JPL AHT. Communication between the laptop and AHT is achieved using a 2.4 Ghz Wireless -G Ethernet link.

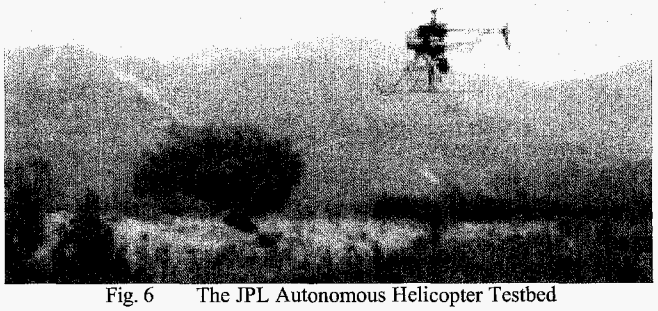


Fig. 6 The JPL Autonomous Helicopter Testbed

An error-state Kalman filter [11] produces state estimates used for the control of the AHT. The state of the filter is initialized using inputs from the compass & inclinometers (orientation) and GPS. (position). Once initialized, the filter state is updated using inputs from the above mentioned sensors as well as the gyro rates and accelerations from the IMU.

Autonomous flight is achieved using a *hierarchical behavior-based* control architecture [7]. A behavior-based controller partitions the control problem into a set of loosely coupled behaviors. Each behavior is responsible for a particular task. The behaviors act in parallel to achieve the overall goals of the system. Low-level behaviors are responsible for functions requiring quick response while higher-level behaviors meet less time-critical needs. For example , the low-level roll control behavior is responsible for maintaining a desired roll angle while the high-level navigation behavior is responsible for achieving a desired GPS waypoint location.

V. SAFE LANDING EXPERIMENTS

A total of four successful autonomous landings were achieved on two separate days, one on the first day of testing and three on the second. The landings were achieved in unknown, hazardous terrain using the following procedure. The helicopter is commanded to fly laterally over the terrain while maintaining its current altitude. While in transit, 40 images of the terrain below the helicopter are gathered over the course of several seconds by the onboard downward-looking camera. Two images for hazard detection are chosen from these 40 images with the criteria being a function of the baseline (larger baseline gives better stereo ranging) and amount of overlapping terrain (larger overlap increases number of features to track) between the two images.

If a safe site is located by the hazard detection and avoidance algorithm, the pixel coordinates of this safe site are transformed into GPS coordinates. This transformation is made possible due to the fact that the 6DOF state of the helicopter plus the laser altimetry range to the ground is all gathered when each image is captured. Once the GPS coordinates are computed, they are passed to the navigation control behavior of the AHT and it guides the helicopter to the desired GPS coordinates. Once the AHT is within a predetermined threshold of these coordinates (currently 2 meters), the AHT descends maintaining a desired vertical descent velocity while continuing to attempt to reduce the error between its current GPS position and desired position. Once the AHT is within a predetermined distance threshold above the ground (determined from laser range measurements and currently set at 1.5 meters), the AHT slowly changes the pitch of the main blades on the helicopter to effect a smooth landing at the safe site.

TABLE II gives the results from the 4 successful landings of the AHT in unknown, hazardous terrain. Unfortunately, the results from the second run are missing but the position error is on the same order of magnitude as the other 3 runs. This position error is the Euclidian distance between the desired GPS northing and easting values and the actual GPS northing and easting measurements provided by the GPS receiver on board the AHT. In addition to the northing and easting measurements, the GPS receiver also provides estimates of the standard deviation for each individual northing and easting measurement. These standard deviations are given because the position error is a direct function of how accurate the GPS measurements are at the time the error is computed. In the results below, dozens of measurements of the values reported are taken after the AHT has successfully landed and averages computed to smooth out variations from one measurement to the next. The table also shows that all run-times took less than 2 seconds.

TABLE II SAFE LANDING ACCURACY RESULTS

Run #	Position Error	Northing Std. Dev.	Easting Std. Dev.	Run time
1	1.21m	0.70m	0.43m	1.8s
2	Data from this run is missing			
3	1.20m	0.61m	0.47m	1.9s
4	0.89m	0.39m	0.35m	1.7s

VI. CONCLUSION

Currently, our system relies on GPS to navigate for landing. We our working on methods for GPS denied safe landing based on using visual navigation. In addition to image-based motion estimation, some of the techniques we are currently pursuing are landmark recognition for position estimation during landing and landing site visual servoing.

The main commercial application of this technology is autonomous navigation of unmanned aerial vehicles for military, search and rescue, fire fighting and surveillance applications. This technology could also be used directly by autonomous underwater vehicles for seafloor exploration with applications to the search for oil and other natural resources and scientific discovery for geology, biology and chemistry.

ACKNOWLEDGMENT

We would like to thank Chuck Bergh for designing and building the Helicopter Avionics and Srikanth Saripalli, for assistance with testing. We would also like to thank Doug Wilson and Alan Butler, our helicopter safety pilots.

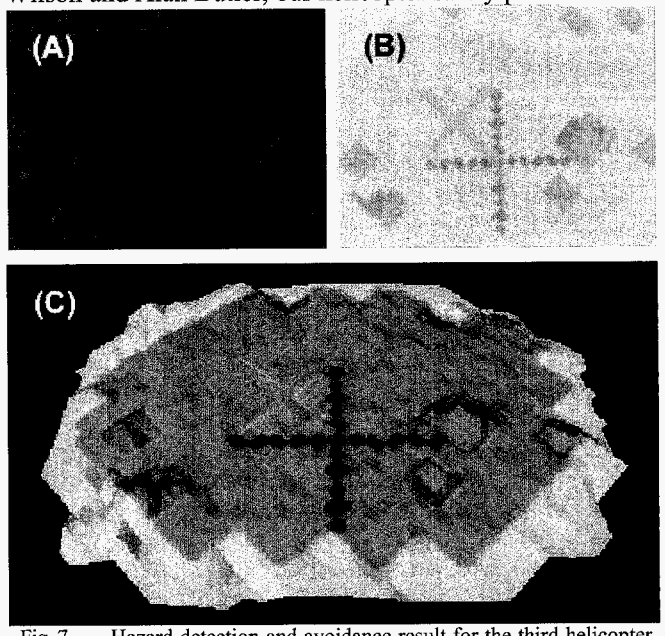


Fig. 7 Hazard detection and avoidance result for the third helicopter run. (A) First image selected from image stream. (B) Safe landing map overlaid on first image. (C) Safe landing map overlaid on terrain map.

REFERENCES

[1] O. Amidi, T. Kanade, and K. Fujita, "A Visual Odometer for Autonomous Helicopter Flight," *Robotics and Autonomous Systems*, **28**, pp. 185-193, 1999.

[2] A. Benedetti and P. Perona, "Real-Time 2-D Feature Detection on a Reconfigurable Computer," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 586-593, 1998.

[3] M. Bosse, W.C. Karl, D. Castanon and P. DiBitetto, "A Vision-Augmented Navigation System," *Proc. IEEE Conf. Intelligent Transportation Systems*, pp. 1028-33, 1997.

[4] P. Garcia-Padro, G. Sukhatme, and J. Montgomery, "Towards Vision-Based Safe Landing for an Autonomous Helicopter," *Robotics and Autonomous Systems*, **38**(1) , pp. 19–29, 2001.

[5] A. Johnson, A. Klumpp, J. Collier and A. Wolf, "Lidar-based Hazard Avoidance for Safe Landing on Mars," *AIAA Jour. Guidance, Control and Dynamics*, **25**(5), 2002.

[6] L. Matthies, *Dynamic Stereo Vision*, Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1989.

[7] J. Montgomery. *Learning Helicopter Control through "Teaching by Showing"*. Ph.D. Thesis, School of Comp. Sci., USC, 1999.

[8] D. Nister, O. Noriditsky, J. Bergen. "Visual Odometry." *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR'04), 2004.

[9] J. Oliensis. "Exact Two Image Structure from Motion." *IEEE Pattern Analysis and Machine Intelligence*, **24**(12), pp. 1618-33, 2002.

[10] S. Roumeliotis, A. Johnson and J. Montgomery. "Augmenting Inertial Navigation with Image-based Motion Estimation" *Proc. Int'l Conf. Robotics and Automation (ICRA'02)*, pp. 4326-33, 2002.

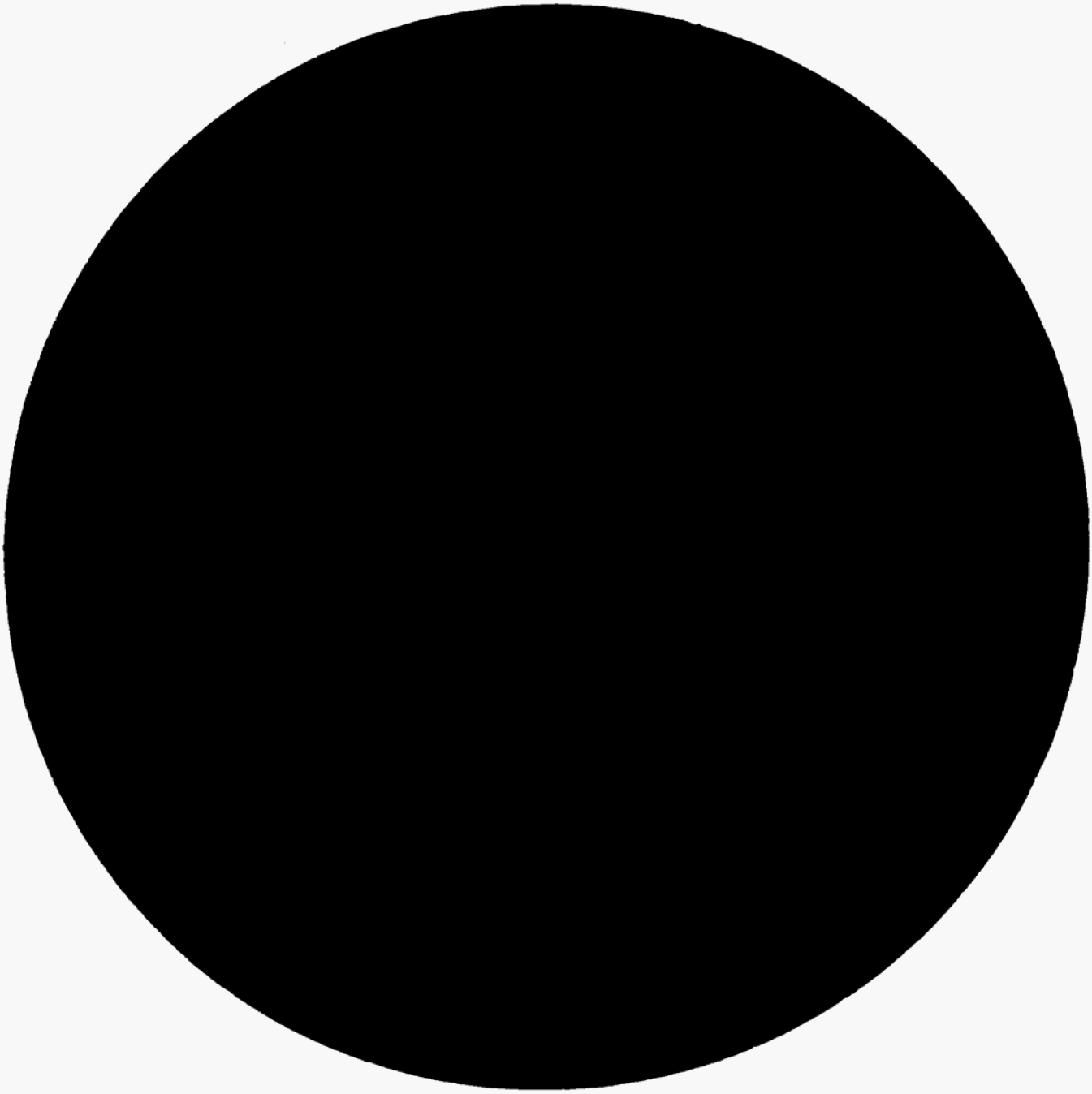
[11] S. Roumeliotis, G. Sukhatme and G. Bekey. "Circumventing Dynamic Modeling: Evaluation of the Error-State Kalman Filter Applied to Mobile Robot Localization." *Proc. Int'l Conf. Robotics and Automation (ICRA '99)*, pp. 1656-63, 1999.

[12] S. Saripalli, J. Montgomery and G. Sukhatme, "Visually-Guided Landing of an Unmanned Aerial Vehicle." *IEEE Transactions on Robotics and Automation*, **19**(3) pp. 371-81, 2003.

[13] O. Shakernia, R. Vidal, C. S. Sharp, Y. Ma, and S. S. Sastry. "Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle." *Proc. Int'l Conf. Robotics and Automation (ICRA '02)*, pp. 2793–2798, 2002.

[14] J. Shi and C. Tomasi. "Good Features to Track." *Proc. IEEE Conf. Computer Vision & Pattern Recognition* (CVPR94), pp. 593-600, 1994.

[15] J. Weng, N. Ahuja and T. Huang. "Optimal Motion and Structure Estimation." *IEEE Pattern Analysis and Machine Intelligence*, **15**(9) pp. 864-884, 1993.

[16] Y. Yakimovsky & R. Cunningham. "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras." *Computer Graphics & Image Processing* **7**, pp. 195-210, 1978.

End of File