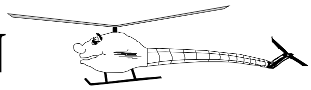
MARVIN



Technische Universität Berlin's Flying Robot for the IARC Millennial Event

M. Musial U. W. Brandenburg G. Hommel

Technische Universität Berlin
Department of Computer Science
Real-Time Systems & Robotics Group
Contact: *http://pdv.cs.tu-berlin.de/MARVIN/*

Abstract

MARVIN is an autonomously flying robot based on a model helicopter. It is designed for participation in the International Aerial Robotics Competition (*IARC*) Millennial Event 1998 – 2000. The competition task consists of a search operation in an unknown environment with possible threats such as fires, water fountains, and smoke. The MARVIN system consists of the helicopter with an on-board computer and a network of Linux laptops serving as a ground station. While flight control is done on-board, mission planning, digital image processing, and the display of the search results take place on ground. Sensors for autonomous operation include carrier phase DGPS equipment, a handcrafted IMU consisting of acceleration, magnetic field, and rotation sensors, an ultrasonic transducer, a radar sensor, and a flame sensor. Image acquisition is done through a digital photo camera.

INTRODUCTION

MARVIN is an abbreviation for **M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntelligent **N**avigation. MARVIN is an autonomously flying robot that can fulfill a complicated search mission purely on the basis of sensor data, without any human interaction.

The requirements to be fulfilled by MARVIN are defined by the mission of the International Aerial Robotics Competition Millennial Event, as described in [1]. The mission task consists of finding and classifying hazards and victims in a simulated disaster area. There are black drums containing dangerous materials, with the contents distinguishable by symbols on the drums' surfaces, and there are fires, water fountains, and smoke threatening the operation of the robot. Victims may be dead or injured persons, the survivors recognizable by motion and sound.

In the 1999 qualifier event, MARVIN was the only participant to fly autonomously during performance judging.

The remaining sections of this paper deal with overall system design, safety and emergency procedures, the computer vision subsystem, and the mission execution strategy.

SYSTEM DESIGN

Figure 1 provides an overview of the MARVIN system. During system development, every detail has been carefully optimized with respect to weight, prize, power consumption, and capabil-

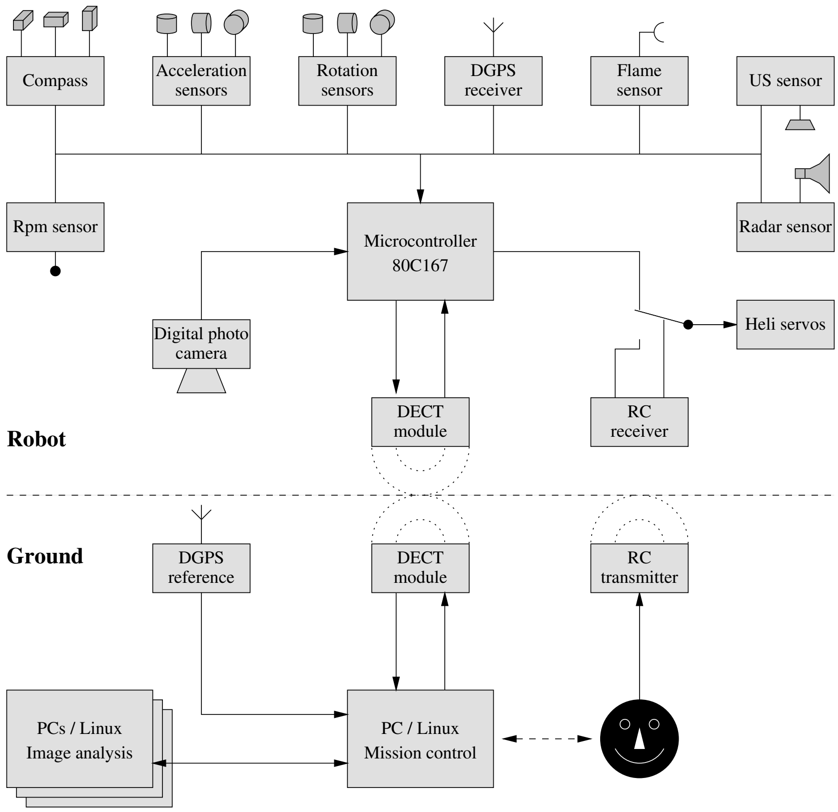


Figure 1: Overview of the MARVIN system

ity. Much care has been taken to select components that fulfill the respective tasks and integrate smoothly into the system, always avoiding things being "better than needed" at the expense of some of the criteria mentioned above. Basically this made MARVIN's success in 1999 possible.

Air Vehicle

The basis of the MARVIN air vehicle is a conventional model helicopter, the "Petrol Trainer" 180 by SSM, Germany. It has a rotor diameter of 1.9 m and is equipped with a two-stroke chain saw petrol engine producing about 2 kW. Its takeoff weight amounts to about 11 kg, operation time is approximately 20 minutes. Since fully autonomous takeoff and landing have been implemented, the search time can be extended to the 60 minutes time-slot in performance judging by refueling during the mission. Figure 2 shows a view of MARVIN in flight.

Sensors

The sensors on board MARVIN are:

- A hand-crafted IMU consisting of three magnetic field sensors, three semiconductor acceleration sensors, and three piezo-electric rotation sensors (material costs below $250).
- A flame sensor that detects certain ultraviolet light characteristic of burning wood, gas, or oil.
- A light barrier rpm sensor for measuring the main rotor rpm.
- An ultrasonic echo sensor looking down.

Figure 2: MARVIN in flight

- A radar sensor, based on frequency modulation and frequency shift detection, looking ahead.
- A digital photo camera looking down.
- A NovAtel RT-2 carrier phase differential GPS receiver (thanks to NovAtel).

Computers

The only source of computational power on-board is a single single-chip computer, an Infineon (formerly Siemens) SAB80C167 microcontroller. This device is perfectly capable of interfacing all the on-board peripherals without additional circuitry. It can, for example, measure or generate all PWM signals to the servos without consuming a single CPU clock cycle for this task. The circuit boards for the on-board computer have been handcrafted by the MARVIN team.

A network of Laptop PCs under Linux serves as a ground station for image processing and user interaction. The number of PCs and the distribution of tasks across the ground station is completely transparent to the software, no single line of code needs to be changed when a new configuration is to be used.

Communication

Data communication between MARVIN and the ground station is performed via two pairs of Siemens Gigaset M101 Data communication modules. Each pair provides a wireless serial "null-modem cable" connection at 107 kbit/s full-duplex, yielding 214 kbit/s of bandwidth.

The communication software implements a *shared memory* approach to communication: Both the on-board computer and the ground station modules use a dedicated memory area containing *all* the variables relevant to the system state. Whenever data are changed within the shared memory area either on-board or on ground, this change is distributed to all remaining copies of the shared memory through the exchange of data packets. To the software, this looks as if all modules were running in a single program and on a single computer. The state variables in the shared memory include (but are not limited to):

Downward communication: MARVIN's position, speed, and orientation; rotor rpm; raw sensor readings; current servo positions; ultrasonics, radar, and fire measurements; CPU load of the on-board computer; current state and solution quality of the GPS receiver; portions of the last taken picture.

Upward communication: Flight phase and target coordinates for the flight control software; take-off or landing command; GPS reference data packets; desired image resolution; parameters of the flight controller; data for compass calibration.

The fact that the controller parameters can be transmitted via the shared memory has proven a great alleviation in the testing process: A new set of controller parameters can be uploaded on the fly – literally.

Since all the state variables are constantly visible at the ground station, they can be logged to one of the laptops' hard disks even during flight phases. This kind of "flight data recording" can be evaluated even before MARVIN is back on ground. Typically, about 15 MB of data are recorded per hour.

SAFETY PROCEDURES

This section describes the safety measures provided in the MARVIN system for cases of failure of operation. There are three stages of such procedures: Failsafe mode in case of GPS malfunction, a procedure for secure retrieval of the robot and the flight termination procedure as required by the competition rules.

Both retrieval and termination are based on a separately powered conventional remote control (RC) unit, which is not used during autonomous operation. The input signals to the servos on-board can be switched between the remote control receiver (manual flight mode) or the on-board computer (autonomous flight mode) via the RC unit (see figure 1). This switching is done through relays that fall back mechanically into the manual position when the on-board computer suffers a power failure or is reset due to any reason.

Failsafe Mode

In case of data communication breakdown, the GPS receiver gets no reference data, which causes a significant loss of GPS accuracy after some seconds. (By the way, the RT-2 shows a tendency to temporarily lose maximum accuracy without obvious reason – this is referred to as "GPS unhappy state" the the MARVIN team.) In such cases, *failsafe mode* is entered, meaning that the on-board flight control software keeps a horizontal orientation without ground station interaction for a time only limited be the fuel reserve. This prevents MARVIN from crashing in the cases mentioned, but the robot typically begins to drift due to the loss of position information.

Secure Retrieval

The first and normal stage of externally triggered safety procedures is that the human backup pilot switches control back to manual mode using an extra switch on the RC transmitter whenever

MARVIN's flight behavior becomes unsatisfactory. Thus, he takes over control and lands MARVIN safely in no other way than a conventional model helicopter. This approach (and the team's pilot) have empirically proven to be reliable in some 100 cases up to now.

Termination Procedure

Throwing a second extra switch on the RC unit, the autorotation switch, combines switching back the servos to manual mode with instant engine shut-off. Its use renders the robot ballistic according to the competition rules. But there are still 11 kg of mass and a rotor revolving at 1,100 rpm in the air – be assured that the first stage of "retrieval" is far more desirable in more or less all possible cases.

COMPUTER VISION

This section describes the approach to image acquisition and outlines the image analysis algorithms used.

Image Acquisition

Since analog video data transmission is expensive, heavy, power consuming, susceptible to interferences, and/or of low image quality, MARVIN carries a usual digital photo camera. Image data are acquired via the camera's serial interface by the on-board computer and transmitted to the ground station via the digital data link. The camera weighs only 200 g and transmits a JPEG-compressed image every 5 seconds at a resolution of 640×480 pixels, or every 13 seconds at 1024×768 pixels. These images are of brilliant quality and free of disturbances. Since a single image is sufficient for the recognition of static objects, an image rate such low does not cause trouble – between the taking of two images, MARVIN should change its position, anyway.

Image Analysis

There are always uncountable approaches to the detection of some sort of objects in an image, and typically, some of them may perform worse or better depending on the respective nature of the depicted scene. Taken into account the unpredictable structure of the disaster scene, the image analysis part of the MARVIN system has been designed in a redundant way: There are a number of computer vision modules running distributed on the ground station network, each module specialized in a certain object class, but for a given object class there may be more than one vision module. If one algorithm encounters difficulties due to adverse environmental conditions, it is probable that another approach to recognizing the same object class will not suffer the very same problem.

Every object recognition by a MARVIN vision module is attributed with a probability expressing the degree of certainty with that the recognition has taken place. This information is used by Mission Control (see below) to handle inconsistent results from different modules.

The following table shows subsections describe the different vision modules implemented in the MARVIN system.

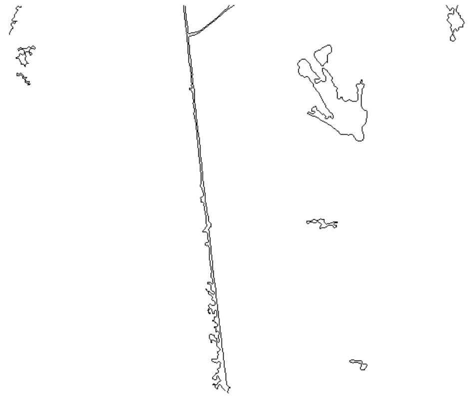


Figure 3: Human body contour in a 1999 competition image

Potential Objects in Overview Images

Vision modules #1 and #2 detect objects which could be drums or bodies in overview images. Fortunately, drums and bodies are similar in size. Module #1 searches for compact segments of a distinguishable color and suitable size, basically using Spann-Wilson-segmentation [4]. Module #2 detects suitably sized image regions the colors of which appear especially seldom in the image. These regions should belong to limited objects distinct from the background.

Drum Labels

Vision modules #3, #4, and #5 are responsible of identifying drum labels. Algorithm #3 computes correlations between pre-defined label masks and the image at all possible positions and orientations. For efficiency, only binary and sparse pixel masks are used. Module #4 collects rectangular segments within the binary image by edge tracing and cuts out the potential labels from the original image using these rectangles. These cut-out regions are validated through cross-correlation with example labels. Algorithm #5 calculates 7 central moments for every image segment, yielding 7-dimensional feature vectors which are classified by an Euclidean distance classifier to identify labels or parts thereof.

Human Bodies

Vision modules #6 and #7 serve for the detection of human bodies. This kind of task has been subject to many research activities shortly, [3] presents a useful survey. Algorithm #6 scans the image for edges and extracts a mathematical representation based on curved line segments. These line segments are than checked against a system of rules describing possible outline of a human body. Module #7 extracts seldom colors similar to algorithm #2, then draws contours around the regions found (see figure 3). These contours are classified using a predefined decision tree of possible body contours and the Chamfer distance [2] for the comparisons. Both of these two modules compare subsequent recognitions of the same body with respect to the respective body models, so that survivors can be identified by the varying shape of their bodies.

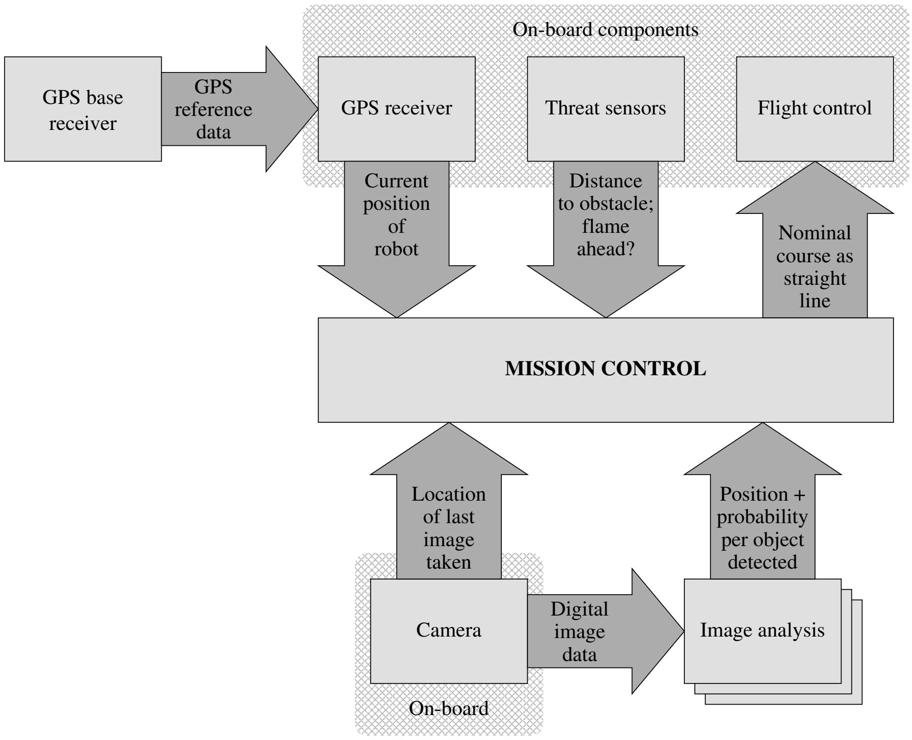


Figure 4: Flow of information between the components involved in mission execution

MISSION CONTROL

This section explains the navigation scheme for fulfilling the competition's search mission, its realization, and how the mission results are presented to the human operators (and competition judges).

Flow of Information

Both the mission strategy and the user interface are implemented in the Mission Control program, which runs on one of the ground station laptops. Figure 4 depicts the information exchange between the units involved in mission execution. All information exchange takes place via the shared memory approach. Mission Control acts as the principal link between the other components. It receives the current position, the threat data, the notifications about the images taken, and the findings of the image analysis modules. It decides about the next flight path segment, which is propagated to the flight control software on-board. Flight control, usually referred to as the *pilot* module, is responsible of autonomously following the current flight path segment. The only assistance to this task from the ground station consists in the GPS reference data stream required by the on-board GPS to function properly.

User Interface

A screen-shot of the graphical user interface to Mission Control is shown in figure 5. The core of the Mission Control windows is a slidable and scalable view of the search area map. This map contains recognized target objects, recognized obstacles and threats, and image targets. The latter are positions from that images shall be taken. The map view shows the current intended flight path,

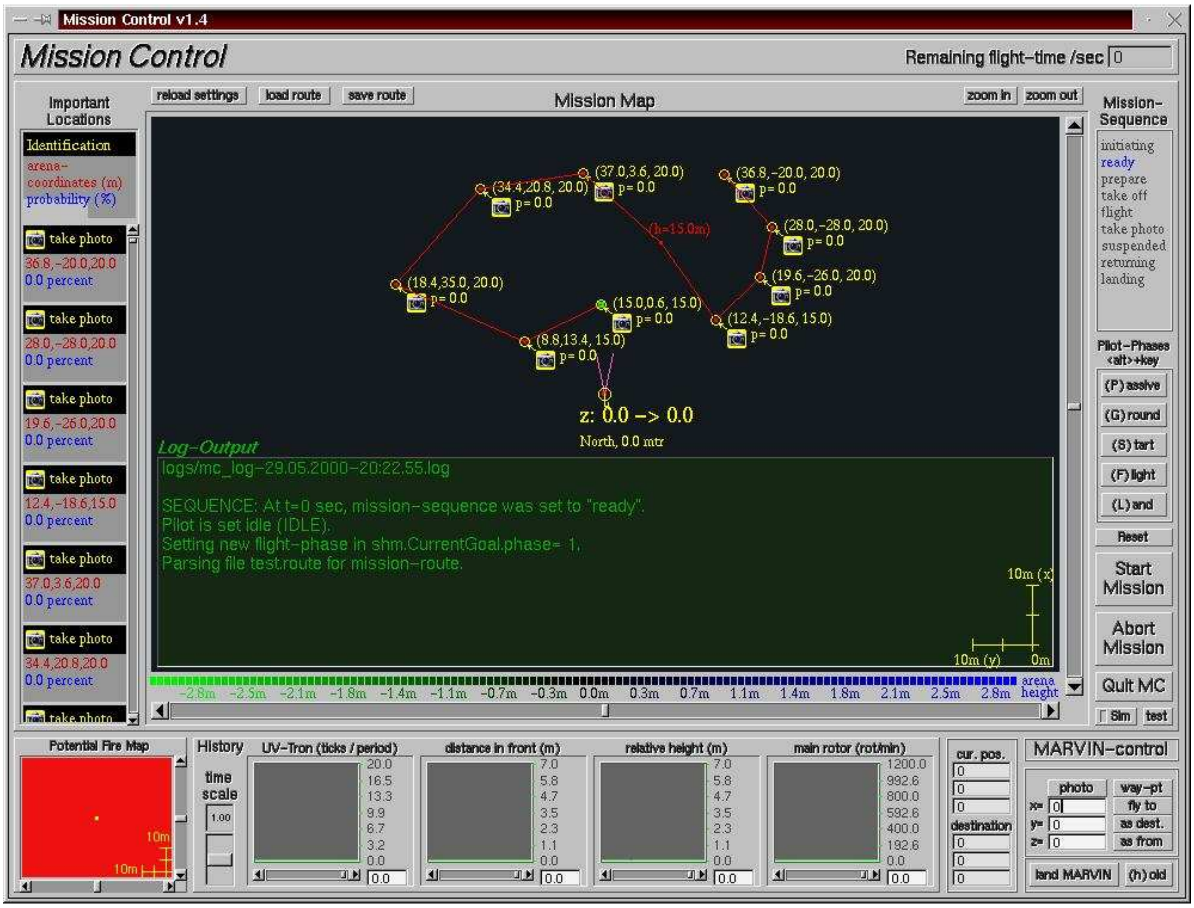


Figure 5: Mission Control, strategy implementation and user interface

connecting all the image target positions via a path optimized with respect to the path length and as few changes of altitude as possible. MARVIN's current position and heading are also depicted in the map. The lower part of the map area displays messages from Mission Control regarding activities planned or executed.

To the left of the map area, there is a scrollable list of all the map entries. This alleviates the reading of the coordinates and types of recognized objects in order to – say – present the search results to the judges.

To the right of the map area, there are a display and some buttons to read and alter the state of the mission and the pilot module. This is mainly for the development and testing process – at the competition, the only interesting thing here will be the "start mission" button.

Below the map area, there are diagrams visualizing some sets of sensor data as a function of time. This includes the ultrasonics and radar distance readings, the information from the fire sensor, and the main rotor rpm. The bottom left corner contains the *potential fire map* as described below in the threat avoidance section.

Basic Mission Execution

When the mission is started by clicking the "start mission" button, the pilot module is instructed to increase the engine throttle until the main rotor revolves at 1,100 rpm. Then, Mission Control issues the *takeoff* command, causing MARVIN to perform a quick vertical launch and stabilize at an altitude of 13 m above the takeoff position.

Then, Mission Controls tells MARVIN to subsequently fly to all of the image target positions present in the search area map as explained above. For mission start, the list of image targets is manually set up as a rough search pattern throughout the search area with target points at an altitude of 50 m. When MARVIN follows the flight path connecting the image target positions, it transmits images to the ground station, where these images are analyzed by the vision modules. As soon as an image target has been reached and the desired image taken, this target position is cleared from the map.

Whenever one of the vision modules recognizes an interesting shape or spot, the respective position is sent to Mission Control, which creates a new image target position above the potential object in question, using the horizontal coordinates of the recognized shape but a substantially lower altitude of 5 m. In this way, the set of image target positions is augmented according to the findings of the vision modules. The process of mission execution is mainly based on the creation of new image target locations and their following and subsequent clearing. Mission Control also take care of fusing multiple vision messages that are probably related to the same physical object.

The images taken from an altitude of 50 m are *overview images* giving just a rough impression of larger parts of the search area. From this altitude, potential drum and body shapes may be found, but the classification of the drum labels and the reliable recognition of body shapes remains to the detailed images taken from 5 m above the object in question. This means MARVIN "looks for interesting things" from 50 m above the search area and descends to 5 m in order to "have a closer look" for verifying if the interesting things are relevant to the search task and for reliable classification. During the 1999 performance judging for example, MARVIN descended to inspect a rectangular drain lid and judged it uninteresting after analyzing the detailed image.

When a predetermined flight time of 17 minutes is over (meaning that MARVIN gets short on fuel) or the set of image target positions becomes empty (meaning that everything interesting has been inspected), Mission Control guides MARVIN back to the takeoff location and initiates an autonomous landing maneuver there.

Threat and Collision Avoidance

Obstacles and threats are also entered into the search area map according to the data from the obstacle and threat sensors. Obstacles are modeled as cylinders, with an x-y-center-point in the search area plane, a height, and a radius. Solid obstacles can be detected by the radar when MARVIN looks towards them. Therefore, when MARVIN operates less than 20 m above the ground, Mission Control regularly lets the robot scan its environment for obstacles by turning itself around the vertical axis while hovering. Based on the distance reading from the radar and the current heading, an obstacle's position is determined and entered into the map.

Flame positions have to be obtained in another way, since the flame sensor does not give any distance information – it is unable to distinguish between far big flames and near small ones. Therefore, Mission Control uses a *potential fire map* map of 1 m^2 raster cells, each of which having one of the three states *probable fire*, *no fire*, or *potential fire*. The map is initialized to *potential fire* everywhere. All the time, the fire sensor's current view sector within the potential fire map is set to *no fire* whenever the sensor reads negative, and *potential fire* cells within the view sector are set to *probable fire* whenever it reads positive. Only *no fire* cells are considered safe for flight path planning, and *probable fire* areas completely surrounded by *no fire* cells are identified as actual fires. The latter are both entered into the search area map and displayed to the user (and judges). For visual control, the potential fire map is part of the Mission Control window (see user interface section).

Water fountains are detected by the radar sensor based upon the Doppler effect caused by the relative movement of the water. Since there are no other quickly moving objects in the search area, such kinds of radar echos are marked as water fountains in the area map.

When Mission Control generates flight path segments to reach an image target location, it checks whether a segment intersects one of the obstacle cylinders in the search area map. In this case, a new intermediate way-point is generated by lengthening the cylinder radius perpendicular to the intersecting path segment by 2 m, a safe distance for passing the obstacle. This process is repeated until no more intersections take place. Additionally, unless both the fire sensor and the obstacle radar yield negative readings ahead, MARVIN's heading is changed clockwise until a view sector clear of obstacles and threats is found, which is safe for moving.

CONCLUSION AND ACKNOWLEDGEMENTS

MARVIN is an autonomously flying robot capable of executing search missions without human interaction. During its development for the participation in the International Aerial Robotics Competition, the careful selection of components according to a principle of "requirement minimalization" has proven critical to successful design. The most important design aspects within the MARVIN system are the digital photo camera, the handcrafted IMU, the selection of the on-board computer, and the flexible shared memory approach to communication.

MARVIN would never have been possible without the enthusiastic engagement of many, many team members: Marc Bartholomäus, Eike Berg, Carsten Deeg, Christian Fleischer, Oliver Klenke, Stefan Pohle, Christian Reinicke, Volker Remuß, Andreas Rose, Roland Stahn, Andreas Wege, and even more. Matthias Jeserich, our (non-software) pilot, is the only system component that has never failed. For lots of valuable support, we wish to thank: NLV Saarmund for welcoming us on their helicopter "airport" for testing; NovAtel Inc. for the loan of the GPS equipment; Prof. Dr. J. Otto of FH Aalen for his support regarding the radar sensor; DaimlerChrysler Aeorospace AG for the first helicopter; Rotor Modellsport Center, Berlin, for their advice and lots of spare parts; Siemens AG for a PC, some 20 microcontrollers, financial support, and their development of the marvelous Gigaset M101 Data module; Tasking BV for the rebate on our C167 development software; TFH Berlin for providing their board production facilities.

REFERENCES

[1] International Aerial Robotics Competition: The Robotics Competition of the Millennium. http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/IARC2000Intro.html.

[2] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, 1996.

[3] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[4] M. Spann and R. Wilson. A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition*, 18(3/4):257–269, 1985.