The JPL Autonomous Helicopter Testbed: A Platform for Planetary Exploration Technology Research and Development

James F. Montgomery, Andrew E. Johnson, Stergios I. Roumeliotis, and Larry H. Matthies[1]

***Abstract* – The JPL Autonomous Helicopter Testbed (AHT), an aerial robot based upon a radio-controlled (RC) model helicopter, provides a small, low-cost platform for developing and field-testing new technologies needed for future space missions. The AHT helps cover the test space in a complementary fashion to other methods such as rocket sleds or parachute drops. The AHT design and implementation is presented as well as experimental results and milestones achieved since its creation in 2001. In addition, technologies we are developing and testing are described. These include image-based hazard detection and avoidance algorithms for safe landing in dangerous terrain and an extended Kalman filter (EKF) that augments inertial navigation with image-based motion estimates to enable pin-point landing.**

Keywords – autonomous helicopter, UAV, image-based hazard detection and avoidance, vision-augmented inertial navigation.

1. Introduction

Numerous future NASA missions have increasingly aggressive goals that require capabilities unavailable today. These include landing safely in hazardous terrain or closer to a particular surface location, i.e., the pin-point landing problem. The technologies to enable these capabilities must be developed and validated through simulation and real-world field tests prior to their use. For field-testing, NASA commonly uses autonomous testbeds (e.g., rovers, aerobots, and helicopters) to demonstrate technology on Earth under mission relevant conditions.

In 2001, the Jet Propulsion Laboratory Autonomous Helicopter Testbed (AHT) (Figure 1) was created. Its purpose is to support the development and field-testing of new technologies needed for future space missions. To date it has been used for planetary landing technology development. Specifically, image-based hazard detection and avoidance has been implemented on the AHT, which has resulted in the first autonomous landing of an unmanned helicopter in unknown, hazardous terrain.

In Section 2, we present the design and implementation of the AHT, including its avionics and software. Capabilities and milestones achieved to date are discussed. The JPL Gantry Testbed (GT) is introduced in Section 3. The GT provides a facility for testing algorithms in a lab-based environment prior to field-testing on the AHT. A comparison of a number of different testbeds, including the AHT and GT, is given in Section 4. Our recent work in vision-based safe and precise landing and vision-augmented inertial navigation, including approach details and experimental results, is described in Section 5. Our current work, Section 6, includes advancing planetary landing technology, as well as increasing the capabilities of the AHT itself. Concluding remarks are made in Section 7.

[1] J.F. Montgomery, A.E. Johnson and L.H. Matthies are with the Jet Propulsion Laboratory, California Institute of Technology {monty|aej|lhm}@jpl.nasa.gov. S. I. Roumeliotis is with the University of Minnesota. stergios@cs.umn.edu.

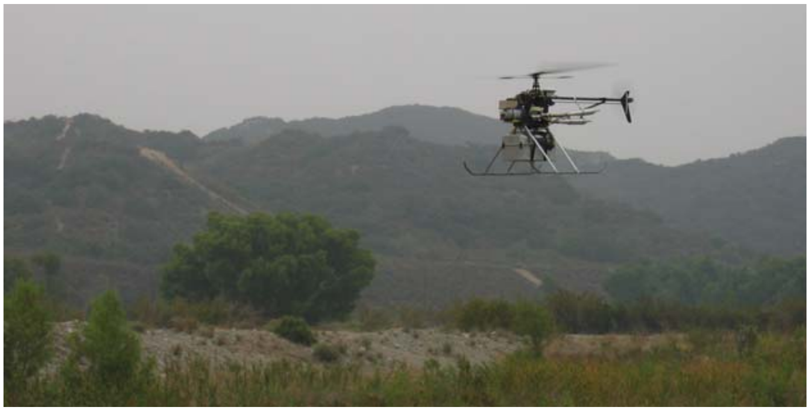

Figure 1 The JPL Autonomous Helicopter Testbed

1.1 Related Work

Before new technology is accepted by a space science mission, its performance must be validated in a relevant environment. For one example technology, terrain relative navigation during landing on Mars, validation requires field-testing to achieve realistic descent dynamics, appropriate altitude and sensing of natural Mars-like terrain. Each field test campaign focuses on the most uncertain aspect of the performance of the algorithm. For the case of landing hazard detection using a scanning lidar, performance at high vertical velocities was the highest concern, so a rocket sled was used to simulate landing (Johnson & Skulsky, 2002). The Mars Exploration Rover Descent Image Motion Estimation System (MER-DIMES) used images to estimate velocity during descent towards the surface of Mars. Prior to launch, a manned helicopter platform was used to fly the MER-DIMES system over three different Mars-like terrains because performance over bland and repetitive terrain needed to be proven (Johnson, Willson, Goguen, Alexander & Meller, 2005). Recently a series of high altitude parachute drop tests were used to collect imagery for testing landmark recognition algorithms. This platform provided useful imagery at very high altitude and under representative parachute dynamics. A complementary alternative to these three testbeds is the autonomous small-scale helicopter. The AHT is representative of this class of testbed.

Autonomous aerial vehicles, helicopters included, have been an area of active research for many years now. An early system, the Stanford Hummingbird (Conway, 1995), with its navigation system based upon GPS alone, demonstrated a small RC model helicopter capable of autonomous hover and low-speed lateral flight. A system for producing position and velocity estimates for a helicopter based upon integration of GPS and IMU measurements is given in (Jun, Roumeliotis & Sukhatme, 1999).

There are numerous examples of systems using vision for both state estimation and control. In (Bosse, Karl, Castanon & DiBitetto, 1997), optical flow-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. The use of optical flow however is restrictive since it is reliable only in domains where the motion between images is expected to be small. Furthermore, it is assumed that the surface being imaged is flat, which may not be the case during landing on planetary surfaces. (Amidi, Kanade & Fujita, 1999), present a visual odometer which estimates 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 are estimated based upon template matching from sequences of stereo vision data. In this approach, attitude estimates are not derived from the vision algorithm and it is assumed that the field of view changes slowly while the helicopter hovers above the same area. New templates are acquired only when the previous ones disappear from the scene. Stereo vision for height and velocity estimation is combined with IMU measurements in an EKF/CF (Complementary filter) in (Corke, 2004). The goal is to minimize the use of GPS and instead rely on velocity estimates from vision for control. Various approaches for vision-based control for autonomous landing are described in (Garcia-Padro, Sukhatme & Montgomery, 2001), (Shakernia, Vidal, Sharp, Ma & Sastry, 2002), (Saripalli, Montgomery & Sukhatme, 2003) (Meingast, Geyer & Sastry, 2004), and (Hintze, 2004). In (Garcia-Padro et al., 2001), (Meingast et al., 2004) and (Hintze, 2004), no autonomous landing is attempted, however vision-based approaches for safe-landing site detection in unknown, unstructured terrain are described. Both (Shakernia et al., 2002) and (Saripalli et al., 2003) describe and demonstrate a vision-based approach for locating a known target and then tracking it while navigating to and landing on the target. However, in these two approaches, the target area is known *a priori* to be flat and safe.

In terms of advanced helicopter control, (Gavrilets, 2003)(La Civita, 2003)(Ng et al., 2004) have all demonstrated novel approaches with actual experimental results. In (Gavrilets, 2003), a control methodology for the autonomous execution of aerobatic maneuvers with small-scale helicopters is presented. In addition, a low-order dynamic model describing a miniature helicopter in a wide range of flight conditions, including aerobatics, is given. An H-infinity controller is developed and used in (La Civita, 2003) to demonstrate a number of maneuvers on a Yamaha R-50 helicopter, including forward and backward flight in circles and nose-in and nose-out pirouettes. A reinforcement learning algorithm is used to generate a controller capable of autonomous inverted hovering in (Ng et al., 2004).

2. JPL Autonomous Helicopter Testbed (AHT)

2.1 Platform

The JPL AHT is based upon the Bergen Industrial Twin, a twin-cylinder, gas powered radio-controlled (RC) model helicopter. It is approximately 2 meters in length, capable of lifting around 9 kg of payload and can fly for about 15-30 minutes per tank of fuel depending upon payload mass and flight profiles. It has five control inputs; main rotor lateral and longitudinal cyclic pitch, tail rotor pitch, main rotor collective pitch and engine throttle. The first three inputs control the roll, pitch and yaw of the helicopter while the last two control its thrust. It can be flown by a human safety pilot in a) a completely teleoperated mode using an RC hobbyist 72 MHz transmitter, b) a completely autonomous mode via the AHT's onboard avionics and control algorithms or c) a mixture of teleoperation and autonomy via an onboard switching system that allows control of AHT servo control actuators on an actuator by actuator basis.

2.2 Avionics

Onboard avionics include a PC/104-based computer stack and numerous sensors all powered by laptop batteries. The computer stack consists of an Advanced Digital Logic MSMP3SEV 700 MHz Pentium III CPU with 128Mb DRAM and a 128 Mb flash disk running the QNX RTOS, Diamond Systems QMM-10 Timer/Counter and EMM-DIO Quad Serial/Digital I/O cards, an Imagenation PX610 grayscale framegrabber card, an Advanced Digital Logic MSMJ104D dual-slot PCMCIA card carrier and the Parvus OnPower 75, a 75 Watt DC/DC power converter card. A custom integration PCB has been designed that routes power and data amongst the various components and also implements the actuator switching logic described above.

The CPU executes all software, written in a mixture of C/C++. The QMM recognizes and generates many waveforms, including pulse-width modulated (PWM) signals. This enables the reading of pilot control inputs from the transmitter for setting actuator trim positions in the control algorithms as well as sending commands to the actuators generated by the control algorithms. The EMM, along with two serial ports on the CPU board, provide RS-232 interfaces to all sensors other than the camera, which is interfaced to the PX610 board. The PCMCIA card carrier holds an Orinoco Silver 2.4 GHz PCMCIA card that provides 11 Mbps wireless Ethernet connectivity to the AHT. The Parvus board is connected to two, in parallel, Molicel ME202A 11.1V 6.6 Ah batteries and it supplies +5V and +/-12V power to the avionics.

Onboard sensors include a NovAtel OEM4 GPSCard receiver, an Inertial Sciences ISIS IMU, a Precision Navigation TCM2 compass & roll/pitch inclinometers, and an MDL ILM200A laser altimeter and Sony XC-55 640x480 pixel grayscale CCD camera, both body-fixed and downward pointing. A Dell Inspiron 8200 laptop functions as a ground station used to send high-level control commands to, and display telemetry from, the AHT as well as being a conduit for differential corrections from a NovAtel PowerPak-4 GPS base-station receiver to the AHT. Communication between the laptop and AHT is achieved using another Orinoco Silver card in a PCMCIA slot in the laptop and a Linksys WRT54G Wireless-G Router.

2.3 Algorithms

Algorithms can be thought of as belonging to one of two categories. Those that give the AHT basic state estimation and control capabilities (briefly described here) and those related to the planetary exploration technologies being developed and tested (describe in detail later).

An error-state Kalman filter (Roumeliotis, Sukhatme, & Bekey, 1999) produces state estimates used by the AHT control system. The filter state is initialized using compass/inclinometers (orientation) and GPS (position) measurements. Once initialized, the filter state is updated using the above-mentioned sensors as well as IMU gyro and accelerometer measurements.

Autonomous flight is achieved using a *hierarchical behavior-based* control architecture (Montgomery, 1999). Briefly, 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.

2.4 Capabilities and Milestones

Since the creation of the JPL AHT in 2001, a number of capabilities have been developed and milestones achieved (Table I). It can operate in a completely autonomous mode, i.e., no human-in-the-loop. It is capable of stable hover, low-speed flight, GPS-waypoint following and landing in unsafe terrain. A variety of sensors were used for each milestone, abbreviated as follows: I=IMU, G=GPS, C/I=compass/inclinometers, L=laser altimeter, C=camera.

Date	**Milestone**
19-May-2001	First teleoperated flight streaming telemetry (I, G, C/I, L, C).
08-Jun-2002	First completely autonomous flight, demonstrating both hover and GPS-based waypoint following (I, G, C/I)
27-Apr-2003	First autonomous landing in known, safe terrain at a GPS-designated location (I, G, C/I, L).
21-Sep-2003	First autonomous vision-guided landing in unknown, hazardous terrain (I, G, C/I, L, C).

Table I JPL AHT milestones

The first milestone was achieved after the stock Bergen helicopter was outfitted with its avionics. Software drivers for each sensor streamed data to the laptop while our safety pilot flew the AHT in a completely teleoperated mode.

The remaining milestones are in fully autonomous mode. The first milestone used a subset of the onboard sensors to achieve autonomous flight, demonstrating the ability to both hover and fly from one GPS-waypoint to another. Beginning with this milestone, our EKF was operational and provided state estimates to the AHT control system. For the next milestone, an altitude controller was implemented that uses GPS measurements to maintain hover at altitude, velocity estimates from the EKF to control vertical descent during landing, and laser altimeter readings to sense the ground prior to landing. Once these pieces were in place, we were able to demonstrate safe landing at a GPS-designated location known *a priori* to be safe. The most recent major milestone for the AHT was the demonstration of autonomous landing in unknown, hazardous terrain using vision. This milestone and the algorithms and sensors required to meet this milestone are described in greater detail in Section 5.

3. JPL Gantry Testbed

The JPL Gantry Testbed (GT) (Figure 2) is a 5DOF testbed that allows intermediate testing of algorithms prior to testing on the AHT. The GT provides a hardware-in-the-loop platform for collecting data above a simulated planetary surface useful for validating algorithms in a controlled environment. It can be commanded to move to absolute x, y and z positions and pan, tilt angles. It can also be given velocity commands in each DOF. The gantry provides ground truth linear measurements with 0.35 millimeter resolution and angular measurements with 0.01 degree resolution at up to a 4 Hz data rate. It can carry payloads weighing up to 9 pounds.

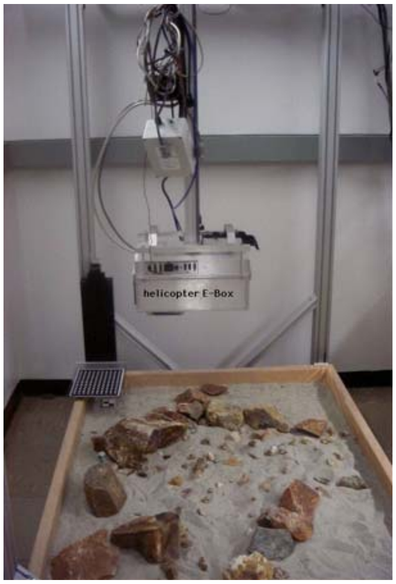

Figure 2 The JPL Gantry Testbed

The electronics box (E-Box) that contains the AHT avionics is detachable and can be mounted directly onto the gantry. The advantage of this is, to the greatest extent possible, that the same algorithms and sensors are used by both testbeds. (Position measurements from the GT are used to simulate GPS measurements since the GT is indoors).

The AHT and GT have complementary characteristics. The GT provides a more controlled environment with excellent ground truth and repeatability in “flight” trajectories at the expense of reduced kinematic and dynamic motion while the converse is true for the AHT. Both can be used for closed-loop testing of sensors

	Rocket Sled	Manned Helicopter	Parachute Drop	Aerobot	Gantry	Autonomous Helicopter
Large Descent Speeds	+		+			
Mission Relevant Horizontal Velocity		+	+	+	+	+
Mission Relevant Altitude	+	+	+	+		
Mission Relevant Attitude and Attitude Rates	+ if instrumented	+ if instrumented	not necessarily		+	
Real Terrain		+	+	+		+
Closed-Loop Pointing	+	+		+	+	+
Closed-Loop Landing				+	+	+
Safe	+	+		+ if safety pilot	+	+ if safety pilot
Repeatable	+	+		+	+	+
Low Cost				+	+	+
Simple Logistics					+	+

Table II Test platform comparisons

and algorithms. Also, using both together allows us to examine various algorithm and system performance criteria more fully than we could with either testbed alone. In the next section, we will compare and contrast the GT and AHT as well as additional testbeds. As will be seen, each platform has its strengths and weaknesses.

4. Test Platform Comparison

Table II provides a comparison of a variety of test platforms using a number of qualitative metrics. A '+' mark in a given location indicates that a testbed meets that metric (some with caveats). The first three columns were described earlier in Section 1. The last two columns describe testbeds similar to the JPL GT and AHT. A representative system for the aerobot (a lighter-than-atmosphere system) is given in (Elfes et al., 2005). For more details, the interested reader is directed to the appropriate reference for each testbed.

The first seven metrics are related to a particular application domain, that of spacecraft Entry, Descent, and Landing (EDL), while the last four are more general in nature. Keep in mind that certain testbeds will be better suited to this domain than others. This domain was chosen for illustrative purposes since most of our research to date has been in the area of EDL technology development.

During entry into the vicinity of a planetary body and prior to landing, large descent speeds are possible. Both rocket sled and parachute drop testbeds can achieve these speeds. However, the rocket sled moves on a track in a horizontal fashion while the parachute drop moves in a largely vertical direction more closely matching the trajectory of a spacecraft during descent. A subset of the testbeds can achieve mission relevant horizontal velocity, altitude, attitude and attitude rates seen during descent. The ability to support operation over real terrain as well as closed-loop pointing and landing allows more extensive and realistic testing of technologies under development. How vulnerable a system under test is to damage is reflected by the safety metric while repeatability refers to whether or not a testbed supports running a test case multiple times with largely similar conditions. The last two metrics provide a measure of how expensive and difficult a given testbed is to operate.

In summary, the AHT provides a low-cost, logistically simple platform that allows repeatable testing in a safe fashion. For the EDL domain, it enables closed-loop pointing and landing using real terrain at mission relevant horizontal velocities. However, it cannot achieve mission relevant descent speeds, altitudes or attitudes and attitude rates.

5. Recent Work and Results

A number of future space missions require landing on the surface of planetary bodies such as Mars, the Moon, asteroids and comets. NASA and JPL are investing significant resources in advancing the state of the art in EDL technologies to support these missions and the AHT has been used for the development and test of some of them.

Recent work includes image-based technologies to enable autonomous safe and precise landing on planetary bodies. It includes machine vision algorithms that use image streams from a monocular, downward-looking camera to a) land safely in unknown, dangerous terrain via hazard detection and avoidance (HDA), b) track a safe site once located to enable more precise landing and c) provide image-based motion estimates to an Extended Kalman filter to augment inertial measurements from an IMU thereby increasing state estimation accuracy and robustness.

5.1 Safe Landing Via Hazard Detection and Avoidance

To date, no space science mission has employed autonomous HDA 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 choose landing sites with a greater emphasis on science return and less on engineering safety criteria.

This section describes a novel algorithm for HDA 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 area correlation 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 approaches, 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.

It should be emphasized that the major technological advance of the work described in this section is the development of a complete system for landing hazard detection and avoidance. One component of the system is a structure from motion (SFM) algorithm that borrows heavily from existing SFM algorithms (Azarbayejani & Pentland, 1995), (Weng, Ahuja & Huang, 1993). It is possible that this algorithm could be replaced with a different existing algorithm. Whatever SFM algorithm is employed, it would still have to be integrated with other the algorithmic components and possibly modified to make it sufficiently robust and computationally efficient to be used in the field. In describing the details of our SFM algorithm we focus on the implementation details that were required to make the algorithm work in practice on an autonomous helicopter.

5.1.1 Terrain Map Generation

The inputs into the 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 and a dense terrain map of the imaged surface. . 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 III.

5.1.1.1 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 (Benedetti & Perona, 1998). This algorithm was chosen over others because in practice it results in good features at low computational cost.

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 with the single best feature selected from each block. As shown in Figure 3, this approach spreads the features evenly across the image.

Once features are selected they are located in the second image using a area 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 (Moravec, 1977).

$$C(r,c) = 2\left(\sum_T \tilde{I}_1(r,c) * \tilde{I}_2(r,c)\right) / \left(\sum_T \tilde{I}_1^2(r,c) + \sum_T \tilde{I}_2^2(r,c)\right) \tag{1}$$

where $\tilde{I}$corresponds to the *I* with the mean subtracted and *(r,c)* correspond to a pixel at row *r* and column *c* in the image, Correlation is applied in a coarse to fine fashion as follows. First, block averaging is used to construct an image pyramid for both images. Feature tracking is then performed at the coarsest level of the pyramid with a template and a search window size scaled to match the coarse resolution. The pixel of highest correlation is used to seed the correlation at the next finer level. After the coarse level, the template size increases as the pyramid level increases while the 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.

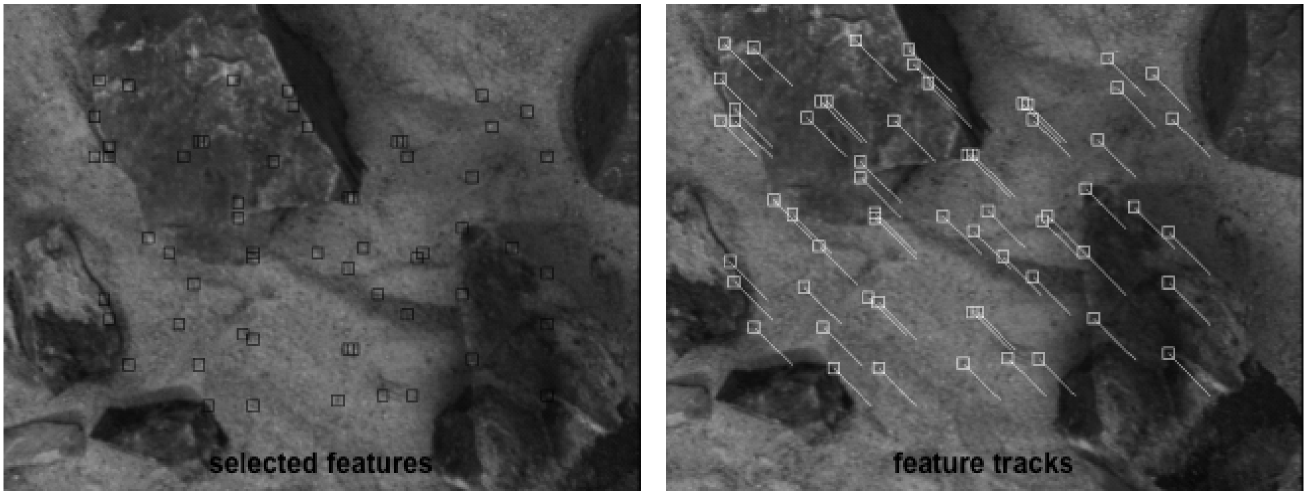

Figure 3 Feature selection and tracking

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 20%.

5.1.1.2 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 (i.e., the motion) of the camera between the images and the depth to the selected features in the first image (i.e., the structure). Structure from motion has been studied for decades, and there are numerous structure from motion algorithms in existence (see (Nister, Noriditsky & Bergen, 2004), (Oliensis, 2002) for the state of the art).

This stage uses a previously reported algorithm (Roumeliotis, Johnson & Montgomery, 2002) that combines techniques presented in (Azarbayejani & Pentland, 1995) and (Weng et al., 1993). The approach uses a robust non-linear least squares optimization to minimizes 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. The altimetry measurement is used to set the initial depths to the features in the scene. This step is crucial because it eliminates the scene scale ambiguity present in other 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. Figure 4 shows three views of the computed motion and structure for the images shown in Figure 3. The two positions of the camera are shown as coordinate axes. The fields of view of the images are shown as rectangles and the 3D position of the feature tracks are shown as white dots.

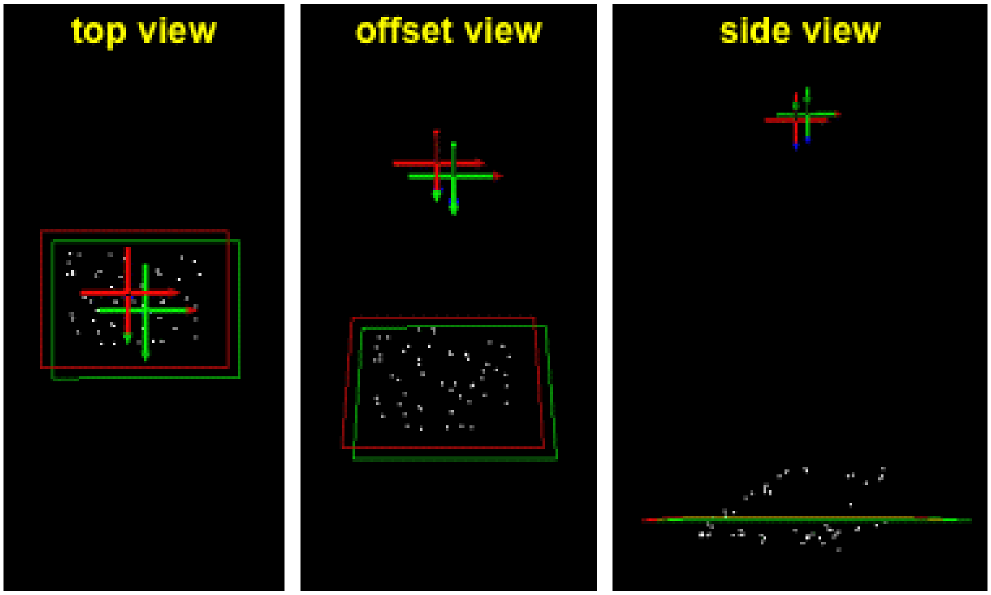

Figure 4 Motion estimation and coarse depth estimation

5.1.1.3 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 known baseline roughly aligned with the image rows, when using a single camera to recover scene structure, the motion between images is arbitrary. Arbitrary motion can result in the focus of expansion being in the image, which makes it difficult to apply standard scan-line rectification algorithms to make surface reconstruction efficient. Consequently, we developed a new approach that takes advantage of the motion and structure recovered in the previous SFM step.

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.

The epipolar segment is determined for each pixel in the grid as follows. Each pixel defines a ray in space. The minimum and maximum depth are used to define two points along this ray. When these points are projected into the second image, they define the epipolar segment. In Figure 5 the segments in the bottom image correspond to the epipolar segments for pixels shown in the top image.

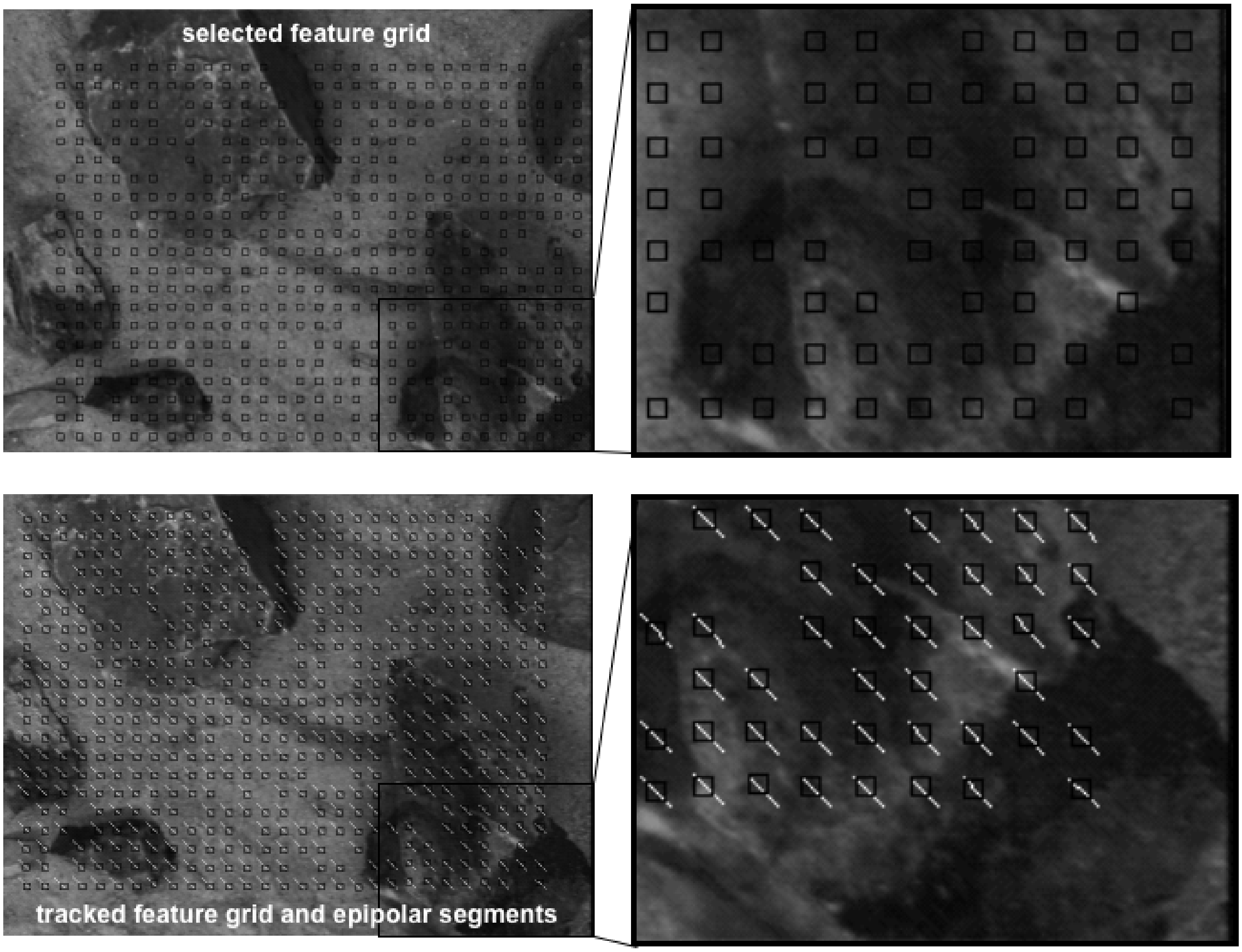

Figure 5 Dense feature tracking on epipolar segments

Next, the best matching location of pixel along the epipolar segment is determined by computing the sum-of-absolute differences (SAD)

$$(2)\quad S(r,c) = \sum_{T} \left| I_1(r,c) - I_2(r,c) \right|$$

along the epipolar segment and selecting the minimum. In a final clean up procedure, correlation values (1) in a 3x3 pixel neighborhood around the SAD minimum are computed and a biquadratic is fit to them. As with the correlation tracker, a sub-pixel correlation peak is obtained from the biquadratic. 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 realized in the initial stage of the algorithm, the search for dense depth is performed 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 the SAD tracker rarely tracks incorrectly. Figure 5 shows the result of SAD tracking where the squares shown in the top image are matched along the epipolar segments with matching locations shown as squares in the bottom image.

Once the grid of feature tracks is established, triangulation, using the method described in (Weng et al., 1993), is applied to establish the depth to each feature. Next, the homogenous coordinates of each feature are scaled by the corresponding depths to produce a cloud of 3D points in the coordinate frame of the first image.

5.1.1.4 Terrain Map Construction

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 (i) local slope relative to gravity can be computed and (ii) 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. To meet these criteria, the point cloud is projected into a digital elevation map (DEM).

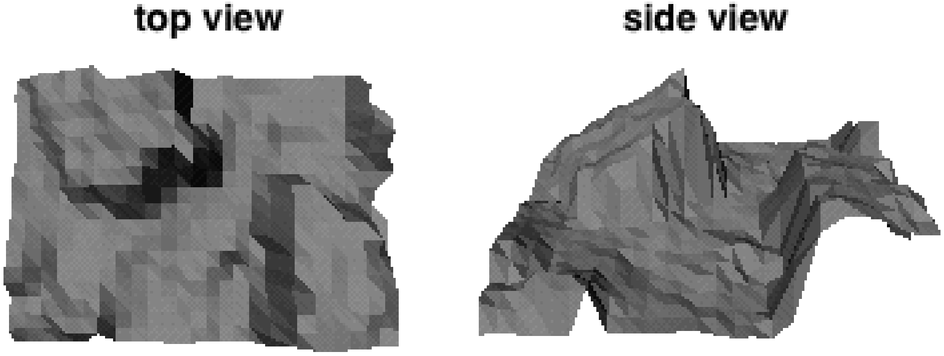

Figure 6 Digital elevation map

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 (from altimetry) and the surface relative roll and pitch angles of the camera (from the EKF). 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 (Johnson, Klumpp, Collier, & Wolf 2002). 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. Figure 6 shows two views of the DEM generated by this process for the feature tracks shown in Figure 5.

5.1.2 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 (Johnson et al., 2002) 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 plane is robustly fit to the DEM using least median squares. A smooth underlying elevation map is generated by bilinearly 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 bilinearly 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 user. At the top of Figure 7 the elevation map, roughness map and slope map are shown for the terrain shown in Figure 6. 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.

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 (Russ 1999) applied to each map. The logical-OR of the grown slope and roughness hazard maps creates the safe landing map. A *safe landing map* is shown in Figure 7 where safe areas are in green and 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 Figure 7 the a-priori landing site is marked as a black X and the selected safe site is shown as a purple +. At the right of Figure 7 the safe landing map is shown texture mapped onto the terrain data from Figure 6. In this figure it is obvious that the safe site was selected in a low slope and low roughness region.

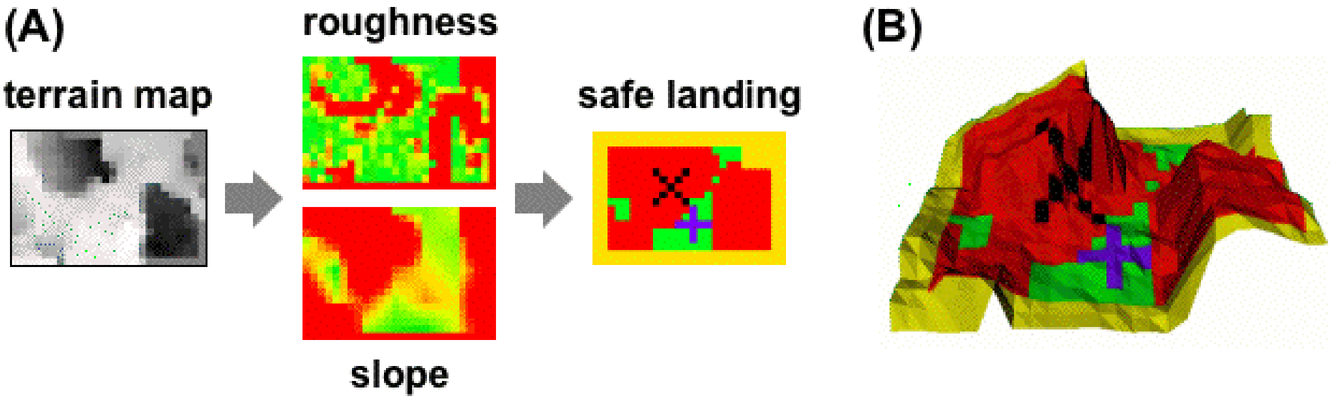

Figure 7 (A) Hazard detection and avoidance maps. (B) Safe landing map texture mapped onto elevation mesh showing correct site selection. (red = unsafe, green = safe, yellow = unknown, black X = a-priori landing site, purple + = selected safe site)

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

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

Table III Example algorithm run times (for given parameters) on a 400 MHz R12000 processor

5.1.3 Safe Landing Experiments

A total of four successful autonomous landings were achieved with the AHT 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 HDA vision algorithms, then GPS is used to land at this site. (Future plans are to demonstrate closed-loop visual tracking of the safe site to enable precise landing, see Section 0 below). The current system transforms the image pixel coordinates of the safe site 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 enable a smooth landing at the safe site.

Table IV gives the results from the 4 successful landings of the AHT in unknown, hazardous terrain and Figure 8 displays visual data from the third landing. Unfortunately, the data from the second run is 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.

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

Table IV Safe landing accuracy results

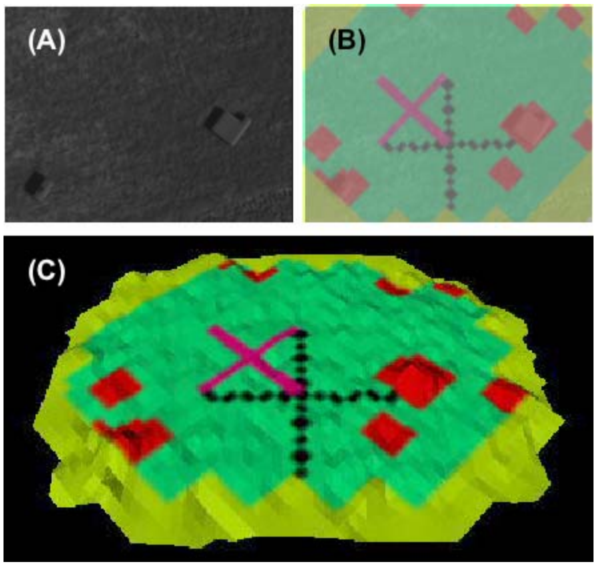

Figure 8 Hazard detection and avoidance result for the third helicopter run. (A) First image selected from image stream. (B) Safe landing map overlaid on first image. (C) Safe landing map overlaid on terrain map.

5.2 Precise Landing Via Safe Site Tracking

As is the case for HDA during landing, no space mission has ever employed safe site tracking for precise landing. This problem is difficult because the large changes in attitude and attitude of a spacecraft during landing make the use of a 2D tracker of limited use. In contrast, a homography transform (Hartley & Zisserman 2003) can represent all image motions when the scene is planar and the camera follows a pin-hole perspective model. Because landing sites are typically flat and spacecraft cameras can be designed to have minimal radial distortion, a homography is sufficient for representing image motion during landing.

Given this insight, our approach to this problem is to represent a located safe site as a set of features. Features are then tracked through a stream of images using a homographic tracking algorithm to account for scale, rotation and perspective changes between images. The algorithm then combines the homography transform from the tracking and an altitude measurement from an altimeter, to compute the position of the lander with respect to the landing site. The steps in the tracking algorithm are as follows. First features are selected in the (initial) image in which the safe landing site was detected (the Shi Tomasi detector (Benedetti & Perona, 1998) is used as described above for hazard detection and avoidance). These features will be tracked through the entire stream of images and consequently the image position of the landing site in each subsequent image can be computed. The next step tracks the selected features in the initial image into the current image in the stream using multi-resolution normalized correlation. This tracking algorithm can only accommodate changes in translation, so there must be small motion between images. Given the feature tracks, an homography transformation is computed that maps the current features back into the initial image. Then each feature in the current image is locally warped using the current homography transformation and these warped features are correlated with the features in the initial image. If the correlation is high, then the feature is kept; otherwise, it is discarded. The remaining features in the current image are then tracked with normalized correlation into the next image. The homography transformation is computed between the next image and the initial image and the process repeats.

Eventually, the number of features tracked will fall below a threshold. At this point, a completely new set of features is selected in the current image and the tracking algorithms is started over. This approach will result in a gradual growth in the landing site position error, but the homographic tracking reduces this growth significantly when compared to a translation only tracker.

Tracking at video rates (30Hz) has been demonstrated with this algorithm on a 400 MHz R12000 processor. We have demonstrated tracking through a 360 degree rotation about the camera optical axis and 2x changes in scale. Figure 9 shows the result of tracking a landing site through 360 frames (72 seconds) of parachute drop test imagery.

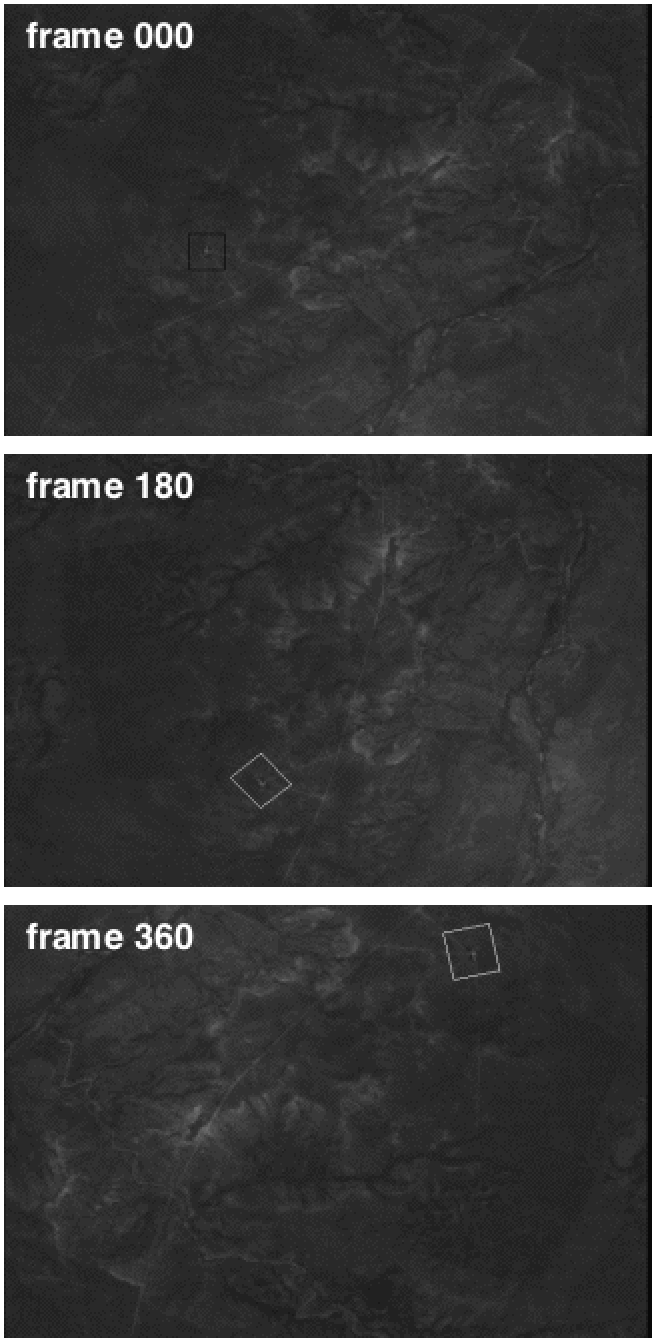

Figure 9 Landing site tracking through 360 frames of parachute drop test imagery

5.3 Augmenting Inertial Navigation with Image-Based Motion Estimation

To land safely and precisely on planetary bodies, accurate and robust state estimation during the descent phase is necessary. Towards this end, we have developed a new approach for improved state estimation by augmenting traditional inertial navigation techniques with Image-Based Motion Estimation (IBME). A Kalman filter that processes rotational velocity and linear acceleration measurements provided from an IMU has been enhanced to accommodate relative pose measurements from the IBME. In addition to increased state estimation accuracy, IBME convergence time is reduced while robustness of the overall approach is improved.

5.3.1 Inertial Sensing and Navigation

Inertial Navigation Systems (INS) have been used for decades for estimating the position and orientation of manned or autonomous vehicles such as aircrafts (Farell, Givargis & Barth, 2000) and spacecrafts (Wertz, 1978), (Lefferts et al., 1982). In recent years, similar systems were employed for localization of autonomous ground vehicles (Sukkarieh, Nebot & Durrant-Whyte, 1999), (Roumeliotis et al., 1999). The core of most INS is an Inertial Measurement Unit (IMU) comprised of 3-axial accelerometers and gyroscopes. Appropriate integration of their signals provides estimates of the location of the vehicle. The quality of these estimates depends primarily on the accuracy and noise profile of the IMU. Such systems can track very accurately sudden motions of short duration but their estimates quickly deteriorate during longer traverses due to the noise that contaminates the IMU signals.

In order to sustain positioning accuracy, inertial navigation systems usually include additional sensors (such as compass, inclinometers, star trackers, GPS, deep space network radio signal receivers, etc.) that provide periodic absolute attitude and position measurements. Since such sensors cannot be used during the Entry, Descent, and Landing (EDL) phase of a spacecraft, we focus on designing INS systems that use instead measurements of displacement (relative position and attitude) provided by a single camera and a laser altimeter. The challenge in this case is that most INS systems cannot directly process *relative* pose measurements unless these are expressed as local velocity measurements and processed as such. As it is discussed in detail in (Roumeliotis, 2001) this can be problematic, especially in cases where the relative pose measurements are available at a lower rate than the IMU signals. For this reason we have developed a variant of a 6DOF Kalman filter that is capable of optimally fusing inertial measurements from the IMU with displacement estimates provided by an IBME feature tracking algorithm.

A fundamental shortcoming of all IBME algorithms is the inability to solve for the magnitude of translational motion ||*t*||, so the output of motion estimation is a 5DOF motion composed of a unit vector describing the direction of heading and the rotation between views. As described in (Johnson & Matthies, 1999) laser altimetry can be combined with the 5DOF motion estimate to compute the complete 6DOF motion of the vehicle. This is also the case in the work presented in this paper. For each image pair, the output sent to the Kalman filter from the IBME is the relative pose measurement $[z_p{}^T z_q{}^T]^T$ (z_p : position, z_q : attitude) and its corresponding covariance R_r .

5.3.2 Fusion of Inertial and Relative Sensor data

In this section, we derive the equations for the modified Kalman filter that processes the *relative pose* measurements from the IBME algorithm. Since our formulation is based on sensor modeling, we use the *indirect* form of the Kalman filter that estimates the errors in the estimated states instead of the states themselves. The interested reader is referred to (Lefferts et al., 1982) , (Roumeliotis et al., 1999), (Roumeliotis, 2000) for a detailed description of the advantages of the Indirect KF vs. the Direct KF.

5.3.2.1 Relative Pose Measurement Error

The Indirect Kalman filter estimates the errors in attitude $\delta\vec{\theta}$, gyroscopes' biases $\Delta\vec{b}_g$, velocity $\Delta\vec{v}$, accelerometers' biases $\Delta\vec{b}_a$, and position $\Delta\vec{p}$. The error state vectors estimated by the filter at times t_k and t_{k+m} for i = 1, 2 are: $\Delta x_i = [\delta\vec{\theta}_i^T, \Delta\vec{b}_{gi}^T, \Delta\vec{v}_i^T, \Delta\vec{b}_{ai}^T, \Delta\vec{p}_i^T]^T$.

The errors in the relative position and attitude (pose) measurements calculated in Eqs. (11) and (19) (Appendix) are:

$$(3)\ \ \Delta\tilde{z}_{k+m} = \begin{bmatrix} \Delta z_p \\ \Delta\tilde{z}_q \end{bmatrix} = \chi \begin{bmatrix} \Delta z_p \\ \Delta z_q \end{bmatrix} = \Gamma[D_1\ D_2]\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \chi\, n_r = H\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \tilde{n}_r$$

with

$$\Gamma = \begin{bmatrix} {}^G_1C^T(\hat{q}_1) & 0 \\ 0 & {}^G_1C^T(\hat{q}_1) \end{bmatrix} \quad , \quad D_1 = \begin{bmatrix} \lfloor {}^G\hat{p}_{1,2} \rfloor & 0 & 0 & 0 & -I \\ (\frac{1}{2}){}^G_1C^T(\hat{q}_1) & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & I \\ (-\frac{1}{2}){}^G_2C^T(\hat{q}_2) & 0 & 0 & 0 & 0 \end{bmatrix} \quad ,$$

$$\chi = \begin{bmatrix} I_{3\times3} & 0_{3\times4} \\ 0_{3\times3} & \Xi^T({}^1_2\hat{q}) \end{bmatrix}$$

Both noise vectors n_r and $\tilde{n}_r$ are assumed to be zero-mean white Gaussian processes with

$$R_r = E[n_r n_r^T] = \begin{bmatrix} R_{pp} & R_{pq} \\ R_{qp} & R_{qq} \end{bmatrix}, \quad \tilde{R}_r = E[\tilde{n}_r\tilde{n}_r^T] = \chi R_r \chi^T$$

As it is evident from Eq. (3), the relative pose measurement error is expressed in terms of the current $\Delta x_2 = \Delta x(t_{k+m})$ and the previous $\Delta x_1 = \Delta x(t_k)$ error-state of the system. Therefore the Kalman filter state vector has to be appropriately augmented to contain *both* these state estimates. Note that t_k and t_{k+m} are the time instants when the two images processed by the IBME were recorded and thus the relative pose (motion estimate) measurement provided by the IBME corresponds to the time interval [t_k, t_{k+m}].

5.3.2.2 Augmented-State Propagation

If $\Delta x_{k/k}$ is the error-state estimate at time t_k (when the first image was recorded) we augment the state vector with a second copy of this estimate:

$$\Delta\breve{x} = \begin{bmatrix} \Delta x_{k/k}^T & \Delta x_{k/k}^T \end{bmatrix}^T$$

Since initially, at time t_k, the two versions of the error-state are identical, the covariance matrix for the augmented system will be:

$$\Delta\breve{P} = \begin{bmatrix} P_{kk} & P_{kk} \\ P_{kk} & P_{kk} \end{bmatrix}$$

where P_{kk} is the covariance matrix for the (error) state of the vehicle at time t_k. In order to conserve the estimate of the state at t_k, necessary for evaluating the relative pose measurement error at t_{k+m}, during this interval, only the second copy of the state estimate is propagated while the first one remains stationary. The propagation equation for the augmented system is:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x \end{bmatrix}_{k+1/k} = \begin{bmatrix} I & 0 \\ 0 & F_{k+1} \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x \end{bmatrix}_{k/k} + \begin{bmatrix} 0 \\ G_{k+1} \end{bmatrix}\vec{w}_k \text{ or } \Delta\breve{x}_{k+1/k} = \breve{F}_{k+1}\Delta\breve{x}_{k/k} + \breve{G}_{k+1}\vec{w}_k$$

where Δx_1 is the static copy of the error state of the vehicle.[2] The covariance of the augmented system is propagated and after m steps it is:

$$(4)\quad \breve{P}_{k+m/k} = \begin{bmatrix} P_{kk} & P_{kk}\mathcal{F}^T \\ \mathcal{F}P_{kk} & P_{k+m/k} \end{bmatrix}$$

where $\mathcal{F} = \prod_{i=1}^{m} F_{k+i}$ and $P_{k+m/k}$ is the propagated covariance of the evolving state at time t_{k+m}.

5.3.2.3 Update Equations

When the relative pose measurement is received, the covariance matrix for the residual is computed as:

$$(5)\quad S = H\breve{P}_{k+m/k}H^T + \tilde{R}_r$$

where $\tilde{R}_r = \chi R_r \chi^T$ is the adjusted covariance for the relative pose measurement and R_r is the initial covariance of this noise as calculated by the IBME algorithm. We define the pseudo-residual covariance matrix as $\tilde{S} = \Gamma^{-1}S\Gamma$ and by substituting from Eqs. (4) and (5):

$$\tilde{S} = D_1P_{kk}D_1^T + D_2\mathcal{F}P_{kk}D_1^T + D_1P_{kk}\mathcal{F}^T D_2^T + D_2P_{k+m/k}D_2^T + \mathcal{R}_r$$

where $\mathcal{R}_r = \Gamma^{-1}\tilde{R}_r\Gamma$. The updated covariance matrix is:

$$(6)\quad \begin{aligned} \breve{P}_{k+m/k+m} &= \breve{P}_{k+m/k} - \breve{P}_{k+m/k}H^T S^{-1} H\breve{P}_{k+m/k} \\ &= \breve{P}_{k+m/k} - \begin{bmatrix} P_{kk}D_1^T + P_{kk}\mathcal{F}^T D_2^T \\ \mathcal{F}P_{kk}D_1^T + P_{k+m/k}D_2^T \end{bmatrix} \tilde{S}^{-1} \begin{bmatrix} D_1P_{kk} + D_2\mathcal{F}P_{kk} & D_1P_{kk}\mathcal{F}^T + D_2P_{k+m/k} \end{bmatrix} \end{aligned}$$

The updated covariance matrix for the new state of the vehicle will be (lower-right diagonal submatrix):

$$P_{k+m/k+m} = P_{k+m/k} - (\mathcal{F}P_{kk}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}(D_1P_{kk}\mathcal{F}^T + D_2P_{k+m/k})$$

The Kalman gain is calculated as:

$$(7)\quad K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \breve{P}_{k+m/k}H^T S^{-1} \text{ with } K_2 = (\mathcal{F}P_{kk}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}\Gamma^T$$

The residual is computed as:

$$\tilde{r}_{k+m} = \Delta\tilde{z}_{k+m} = \chi\Delta z_{k+m} = \begin{bmatrix} z_p - \hat{z}_p \\ \Xi^T({}^{1}_{2}\hat{q})(z_q - \hat{z}_q) \end{bmatrix}$$

where z_p, z_q are the relative position and orientation measurements provided by the IBME, $\hat{z}_p = {}^{1}\hat{p}_2 = {}^{G}_{1}C^T(\hat{q}_1)({}^{G}\hat{p}_2 - {}^{G}\hat{p}_1)$, $\hat{z}_q = {}^{1}_{2}\hat{q}$, and $\Xi^T({}^{1}_{2}\hat{q})\hat{z}_q = \Xi^T({}^{1}_{2}\hat{q}){}^{1}_{2}\hat{q} = 0_{3\times 1}$.

Thus

$$\tilde{r}_{k+m} = \begin{bmatrix} z_p - {}^{G}_{1}C^T(\hat{q}_1){}^{G}\hat{p}_{1,2} \\ \Xi^T({}^{1}_{2}\hat{q}))z_q \end{bmatrix}$$

Finally, the updated augmented state is computed as:

[2] The discrete-time state and system noise propagation matrices F_{k+1}, G_{k+1} are described in detail in (Wertz, 1978), (Friedland, 1978), and (Roumeliotis, 2000).

$\breve{x}_{k+m/k+m} = \breve{x}_{k+m/k} + K\tilde{r}_{k+m}$

From Eq. (7) the (evolving) state will be updated as:

$x_{k+m/k+m} = x_{k+m/k} + (\mathcal{F}P_{kk}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}(\mathcal{Z}_{k+m} - \begin{bmatrix} {}^G\hat{p}_{1,2} \\ 0 \end{bmatrix})$, where $\mathcal{Z}_{k+m} = \begin{bmatrix} {}_1^GC^T(\hat{q}_1)z_p \\ {}_1^GC^T(\hat{q}_1)\Xi^T({}_2^1\hat{q})z_q \end{bmatrix}$, is the modified measurement of the relative displacement (pose) expressed in global coordinates. The quantities $\hat{q}_1^{-1}$, ${}_2^1\hat{q} = {}_G^1\hat{q} \otimes {}_G^2\hat{q}^{-1} = \hat{q}_1 \otimes \hat{q}_2^{-1}$, and $\hat{p}_{1,2} = {}^G\hat{p}_2 - {}^G\hat{p}_1$ are computed using the previous and current state estimates from the filter. Note that the current state estimates at time t_{k+m} are calculated by propagating the previous state estimates at time t_k using the rotational velocity and linear acceleration measurements from the IMU.

The same process is repeated every time (for $\lambda = 1,2,...$) a new set of relative pose measurements $z(t_{k+\lambda m}) = \left[z_p^T(t_{k+\lambda m}) \quad z_q^T(t_{k+\lambda m})\right]$ becomes available. The previous treatment makes the assumption that the measurements $z(t_{k+\lambda m})$ are mutually independent, i.e., $E\{z(t_{k+\lambda_i m})z^T(t_{k+\lambda_j m})\} = 0$. If the IBME algorithm uses the same set of features from an intermediate image to track the pose of the vehicle through two consecutive steps then these measurements are usually loosely correlated:

$E\{z(t_{k+\lambda m})z^T(t_{k+(\lambda+1)m})\} \neq 0$, $E\{z(t_{k+\lambda m})z^T(t_{k+(\lambda+1)m})\} << E\{z(t_{k+\lambda m})z^T(t_{k+\lambda m})\}$.

If this is not the case, the correlations have to be explicitly addressed by the estimation algorithm. Alternatively, if two disjoint subsets of features are used from the intermediate image to track features in the previous and the following ones, then the IBME motion estimates will be independent.

5.3.3 Vision Augmented Inertial Navigation Experiments

Experiments were performed on the GT and the AHT, but were only successfully demonstrated on the GT. We attached the AHT avionics package containing the sensors, onboard computers, and battery power to the gantry. Only the IMU and CCD camera were used for these experiments. The laser altimeter will be integrated into the approach in the future. Instead, a synthetic measurement of distance from the ground was provided from the gantry with the addition of noise. Also, at the time these experiments were performed, a lower quality IMU, the Crossbow DMU-VGX, was used. Since these experiments were performed, we have upgraded the AHT avionics with the higher performance ISIS IMU described earlier.

In the results presented here the motion of the vehicle is tracked after it has been accelerated to a speed of v = [42.7 -42.7 0]^T mm/sec at t = 5.5 sec. For the rest of the time the vehicle follows a straight line, almost constant velocity, trajectory until time t = 18 sec when it decelerates to a full stop at t = 19.6 sec. In order to extract the actual body acceleration during this motion, the local projection of the gravitational acceleration vector ${}^L\vec{g} = {}_L^GC(q)\,{}^G\vec{g}$ has to be subtracted from the accelerometer signals. Even small errors in the attitude estimate $\hat{q}$ can cause significant errors in the calculation of the actual body accelerations. This is more prevalent during slow motions with small body accelerations as the ones during this experiment. The estimated velocities and positions through the integration of the IMU are susceptible to large errors due to the magnitude of the gravitational acceleration compared to the minute body accelerations that the vehicle experiences during its actual motion. With ${}^L\vec{g}$ being the dominant acceleration in the measured signals, error analysis based on the propagation equations of the IMU integrator has shown that even for small attitude errors such as $\delta\theta = 1°$ the errors in position can grow as $\Delta p_x \cong \|g\| \; \delta\theta \; t^2 = 171t^2$ mm while the vehicle only moves as $p_x \cong v_x t = 42.7t$ mm. This is evident in Figure 10 where the position estimates calculated by appropriate integration of the IMU signals are valid for only a short period of time before the errors grow unbounded (e.g., for the y component of position the final error is over 1500 mm). Note also that during this relatively slow motion the integration of the IBME estimates provides significantly better results with only small errors introduced at the beginning of the experiment (end of the acceleration phase).

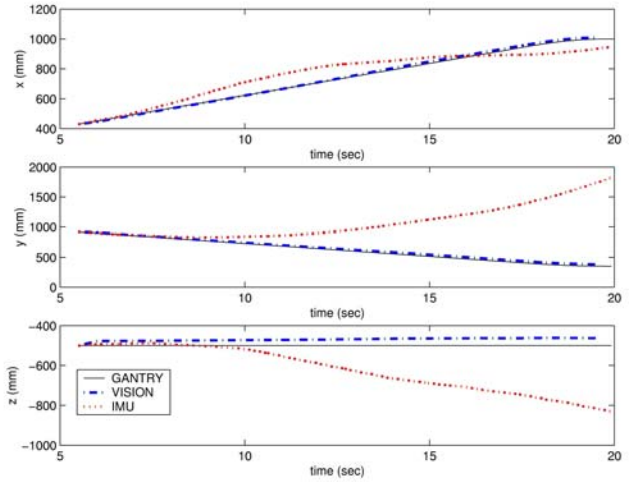

Figure 10 Trajectories estimated by integration of (i) the relative pose measurements provided by the IBME (vision) algorithm, (ii) the IMU signals, and (iii) the gantry

By combining the IBME relative pose measurements with the IMU signals within the Kalman filter, positioning errors are substantially reduced. The vehicle trajectory estimated by the KF follows more closely the ground truth path (recorded by the gantry) when compared to the trajectories estimated previously by either the IMU or the IBME. The position estimates for (*x* - *y*) and (*z* - *t*) are shown in Figure 11 and Figure 12(a), respectively. The average (absolute) errors in these estimates were $|\Delta p|_{avg} = [4.5 \quad 4.7 \quad 4.2]^T$ mm for the KF, $|\Delta p|_{avg} = [17.4 \quad 41.4 \quad 29.9]^T$ mm for the IBME and $|\Delta p|_{avg} = [53.5 \quad 464.7 \quad 126.1]^T$ mm for the IMU.

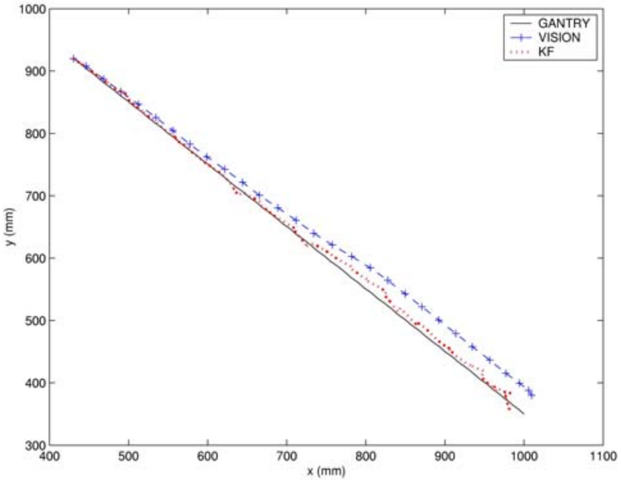

Figure 11 Trajectories estimated by (i) the integration of the relative pose measurements provided by the IBME (vision) algorithm, (ii) the Kalman filter, and (iii) the gantry (x-y)

The availability of intermittent (relative) positioning information enables the filter to also update the estimates of the biases in the accelerometer and gyroscope signals as depicted in Figure 12(b). This in effect reduces the errors in the linear acceleration and rotational velocity measurements and allows the KF estimator to operate for longer periods of time before an external absolute pose measurement is necessary. Finally, we should note that since the information from the IBME corresponds to *relative* and not *absolute* pose measurements the filter estimated errors will continue to grow, albeit at a lower rate. This rate is determined by the frequency and quality of the IBME measurements.

6. Current Work

Our current work includes new EDL technology research and development and an effort aimed at improving the AHT control system while giving it the ability to emulate spacecraft landing dynamics.

6.1 Coupled Vision and Inertial Navigation for Pin-Point Landing

Future space missions will require landing much nearer to desired locations than is possible with current EDL technology. For example, the Mars Exploration Rovers had landing error ellipses with axes on the order of 100's of kilometers in size, while some future missions will require landing within 10's or 100's of *meters.* This is known as Pin-Point Landing (PPL) and we are developing a coupled vision and inertial navigation (CVIN) system to enable PPL.

The objectives of this work are to develop the real-time vision and estimation algorithms for PPL and then demonstrate them on the JPL GT and AHT. The central component of the algorithms is a Kalman filter estimator that integrates inertial measurements with vision-inferred measurements of spacecraft pose (position and attitude) and pose displacement. The KF updates the estimates for the state of the vehicle and the biases in the inertial sensors. This is an extension of our work described above that fuses just inertial and relative pose information. The extension is in the fusion of updates of absolute pose estimates from a vision system that matches features extracted from imagery against a database of known landmarks.

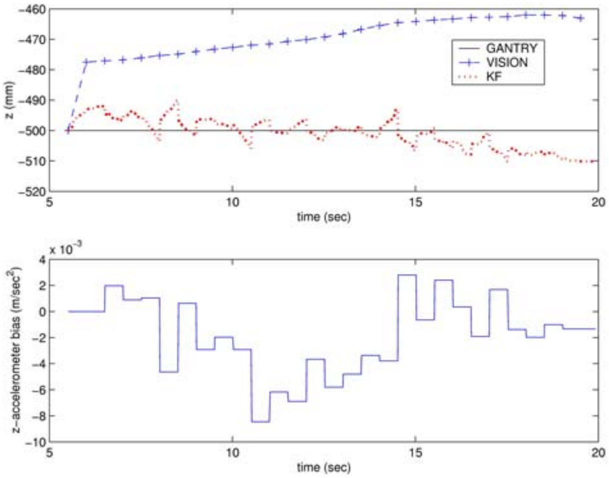

Figure 12 (a) Trajectories estimated by (i) the integration of the relative pose measurements provided by the IBME (vision) algorithm, (ii) the Kalman filter, and (iii) the gantry (z-t). (b) z-axis accelerometer bias estimate

We are one year into a three year task and so far have been focused on KF and vision algorithm development. The KF and vision algorithms have been tested and demonstrated independently in simulation. We have ported the algorithms to our AHT avionics and have reproduced the results shown in simulation using canned input data. We are in the process of integrating the KF and vision algorithms, interfacing the algorithms to real sensor data instead of canned input data and readying the GT to support algorithm testing. Testing and additional algorithm development will continue on the GT throughout year two and three of the task. In year three, all of the algorithms will be field-tested on the AHT.

6.2 Advanced Helicopter Control

To date, research on the AHT has been on the development and testing of vision and state estimation algorithms. We have devoted little effort to the control system itself. This work, and the work described in the next section, is part of the same task meant to increase the capabilities of the AHT.

This part of the task involves replacing the current control system with its PI and PD loops and hand-tuned gains with a model-based nonlinear controller. The goal is to make an autonomous platform that can be controlled in a larger flight envelope, e.g., higher lateral speeds with more aggressive banking and turning.

6.3 Emulation of Spacecraft Landing Dynamics

Currently, the AHT flies like a helicopter. It is given helicopter input commands, e.g., collective pitch, and it is allowed to fly within its dynamic constraints. However, we would like to make the AHT a more realistic testbed for the development and testing of algorithms meant for spacecraft landing. Therefore, we are starting an effort to emulate spacecraft landing dynamics with the AHT. This work will be built upon the advanced helicopter control work described above.

The controller will accept thruster inputs (like those on a spacecraft) and convert them into appropriate helicopter stick controls such that the resulting trajectory of the helicopter is close to the trajectory that would have been achieved by simply providing the same thruster inputs to a spacecraft. The approach relies on a simplified model of the spacecraft and helicopter dynamics. Preliminary results in simulation (Saripalli et al., 2002) indicate that the approach is feasible, with tracking accuracies in the order of 5 m distance between the desired and actual points along the trajectory. Admittedly, the AHT is only able to coarsely approximate spacecraft motion since a helicopter must roll or pitch to move laterally, while a well-placed spacecraft thruster can move the craft laterally without rolling or pitching first. During the course of this research we will seek to determine whether or not a helicopter can be an effective surrogate for the spacecraft landing domain.

7. Conclusions

The purpose of the JPL Autonomous Helicopter Testbed is to provide a platform for the research and development of technologies for planetary exploration. Since its creation in 2001, the AHT has shown itself to be capable of meeting that role and covering a subset of the technology testspace as described earlier. To date, it has been used to develop and test EDL technologies, but it has characteristics that lend itself well to supporting other applications. The ability to hover and move in any direction provides a useful platform for testing small body exploration technologies such as image-based position estimation and station-keeping around comets and asteroids. Our ongoing work to improve the helicopter control system and emulate spacecraft dynamics with the helicopter will make the AHT a more capable and valuable platform in the future.

Acknowledgements

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The Explorations Systems Research & Technology (ESR&T) Development Program, NASA Exploration Systems Mission Directorate funded our work in safe and precise landing, the Mars Technology Program, NASA Science Mission Directorate funds our CVIN work and the NASA/JPL Director’s Research and Development Fund is sponsoring our advanced helicopter control and emulation of spacecraft landing dynamics work. Additional funding for this work was provided by the Mars Technology program for the project “Coupled Vision and Inertial Navigation for Pinpoint Landing”. S.I. Roumeliotis’ work was also funded by NSF (ITR-0324864, MRI-0420836). We would like to thank Chuck Bergh for designing and

building the JPL AHT Avionics and the JPL GT, Srikanth Saripalli for his assistance in software integration and field-testing and Doug Wilson and Alan Butler, our helicopter safety pilots.

8. Appendix

In what follows, we assume that at time t_k the vehicle is at position ${}^{G}p(t_k) = {}^{G}p_1$ with (quaternion) attitude ${}^{1}_{G}q(t_k) = q_1$ and after m steps it has moved to position ${}^{G}p(t_{k+m}) = {}^{G}p_2$ with attitude ${}^{2}_{G}q(t_{k+m}) = q_2$. Frames $\{G\}$, $\{1\}$, and $\{2\}$ are the frames of reference attached to the vehicle at times t_0, t_k and t_{k+m} correspondingly.

8.1.1.1 Relative Position Measurement Error

The relative position measurement z_p between the two locations {1}, and {2} can be written as:

(8) $z_p = {}^{1}p_2 + n_p = {}^{G}_{1}C^T(q)({}^{G}p_2 - {}^{G}p_1) + n_p$

where n_p is the noise associated with this measurement assumed to be a zero-mean white Gaussian process with covariance $R_p = E[n_p n_p^T]$. ${}^{G}_{1}C(q)$ is the rotational matrix that expresses the orientation transformation between frames $\{G\}$ and $\{1\}$.

If Δp_i is the error in the estimate of position p_i and δq is the error in the estimate of attitude q then:[3]

$$p_i = \Delta p_i + \hat{p}_i,\ i = 1,2,\ q = \delta q \otimes \hat{q}$$

Eq. (8) can now be written as:

(9) $z_p = {}^{G}_{1}C^T(\delta q \otimes \hat{q})({}^{G}\hat{p}_2 + \Delta p_2 - {}^{G}\hat{p}_1 - \Delta p_1) + n_p$

The estimated relative position measurement is:

(10) $\hat{z}_p = {}^{G}_{1}C^T(\hat{q})({}^{G}\hat{p}_2 - {}^{G}\hat{p}_1) = {}^{G}_{1}C^T(\hat{q})\,{}^{G}\hat{p}_{1,2}$

The error in the relative position measurement is defined as:

$$\Delta z_p = z_p - \hat{z}_p$$

Substituting from Eqs. (9), (10) and employing the small angle approximation $\delta q \cong [\delta\vec{q}\ \ 1]^T \cong [\frac{1}{2}\delta\vec{\theta}_1\ \ 1]^T$, it can be shown (Roumeliotis, 2001) that:

(11) $\Delta z_p \cong {}^{G}_{1}C^T(\hat{q}_1)\lfloor {}^{G}\hat{p}_{1,2} \rfloor \delta\vec{\theta}_1 + {}^{G}_{1}C^T(\hat{q}_1)\Delta p_2 - {}^{G}_{1}C^T(\hat{q}_1)\Delta p_1 + n_p$

where $\lfloor\ \ \rfloor$ denotes the cross-product matrix of a vector

$$\lfloor \vec{V} \rfloor = \begin{bmatrix} 0 & -V_3 & V_2 \\ V_3 & 0 & -V_1 \\ V_2 & V_1 & 0 \end{bmatrix}$$

In Eq. (11) the first term expresses the effect of the orientation uncertainty at time t_k on the quality of the estimated measurement. Note that if at time t_k there was no uncertainty about the orientation of the vehicle

[3] Note that from here on, q refers to q_1 and δq refers to δq_1. We have also dropped the vector symbol from the real, measured, estimated, and error position to simplify notation.

(i.e., $\delta\vec{\theta}_1 = 0$), the error in the relative position measurement would depend only on the errors in the estimates of the previous and current position of the vehicle.

8.1.1.2 Relative Attitude Measurement Error

The relative attitude measurement error between the two locations {1}, and {2} is:

(12) $$\Delta z_q = z_q - {}^1_2\hat{q} = {}^1_2 q + n_q - {}^1_2\hat{q}$$

where n_q is the relative attitude measurement noise. We assume that n_q is a zero-mean white Gaussian process with covariance $R_q = E[n_q n_q^T]$. Since ${}^1_2 q = {}^1_0 q \otimes {}^2_0 q^{-1} = q(t_k) \otimes q^{-1}(t_{k+m}) = q_1 \otimes q_2^{-1}$ and $q_i = \delta q_i \otimes \hat{q}_i$, $i = 1,2$, ${}^1_2 q$ can be written as:

(13) $${}^1_2 q = \delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1}$$

By substituting Eq. (13) in Eq. (12) we have:

(14) $$\Delta z_q = \delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1} - {}^1_2\hat{q} + n_q$$

with

$$\delta q_1 = \begin{bmatrix} \delta\vec{q}_1 \\ \delta q_{1_4} \end{bmatrix}, \quad \delta q_2 = \begin{bmatrix} \delta\vec{q}_2 \\ \delta q_{2_4} \end{bmatrix}, \quad \delta q_2^{-1} = \begin{bmatrix} -\delta\vec{q}_2 \\ \delta q_{2_4} \end{bmatrix}$$

In order to simplify the notation we set:

$${}^1_2\hat{q} = q = \begin{bmatrix} \vec{q} \\ q_4 \end{bmatrix}$$

For small attitude estimation errors δq_1 and δq_2 we make the following approximations: $\delta q_{1_4} \cong 1$, $\delta q_{2_4} \cong 1$, $\delta\vec{q}_1 << 1_{3\times 1}$, $\delta\vec{q}_2 << 1_{3\times 1}$. The first term in Eq. (14) can be written as:

(15) $$\delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1} \cong \begin{bmatrix} \vec{q} + q_4(\delta\vec{q}_1 - \delta\vec{q}_2) + \lfloor \vec{q} \rfloor(\delta\vec{q}_1 + \delta\vec{q}_2) \\ q_4 - \vec{q}^T(\delta\vec{q}_1 - \delta\vec{q}_2) \end{bmatrix}$$

By multiplying both sides of Eq. (14) with the matrix

(16) $$\Xi^T({}^1_2\hat{q}) = \Xi^T(q) = [(q_4 I - \lfloor \vec{q} \rfloor) \quad -\vec{q}]$$

we define the *vector* attitude measurement error as:

(17) $$\Delta\tilde{z}_q = \Xi^T({}^1_2\hat{q})\Delta z_q = \Xi^T({}^1_2\hat{q})({}^1_2 q + n_q) - 0 = \Xi^T({}^1_2\hat{q})(\delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1}) + \Xi^T({}^1_2\hat{q})n_q$$

Substituting from Eqs. (15), (16), the first term in the previous expression can be written as:

(18) $$\Xi^T({}^1_2\hat{q})(\delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1}) \cong \delta\vec{q}_1 - {}^1_2 C({}^1_2\hat{q})\delta\vec{q}_2$$

Eq. (17) is now expressed as:

(19) $$\Delta\tilde{z}_q \cong \frac{1}{2}(\delta\vec{\theta}_1 - {}^1_2 C({}^1_2\hat{q})\delta\vec{\theta}_2) + \tilde{n}_q$$

where we have used the small angle approximation $\delta\vec{q}_i = \frac{1}{2}\delta\vec{\theta}_i, i = 1,2$ and $\tilde{n}_q = \Xi^T({}^1_2\hat{q})n_q$ with

$$\tilde{R}_q = E[\tilde{n}_q \tilde{n}_q^T] = \Xi^T({}^1_2\hat{q}) R_q \Xi({}^1_2\hat{q})$$

References

1. Amidi, O., Kanade T. & Fujita K. (1999). A Visual Odometer for Autonomous Helicopter Flight. Robotics and Autonomous Systems, 28, 185-193.
2. Azarbayejani, A. & Pentland A. (1995). Recursive Estimation of Motion, Structure, and Focal Length. IEEE Pattern Analysis and Machine Intelligence, 17(6), 562-575.
3. Benedetti, A. & Perona, P. (1998, June). Real-Time 2-D Feature Detection on a Reconfigurable Computer. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ‘98), Santa Barbara, CA, USA.
4. Bosse, M., Karl, W.C., Castanon, D. & DiBitetto, P. (1997, November). A Vision-Augmented Navigation System. Paper presented at the IEEE Conference on Intelligent Transportation Systems (ITS ’97), Boston, MA, USA.
5. Conway, A. (1995) Autonomous Control of an Unstable Helicopter Using Carrier Phase GPS Only. Ph.D. Thesis, Stanford University, Stanford, CA.
6. Corke, P. (2004). An Inertial and Visual Sensing System for a Small Autonomous Helicopter. Journal of Robotic Systems, 21, 43-51.
7. Elfes, A., Montgomery, J., Hall, J., Joshi, S., Payne, J. & Bergh, C. (2005, September). Autonomous Flight Control for a Titan Exploration Aerobot. Paper presented at the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS ’05), Munich, Germany.
8. Farrell, J.A., Givargis, T.D. & Barth, M.J. (2000). Real-Time Differential Carrier Phase GPS-Aided INS. IEEE Transactions on Control Systems Technology, 8(4), 709-721.
9. Friedland, B. (1978). Analysis Strapdown Navigation Using Quaternions. IEEE Transactions on Aerospace and Electronic Systems, AES-14(5), 764-768.
10. Garcia-Padro, P., Sukhatme, G. & Montgomery, J. (2001). Towards Vision-Based Safe Landing for an Autonomous Helicopter. Robotics and Autonomous Systems, 38(1), 19–29.
11. Gavrilets, V. (2003). Autonomous Aerobatic Maneuvering of Miniature Helicopters. Ph.D. Thesis, Massachusetts Institute of Technology, Boston.
12. Hartley, R. & Zisserman, A. (2003). Multiple View Geometry in Computer Vision. Cambridge, UK: Cambridge University Press.
13. Hintze, J. (2004). Autonomous Landing of a Rotary Unmanned Aerial Vehicle in a Non-cooperative Environment Using Machine Vision. Master’s thesis, Brigham Young University, Provo, UT.
14. Johnson, A.E., Klumpp, A., Collier, J. & Wolf, A. (2002). Lidar-based Hazard Avoidance for Safe Landing on Mars. AIAA Journal Guidance, Control and Dynamics, 25(5).
15. Johnson, A.E. & Matthies, L.H. (1999, June). Precise Image-Based Motion Estimation for Autonomous Small Body Exploration. Paper presented at the 5th International Symposium On Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS ’99), Noordwijk, The Netherlands.
16. Johnson, A.E. & Skulsky, E.D. (2002, February). Descent Speed Testing of a Hazard Detection System for Safe Landing on Mars. Paper presented at the 25th AAS Guidance and Control Conference, Breckenridge, CO, USA.
17. Johnson, A.E., Willson, R., Goguen, J., Alexander, J. & Meller, D. (2005, April). Field Testing of the Mars Exploration Rovers Descent Image Motion Estimation System. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA ‘05), Barcelona, Spain.

18. Jun, M., Roumeliotis, S.I. & Sukhatme, G. (1999, October). State Estimation of an Autonomous Flying Helicopter. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '99), Seoul, Korea.
19. La Civita, M. (2003). Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.
20. Lefferts, E.J., Markley, F.L. & Shuster, M.D. (1982). Kalman Filtering for Spacecraft Attitude Estimation. Journal of Guidance, Control, and Dynamics, 5(5), 417-429.
21. Lefferts, E.J. & Markley, F.L. (1976). Dynamics Modeling for Attitude Determination. AIAA Paper 76-1910.
22. Meingast, M., Geyer, C. & Sastry, S. (2004, December). Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles. Paper presented at the IEEE Conference on Decision and Control (CDC '04), Atlantis, Paradise Island, Bahamas.
23. Montgomery, J. (1999). Learning Helicopter Control through "Teaching by Showing". Ph.D. thesis, University of Southern California, Los Angeles, CA.
24. Moravec, H. (1977, August). Towards Automatic Visual Obstacle Avoidance. Paper presented at the 5th International Joint Conference on Artificial Intelligence, Cambridge, MA, USA.
25. Ng, A., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E. & Liang, E. (2004, June). Inverted Autonomous Helicopter Flight via Reinforcement Learning. Paper presented at the International Symposium on Experimental Robotics (ISER '04), Singapore.
26. Nister, D., Noriditsky, O. & Bergen, J. (2004, June). Visual Odometry. Paper presented at the IEEE Conference Computer Vision and Pattern Recognition (CVPR'04), Washington, DC, USA.
27. Oliensis, J. (2002). Exact Two Image Structure from Motion. IEEE Pattern Analysis and Machine Intelligence, 24(12), 1618-33.
28. Roumeliotis, S. I. (2001). A Kalman Filter for Processing 3-D Relative Pose Measurements (Tech. Rep.), Pasadena, CA: California Institute of Technology, Robotics Laboratory, http://robotics.caltech.edu/~stergios/tech_reports/relative_3d_kf.pdf.
29. Roumeliotis, S.I. (2000). Robust Mobile Robot Localization: From Single-Robot Uncertainties to Multi-Robot Interdependencies. PhD thesis, University of Southern California, Los Angeles, CA.
30. Roumeliotis, S.I., Johnson, A.E. & Montgomery, J. (2002, May). Augmenting Inertial Navigation with Image-based Motion Estimation. Paper presented at the International Conference on Robotics and Automation (ICRA '02), Wahsington, DC, USA.
31. Roumeliotis, S.I., Sukhatme, G. & Bekey, G. (1999, May). Circumventing Dynamic Modeling: Evaluation of the Error-State Kalman Filter Applied to Mobile Robot Localization. Paper presented at the International Conference on Robotics and Automation, (ICRA '99), Detroit, MI, USA.
32. Russ, J.C. (1999). The Image Processing Handbook. London UK: CRC Press and New York, NY, USA: IEEE Press.
33. Saripalli, S., Montgomery, J. & Sukhatme, G., (2003). Visually-Guided Landing of an Unmanned Aerial Vehicle. IEEE Transactions on Robotics and Automation, 19(3), 371-81.
34. Saripalli, S., Sukhatme G. & Montgomery, J. (2002, October). A Testbed for Mars Precision Landing Experiments by Spacecraft Dynamics on a Helicopter. Paper presented at the IEEE/RSF International Conference on Intelligent Robots and Systems (IROS '02), EPFL, Switzerland.
35. Shakernia, O., Vidal, R., Sharp, C.S., Ma, Y. & Sastry, S.S. (2002, May). Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle. Paper presented at the International Conference on Robotics and Automation, (ICRA '02), Washington, DC. USA.
36. Sukkarieh, S., Nebot, E.M. & Durrant-Whyte, H.F. (1999). A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications. IEEE Transactions on Robotics and Automation, 15(3), 572-578 .

37. Weng, J., Ahuja, N., & Huang, T. (1993). Optimal Motion and Structure Estimation. IEEE Pattern Analysis and Machine Intelligence, 15(9), 864-884.

38. Wertz, J.R., editor (1973). Spacecraft Attitude Determination and Control, volume 73 of Astrophysics and Space Science Library. Dordrecht, The Netherlands: D. Reidel Publishing Company.