Full Mission Simulation of a Rotorcraft Unmanned Aerial Vehicle for Landing in a Non-Cooperative Environment

Colin Theodore
Steve Shelden
Dale Rowley
San Jose State Foundation
NASA Ames Research Center, CA

Weiliang Dai
UARC, UC Santa Cruz
NASA Ames Research Center, CA

Tim McLain
Brigham Young University
Provo, UT

Marc Takahashi
QSS, Computational Sciences Division
NASA Ames Research Center, CA

ABSTRACT

This paper describes the development and testing of an integrated simulation environment for the autonomous landing of a rotorcraft unmanned aerial vehicle at non-cooperative sites without the aid of GPS. The simulation includes a graphical display element that provides sensor modeling capabilities, a ground station user interface for mission setup and runtime, a vehicle dynamics model and control laws for a Yamaha RMAX, and a mission manager that provides autonomy functions and ties all of these elements together. The mission manager implements a landing procedure that incorporates stereo ranging and safe landing area determination algorithms to identify a suitable obstacle-free landing point, and monocular feature tracking with position estimation algorithms for navigation to the landing point without GPS. Results are reported from a performance evaluation of the individual machine vision algorithms using the simulation, and these results are compared with requirements for autonomous landing flight trials.

INTRODUCTION

Accurate, reliable autonomous landing of Rotorcraft Unmanned Aerial Vehicles (RUAVs) remains a challenging and important capability for operational systems to achieve greater mission flexibility, less operator involvement, and more rapid sortie turnaround. However, current technologies for the landing of UAVs are most often limited to using an external pilot, recovery net, or auto-land capability requiring landing site based instrumentation and/or radar. These technologies preclude UAVs from landing in un-prepared environments where the terrain profile is uncertain and possibly cluttered. In addition to this, in a cluttered environment such as an urban canyon, GPS signals may be intermittent (due to occlusion or jamming) and cannot be solely relied upon for uninterrupted guidance and navigation.

One method of landing RUAVs autonomously is to specifically prepare the landing site with instrumentation or markings. One such landing system in current use is the UAV Common Automatic Recovery System (UCARS) developed by Sierra Nevada Corporation [1]. Derivatives of this system include the Tactical Automatic Landing System (TALS) that is used by the US Army's Shadow UAV, and UCARS-V2 that has seen success with Northrop Grumman's RQ-8A Fire Scout. These systems use a ground-based millimeter-wave radar tracking system with a transponder mounted on the vehicle. The systems can be used to provide automatic landing only at cooperative sites, since both ground and vehicle-based equipment is required. Another approach is to mark the landing point with a pattern and then track the pattern with sensors and algorithms on the vehicle. This approach was taken by researchers at the Jet Propulsion Laboratory (JPL) [2, 3] who marked an "H" target on the landing pad and used machine vision to identify and track the target for landing. A combination of vision, Inertial Measurement Unit (IMU) and GPS measurements were used to navigate the helicopter to the cooperative landing pad. When considering non-cooperative landing sites, various methods are available for sensing the terrain, including stereo ranging and structure from motion, laser scanning, and millimeter-wave radar. A vision-based approach has also been used with a safe landing area detection algorithm running on the vehicle to identify safe landing points at non-cooperative sites [4]. Researchers at UC Berkeley have coupled a multiple view motion estimation algorithm

Presented at the American Helicopter Society 61st Annual Forum, Grapevine, TX, June 1-3, 2005.

with a helicopter controller to achieve autonomous landings [5].

This paper describes an integrated approach to the autonomous landing of RUAVs that was developed as part of the US Army's Precision Autonomous Landing Adaptive Control Experiment (PALACE) at NASA Ames Research Center. This approach uses machine vision technologies for the determination of a safe landing point in a non-cooperative environment, and to provide vehicle positioning information for navigation around the landing zone without GPS. Previous work on this program [6] independently demonstrated each of the core machine vision technologies, and validated their utility for autonomous landings of RUAVs in both simulation and flight.

The efforts described here expand on this previous work by combining the machine vision technologies into an integrated architecture that is implemented in simulation [7]. This integrated simulation environment includes: a graphical interface for an operator to setup, monitor and re-task the vehicle during the autonomous mission; a graphical simulation environment; Yamaha RMAX flight dynamics with inner-loop and waypoint navigation control modes [8]; Jet Propulsion Laboratory (JPL) stereo machine vision and monocular feature tracking algorithms [9]; a Monocular Position Estimation (MPE) algorithm; and a mission manager that is responsible for mission autonomy and handles coordination between the machine vision algorithms, the operator interface, and helicopter. The integrated simulation was constructed to evaluate the PALACE landing procedure, and to quantify the performance of the individual machine vision algorithms. It also provides a level of risk reduction when transitioning the landing system and machine vision algorithms to flight trials, which is the ultimate goal of the PALACE program.

The first section of this paper gives a brief description of the PALACE program. The next section details the integrated simulation environment, including the simulation architecture and the various simulation elements. The third section presents results from the integrated simulation, including stereo range mapping, Safe Landing Area Determination (SLAD), and monocular feature tracking evaluations. This same section also compares the measured performance of the vision algorithms against the requirements of the PALACE program and discusses their suitability for transition to autonomous landings in flight. The final section presents concluding remarks.

PALACE PROGRAM DESCRIPTION

The PALACE research program is a US Army Science and Technology Objective (STO) formulated to address some of the current limitations of landing RUAVs. These limitations include reliance on an external pilot, or auto-land system with ground-based instrumentation, and reliance on GPS for navigation around the landing zone. The program is targeted towards missions that require precision autonomous landings at unprepared sites for perch and stare surveillance, precision UAV supply delivery and Forward Area Arming and Refueling Point (FARP) operations.

PALACE is a three-year program, begun in FY02, that seeks to mature and integrate vision-based guidance and control technologies for the autonomous landing of RUAVs at non-cooperative landing sites, without the aid of GPS. The first year of the program defined the system architecture and independently validated that the core machine vision technologies could be used in both simulation and flight. The second year involved the construction of a full-mission simulation environment that includes realistic helicopter dynamics and controls, an operator interface, and a mission manager for the coordination of the machine vision technologies. The simulation environment was used to evaluate the performance of the machine vision algorithms, and assess their applicability for flight trials on a Yamaha RMAX. The evaluation of the machine vision algorithms within the integrated simulation is the focus of this report. The third year involves flight trials of the autonomous landing technologies on a Yamaha RMAX RUAV. The in-flight performance of the individual algorithms will be assessed, as well as the performance of the overall landing procedure. Final flights will demonstrate the complete landing system with the RMAX landing within obstacle fields without GPS.

Table 1. PALACE program performance metrics and objective values.

Quantitative Metric	Project Objective
Landing Site Size	< 6.25 m
Landing Surface Slope	< 15 deg
Landing Surface Roughness	< 10 cm
Landing Accuracy	< 1.25 m
Feature-Tracking Cycle Time	< 100 msec
SLAD Calculation Time	< 5 sec
SLAD Success Rate	> 98%

Table 1 lists the quantitative metrics and the target performance objectives for the PALACE program. The first three metrics are the target constraints on the landing site selection algorithm and are a function of the flight vehicle (for this paper, the target vehicle is the Yamaha RMAX). The following metric specifies the landing accuracy and accounts for the amount of drift in vehicle position during the vision-based descent. The next requirement specifies that the feature-tracking algorithms

should run with a processing time of less than 100 msec. to produce a position estimate at 10Hz. The final two objectives specify that the SLAD algorithms should run in under 5 seconds with an accuracy of greater than 98% in choosing a safe landing site. The ability of the machine vision algorithms to meet this set of objectives is discussed in this paper.

For demonstration purposes, a complete UAV mission was defined that consists of pre-flight planning, autonomous take-off and waypoint navigation, and autonomous landing at a non-cooperative site. This PALACE mission facilitated the development and testing of an integrated landing procedure and the evaluation of machine vision technologies that were selected as part of the landing procedure. Figure 1 shows a schematic of the complete PALACE mission from pre-flight planning to autonomous landing.

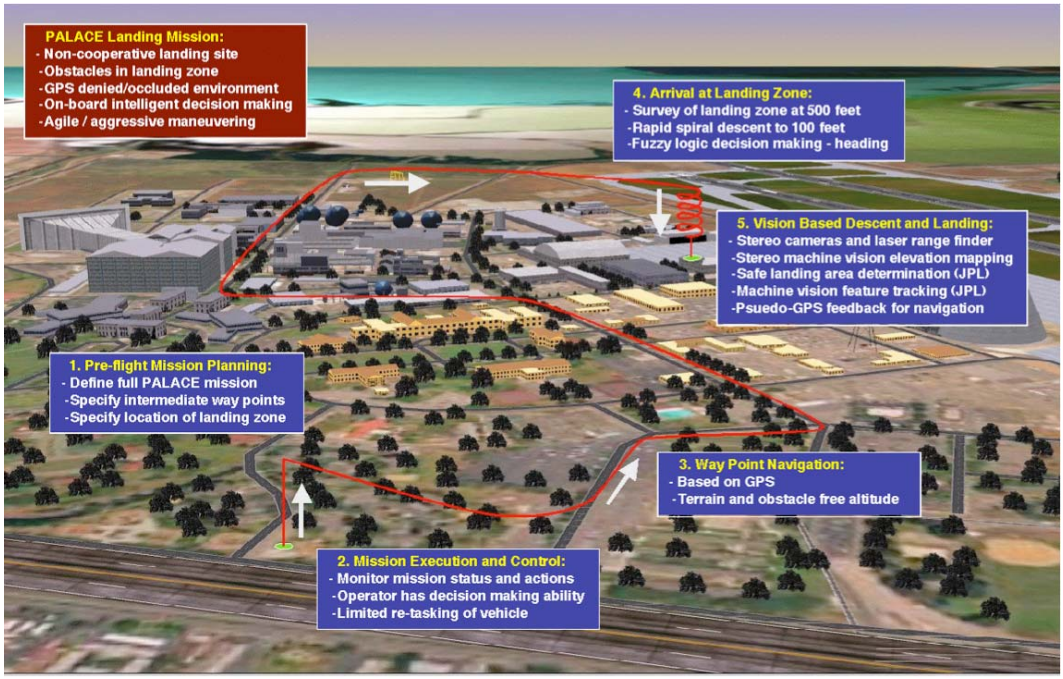

Figure 1. PALACE mission segments.

PALACE INTEGRATED SIMULATION

The PALACE integrated simulation environment was developed to combine and evaluate the various elements of the PALACE mission in a single simulation. The simulation allows an operator to plan and execute a complete mission, as well as monitor and change mission parameters during execution. The simulation also serves as an engineering tool for the evaluation and testing of the individual machine vision technologies, and also allows for the extraction of both qualitative and quantitative performance metrics. Finally, the integrated simulation allows for the assessment of the overall landing system to resolve any integration issues and provide a level of risk reduction when taking these technologies to flight.

Figure 2 shows a schematic of the PALACE integrated simulation architecture. Each element of this architecture is described in the following sections.

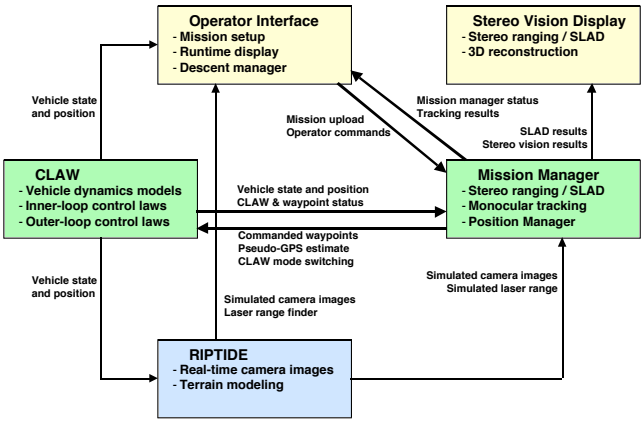

Figure 2. PALACE integrated simulation architecture.

RIPTIDE Simulation Environment

The Real-time Interactive Prototype Technology Integration/Development Environment (RIPTIDE) [10] is a product of the Army/NASA Rotorcraft Division at NASA Ames Research Center and provides a real-time 3-D

graphical environment. For the PALACE integrated simulation, RIPTIDE provides simulated camera images and a simulated laser range based on vehicle state and position information that is supplied by the CLAW block (see Figure 2). The camera images are used in the machine vision algorithms for stereo ranging and monocular tracking, and for display to the operator on the ground station. The laser range is used in the MPE algorithms and is used to compute an equivalent height-above-terrain measurement for display in the user interface.

Figure 3 shows a RIPTIDE screenshot from the PALACE integrated simulation environment. The parking lot shown in the figure was constructed to assist in the development and testing of the PALACE landing technologies. In an urban environment, parking lots are potentially suitable sites for landing since they are typically flat and fairly open. The parking includes cars, light poles, trees, and adjacent buildings, objects that are typically be encountered in a realistic landing scenario. The upper-right region of Figure 3 shows a number of objects, including cars, trucks, boxes, etc. that can easily be re-configured to change obstacle spacing and density. This provides for a variety of obstacle fields in which to evaluate the landing technologies.

Figure 3. Parking lot scenario in RIPTIDE.

CLAW

The target vehicle that will be used for the PALACE demonstrations is a modified Yamaha RMAX, which is the flight hardware used as part of the Autonomous Rotorcraft Project (ARP) [8]. The Yamaha RMAX is a small-scale helicopter with a rotor diameter of 3.12 m and an empty mass of 66 kg. The maximum payload mass is 28 kg.

The Yamaha RMAX operated by ARP is shown in Figure 4 and has been modified to include an avionics payload and various sensors. The avionics payload includes: a navigation and flight control computer, an experimental computer for vision processing, an IMU, a GPS receiver, and radio communications equipment. A pair of monochrome 640x480 resolution stereo cameras are mounted on vibration-isolated stub wing, that provides a stereo baseline of one meter (shown in Figure 5). Finally a Reigl laser range finder is mounted under the nose to provide accurate distance measurements.

Figure 4. ARP RMAX research aircraft.

Figure 5. ARP RMAX avionics payload.

The CLAW block of the integrated simulation (see Figure 2) contains the vehicle dynamics model as well as the inner- and outer-loop control laws. The vehicle dynamics model was obtained using CIFER® system identification techniques [11] from a set of hover flight test data collected on the ARP RMAX. This highly accurate vehicle model includes rigid body dynamics, as well as rotor and stabilizer bar dynamics. Sub-system dynamic models for the actuators (including non-linear rate and position limits) and sensors (gyros, accelerometers, etc.) are also included.

A model of the landing gear is included that allows the vehicle to take-off and land in the simulation environment. Environmental effects are built into the vehicle model and include steady winds as well as gusts and turbulence of varying severity. The outer-loop control system implemented in CLAW provides a means to command the vehicle to specified waypoints. The CLAW block provides the vehicle state and position information to the other elements of the simulation architecture.

PALACE Ground Station

The User Interface (UI) component of the PALACE integrated simulation was addressed by the Human Systems Integration (HSI) group of the Army/NASA Rotorcraft Division at NASA Ames Research Center. The HSI group designed two graphical UI modules to facilitate the operator interaction with the PALACE system.

The Mission Planner module (see Figure 6) allows an operator to plan, preview and save (for later execution), a complete PALACE mission. The Mission Planner module requires the operator to identify a nominal take-off location, optionally enter intermediate waypoints, designate the landing zone, and set a "rally point" to which the vehicle would fly under specific, off-nominal circumstances. If desired, the operator can pair a vehicle action with a waypoint – for example, pause and await a "continue" command – and select from a short list of broad contingencies with user defined parameters, that would trigger automated navigation of the helicopter to the rally point – for example, in the event of a loss of communications with the ground control station exceeding 30 seconds, or if no touch-down site is found to be suitable within the landing zone boundary.

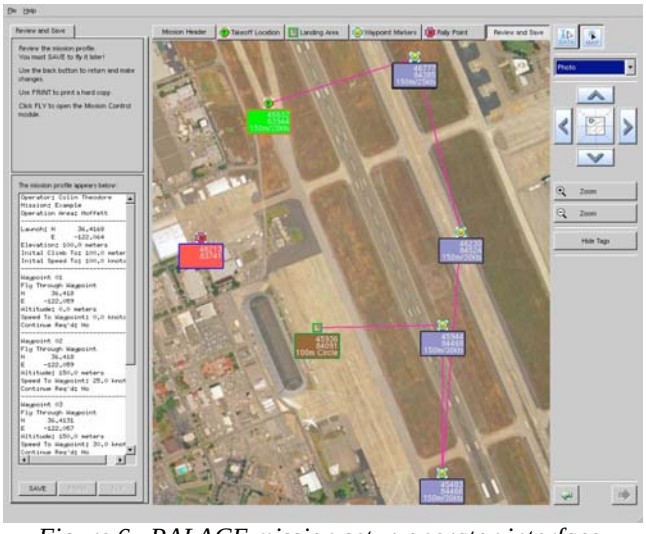

Figure 6. PALACE mission setup operator interface.

The Mission Control module (see Figure 7) allows the operator to select a saved mission to execute, monitor the flight, and if circumstances dictate, make limited in-flight vehicle/mission modifications. The Air Vehicle Performance panel provides the operator with real-time data on the RUAVs location, speed, heading altitude, etc. The Mode Control panel indicates the vehicle's current and next operational phase or activity, if an action has been assigned to a waypoint, and presents any necessary operator controls. A track-up Moving Map provides information and feedback on the vehicle's location and future mission navigation. The Camera View provides the operator with a real-time view as seen from the helicopter's left camera. During descent to landing, this view also indicates the point being tracked by the machine vision feature-tracker.

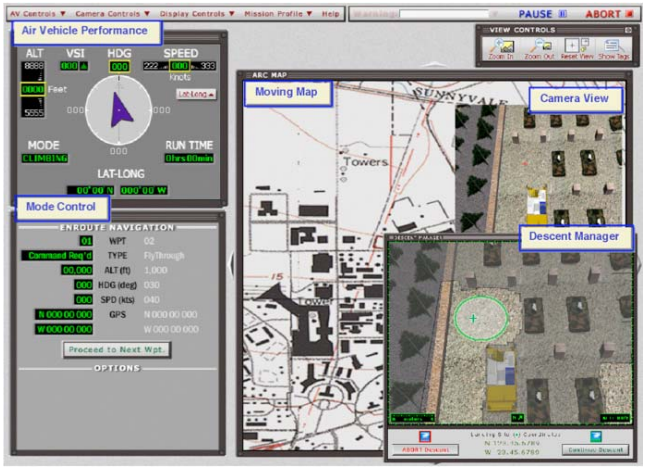

Figure 7. PALACE mission run-time operator interface.

The Descent Manager panel is presented to the operator toward the end of the mission, as the RUAV nears the proposed landing zone. This panel displays the landing zone, overlaid with the results of the SLAD algorithms, and an indicates the precise touchdown point and clearance required by the vehicle, for verification by the operator. At this time, the mission can continue to the chosen landing point, or the operator can designate a new landing point. Such a modification might be necessitated by SLAD designating a very "tight" landing site, or exigent circumstances related to the mission.

A final display, used for development and validation purposes only, shows stereo ranging and SLAD results. Figure 8 (left) is a screen shot of the 3-D reconstruction produced by the stereo ranging algorithms. Figure 8 (right) is a screen shot of intermediate and final results produced by a SLAD algorithm computation.

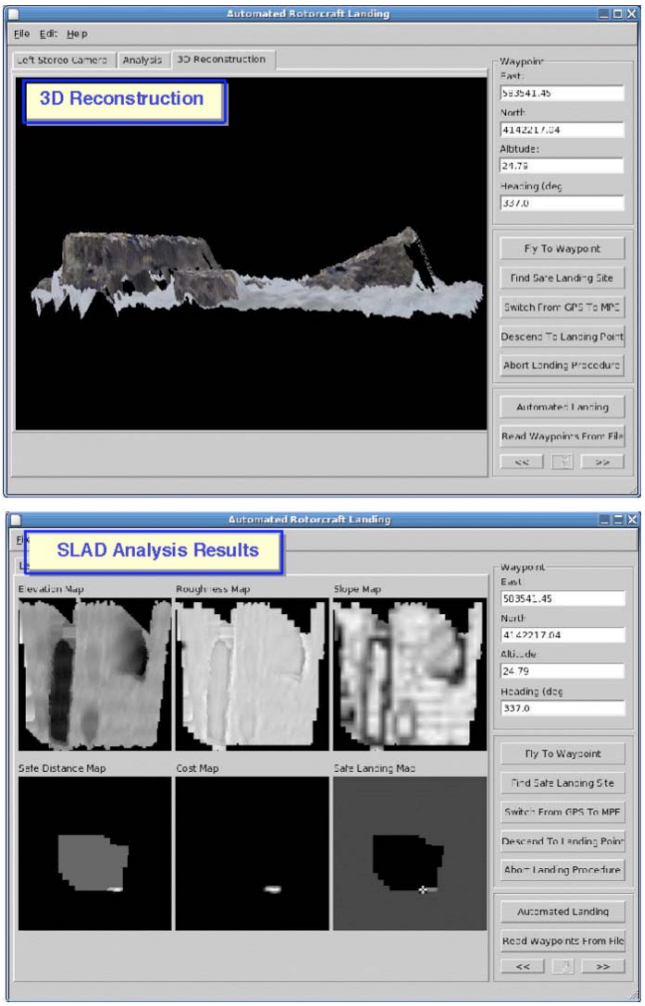

Figure 8. PALACE stereo-vision display.

Mission Manager / Landing Procedure

The mission manager is the heart of the integrated simulation and unifies the machine vision algorithms as well as providing the decision-making and coordination capabilities required for complete PALACE missions to be flown autonomously. The core features of the mission manager require it to process mission commands from the user interface, and communicate instructions to the various elements of the system.

The mission manager first receives a mission from the PALACE ground station and begins to execute the mission, once a command to do so is given. In general terms, a PALACE mission consists simply of a series of waypoints with various actions to be performed at each waypoint depending on the waypoint type. Waypoint types include: take-off, hover and wait, fly through, landing, rally, etc. The mission manager generates additional waypoints that are used as part of the landing descent. The mission manager also provides a series of commands to CLAW to switch between the three navigation modes: GPS, MPE, and dead reckoning.

Figure 9 shows the steps in the PALACE landing procedure as the vehicle descends from 30 meters to the ground. At 30 meters, the first step is to calculate the optimum heading to perform the descent and landing. A fuzzy logic rule set was constructed for the optimum heading determination that takes into account the wind speed and direction, the level of turbulence, the direction and elevation of the sun, and the camera angle on the helicopter. The rules ensure that the sun and the helicopter shadow are not in the field of view of the camera and enforce a preference to land into the wind, especially with higher wind speed and turbulence levels. The vehicle then moves under GPS navigation to a point where the nominal landing location specified by the operator is in the center of the camera image at this new heading. At this point, the stereo ranging and SLAD algorithms are run to calculate the optimum safe landing point. The mission manager then moves vehicle so that the new landing point is in the center of the image and starts tracking the landing point with the feature-tracker. CLAW is then instructed to switch from using GPS for navigation to a pseudo-GPS position estimate returned by the MPE algorithm. A waypoint is set at an altitude of 24 meters along the glideslope defined by the camera angle and the vehicle is commanded to descend to this waypoint.

Altitude:	Action:
30 meters	- Determine optimum heading for landing - Move to bring point to center of image at new heading - Run SLAD and estimate location of landing point - Move vehicle to bring point to center of image (GPS) - Switch on feature-tracker, MPE for positioning - Descend under waypoint control along glideslope to 24m
24 meters	- Re-run SLAD to refine the landing point - Move vehicle to bring landing point to center of image - Begin tracking new point - Descent under waypoint control to 18 meters
18 meters	- Re-run SLAD to refine the landing point - Move vehicle to bring landing point to center of image - Begin tracking new point - Descent under waypoint control to 12 meters
12 meters	- Run SLAD to calculate final landing point - Move vehicle to bring landing point to center of image - Begin tracking new point - Descent to 3 meters under MPE
3 meters	- Switch to Dead Reckoning for vehicle positioning - Turn off machine vision tracker - Descend to ground under Dead Reckoning
Ground	- Final landing/touchdown procedure - Weight on wheels switches begin vehicle shutdown

Figure 9: Steps in PALACE landing procedure as the vehicle descends from 30 meters to the ground.

At 24 meters, the mission manager re-runs the SLAD algorithm to update the landing point. The mission

manager again moves the vehicle (under MPE) so that this updated landing point is in the center of the image and the tracker locks onto this new landing point. Another waypoint is set at an altitude of 18 meters along the glideslope and the vehicle descends further. The same procedure happens as the vehicle descends through 18 meters to an altitude of 12 meters.

At 12 meters, the SLAD algorithms are run for the final time to pick the final landing point. The mission manager sets a waypoint at 3 meters and instructs the vehicle to descend to this waypoint.

At 3 meters, the mission manager instructs CLAW to switch from MPE to dead reckoning for positioning of the vehicle during the last portion of the descent. Dead reckoning is required since feature tracking becomes unreliable below about 3 meters. The mission manager then sends an "initiate landing" command to CLAW, which plots a landing waypoint on the ground and descends to this final waypoint. Weight-on-wheels switches are triggered when the vehicle touches down on the ground to complete the landing.

TEST AND EVALUATION RESULTS

The overriding objective of the integrated simulation is to evaluate how well the system performs during autonomous landing missions at non-cooperative sites without GPS. Although many factors have the potential for contributing to the success or failure of the landing mission, the most important components are the machine vision algorithms. For this reason, the test and evaluation results discussed here focus on the performance of the individual machine vision algorithms. In each case, a discussion is presented on the applicability of the machine vision algorithms to meet the PALACE objective of autonomous landing demonstrations with the ARP RMAX, as listed in Table 1.

Stereo Ranging Evaluations

The generation of an accurate range map from a pair of stereo camera images is the key element in the selection of a safe landing area. If the range map does not represent the terrain with sufficient accuracy, then the SLAD algorithms may not select the optimum, or even a valid landing point. The JPL stereo ranging algorithm used for PALACE receives a pair of stereo images, along with information about the pose of the camera, and produces a stereo range map [6]. There are a number of configuration parameters that can be modified to affect the performance of the algorithm. This section shows some results from the stereo ranging evaluations.

The PALACE requirement for flight demonstrations (see Table 1) on the ARP RMAX is for the stereo ranging algorithm to distinguish, and for SLAD to reject, obstacles as small as 10cm. This value is based on the assumption that the RMAX could successively land on terrain where there are obstacles or surface roughness of less than 10cm. Here the surface roughness is equivalent to the height difference between peaks and valleys of an uneven, or rippled surface.

Effect of Image Texture

The stereo ranging algorithm is based on the assumption that a feature on one camera image can be uniquely matched with the same feature in a second camera image. This assumption clearly breaks down in regions where pixels are indistinguishable and features cannot be uniquely identified in the second image. In such cases, "holes" are produced in the range map where no range information could be deduced.

Figure 10 shows the effect of image texture on the stereo ranging results for three different scenes in simulation. The top graphics show the left camera images, and the lower graphics show the stereo range results. For the two range maps on the right there are a large number of "holes" where no range information could be determined. These "holes" are a result of the lack of image texture on the terrain. The implication is that the SLAD algorithm would not find a safe landing area even though the actual terrain is free of hazards. For the parking lot scene on the left, the ground has a detailed texture that does not produce any "holes" in the range map. On the other hand, the car has only a single-shaded texture with no detail, and "holes" are evident on the surfaces of the car. However the outline of the car is clearly visible and it would be correctly recognized as an obstacle and accounted for by the SLAD algorithm.

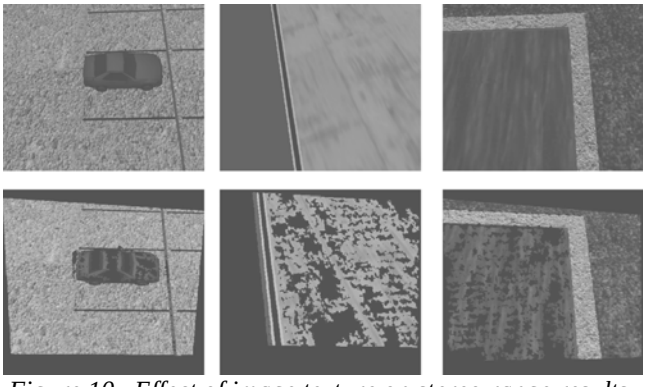

Figure 10. Effect of image texture on stereo range results. Upper row shows left camera images, lower row shows range maps.

These results provide some confidence that it is possible to generate range maps from simulated imagery as long as the surfaces have sufficiently detailed textures. Also, particular attention should be given to the texturing of large

flat surfaces since these would represent potential landing areas and should not contain range map "holes". The texturing of obstacles is less important since the obstacle edges are normally clearly distinguishable from the texture of the surrounding terrain.

Range Map Accuracy and Resolution

Two quantitative measures of the performance of the stereo ranging algorithms are the absolute accuracy and resolution of the stereo range calculations. The resolution is an important metric in the landing task since it indicates the minimum size of objects that can be resolved by the stereo algorithms.

Figure 11 shows the setup for the stereo range accuracy evaluations in simulation and with the ARP RMAX hardware. For the simulation results, the actual height of the vehicle above the terrain was varied and compared with the range calculated from the stereo analysis. For the ARP RMAX, ground tests were performed by moving the vehicle to different distances from the hangar door. The measured distances are compared with the distance from the stereo ranging calculations.

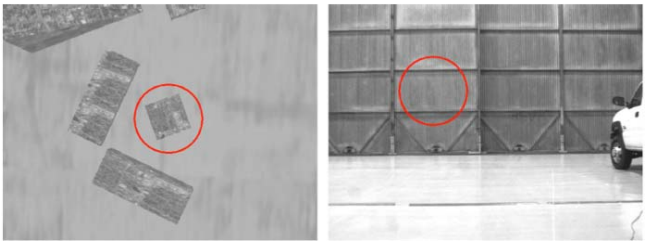

Figure 11. Setup for stereo range accuracy tests in simulation (left) and with the ARP RMAX (right).

Figure 12 shows the stereo ranging error versus the actual distance in simulation and from the ARP RMAX. These results show that the range map accuracy is within 2% of the actual distance in simulation and within 3% of the actual distance using the ARP RMAX. It should be pointed out that these results are limited by the fact that the vehicle is stationary and the image surface is orthogonal to the sensor. A more complete characterization of the stereo ranging accuracy would require the determination of the error as a function of range, angle, and feature location in the sensor field of view and with vehicle vibration and errors in the sensor pose information from the IMU. For the PALACE landing procedure, stereo ranging is used only to provide an input to the SLAD algorithms (a laser range finder is used to measure the distance to the ground) and not specifically to provide highly accurate ranging information.

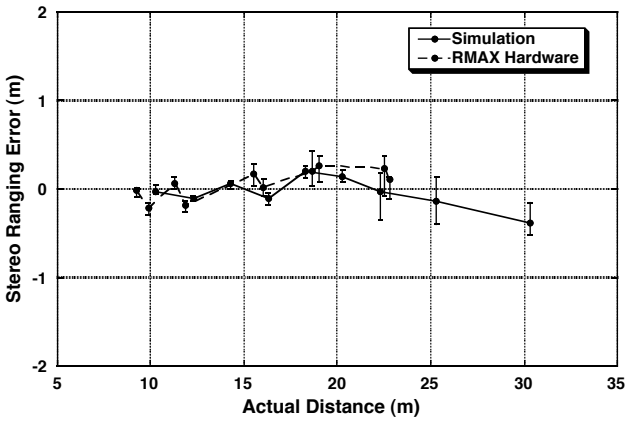

Figure 12. Stereo range error versus distance from the simulation and the ARP RMAX.

Figure 13 shows the setup to evaluate the resolution of the stereo ranging algorithms by calculating the height of a box in simulation. The actual height of the box is varied and is estimated by subtracting the range to the top of the box from the range to the ground. A number of points on the top of the box are used for the calculation to get the minimum, maximum and average box heights.

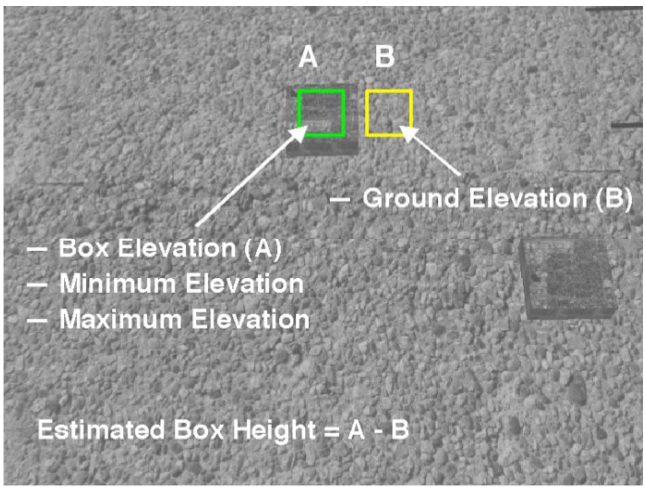

Figure 13. Setup for box height tests to measure stereo range resolution in simulation.

Figure 14 shows the results of the box height tests from a distance of 14 meters with the stereo cameras angled at 30 degrees from vertical. From this height, the resolution of the stereo ranging is about 15cm since boxes under about 15cm are generally not distinguishable from the flat ground. Inspection of the range maps revealed a spurious ripple, which is typical of stereo range results, that in this case is about the size of the 10cm box. It is the presence of this spurious ripple that limits the stereo ranging resolution that can be achieved with this configuration.

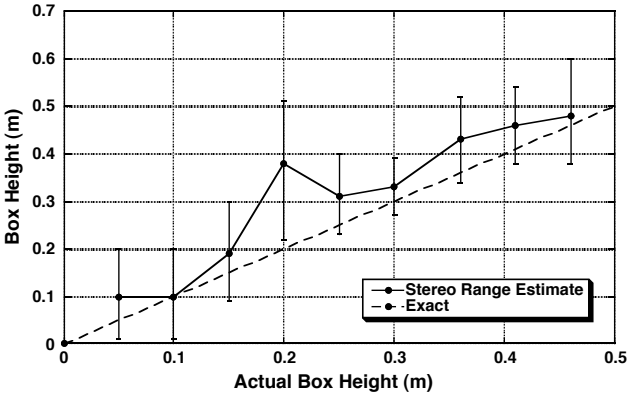

Figure 14. Range map resolution results from simulation with 50% image reduction (convert 640x480 pixel images to 320x240 pixel range map).

It should be noted that these results shown in Figure 14 were based on a 50% reduction in image size (640x480 image pixels converted to 320x240 range map pixels). The processing time (on a Pentium 4, 3.2GHz) required to calculate the range map (with a 50% reduction in image size) was 2 seconds, compared with 6 seconds for the full images with no reduction. The reduced execution time with reduced images comes at the cost of the overall range map resolution, and the configuration can be tailored to the required resolution. This indicates that there is a trade-off between the required stereo resolution and the amount of processing power available to calculate the range maps.

For the PALACE flight demonstrations, the requirement is that the stereo ranging algorithms be able to distinguish objects as small as 10cm. It is clear that this cannot be achieved from 12 meters with images reduced by 50% (320x240) as shown here. In order to achieve the required resolution of 10 cm in flight at 12 meters, the stereo ranging will have to be run with the full image size (which produces a smaller spurious ripple). If the desired resolution is still not achieved, then the stereo ranging will have to be run at altitudes lower than 12 meters. The height range to achieve the 10cm resolution will be determined as part of the flight trials.

Safe Landing Area Determination (SLAD) Evaluations

The SLAD algorithm takes a stereo range map and combines it with a set of landing point constraints to calculate a safe landing map and to choose the optimum landing point. The safe landing map highlights the regions that satisfy the safe landing constraints, and the optimum landing point is the point that best satisfies the set of constraints. The SLAD constraints ensure that the safe landing areas have slope and surface roughness characteristics that are below specified limits, and that there is a minimum distance from hazards.

The values of the constraints used for the PALACE program are based on the geometry and performance limits of the ARP RMAX. It has already been mentioned that the RMAX can land on obstacles up to 10cm in height, so the roughness constraint for the flight trials has to be set accordingly. For these simulation tests, the roughness constraint is set at 15cm, since this is the resolution of the current configuration from 12 meters (with a 50% reduction in image size from 640x480 pixels to 320x240). The size of the minimum obstacle-free area (see Table 1) is set at 6.25 meters, which is double the rotor diameter and leaves a margin of 1.5 meters from the rotor tip to the nearest obstacle. This margin is required to account for drift associated with the monocular tracker and dead reckoning systems in the final descent from 12 meters to the ground. The margin would need to be made larger if the monocular tracker and dead reckoning systems experience more drift than 1.5 meters. The final requirement is for SLAD to reject any slopes that are larger than 15 degrees. For the flight demonstrations, the vehicle will always land on level surfaces (no slope), but the system will be evaluated to reject slopes greater than 15 degrees.

The performance measures for the SLAD algorithm are based on the success or failure rates in choosing valid landing sites from 30 repeat runs. A "success" is where the algorithm correctly identifies a safe landing point when one exists, or correctly identifies that there are no safe landing sites when none exist. There are two failure modes for the SLAD algorithm. The first is a "false negative" when the algorithm returns that there is no safe landing point when one actually exists. The second is a "false positive" when the algorithm returns a landing point that does not satisfy all of the constraints.

Effect of Number of Safe Landing Areas

The first set of SLAD tests are designed to observe the effects of having different numbers of safe landing areas. Four different scenes were constructed in simulation with 0, 1, 2 and 3 safe landing areas that satisfy all of the SLAD constraints. Figure 15 shows the SLAD test results with different numbers of available safe landing sites. The results show that there is greater than an 85% success rate regardless of how many safe landing sites are available. For the cases where there are no safe landing points, the algorithm produces about 10% "false positive" results where the algorithm indicates that there is a safe landing point when none actually exist.

For the flight demonstrations on the ARP RMAX, it is clear that any "false positives" are unsuitable and could lead to the vehicle attempting to landing at an unsafe site. Additional research is underway to determine why the

SLAD algorithms are returning these "false positives" and what steps can be taken to reduce these results.

From an operational standpoint, one way to improve the performance of the SLAD algorithm is to use the full size images (640x480) in the stereo ranging rather than the 50% reduced (320x240) images used for these results. If the performance is still not sufficient in flight, then the SLAD algorithm can be run a number of times (say 3-4 times) to statistically reach a consensus on the safe landing point. The PALACE objective (listed in Table 1) is to have a 98% "success" rate with the SLAD algorithm.

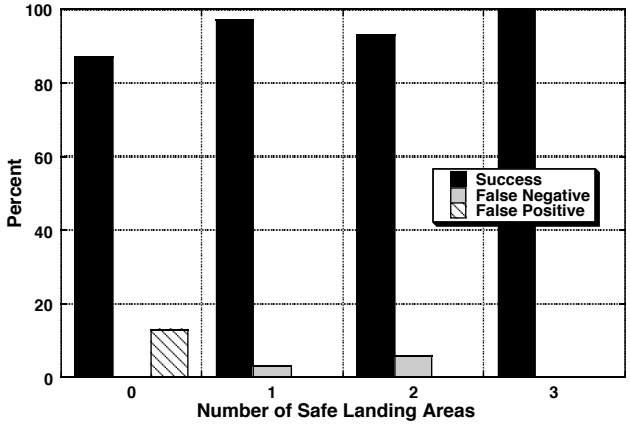

Figure 15. SLAD test results with different numbers of safe landing sites.

Effect of Obstacle Spacing

The next set of SLAD results focus specifically on the constraint of minimum distance from obstacles. Figure 16 shows the SLAD success and failure rates against the percent of the open space (minimum distance) from obstacles. When the open space size is below the constraint value (<100%) there are no safe landing areas and a success is represented where no safe landing area is reported. For this case, there is at least an 85% success rate, similar to the previous tests with 0 safe landing points. When the open space size is above the constraint value (>100%) there is a single safe landing area and a success indicates that this point is identified correctly. The rate of success for these cases increases as the open area increases. Good performance with a success rate of greater than 85% is achieved when the open space is larger than about 5-10% of the constraint value.

These obstacle spacing results also show that there is degraded performance when the actual distance from obstacles is close to the constraint value. These results would indicate that a margin of 5-10% should be included in the minimum distance constraint value to ensure an acceptable success rate.

For the ARP RMAX flight trails, an additional 10% margin will be added to the minimum distance from obstacles constraint, requiring that the obstacle-free region have a minimum diameter of about 7 meters. This gives a margin of about 2 meters from the edge of the rotor disk to the nearest obstacle.

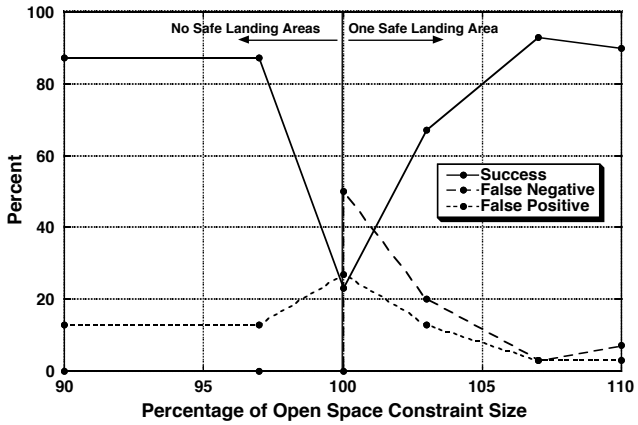

Figure 16. SLAD test results with varying obstacle spacing.

Effect of Altitude

The final set of SLAD results examine the effect of altitude on the SLAD success and failure rates. For these tests, the landing scene contains one possible safe landing point and the altitude of the vehicle is varied. Figure 17 shows the SLAD success/failure rates against the stereo ranging altitude. There is reasonable performance of the SLAD algorithm at and below about 12 meters with success rates around 70%.

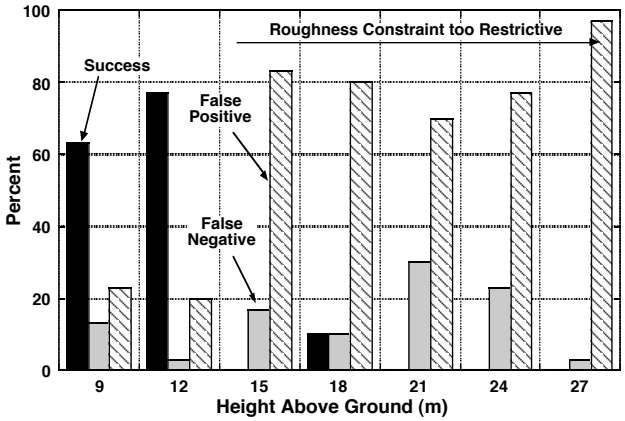

Figure 17. SLAD test results with variations in the height above the ground.

For altitudes above 12 meters the success rates are severely degraded with the results being dominated by spurious "false positives" in which the chosen landing sites clearly violate the SLAD constraints. Further examination of the

results for altitudes above 12 meters revealed that the size of the spurious ripple in the stereo ranging results increased with altitude from about 15cm (see Figure 14) at 12 meters. The presence of the spurious ripple should logically violate the roughness constraint (set at 15cm) for all cases above 12 meters, and should lead SLAD to produce "false negatives". However the algorithm produces spurious "false positives" for these cases. Further investigation is underway to determine why the SLAD algorithm is reporting "false positives" when the roughness constraint is being violated.

These results for variations in altitude indicate the need to dynamically specify the roughness constraint value based on the height above the ground. For the flight demonstrations, the roughness constraint value would be set at 10cm to reject any obstacles and rough areas greater than this value, only at the lowest altitude at which the SLAD algorithm is run (nominally 12 meters). For higher altitudes, the roughness constraint value will be dynamically scaled so that the constraint value is not beyond the resolution of the stereo ranging at that altitude.

Feature Tracker / Monocular Position Estimation (MPE) Evaluations

The JPL monocular feature-tracker and MPE algorithms take left camera images and information about the pose of the camera to estimate the position of the vehicle relative to a fixed point on the ground that the machine vision algorithms are tracking [6]. The MPE algorithm is used during the descent and landing portions of the PALACE mission where the GPS signal may be intermittent or occluded and cannot be relied upon for navigation. To minimize the number of changes to the control system, the intent was to replace the GPS signal input to the Kalman filter with a pseudo-GPS signal calculated by the MPE algorithms. This required that the tracker run in real-time and provide a position estimate at the same rate and in the same format as GPS. The MPE algorithms were therefore required to produce a pseudo-GPS signal at 10Hz (same as GPS rate), which meant that the maximum processing time for the MPE calculations was 100 msec. Careful attention was also required in the simulation to minimize the time delays of the system and to synchronize the data and images. This ultimately allowed for a common set of inner-loop control system gains and Kalman filter parameters to be used for operation with both the GPS and pseudo-GPS signals.

The feature-tracking algorithm functions by first saving a small template window containing the feature that is being tracked. The following image is scanned to find the location of the best match to the template. The template window is then updated from the new image and the algorithm returns the pixel location of the feature and a confidence measure for the template match. Finally the MPE algorithm calculates the position of the vehicle relative to the feature on the ground using the pixel location of the feature from the feature-tracker, a laser range measurement, and the pose information of the camera.

The performance metrics for the position estimation algorithms include the frame-to-frame tracking performance, the amount of tracking drift over time, and the processing time required. The frame-to-frame tracking performance is given by the confidence measure in matching the template window in each new frame. A low confidence value could indicate that the tracker has jumped to a new feature or has lost the ability to track the feature.

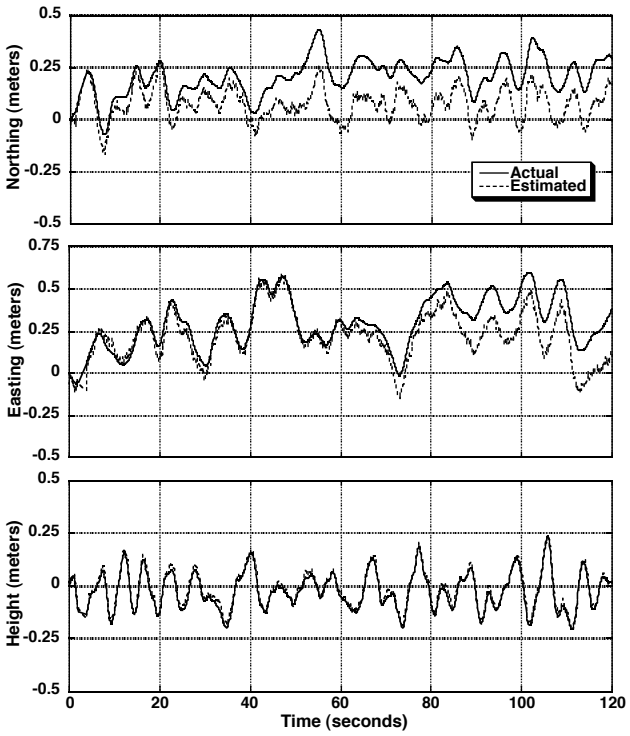

Figure 18. Example feature tracking results comparing actual position and estimated position (MPE).

The tracked position drifts over time due to the fact that the feature template is updated at each cycle of the tracker. Updating the feature template at each step has the advantage of adapting to changes in the camera position and view of the scene during the descent, but is prone to drift since the original feature that was being tracked is not kept in memory. For these tests, the amount of tracker drift is evaluated as the difference between the true position of the vehicle, and the estimated position (from MPE) after a two-minute tracking run. An example of these data is shown in Figure 18, which shows the northing, easting and altitude time histories of the true and estimated positions for two minutes of hover. Both sets of data start at the same point and diverge over time due to the drift inherent

in the tracking algorithm. The difference between the true and estimated positions after two minutes is the amount of tracking drift.

The requirement of the PALACE program (listed in Table 1) is for a landing accuracy of 1.25m, which is evaluated by considering the combined drift in the feature tracker and dead reckoning algorithms from the lowest SLAD altitude (nominally 12 meters) to the ground. The metric for the landing accuracy includes only this final part of the descent since the SLAD operation effectively resets the drift at each altitude from which it is run. This is because a new optimum landing point is selected at 30, 24, 18, and 12 meters by the SLAD algorithms and the system begins tracking this new landing point.

Effect of Atmospheric Conditions

This first set of MPE results are designed to evaluate the tracker performance under various atmospheric wind speeds, directions, and levels of turbulence. Figure 19 shows the effect of wind speed, wind direction and level of turbulence on the tracking performance in simulation. The plot shows the maximum tracking drift (from five repetitions of the two-minute tracking task) versus wind direction for calm and windy conditions. Calm conditions are defined as a wind speed of 1 m/sec with low turbulence, and windy conditions are defined as a wind speed of 4.5 m/sec with high turbulence. The amount of tracking drift is higher during windy conditions because the larger disturbances to the vehicle result in the tracked feature moving farther frame-to-frame than for calm conditions. The maximum drift after two minutes of tracking is less than two meters for calm conditions and less than three meters for windy conditions.

Another conclusion from Figure 19 is that the amount of drift is a function of the wind direction. The best tracking performance is achieved with the vehicle facing into the wind, and the worst tracking performance is with the wind from the side. For calm conditions, the maximum tracking drift experienced is less than 0.5 meters (with the wind from the front), compared with tracking drift approaching 2 meters for wind from the side. Also for windy conditions, the tracking drift can be reduced to below 1 meter by pointing the vehicle into the wind. These results make sense since winds and gusts from the side result in greater disturbances to the vehicle than gusts from the front because of the lower inertia in roll than in pitch, the greater yaw response due to gusts from the side, and the greater flat-plate area for lateral gusts.

For the PALACE flight trials on the ARP RMAX, this tracking performance is sufficient to meet the requirement of a 1.25m landing accuracy, as long as the heading is chosen with consideration to the wind direction. Tracking drifts of less than 0.5 meters for calm conditions, and less than 1 meter for windy conditions can be achieved with a vehicle heading to within +/-45 degrees of the prevailing wind. This +/-45 degree margin on the heading is important from an operational perspective since it allows some margin in the choice of vehicle heading while still maintaining a drift of less than 1 meter.

If the amount of drift experienced in flight happens to be larger than that seen in simulation, there are two possible methods mitigation. The first is to alter the inner-loop control system gains to improve the disturbance rejection performance of the vehicle. This would decrease the vehicle response to disturbances (and the camera motion) and therefore decrease the tracking drift, but comes with the expense of higher control crossover frequencies and increased actuator motion. The second method is to keep the template window for a few cycles of the tracker (say 3-5) rather than discarding the old template window at each step. The reduction in drift for this case comes from the fact that the template window is updated less frequently.

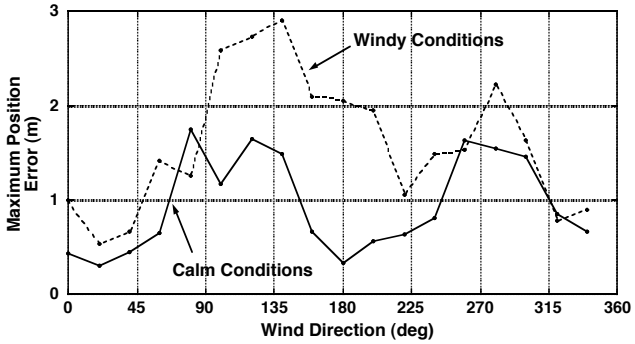

Figure 19. Effect of atmospheric wind speed, direction and level of turbulence on the maximum tracking drift.

Effect of Altitude

The next set of simulation results examines the tracking performance as a function of the altitude. The effect of altitude on the tracking performance is important as the machine vision tracking algorithms will be used to give vehicle positioning information as the vehicle descends from 30 meters, and the tracker must work well over this altitude range. These altitude tests will also indicate the lower altitude limit under which the tracking performance begins to degrade to the extent that the MPE algorithm cannot reliably produce an accurate position estimate. For the PALACE landing mission, once the tracker becomes unreliable, vehicle positioning switches to a dead reckoning system to descend the last few meters to the ground. The transition point from MPE to dead reckoning should be as low as possible since it is expected that the dead reckoning system will be more prone to drift than the feature-tracking system.

Figure 20 shows some simulation test results for calm conditions with the vehicle hovering at various altitudes. These results first show that the amount of tracking drift increases as the altitude increases. This is explained by the fact that each pixel of drift translates into a greater physical drift as the altitude increases. For calm conditions, the maximum drift experienced over five repetitions of the two-minute test run at 30 meters was about 1.25 meters.

Under 5 meters, there are instances where the feature was lost by the tracker as indicated by the low minimum confidence values and the number of times that low confidence values were recorded (Low Confidence Count). The results indicate that tracking starts to degrade below 5 meters and becomes marginal at about 3 meters. Based on these results, the dead reckoning algorithms would be started when the vehicle descends below 3 meters. For the ARP RMAX flight trials, the frame-to-frame tracking performance (as measured by the confidence values) would have to be monitored to ensure good tracking performance as the vehicle descends. If the performance starts to degrade, positioning should be switched to dead reckoning even if the vehicle is at an altitude greater than 3 meters.

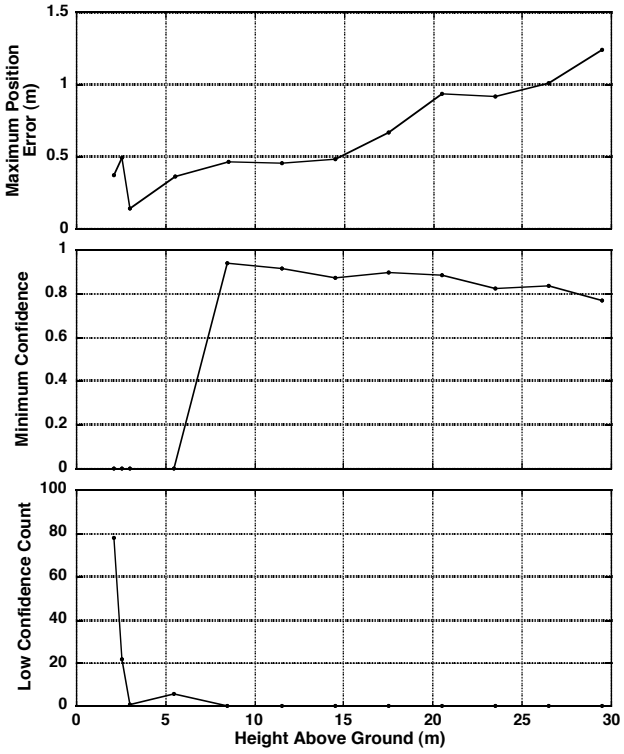

Figure 20. Effect of altitude on tracking performance.

Effect of Search Window Size

There are two configuration parameters that have an effect on the performance of the feature-tracking algorithm. The first is the template window size, which is the size of the window is carried over to match in the next frame. The second is the search window size, which is the amount of the image that is searched to find the best match to the template window. A combination of the search window size and template window size defines an 'effective' search window size, which is the maximum distance (in pixels) that a feature can move from one frame to the next and still be tracked.

Figure 21 shows the processing time and tracking confidence in simulation for various search window sizes in calm conditions. The tests were conducted at an altitude of 12 meters with a template window size of 20 pixels (previous simulation tests have shown that the best frame-to-frame feature tracking performance is achieved with a 20x20 template window). The tracking performance is very good for search window sizes of 80 pixels and greater as indicated by the near-perfect confidence values shown in the lower plot. Below 80 pixels, the minimum confidence value drops to zero, indicating that the feature has been lost at least once during the tracking tests, possibly resulting in the feature-tracker 'jumping' to a new feature. The conclusion is that for these calm conditions, a search window size of 80 pixels is enough to give good performance and there is no performance gain by using a larger search window. Clearly, as the atmospheric conditions degrade, with increases in the wind speed and wind direction and the level of turbulence, the search window size would have to be increased to gain the same level of performance.

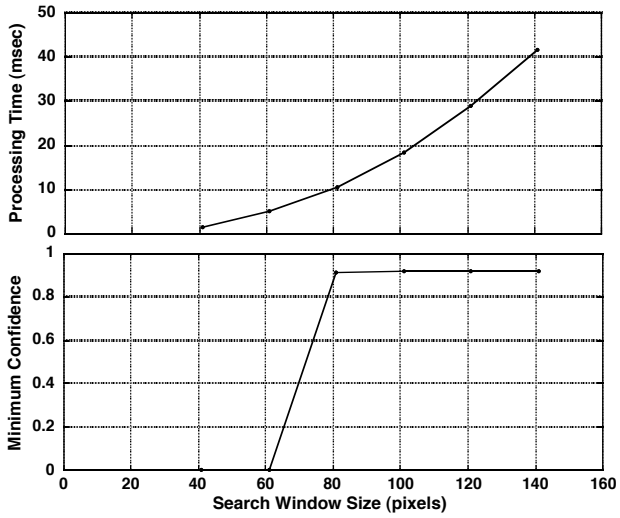

Figure 21. Effect of search window size on tracking performance.

The other factor in selecting the search window sizes is the processing time required for the feature-tracking algorithm. The top plot in Figure 21 shows the processing time required for different search window sizes (as measured on

a Pentium 4, 3.2 GHz machine). This shows how the processing time increases as a function of the search window size. Here there is a trade-off between the required tracking accuracy (which requires a large search window size) and the processing power available for tracking (which limits the size of the search window).

For the ARP RMAX flight trials, careful selection of the template window and search window sizes will be important to gain the required level of performance of the PALACE program given the amount of available processing power on-board the vehicle. The size of the search window must be large enough to ensure that the tracked feature does not move further than the 'effective' search window size from one frame to the next. The search window could also need to be dynamically increased as the vehicle approaches the ground in order to minimize the altitude at which the tracking performance degrades.

CONCLUDING REMARKS

This paper provided an overview of the PALACE program and the methods of integrating machine vision technologies with realistic vehicle dynamics and control laws for autonomous landings of RUAVs. An integrated simulation environment was constructed for the testing and evaluation of the PALACE landing procedure, as well as for the performance testing of the individual machine vision algorithms. Results for the testing in simulation of the stereo ranging algorithm, SLAD, monocular feature-tracking algorithm and MPE algorithm were presented for various conditions that would be expected during typical RUAV landings. Each set of results also included a discussion of the applicability of the simulation results to upcoming flight trials on the RMAX to meet the specific PALACE requirements.

A comparison of the between the quantitative objectives of the PALACE program (listed in Table 1) and the actual values as measured in simulation are shown in Table 2. The feature-tracking and MPE algorithms are able to meet the requirements of drift and processing time with simulation results, as long as the vehicle heading and configuration parameters are chosen carefully. The SLAD algorithm was shown to produce a number of spurious results that limit the ability to meet the required metrics for the PALACE program. Further investigation is underway to determine the cause of the spurious results. Other methods are available to improve the success rate for the SLAD algorithm to choose suitable landing points. These methods were discussed with the SLAD results.

Table 2. PALACE program quantitative metrics. Project objectives and actual values.

Quantitative Metric	Project Objective	Measured Values
Landing Site Size	< 6.25 m	< 7 m
Landing Surface Slope	< 15 deg	< 15 deg
Landing Surface Roughness	< 10 cm	< 15 cm
Landing Accuracy	< 1.25 m	< 1.00 m
Feature-Tracking Time	< 100 msec	< 45 msec
SLAD Calculation Time	< 5 sec	< 6 sec
SLAD Success Rate	> 98%	> 85%

Based on the results of the testing in the integrated simulation, the following conclusions can be drawn:

1. The integrated simulation environment proved to be an effective tool for the performance evaluation of the machine vision algorithm even though the images used by the algorithms were computer generated. However, this required that special attention be given to the texturing of the 3D terrain, particularly the texturing of large, flat surfaces. The simulation environment was also effective for evaluating how well the individual technologies could be integrated into a complete system to for the autonomous landing task.

2. The amount of drift in the position estimation algorithm was seen to vary as a function of the wind speed, wind direction and level of turbulence. Performance typically degraded as the wind speed and turbulence increased due to the fact that the vehicle response to disturbances (and camera motion) increases under these conditions. For a given wind speed and turbulence level, the best tracking performance was achieved with the vehicle pointing to within +/- 45 degrees of the prevailing wind.

3. For each of the machine vision algorithms, there is a trade-off between the performance of the algorithms and the processing time required. The testing conducted with the integrated simulation illustrated how the choice of configuration parameters affects the performance and processing time of the algorithms. Careful selection of these configuration parameters is important to achieve the required level of performance of the vision algorithms with the available processing power. This is will be especially true when using these algorithms on flight hardware where processing power is limited.

ACKNOWLEDGEMENTS

The authors would also like to acknowledge James Montgomery, Larry Matthies, Adnan Ansar, Andrew

Johnson and Steve Goldberg from the Robotics Laboratory at JPL for their assistance with the machine vision algorithms.

REFERENCES

[1] Sierra Nevada Corporation, "UAV Common Automatic Recovery System (UCARS)," http://www.sncorp.com/uav1.html.

[2] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Visually-Guided Landing of an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 371-381, June 2003.

[3] Saripalli, S., and Sukhatme, G. S., "Landing on a Moving Target using an Autonomous Helicopter," Proceedings of the International Conference on Field and Service Robotics, July 2003.

[4] Garcia-Padro, P. J., Sukhatme, G. S., and Montgomery, J. F., "Towards Vision-based Safe Landing for an Autonomous Helicopter," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 19-29, 2001.

[5] Shakernia, O., Vidal, R., Sharp, C. S., Ma, Y., and Shastry, S. S., "Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle," Proceedings of the International Conference on Field and Service Robotics, May 2002.

[6] Hintze, J., Christian, D., Theodore, C., Tischler, M., McLain, T., and Montgomery, J., "Simulated Autonomous Landing of a Rotorcraft Unmanned Aerial Vehicle in a Non-cooperative Environment," Proceedings of the American Helicopter Society 60th Annual Forum, Baltimore, MD, June 2004.

[7] Rowley, D. D., "Real-time Evaluation of Vision-based Navigation for Autonomous Landing of a Rotorcraft Unmanned Aerial Vehicle in a Non-cooperative Environment," Master's thesis, Brigham Young University, Provo, UT, 2005.

[8] Whalley, M., Takahashi, M., Schulein, G., Freed, M., Christian, D., Patterson-Hine, A., and Harris, R., "The Army/NASA Autonomous Rotorcraft Project," Proceedings of the American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.

[9] Johnson, A., Klump, A., Collier, D., and Wolf, A., "LIDAR-Based Hazard Avoidance for Safe Landing on Mars," AAS/AIAA Space Flight Mechanics Meeting, Santa Barbara, CA, Feb 2001.

[10] Mansur, M. H., Frye, M., Mettler, B., and Montegut, M., "Rapid Prototyping and Evaluation of Control System Design for Manned and Unmanned Applications," Proceedings of the American Helicopter Society 56th Annual Forum, Virginia Beach, VA, May 2000.

[11] Tischler, M. B., and Cauffman, M. G., "Frequency-Response Method for Rotorcraft System Identification: Flight Application to BO-105 Coupled Rotor/Fuselage Dynamics," *Journal of the American Helicopter Society*, vol. 37, no. 3, pp. 3-17, 1997.