# The Development of a Small Autonomous Helicopter Robot for Search and Rescue in Hostile Environments

**June 1, 1999**


| | | |
|---|---|---|
| Jacob Beal[†] | Paul Eremenko[†] | Manway Liu |
| Carl Blaurock | Eric Feron[‡] | Stefan Marti |
| Kieth Bonawitz | Emilio Frazzoli | Joshua Napoli |
| Kyrilian Dyer | Benjamin Ingram | Kailas Narendran |
| Paul Elliott | Michael Lester | Scott Rasmussen[†] |

Aerial Robotics Group
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139



## ABSTRACT

An interdisciplinary group of MIT undergraduates has engaged in the design and construction of a small autonomous aerial robot based on a helicopter platform to achieve the objectives of a search-and-rescue scenario posed by the International Aerial Robotics Competition. The helicopter is capable of fully autonomous flight operations and is equipped with an on-board vision system that locates and identifies ground objects. In its present state, the helicopter will compete in the 1999 qualifying round of the International Aerial Robotics Competition Millennial Event, but is in a preliminary development stage for the year 2000 contest. An overview of the target system and its components is presented. The robot, in its present state, is described in detail.


## INTRODUCTION

The Association for Unmanned Aerial Vehicle Systems International (AUVSI), in cooperation with the U.S. Department of Energy (DOE) and the Department of Defense has created a simulated five-acre urban disaster area at the DOE's Hazardous Materials Management and Emergency Response (HAMMER) facility adjacent to the Hanford Nuclear Site in Washington. The objective of the International Aerial Robotics Competition is for an autonomous flying robot to traverse the disaster zone, avoiding obstacles such as flame jets, water jets, aerosol obscurants, and tall structures, and map the location of these hazards as well as human victims -- simulated by animatronics -- to a 2 m SEP accuracy [7]. Several restrictions are placed on the aerial robot: it must weigh less than 90 kg, it must fly untethered out of ground effect, it must receive no intervention of any kind from a human for the duration of the mission, but it may communicate with ground-based computers, if necessary. Multiple aerial robots can also be used, but must collectively fall under the 90 kg weight limit. Furthermore, competitors are permitted to introduce ground robots. These accrue points by aiding the situation on the ground (e.g. putting out fires, marking victim locations, etc.), but must also operate autonomously, and must be completely subservient to the aerial vehicles – so as to maintain the focus of the competition on aerial robot development. This paper describes the system concept and subsystem details being pursued by MIT for the Millennial Event, and a scaled-back version of the robot for the 1999 qualifier.

## MILLENNIAL SYSTEM DESIGN

The details of the components and associated datapaths can be seen in Figure 1. The blocks at the top of the figure will be ground-based systems and will include: the safety pilots, a flight status monitor, and a team of ground robots. All other subsystems will be airborne. All datapaths will be digital.

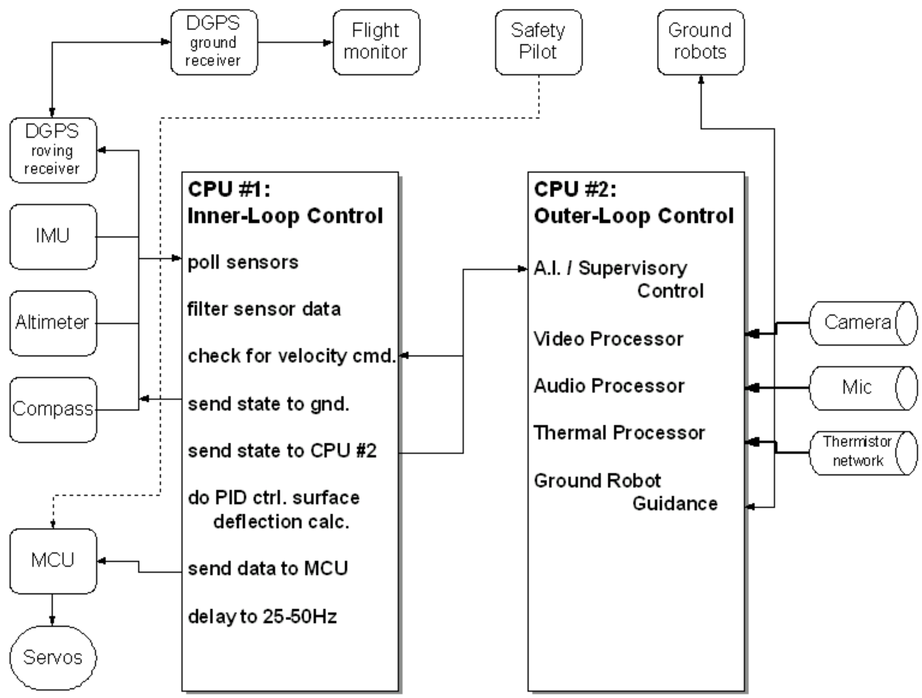


**Figure 1**

## The Flight Platform

The airborne electronic components and sensors will be mounted on a Bergen Industrial Twin Helicopter. The helicopter is powered by a modified Zenoa G23 2 cycle, 2 cylinder engine. The maximum gross weight of the helicopter is estimated at 30 lb, with a 15 lb estimated useful payload.

Electronics and sensors will be mounted in four aluminum boxes, two on either side of the helicopter, and two in the front. The boxes and their mountings will be designed to shield the equipment from radio interference, high-frequency vibrations, and potential ground or obstacle impact. The vision and directional audio sensors will be attached to a special pod underneath the forward electronics box. Thermal sensors (which are quite small) will be distributed around the entire helicopter frame. The roving GPS antenna will be mounted directly on the tail boom. Electrical power will be supplied by a lithium-ion battery pack placed aft to balance the helicopter's center of gravity. The fuel tank will also be slightly aft of the rotor-shaft, underneath the tail boom.

**Navigation Sensors and Sensor Processing**

The navigation sensor package will include a differential global positioning system (DGPS) which provides high-accuracy position and velocity information from the Navstar satellite navigation constellation. A Novatel RT-2 DGPS system will be employed, providing a position accuracy of 0.02 m CEP, and velocity accuracy of 0.03 m/s RMS with differential, carrier-phase, and multipath corrections at a 1 Hz refresh rate. An inertial measurement unit (IMU) will provide high-bandwidth angular rates and accelerations resolved in body-fixed axes. A Crossbow DMU-6 IMU will be used, providing acceleration and angular rate updates at 30 Hz. To compensate for poor GPS accuracy in the vertical direction, a sonar altimeter (based on a Polaroid 6500 ultrasonic transducer) will be employed. It is capable of providing altitude information at a reasonable accuracy up to 10-12 m. Since long-term heading drifts are poorly observable from the IMU, and the GPS-derived velocity does not provide enough information to resolve heading angle at slow speeds, a digital compass will be used to establish the helicopter yaw angle to about $\pm 1°$.

All navigation sensors require a power input and produce a serial (RS-232C standard) output. All these serial outputs will be fed to a PC104-based AMD486DX4 CPU running at 100 Mhz (labeled CPU #1 in Figure 1). Additionally, the airborne DGPS receiver will have a second serial interface connected to an RF modem with 19,200 baud data rate. The RF modem, which has a line-of-sight range of several hundred feet, will link to a similar modem on the ground, also serially connected to an identical DGPS receiver which will act as the ground station and will transmit differential corrections to the helicopter. The same RF link can be employed for low-rate telemetry transmission and other auxiliary purposes.

The outputs of all the navigation sensors will be read (through the appropriate device drivers, of course) by an extended Kalman filter (EKF) [5] running as a part of the inner-loop control (described in the section titled "Navigation Sensors and Sensor Processing") software. The EKF, then produces its optimal estimate for the helicopter state.

**Mission Sensors and Sensor Processing**

While the navigation sensors and the associated inner-loop control software is able to have the helicopter take-off, land, and fly from waypoint to waypoint, the robot must be capable of establishing these waypoints. Since the competition environment is dynamic, this must be done in real-time. Consequently, the aerial robot will possess a suite of mission sensors that help it accomplish tasks necessary for the mission. These sensors will include a video camera, a directional audio microphone, and a network of infra-red sensors known as thermisters.

The video camera will be mounted on a servo so as to permit rotation about the pitch axis. This will permit the use of the video stream to observe ground objects, as well as detect obstacles at the helicopter's flight altitude. The camera's NTSC output will be processed with a PC104 digital capture board and directed to a PC104 Pentium-class on-board computer for processing (labeled CPU #2 in Figure 1). The sequence of frames produced by the capture board will be sampled at the highest rate permissible by CPU speed. The images will then go through a series of processing steps. These can be broken down into two functional categories: ground object processing and collision avoidance.

When focusing on ground objects, the video stream will undergo three basic steps: fixation, classification, and geolocation. The first one is object fixation (also known as tracking) [9]. The basic function of this algorithm is to take a video frame and find things that it considers to be distinct from the background, i.e. objects, and then track these objects within each subsequent video frame (as long as they stay in the field of view) despite the helicopter's motion. Not only will this algorithm determine candidate objects for classification, but it also permit the determination of the distance to a given object by triangulation (where the motion of the helicopter transverse to the object creates the base for triangulation). The second algorithm applied to the video stream is object classification [10], [8]. This will be accomplished by a method known in the vision field as geons. Basically, each potential target object (i.e. human, barrel, etc.) is modeled as a combination of simple geometric shapes (e.g. spheres, cylinders, etc.), and when a collection of these geometric shapes is found in close proximity, a successful detection of a human victim or other object is made with a given probability. The final portion of the video stream processing consists of a geolocation algorithm. Having determined the distance to a given object (as part of fixation) and having identified the object as one of interest, the geolocation filter will use simple trigonometry (taking into account the orientation of the helicopter in space and the position of the camera) to compute the position of the object in the earth-fixed 3D reference frame.

An algorithm estimating the optical flow of the camera image will also be employed for collision avoidance purposes [1]. Any object near which the optical flow diverges rapidly is likely to be in the collision path of the helicopter.

The audiolocation system will be based on a directional microphone. The constant ambient noise (e.g. rotor noise) will be filtered out, and unusual sounds such as human screams will be detected. The directional characteristics of the microphone will permit the determination of the approximate direction from which the sound is emanating. Then, using the motion of the helicopter as a base, triangulation to determine sound source location can be accomplished by an appropriate simple software algorithm.

The thermister network is a series of several dozen small devices that change resistance on the basis of temperature. Their associated software can be calibrated according to the "ambient" temperature due to their location. The differential reading of these devices scattered around the helicopter will permit the determination of strong heat or infra-red sources – flame jets. This information will be crucial for hazard avoidance.

Thus, the sensor processing software, at the highest level of abstraction, will yield a series of object types (e.g. victim, barrel, tree, etc.) with associated 3D positions, a series of locations of probably survivors emanating screams, and a direction, relative to the helicopter, of any strong sources of heat. This information will then be used by the outer-loop control to plan waypoints for the helicopter to traverse.

### Inner-Loop Control

The low-level control of the helicopter will be attained by closed-loop PID control software [6]. At the top abstraction level, a velocity input and a helicopter state input will be required for the inner-loop control, and appropriate servo deflection will be produced. The velocity input, however, will only be needed for one of the control loops.

The first control loop is for yaw stabilization. It will utilize a PD (proportional-derivative) controller. Roll and pitch cyclic loops will function likewise. The altitude (collective) control loop will employ a PID controller (with the integral term used to compensate for steady-state altitude errors). Finally, the helicopter position loop is closed with a PD controller as well [4]. All the control loops, with the exception of the latter, are delayed to run at approximately 25 Hz. The position loop runs at approximately half that rate in order to avoid frequency mix-up and undesired instability.

### Outer-Loop Control

The outer-loop control software, also known as supervisory control, mission planner, or AI (artificial intelligence), will be responsible for planning waypoints (velocity vectors, actually) for the helicopter to traverse and commanding various high-level behaviors (such as take-off, search, land, etc.). A set of genetically evolved subsumption architecture [3] augmented finite state machines (AFSM) [2] will take sensory information and produce velocity commands in an Earth parallel, heading fixed axis system. These velocity commands will be presented to the inner-loop control algorithm that performs the necessary pre-filtering to ensure the flight vehicle operates in its accepted performance envelope.

The subsumption architecture will be used to program some basic behaviors which, at a hand coded level, basically perform the desired mission. An evolutionary strategies approach will then used to genetically tune the AFSM network and improve its performance on simulated missions. Some of the basic behaviors that will be implemented include the following: a desire to keep away from anything solid, a desire to seek items of interest, a set of stooping behaviors (in which the network will direct the vehicle to hunt at a high altitude until something of interest is located and then descend to take a closer look at whatever was found) and a return procedure. The subsumption architecture means that the AFSM network will be set up in layers such that at any point if a higher priority behavior decides that it is important, it will inhibit or suppress the lower priority behavior. For example, the “avoid solid objects” behavior will take precedence over the “seek interesting objects behavior,” thus if the vehicle encounters a building while trying to take a closer look at something it believes is a human body, the “avoid” behavior will be activated and it will send commands which maneuver the vehicle around the obstacle.

The “scents” which will form the sensor input to the AFSM network are derived from the vision processor's probability matches on objects in the visual field, the vision divergence algorithm, and the audiogeolocation and thermal sensor software. As the vehicle approaches the location of a potential object the scent will increase, also as the vehicle turns towards the direction from which the scent is emanating the scent will also increase. As the probability that the object is real increases (because the vision system processes more images and obtains better matches from an interesting feature) the scent will also increase. Thus the behaviors that will be programmed into the AFSM network are set to follow the vector of increasing scent. When the scent increases sufficiently the vehicle can be directed to descend upon the object so that it will be better positioned within the field of view of the camera and thus permit the vision system to obtain a more accurate estimate of its identity.

**Flight Monitor**

Each time that a new helicopter state vector is computed by the navigation sensor processing software, the inner-loop control algorithm transmits a copy of the state vector to the ground via a wireless ethernet link (running at approximately 2 Mbps). This allows a ground station to monitor the status of the helicopter at all times, and plot the helicopter's position, velocity, and other information of interest for human monitoring. The use of wireless ethernet allows seamless integration of additional vehicles (in the air or on the ground) or nodes – calling simply for the assignment of more IP addresses. Furthermore, the ability to reference and communicate with each vehicle simply via IP permits monitoring of vehicle status from anywhere using a simple Web browser-type interface, thereby avoiding the development of a custom groundstation of unneeded complexity.

**Safety Pilot**

When the inner-loop control receives a velocity command, and then determines how various control surfaces need to be deflected so as to respond to the command and maintain vehicle stability – these control surface actuations will be turned into servo deflections and transmitted via a serial line to a Motorola 68HC12 microcontroller. The microcontroller will have two input channels: one is the serial line from the on-board inner-loop control CPU, and the other is from the standard RC helicopter receiver, also carried on-board. It will also have output to the servos. Under normal autonomous operation, the 68HC12 will simply redirect the serial CPU input to the servo output. A safety pilot will monitor the health and status of the helicopter at all times during autonomous operation. Should he or she feel inclined to do so, they will be able to activate a reserved channel on their RC transmitter. The 68HC12 then will respond to the activation of this channel by switching off the serial input, and redirecting all input from other RC receiver channels to the servos, thereby giving the safety pilot control of the helicopter. For safety and robustness reasons, the 68HC12 and the airborne RC receiver will be powered by a battery separate from the main on-board power supply.

**Ground Robots**

The previously described hardware and software architecture of the aerial robot has been specifically designed to be flexible enough to permit the introduction of an arbitrary number of supporting autonomous ground robots. A description of all possible ground robot schemes and issues is beyond the scope of this paper, but a simple example is offered.

A ground robot whose sole function will be to deliver first-aid kits will be developed. The robot will consist of a simple electric vehicle platform with typical features (e.g. steering, speed control, etc.). The vehicle will be equipped with a mechanism that will store several first aid kits, and will be able to dispense them (e.g. round bottles rolling off a platform when a servo-controlled gate opens) on command. The robot will possess a small microcontroller (Motorola 68HC12, for instance), a wireless ethernet card, and a compass. The idea is to use the minimal amount of sensors and electronic to keep per-vehicle cost down while still be able to accomplish some useful component of the overall mission. The navigation and control of the ground robot will be almost completely an airborne task. The helicopter will sight the ground robot using its vision system (the ground robot can be painted pink or have some other visually distinguishing feature to make its identification from the air trivial). The aerial robot will then proceed to compute the position of the ground robot using the previously described geolocation algorithm. Based on its mapping mission, the aerial robot will be able to decide on an appropriate waypoint for the ground robot, and transmit the appropriate heading and velocity via the air-ground data link. The ground robot will then turn on the desired course and accelerate to the specified velocity. If it an obstacle that the aerial robot is aware of, the latter will transmit a course correction (i.e. create a new waypoint and issue a new heading and velocity) to the ground robot. When the aerial robot visually ascertains that the ground robot has reached its destination, a command to dispense the first aid kit or perform some other task will be issues via the data link.

Using this or a similar ground robot paradigm, a large number of relatively inexpensive ground robots will be deployed. The accidental loss of one such robot to an unexpected flame jet, or an unmapped water jets will, therefore, not compromise the entire mission. Furthermore, such a large team of robots with heterogeneous capabilities can serve as a unique and invaluable field testbed for many concepts in cooperative robotics previously unexplored.

## 1999 SYSTEM OVERVIEW

The 1999 system is a predecessor for the Millennial design, and incorporates many features of the latter. Some subsystems, however, are not sufficiently mature at present to facilitate field deployment, and are replaced by simplifications or off-the-shelf products which offer limited but sufficient capabilities to accomplish the goals of the 1999 IARC qualifier.

### The Flight Platform

The flight platform and equipment mounting system is identical the one planned for the Millennial Event.

## Navigation Sensors and Sensor Processing

The navigation sensor suite used in 1999 is identical to that planned for the Millennial Event and described above. The processing software, however, does not presently include a Kalman filter. Instead, the control loops for various axes rely exclusively on one navigation sensor. An effort has been made to use sensor readings instead of numerical calculation for each (proportional, integral, and derivative) term. For yaw control, the angle is measure directly by the compass and the yaw rate is measured directly by the IMU (and R/C gyro). For pitch and roll control, the angle is measured by the IMU accelerometers which tell us the direction that gravity points relative to the helicopter. For x- and y- position, the DGPS provides both the proportional and derivative terms. For altitude control, unfortunately, the sonar altimeter readings will need to be numerically differentiated and integrated up to 10 meters altitude, and the DGPS z-coordinate will be used above 10 meters. Because of high-frequency engine vibration, IMU readings are filtered using a software implementation of a second-order Bessel low-pass filter.

## Mission Sensors and Sensor Processing

The vision and audiogeolocation system are in early development stages, and consequently are not ready for field deployment. Instead, a Cognachrome vision board is used for object detection and recognition [11]; no audio capabilities presently exist. Geolocation is still accomplished by simple trigonometry – simplified even further by the fact that the camera does not have tilting capabilities (i.e. it is fixed). The thermister network is employed for flame avoidance as previously described.

## Inner-Loop Control

Inner-loop control functions as described in the Millennial design above. It must be noted that filtering of navigation sensor data does not use Kalman filtering for the state vector, as described in “Navigation Sensors and Sensor Processing” above.

## Outer-Loop Control

In order to accommodate the reduced object detection and identification capabilities offered by the Cognachrome vision system a simple, deterministic mission planning algorithm is being used instead of a subsumption architecture. The aerial robot will perform a simple rectangular search pattern over the area, and then return to the landing zone.

## Flight Monitor

The communication of helicopter state information is accomplished via the DGPS RF modem link, not a wireless ethernet. This will not easily permit the flexibility offered by IP addressing the helicopter, but with a single vehicle this flexibility is unnecessary.

Otherwise, the flight monitoring software is identical with what is expected for the Millennial Event.

### Safety Pilot

The safety pilot interface is identical with that designed for the Millennial Event.

### Ground Robots

No ground robots nor any supporting software capability is introduced in the 1999 competition.

## SYSTEM TESTING AND INTEGRATION BY SIMULATION

A general simulation environment has been developed in order to support the design of the autonomous helicopter control logic, as well as other current research activities on autonomous vehicle control at MIT and Draper Laboratory. The main advantages of this environment are its flexibility and its low cost, which make it a very useful research and training tool.

The simulation software architecture is based on a multi-process client-server approach, where MATLAB, a graphical 3D rendering process and external systems, like a remote controller or the on-board computer, interact to give a real-time simulation, with hardware-in-the-loop simulation capabilities (see Figure 2, below). This approach is particularly appropriate for the control law development and validation, as it allows the controls engineer to have full access to the MATLAB/Simulink environment and its facilities, while at the same time allowing for visual inspection of the behavior of the simulated system.  At the same time, it can be used to record and analyze flight data from a ‘‘manual mode” test flight, as well as to train prospective pilots with a direct R/C interface.

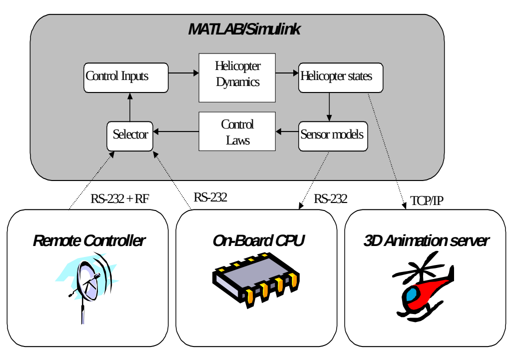


**Figure 2**

The flight vehicle control software development time is reduced to a minimum by relying on existing software for the handling of numerical data, user input/output, and control system design, analysis and evaluation.  The same simulation framework is currently used for nonlinear simulations of the autonomous helicopter, a fixed-wing UAV (WASP), and of the F-18 HARV.

## SUMMARY

The MIT Aerial Robotics Group has developed an autonomous aerial vehicle for performing a search mission in a simulated disaster zone as outlined in the competition requirements for the 1999 International Aerial Robotics Competition. The basic system architecture chosen involves a proven flight vehicle combined with an integrated avionics suite consisting of COTS products that are networked and controlled via a series of software and hardware units created by the MIT team.

The system design and development was based on creating mini-product teams that were chosen to minimize subsystem interactions, thus reducing overall system complexity. A systems interface engineer was assigned to ensure that all sub-system blocks would come together as intended and that signal and power protocols were established and adhered to early in the system development process. This allowed for a rapid prototype development process and enables easy subsystem reuse for the Millennial Event.

The MIT team goal of establishing a fully functional autonomous system for the 1999 competition was met and the use of simulation training and hardware in the loop simulation testing provided the necessary tools to fully develop and test the flight vehicle system on the ground. Coupled with the genetically evolved supervisory control system, it is felt that this aerial robotic vehicle will prove effective and competitive in the 1999 International Aerial Robotics Competition.

## ACKNOWLEDGEMENTS

The support of the following companies and organizations for this project is gratefully acknowledged: MIT Department of Aeronautics and Astronautics, United Technologies, Sideband Systems, Aironet, Novatel, Crossbow, and the Charles Stark Draper Laboratory. Furthermore, the following individuals' assistance is greatly appreciated: Ed Crawley, Steve Hall, Phyllis Collymore, Dick Perdichizzi, Bill Patrick, Jack Davis, Brent Appleby, and all of the MIT faculty and Draper staff who have provided invaluable guidance and advice to the students on this project.

## REFERENCES

[1] Ancona, N. and Poggio, T., 1995, "Optical flow from 1-D correlation: application to a simple time-to-crash detector", *International Journal of Computer Vision*, 14(2), pp. 131-146.

[2] Beal, J. and Rasmussen, S., 1998, "Hawk: Embodied Intelligence for Mission Planning in the 1998 International Aerial Robotics Competition", MIT Aerial Robotics Memo, Cambridge, MA.

[3] Brooks, R., 1985, "A Robust Layered Control System for a Mobile Robot", MIT Artificial Intelligence Laboratory Memo 864, Cambridge, MA.

[4] Gavrilets, V., 1998, "Avionics Systems Development for Small Unmanned Aircraft", MIT Aero/Astro Master's Thesis, Cambridge, MA.

[5] Gelb, A., 1974, *Applied Optimal Estimation*, MIT Press, Cambridge, MA.

[6] Johnson, E. et al., 1996, "The Draper Small Autonomous Aerial Vehicle Technical Description", Draper Laboratory Technical Report, Cambridge, MA.

[7] Michelson, R., 1999, "International Aerial Robotics Competition Millennial Event", Georgia Tech Robotics Institute Web, Atlanta, GA.

[8] Negahdaripour, S., 1987, "A Direct Method for Locating the Focus of Expansion", MIT Artificial Intelligence Laboratory Memo 939, Cambridge, MA.

[9] Taalebinezhaad, M., 1992, "Robot Motion Vision by Fixation", MIT Artificial Intelligence Laboratory Technical Report 1384, Cambridge, MA.

[10] Watt, R., 1986, "Feature-Based Image Segmentation in Human Vision", *Spatial Vision*, Vol. 1, pp. 243-256.

[11] Wright, A. et al., 1996, "Cognachrome Vision System User's Guide", Newton Research Labs, Redmond, WA.

## ENDNOTES

† Also affiliated with the Charles Stark Draper Laboratory.

‡ Associate Professor of Aeronautics and Astronautics, faculty advisor.