

# **Towards Vision-Based Autonomous Landing for Small Unmanned Aerial Vehicles: Image Processing Hardware Development**

H.-W. Schulz\*

*Technische Universität Braunschweig, Braunschweig, 38108, Germany*

DOI: 10.2514/1.35127

**This paper describes the development of intelligent vision capabilities for small-unmanned aerial vehicles as part of the program CAROLO at the Technische Universität Braunschweig, Germany. The development's objective is to create appropriate hardware that suits the requirements of small- unmanned aerial vehicles (5 kg maximum takeoff weight). Enabling vision-based flight guidance during landing was chosen as the first task and acts as a technology demonstrator for future payload-driven vision capabilities. The paper describes the developed image processing hardware and its capabilities. Attention is also paid to implementing image processing algorithms for detecting a known landing point marker and computing its relative position and attitude.**

## **I. Introduction**

**T**HE Institute of Aerospace Systems of the Technische Universität Braunschweig, Germany, has developed a family of mini aerial vehicles and small unmanned aerial vehicles (UAVs) ranging from 49 cm up to 330 cm wingspan. The ultimate goal of the program CAROLO was to build an aircraft capable of flying fully autonomously with the smallest wingspan possible. This was accomplished in 2004 using CAROLO P50 (49 cm wingspan). All aircraft are capable of flying fully automatically from takeoff to landing using preprogrammed waypoint navigation and an autopilot, which was also developed at the Institute [1].

To enhance the mission capabilities, the Institute decided to develop an onboard image processing system which fulfills challenging requirements regarding mass, dimensions, and power consumption of small UAVs such as CAROLO T200. The feasibility of such a system should be demonstrated by a fully autonomous vision-based landing.

### **A. Motivation**

In the field of UAVs, image processing is used either for flight guidance and control (e.g. navigation) or for mission related purposes, as part of the payload. Currently, the latter rarely found in small UAVs, but a good example is establishing the relative position of a target from a video signal received on the ground and sending it back to the aircraft for camera control [2]. More commonly, vision-based systems are used to extract horizon information [3–6], to support landings [for example 7–11] or, with increasing importance, to act as a sense and avoid system [12]. Two approaches can be distinguished: image processing is either done on the ground and information is sent back to the aircraft or the task is addressed by onboard hardware. However, in most cases where image processing is done onboard, mass, dimensions and power consumption of current systems prohibit use of this hardware on board particularly small UAVs. A gap in capabilities exists for unmanned aircraft too small to use currently available image

---

Received 14 October 2007; accepted for publication 1 May 2008. Copyright © 2008 by H.-W. Schulz. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1542-9423/08 \$10.00 in correspondence with the CCC.

\* Former Research Associate, Institute of Aerospace Systems, AIAA Member. Since June 2007 the author is a Senior Aerospace Systems Engineer with ESG in Munich, Germany, ha.schulz@tu-bs.de

processing hardware onboard. This gap cannot be closed by doing the image processing on the ground because this would decrease range and safety.

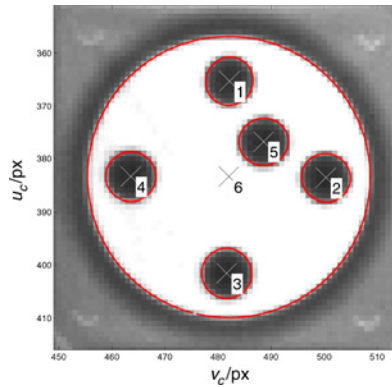
### B. Vision-Based Landing Procedure

For vision-based landing, a known marker, indicating the preferred landing point, is recognized by an onboard camera and its position is determined. The system described here will be implemented in the fully autonomous small UAV “CAROLO T200” depicted in Fig. 1. Aircraft of this category often land directly on the fuselage without need for landing gear and then come to a halt within less than 5 m. Assuming a reasonably flat and level landing field, it should be sufficient to specify a landing point rather than a landing area. To match the high mobility of mini and small UAV systems, the marker is made of a synthetic canvas cover, which is clamped to a foldable frame. This provides a lightweight but dimensionally stable marker which can be laid out easily at the desired landing point by the UAV’s operator (Fig. 2).

Ideally, the camera used to determine the altitude above the landing point will be the same camera used for surveillance during the aircraft’s mission. Most of these cameras are downward-facing cameras. The landing procedure and the image processing algorithms have to take this into account. With a downward-facing camera the marker will be in the field of view of the camera for a short period of time only. That is why, contrary to other current approaches that use an onboard camera and visual ground aids, the marker does not have to be in view constantly. Theoretically, one measurement is sufficient to determine the navigation solution. Nevertheless, the flight path before landing is planned in a way that the aircraft passes the marker several times. Starting at a higher altitude, where the field of view is larger, the UAV will fly over the assumed landing point position. Using the measurements of the first pass, a new flight path can be computed to pass the marker at a lower altitude, providing even better measurements. Simulation [13] and experimental [14] results show that three passes at 60 m, 45 m and 30 m will be sufficient for a safe landing.



**Fig. 1 CAROLO T200: a fully automatic small UAV, 2 m wingspan, 5 kg maximum takeoff weight, 2 kg payload (pictured here with wind measuring 9-hole-probe and other meteorological sensors).**



**Fig. 2 Identified landing point marker (2 m diameter), detail of original image taken from an altitude of 40 m.**

## II. Image Processing Hardware Development

At the 2005 Infotech@Aerospace Conference the current author presented algorithms to extract features of a landing point defining marker known a priori and to compute its position and attitude relative to the camera coordinate system [14, 15]. An example of the identified marker can be seen in Fig. 2. The achievable accuracy had been verified through experiments and was published in the same paper. These ground-based experiments were based on the camera which will be used for the flight tests as well. Image processing was done using off-the-shelf PC hardware.

This paper extends the previous work by describing the development of image processing hardware for the image-based autonomous landing procedure. In the future, only the aircraft's onboard hard- and software shall be used to recognize the desired landing point and compute its position and attitude.

### A. Overall System Requirements

CAROLO T200 is a good example of a small UAV, namely of the mini aerial vehicle class. It was therefore chosen as the test platform and all aircraft related requirements were derived from CAROLO T200. With 2 m wingspan and a maximum takeoff weight of 5 kg this unmanned aircraft offers up to 2 kg of payload capacity. The cross section of this aircraft, dominated by the overhead payload bar as illustrated in Fig. 3, defines the maximum dimensions of the image processing system. Maximum mass is restricted to 800 g to enable the use of additional sensors (e.g. tactical grade IMU at 800 g) at the same time. Nevertheless, a maximum mass of 500 g would be preferred, to enable the possibility this image processing hardware could be used on smaller aircraft. A power consumption requirement can be set analyzing the propulsion system of CAROLO T200. As with most of the mini aerial vehicles this aircraft has an electrical propulsion system and its accumulators make up most of the aircraft's mass. To minimize accumulator mass needed for the payload, power consumption of the image processing system is restricted to 10% of cruise power. Power consumption of the two three-phase alternating current motors is 100 W during cruise. Therefore, 10 W is defined as maximum power consumption of the image processing payload system. The overall system requirements are summarized in Table 1.

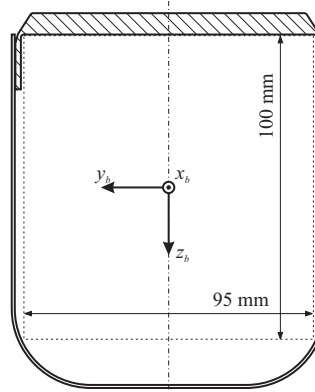


Fig. 3 Body cross section of CAROLO T200.

Table 1 Overall system requirements

Mass	<800 g	F
	<500 g	S
Length	<150 mm	S
Width	<95 mm	F
Height	<100 mm	F
Volume	<(90 mm) <sup>3</sup>	S
Power consumption	<10 W	F

F: fixed requirement

S: soft requirement

## B. Camera System

For the vision-based landing procedure presented in references [14] and [15], a camera system is needed. This system must be as versatile and flexible as possible to study different approaches of vision-based flight guidance and control as well as vision-based, more payload oriented tasks. Currently, cameras in micro and mini aerial vehicles are almost exclusively digital cameras with analog output signals which are relatively easy to transmit. To process these images by means of machine vision these analog signals must be converted back to digital signals. Conversions, especially a potential analog radio transmission, increase signal noise. To avoid low signal-to-noise ratios a camera with digital output signals should be used. Using a standardized interface, the camera could later be exchanged for a different imaging sensor, for example an infrared camera for meteorological use. An image from a black and white camera is sufficient for vision-based landing, where the landing point marker must be identified reliably.

From the experiments in references [14] and [15], must be identified reliably it is known that the marker must be resolved by at least 40 pixels. This information can be used to derive the following requirement: the camera sensor should feature roughly 1 Megapixel and the focal length of the lens should be bigger than 1000 pixels. This configuration maximizes the field of view of the camera, provides adequate resolution for altitudes of up to 60 m and limits the number of pixels. The latter is essential if the new image processing system needs to process the image information in or near real time. All requirements for camera and lens are summed up in Table 2.

Based on the requirements in Tables 1 and 2, camera and lens were chosen. The camera is a Sony XCL-X700 (XGA), which is a black and white-camera with a 1/3" charge-coupled device (CCD)-sensor. This sensor has  $1024 \times 768$  sensitive pixels and color depth is 10 bit, that is 1024 shades of gray. These signals are output digitally according to the Camera Link standard. This standard was specified for industrial machine vision applications, especially for fast digital video data transmission. The camera uses a variant of the Camera Link, called Camera Link Base, which allows for up to 24 bits per pixel. The Sony XCL-X700 outputs the digital pixel data in parallel with a data rate of 30 MHz. Maximum image rate is 30 Hz and single images following a trigger are supported. To use the Camera Link interface, the image processing hardware must feature an appropriate low-voltage differential signaling (LVDS) interface. Using the standardized Camera Link interface for the new computer, different imaging sensors and off-the-shelf hardware can be supported. For example, during the aforementioned experiments a commercial frame

**Table 2 Camera and lens requirements**

Camera type	Visible light	F
Image rate	Min. 10 hz	F
	Max. 30 hz	S
Exposure control	Automatic	S
Exposure time	<1/500 s	F
	<1/4000 s	S
Trigger	Extern	F
	single image	S
Color depth	256 scales	F
Resolution	Min. 1024 px $\times$ 768 px	F
	Max. 1024 px $\times$ 1024 px	F
Output signal	Digital	F
	standardized	F
Lens mount	Standardized	F
lens type	fixed focal length	F
Focal length	$\geq 1000$ px	F
Focus	Manual	F
Imaging quality	High	S
Max. aperture	min. 1:2.0	S

F: fixed requirement

S: soft requirement

grabber with Camera Link interface was used to connect the camera with PC-hardware to test algorithms, while the new image processing system was still under development.

The Camera Link interface includes a serial interface to communicate with the Sony XCL-X700. Unfortunately, the camera does not come with an automatic exposure control. Exposure time must be provided by the image processing computer. Lacking an exclusive sensor for exposure, the imaging sensor has to become part of a control loop: the exposure of every image taken is evaluated and a corrected value for the exposure time computed. Provided that the image processing sensor can process the exposure evaluation fast enough, the camera's image rate of 30 Hz provides sufficient speed for a fast exposure control. As will be shown later, the design of the image processing computer respects this requirement.

Despite this slight disadvantage the chosen camera brings one big benefit: currently, the Sony XCL-X700 is the smallest camera available with standardized digital interface. Measuring 50 mm  $\times$  30 mm  $\times$  30 mm and with a mass of 50 g only this camera is well suited for integration in small unmanned aircraft.

Using the specifications of the camera, the lens can be chosen next. Knowing that the CCD-chip features 1024 pixels  $\times$  768 pixels at a chip width of 5.8 mm, the minimum focal length of the lens can be computed as 5.66 mm. Choosing a focal length close to this theoretical minimum, the field of view can be maximized while still ensuring sufficient resolution of the landing point marker at the same time. The lens chosen is the Cosmimar Television Lens H612A made by Pentax. This lens has a focal length of 6 mm, resulting in a field of view of 48.3 m  $\times$  41 m at an altitude of 50 m. Focus and aperture have to be operated manually. The lens is fast with a maximum aperture of 1:1.2. Developed for industrial machine vision the lens delivers high imaging quality. However, this quality comes at a cost: even though the lens is relatively small compared to most other lenses from the field of industrial machine vision, dimensions, and mass are quite high compared to other elements of the new image processing system. Choosing a lens is always a compromise between high quality and price on the one hand and low mass and dimensions on the other. For this first prototype quality was valued higher than mass and dimensions, but with the standardized lens mount the lens can be exchanged easily in the future.

Table 3 sums up the specifications of camera and lens whereas in Fig. 4 camera and lens are shown during the experiment mentioned earlier.

**Table 3 Camera and lens specifications**

Camera type	Sony XCL-X700 (XGA)
Image rate	Max. 30 Hz
Trigger	Extern, freely configurable single images enabled
Exposure time	Max. 1/4 s Min. 1/100.000 s
Color depth	1024 gray scales
Resolution	1024 px $\times$ 768 px
Output signal	Camera Link Base (LVDS)
Software interface	RS232
Lens mount	C-Mount
Camera length	50 mm
Camera width	30 mm
Camera height	30 mm
Camera mass	50 g
Power consumption	2.2 W at 12 V
lens type	Cosmimar Television Lens H612A
Lens focal length	6 mm
Lens focus	manual
Max. lens aperture	1:1.2
Lens length	45 mm
Lens diameter	43 mm
Lens mass	120 g



**Fig. 4 Camera Sony XCL-X700 and lens Pentax H612A.**

### C. Selection of the Image Processing Computer

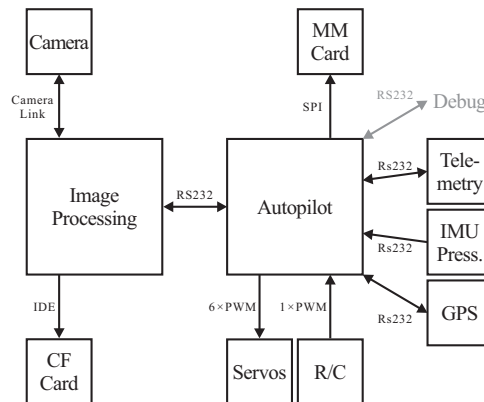
The image processing computer shall be used on board small UAVs to process image data in real time. Besides the requirements regarding mass, size, and power consumption posed by these small aircraft, the new computing system needs to be as flexible as possible to not only support the vision-based landing procedure but image or signal processing tasks in general.

Two different approaches are feasible: the image processing computer could be seen as a modular component that works in parallel with the autopilot, or the functions of the autopilot and the image processing computer could be highly integrated and share the same platform. Only the first approach is described here. However, it will be shown later that the resulting computer is well suited to implement the integrated approach as well.

As an independent module the image processing system could, depending on the mission, communicate with the onboard autopilot or a ground station. The system would be an intelligent payload sensor, of interest for other developers/manufacturers as well. Integration into other airborne systems would be relatively easy.

For the task at hand—vision-based landing—position and attitude of the aircraft at the time of exposure must be provided by the CAROLO autopilot. Therefore, a connection between the imaging computer and autopilot is needed. This connection is also used to transmit the position of the landing point marker and the target landing track angle to the autopilot where this information is used as input for flight guidance and control algorithms during landing.

Figure 5 shows a schematic representation of the image processing computer working in parallel with the CAROLO autopilot. The system consists of a camera connected via Camera Link, a compact flash card for data storage and the autopilot for remote control, servo control, navigation, flight guidance and control as well as communication with the ground station. This configuration leads to the interface requirements for the image processing computer: Camera Link Base, CF-Card (IDE) and serial interfaces (RS232) are needed.



**Fig. 5 Image processing system as single add-on module.**

Additional requirements apply to all image processing systems in general: large amounts of data have to be processed efficiently. If the processing must be done in real time, high data throughput is required as well. For these requirements special integrated circuits exist: digital signal processors (DSP) and field programmable gate arrays (FPGA). Both are specially designed for signal processing but have quite different characteristics.

DSPs are optimized for the mathematical operations that are typical for signal processing. One such characteristic operation is convolution for filtering the image signal. For every pixel, several multiplications and additions have to be made and a DSP is specifically designed to execute several of these tasks in parallel. Memory access is also optimized and is often executed within the same clock cycle. Therefore DSPs achieve high data throughput. For several signal processing tasks such as audio, image, or video compression highly specialized DSPs are available.

FPGAs, on the other hand, are integrated circuits consisting of programmable logic cells and programmable connections between those logic cells. Every logic cell can be used to represent a basic logic function such as AND, OR, NOT, NAND, and NOR, or to represent a basic memory cell. The combination of multiple logic cells can be used to generate complex and arbitrary functions. These functions can be truly complex—for instance mapping the circuitry of a complete processor onto the FPGA is possible. Such a processor is called soft core and it uses a lot of logic cells without using the inherent high flexibility of the cells. This is why special FPGAs are available, which contain hard-coded processors. These hard core processors operate faster and use less energy.

Of special interest for image processing is the FPGA's capability to process hundreds of multiplications and additions in parallel using the logic cells. Additionally, modern FPGAs offer hard-coded functionalities, for example memory and multipliers. Again, these are much faster than circuitry made of logic cells.

Even this short introduction proves that DSPs and FPGAs have different characteristics, even though both integrated circuits are predestined for signal processing. To come to a well-grounded decision, a rating matrix, shown in Table 4, was used.

For every criterion a weighting factor between 1 and 100 is given, indicating the importance of the criterion for the overall design. Next, DSP and FPGA are rated using values between 1 and 100. A higher value means that a particular solution is favored. Finally a weighted sum can be derived, describing the qualification of a particular approach for forming the basis for the image processing computer system.

Based on the high parallelism of the FPGAs operations, FPGAs are predestined for high data volumes and high data throughput which are needed by image processing. A DSP, on the other hand, offers benefits when it comes to navigation and flight guidance. Here, programs have to be executed in a more sequential structure. This requirement was added to the analysis bearing in mind the medium to long-term hardware development goals at the institute, which possibly include the integration of navigation, flight control, and guidance as well as some payload functions into one hardware platform.

The rating includes the criteria scalability, flexibility, and modularity. FPGAs can be scaled especially well, as the manufacturers frequently offer FPGAs with a different quantity of logic cells. Additionally, inside one family of FPGAs these chips feature identical packages with nearly identical pin out. Therefore, a FPGA with different performance can be easily integrated into an existing design.

**Table 4 Rating matrix DSP/FPGA**

Criterion	Weighting	DSP	FPGA
Performance image processing	100	30	70
Performance navigation/flight guidance	80	60	40
Flexibility	80	10	90
Scalability	70	30	70
Modularity	50	50	50
Power consumption	70	65	35
Weight/size	50	60	40
Cost	25	80	20
Development effort	100	65	35
		47	53

High flexibility is the most important advantage of FPGAs. Through programming, the functions of a FPGA can be changed dramatically. This is most noticeable with the integration of interfaces: Using FPGAs, arbitrary interfaces can be created for changing peripherals. Here, every DSP-based design is inferior, as its scope of operation is defined during the preliminary design phase.

No differences can be found regarding the modularity. If the distribution of functions of the overall system is required, multiple FPGAs as well as DSPs can be used. With the different requirements of signal processing and flight control/guidance discussed before, using both types of integrated circuits using their respective advantages might be reasonable as well.

As the software-defined circuitry of FPGAs must be programmed and then kept during run time, FPGAs suffer from higher power consumption than DSPs. The latter save energy being optimized for a limited set of functions. For a small unmanned aircraft with limited power supply, this aspect cannot be neglected. Therefore its weight is relatively high.

The weight and size of FPGAs and DSP are acceptable, even though using a DSP offers slight advantages.

Clear differences can be found regarding the cost of both devices. Higher complexity and lower quantities produced lead to high prices of FPGAs. Because of the high flexibility offered, FPGAs are often used for prototypes or small series. Often the functions verified with FPGAs are transferred to cheaper, application-specific integrated circuits (ASICs), if higher quantities are needed. In principle, DSPs are a special kind of ASIC. The lot size expected for the proposed image processing system is comparatively low. As in many cases of specialized small series the development costs outweigh the component costs. This is why cost is weighted relatively low.

Accordingly, the expected development effort has to be an important aspect of the hardware selection. The decisive factor here was the lack of experience at the institute using DSPs or FPGAs. Using a DSP was associated with decreasing the development effort, as there are similarities in programming DSPs and programming the processors used during the previous autopilot development.

Finally, the resulting weighted sum shows a slight advantage for the use of a FPGA based solution. Table 5 shows the specifications of the chosen FPGA, which forms the core of the new image processing system. In a first approach, the image processing system will be a stand alone module, which works in parallel to the existing autopilot of all CAROLO unmanned aircraft (Fig. 5).

#### D. Development of the Image Processing Computer

The core of the new processing system is a Xilinx Virtex II Pro chip. This hybrid FPGA not only has large amounts of free programmable logic gates, but block-RAM and hard core multipliers as well. More important, however, it features two hard core Power-PC 405 processors with 300 MHz clock rate. Like all hard core components they work much more efficiently than soft core components, that is components (i.e. components programmed into the logic cells). These processors compensate the immanent disadvantage of FPGAs by processing sequentially defined programs more efficiently. The processors are inside the FPGA's fabric and must be connected via circuit paths to external components such as RAM, flash memory, or the clock generating quartz oscillator. The manufacturer offers generic blocks for interfaces, data busses, and drivers. Those can be implemented using a graphical development environment, which is part of the Xilinx Embedded Development Kit.

**Table 5 FPGA specifications**

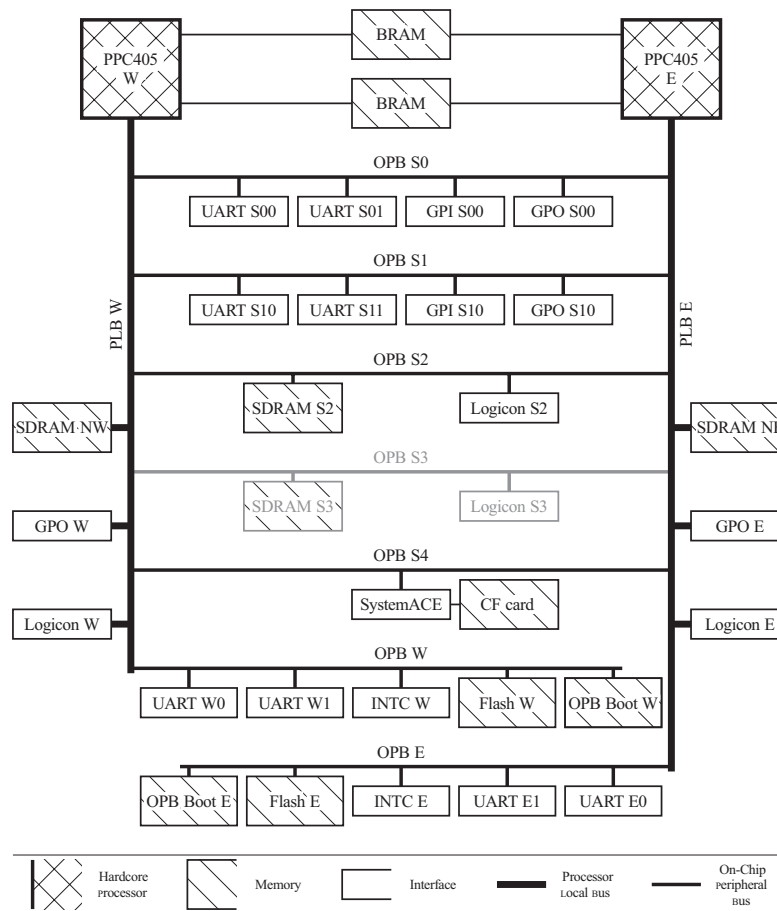
FPGA family	Xilinx Virtex II Pro
FPGA type	xc2vp50
Package	ff1152
Speed grade	-5
Logic cells	53.136
block-RAM	4.176 Kbit
max. user I/O	692
Digital clock master	8
Power-PC processors	2
Multipliers	232



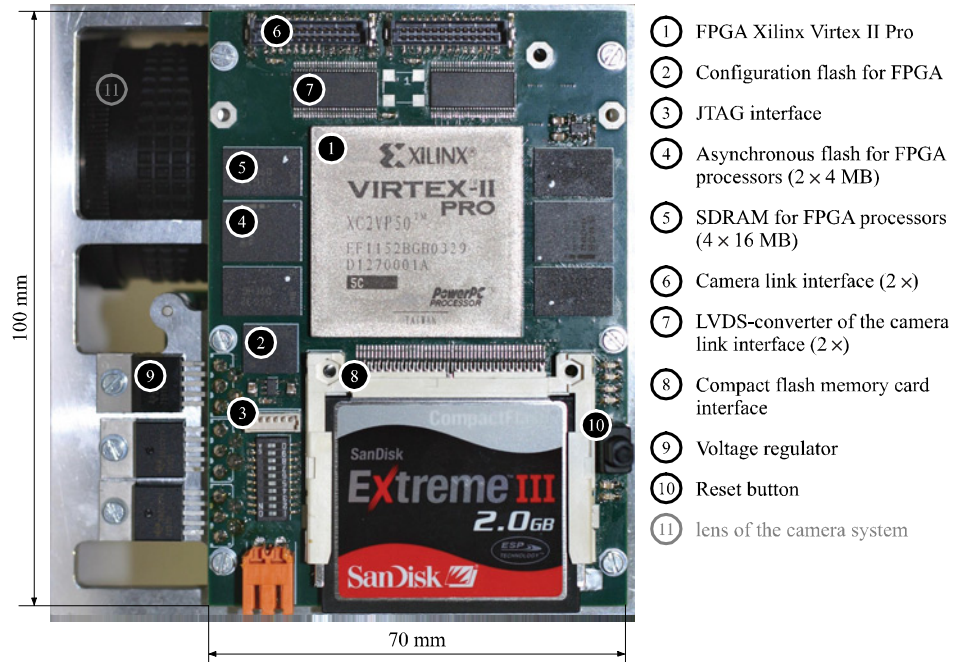
Additionally, user-defined functions can be implemented inside the fabric using either Very High Speed Integrated Circuit Hardware Description Language (VHDL)-code or graphically defined electronic circuits. Based on this so-called user logic, net lists and ultimately the specific circuitry are generated. Those are copied to the FPGA via a Joint Test Action Group (JTAG)-interface, from onboard nonvolatile flash memory, or using the System-ACE chip (System-ACE is a system-level configuration device that can configure all Xilinx FPGAs in a system and even across multiple boards. It includes a built-in microprocessor interface and supports most CompactFlash cards, including Microdrive storage technology.). The latter features special functions to load the FPGA's circuitry from a compact flash memory card. The very same memory card will be used to save the captured image data.

However, to use the application specific user logic additional interfaces are needed. These are external interfaces such as serial interfaces for communication with the autopilot or interfaces to connect a camera. Internal interfaces are needed as well to connect user-defined logic with the processors. Those interfaces are called Logicon, from logic connection, and include a configurable number of General Purpose Input/Output ports (GPIO)-pins, interrupt-pins, data registers, and First In-First Out (FIFOs). Figure 6 shows the functional structure of the image processing system's core, including all interfaces.

The processors are named West and East. Busses and memory chips are labeled accordingly. Every processor has its own local bus (64 bit) with exclusive access. Connected to these are the peripheral busses (32 bit). The on-chip peripheral busses can be used exclusively by each processor (OPB W, OPB E) or in common access by both processors (OPB S0-4). Every processor has its own working memory (SDRAM), flash memory, exclusive access to user-defined logic (Logicon), output-pins (GPO), serial interfaces (UART), interrupt controller (INTC), and memory



**Fig. 6 Schematic representation of Xilinx Virtex II Pro FPGA and implemented internal and external interfaces.**



**Fig. 7 Image processing system for UAVs based on a Xilinx Virtex II Pro.**

for the boot-loader (part of the block RAM). Additional serial and Logicon interfaces are located at the common on-chip peripheral busses. Read and write access to the compact flash memory is possible via the SystemACE interface from both processors. Direct data exchange between processors West and East uses common block-RAM. The Camera Link interfaces have no direct connection to the processors. They are connected to user logic and can be accessed via Logicons.

The hardware design not only takes into account the requirements of the embedded processors, but also the requirements of image processing in general and especially of the image processing algorithms for detecting and extracting the landing point marker characteristics [14, 15]. Section III focuses on software implementation, to show how this new system is capable of determining the landing point.

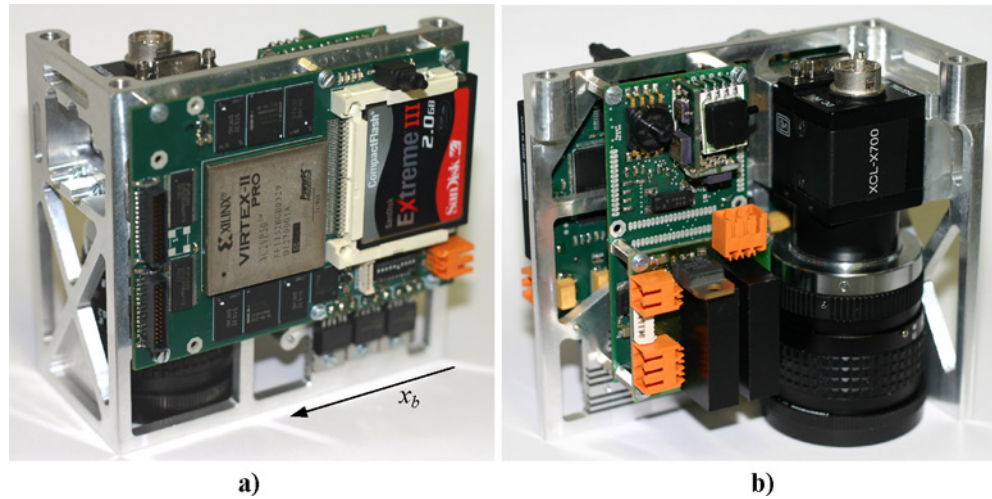
Based on the layout presented in Fig. 6 an electronic circuit was created and a printed circuit board with eight layers was designed. The resulting powerful image processing computer with a footprint of 100 mm  $\times$  70 mm and a mass of 64 g only is shown in Fig. 7. The individual components are labeled. The SystemACE-chip and four LVDS-converters which are part of the Camera Link interface are not visible. They are on the rear side of the Printed Circuit Board (PCB).

### E. Integration of the Overall System

To integrate the image processing system into the test aircraft CAROLO T200, it has to be fastened to the overhead payload bar (Fig. 3). The camera's line of sight must be aligned with the aircraft's body fixed  $z_b$ -axis. Therefore, an aluminum frame was designed to:

- carry all components (image processing computer, camera system, autopilot, power supply);
- precisely align the Inertial Measurement Unit (IMU) of the autopilot and the camera's line of sight (assembly in vertical format or landscape format possible);
- protect all components (especially to absorb dynamic forces of the heavy lens in  $x_b$ -direction during landing);
- cool the voltage regulators using cooling fins of the frame.

Figure 8 shows the assembled overall system with the aircraft's body fixed  $x_b$ -axis to demonstrate the direction of flight. The image processing system measures 105 mm in length, 65 mm in width, and 100 mm in height and fulfills the set requirements regarding its dimensions.



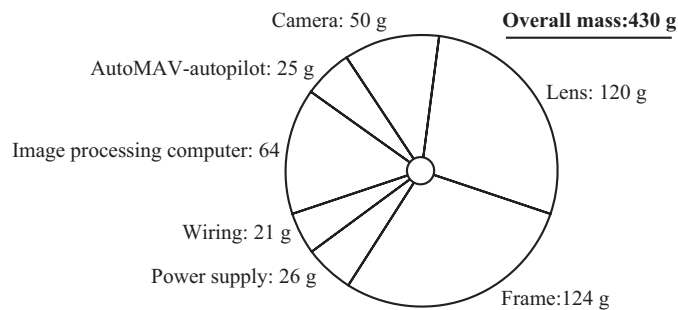
**Fig. 8** Image processing system for vision-based autonomous landings of UAVs: a) image processing computer; b) camera system, AutoMAV-autopilot, power supply.

**Table 6** Specifications of the overall system

	Requirements	Specifications
Length	<150 mm	105 mm
Width	<95 mm	65 mm
Height	<100 mm	100 mm
Volume	<(90 mm) <sup>3</sup>	≈(88 mm) <sup>3</sup>
Mass	<500 g	430 g
Power consumption	<10 W	≈ 6 W

With a mass of 430 g the overall system clearly fulfills the mass requirements restated in Table 6 as well. Nevertheless, analyzing the mass distribution shown in Fig. 9 reveals some interesting facts: the lens and the aluminum frame account for more than 50% of the overall system's mass. Bearing in mind that the aluminum frame could be designed much lighter with a different lens, using a lighter lens has a lot of mass saving potential. Assuming that 70 g could be saved with a different lens and approximately 60 g with a lighter frame, a future system might weigh around 300 g only. Special care has to be taken, however, regarding the quality of the lens: it must be acceptable for the machine vision task at hand.

Maximum power consumption was specified at 10 W. During tests with both processors, the FPGA logic gates and the camera, power consumption was measured as 6 W at 12 V.



**Fig. 9** Mass distribution of the overall system.

### III. Implementation of Image Processing Algorithms

After gathering hardware requirements for the new image processing system and developing hardware, which fulfills these requirements, the next task is implementing the algorithms for the vision-based extraction of the landing point marker and computing its relative position and attitude. It is important to use the advantages the specially developed hardware offers to full extent. Therefore, the implementation of the algorithms will be examined from different points of view. First, the different requirements of the algorithms will be evaluated and compared to the capacities of the image processing hardware. In a next step a more hardware-dominated view is adopted. This approach results in a detailed plan for the software implementation.

#### A. Software Implementation Concept

To determine relative position and attitude, highly disparate sub-problems must be solved. As part of the image preprocessing, the gray scale image must be converted into an edge image. For every pixel two convolutions must be computed. Using these results, the contour improvement can be conducted and every pixel can be categorized as a strong, a weak, or not an edge pixel. Despite not being a mathematically challenging task, the computational cost is high, as every step has to be done separately for each of the  $\sim 800,000$  pixels. Because all computation steps are identical for each pixel, the freely programmable fabric of the FPGA is well suited for implementation of this part of the image processing algorithms. Here, the analogy of a conveyor belt fits. While for one pixel the two gray scale gradients are computed, the norm of the gradient can be computed for a pixel received before, and for another pixel the bi-linear interpolation for contour improvement can be carried out. All these operations take place simultaneously in different parts of the logic fabric. This is the immanent strength of FPGAs.

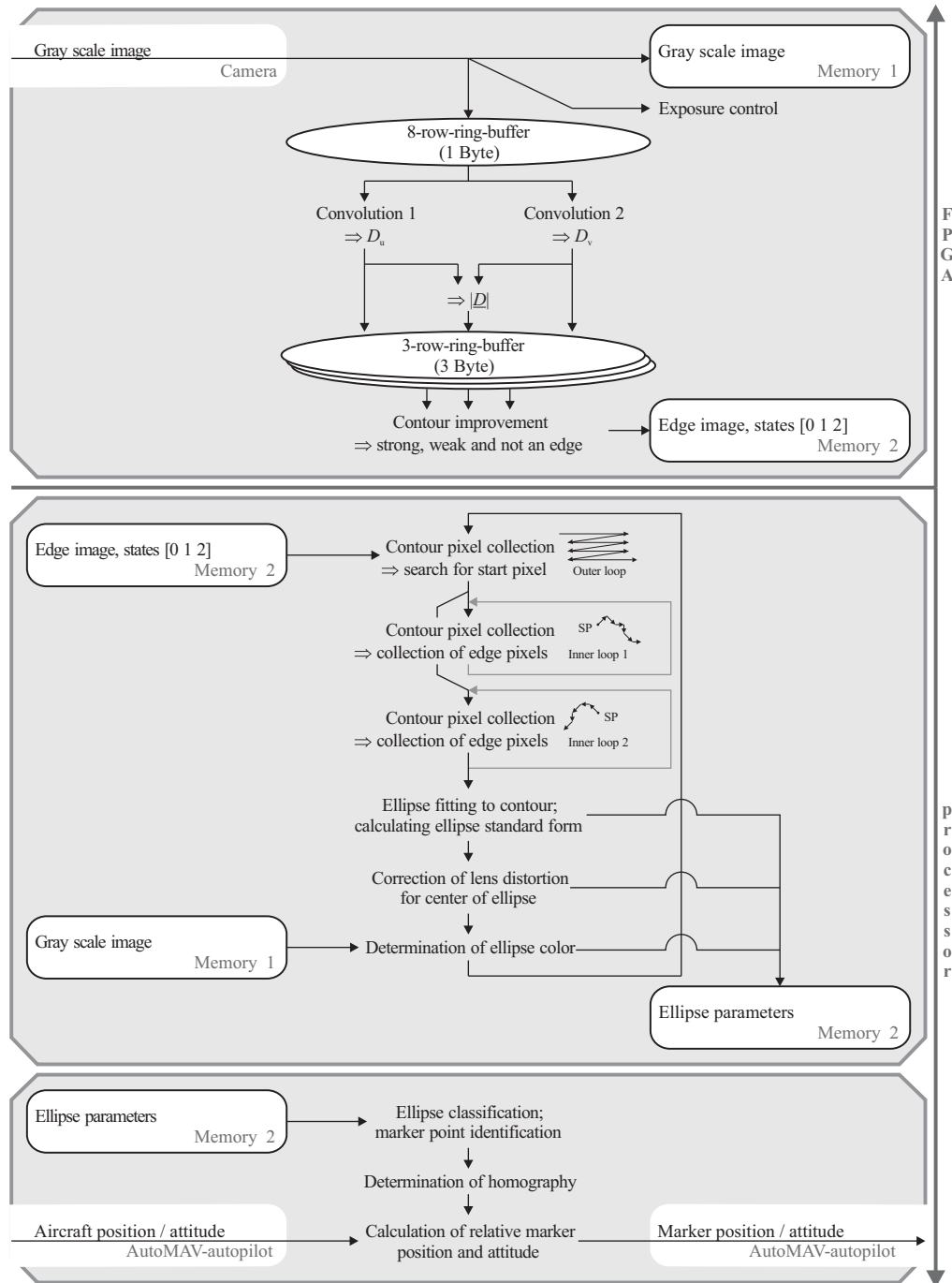
While this first phase transforms every gray scale pixel into the corresponding edge pixel, the next step is the first of two important abstractions. The relevant edge pixels are transferred to lists of contour pixels. Two different search algorithms are combined, one of them nondirectional. To support the nonpredictable memory access of the nondirectional search, the complete edge image has to be computed beforehand. It is not sufficient to consider the local neighborhood of the relevant edge pixel only. Even though this algorithm could be implemented inside the FPGA's fabric, using the processors is recommended. Another benefit comes with using the efficiently implemented memory management of the processors.

All these tasks have to be solved sequentially: abstraction of the contours by ellipses, identifying the marker, computing the homography, and computing the relative position and attitude of the landing point marker. They can be realized much better by a program rather than a fixed electronic circuit. Therefore, these tasks should be implemented on the two PPC-405 processors. Figure 10 shows an overview of the assignment of the different tasks as well as a first memory allocation strategy.

Based on the strategy presented before, a more detailed planning can be realized taking into account the existing hardware resources. Figure 11 shows the complete data flow inside the FPGA and all external connections to camera, compact flash memory card, and AutoMAV-autopilot. Memory chips are quasi-external components as well, which are connected to the FPGA via processor local bus and on-chip peripheral bus.

The gray scale image exits the camera as a parallel data stream with 10 bit color depth. This image is transferred to the user-logic function "Exposure Control" as well as to the FIFO of Logicon S2. From the FIFO, connected to OPB S2, the gray scale image is copied to SDRAM S2 using direct memory access. Via the OPB S2, the processor PPC East can access the image data to copy it to the compact flash memory card. The image data is saved in uncompressed BMP-file format. A modified file header includes user specific data, e.g. the navigation solution to each exposure. PPC East receives the navigation solution from the autopilot via PPC West and the common block RAM. Currently, PPC East only saves the image data to compact flash memory and is therefore available for other tasks in the future.

One of the most important functions is the user-logic function "image pre-processing." According to the definitions of Fig. 10 this function computes the edge image. The edge image data stream is written to SDRAM NW via the Logicon W. So memory access of PPC West to the edge image is exclusive. Therefore, this processor can be used to further process the edge image without interfering with saving the image data to compact flash by PPC East.



**Fig. 10 Distribution of algorithms between logic gates and processors for computing marker position and attitude.**

PPC West, in addition to conducting the high level image processing, is used for sequence control. Serial interfaces connect PPC West with the AutoMAV-autopilot, with user-defined functions and finally with the camera. Using the OPB S2, PPC West not only has access to the edge image but the original gray scale image saved in SDRAM S2 as well. This access is used to determine the color of the computed ellipses—an important task of marker identification.

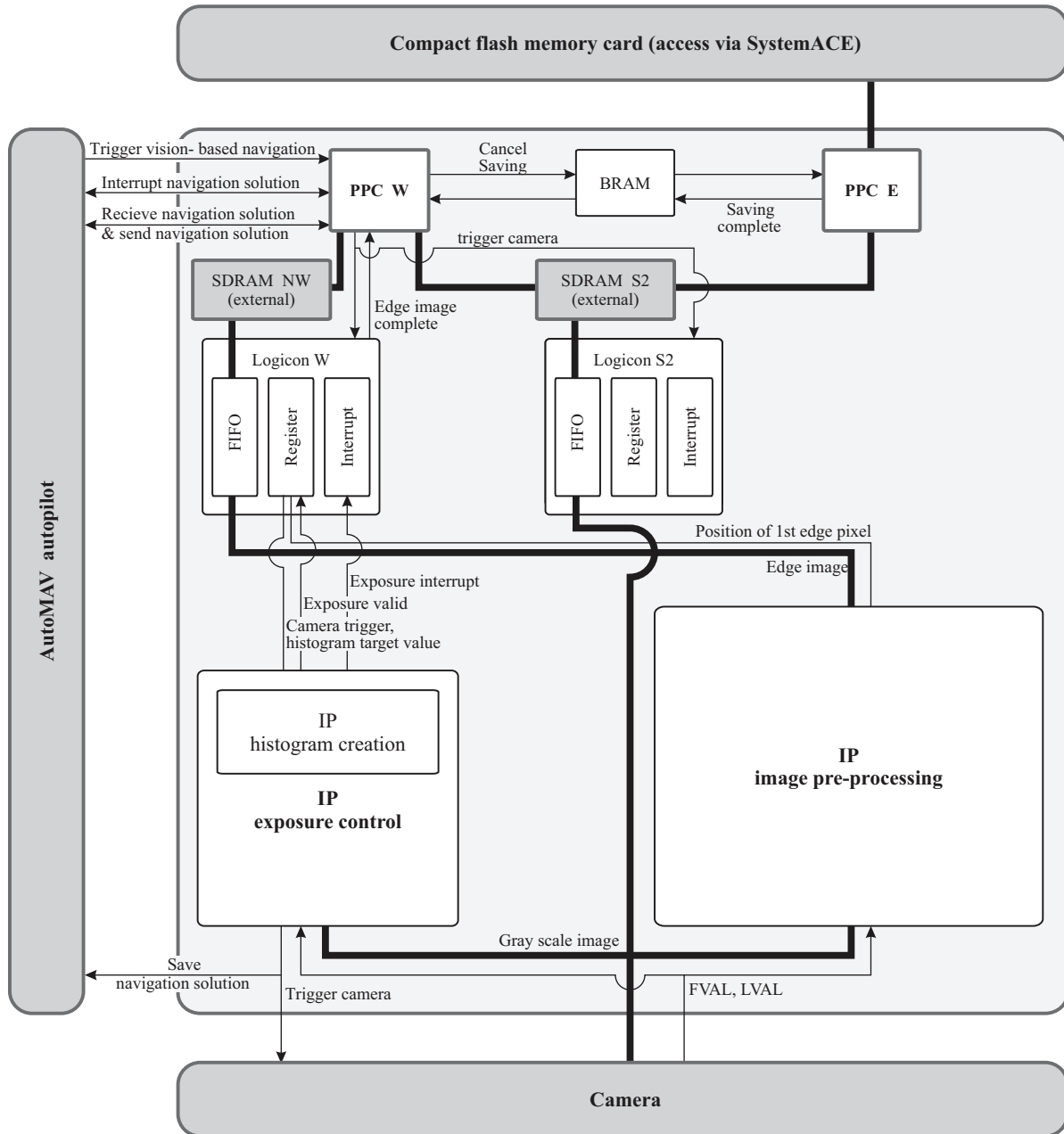


Fig. 11 Functions and data flow inside the FPGA.

## B. First Results of Software Implementation

To test functioning of the overall image processing chain the following first test was carried out: the camera was triggered by a program running on PPC West. The trigger signal was routed through the Logicon and the FPGA's fabric. The original image data output by the camera were used to compute the edge image and at the same time written to the FIFOs of two Logicons. The arrival of the first pixel in the memory of the FIFO triggered a direct memory access (DMA), which copied the edge image data to a SDRAM until the edge image was completed. Finally, this image was processed by PPC West. The marker was identified and its position and attitude computed. At the

same time, the original image was copied to compact flash memory card. Using this memory card the error-free saved image could be read at a PC. This image was used to compute a reference solution which proved the correct implementation of the image processing algorithms.

This successful test was significant in proving the complete operability of the newly developed hardware. These first tests showed promising indications that image processing is fast enough for the proposed landing procedure. These results are preliminary and have to be examined more thoroughly.

With this last result the following can be stated: As the developed hardware fulfills all set requirements, a powerful and highly flexible computing system for image processing onboard small UAVs is now available.

#### IV. Conclusion

A detailed analysis of the capabilities of small UAVs showed weaknesses in the area that is most important for these aircraft: surveillance. Usually, the data transmission bandwidth of these small aircraft is not big enough to transmit image or video data over great distances. Also, onboard computing power of small unmanned aircraft is not sufficient to process image data on board. The work presented here tries to solve this problem by developing a highly flexible onboard image processing system. Thus the mission spectrum of small UAVs and their usefulness shall increase considerably. To achieve this, the image processing system shall be used first as a sensor for flight guidance during a vision-based landing.

Based on the experimental results of the verification of image processing algorithms to extract the features of a known landing point marker and compute its relative position and attitude, the requirements for the new image processing hardware were defined. These take into account the special requirements regarding mass, size and power consumption imposed by small aircraft like CAROLO T200, an aircraft with 2m wingspan and 5 kg MTOW.

The developed image processing computer is based on a hybrid FPGA with two power-PCs and weighs only 64 g with a footprint of 100 mm  $\times$  70 mm. It features two standardized Camera Link interfaces to connect high quality digital cameras. Together with the components camera, lens, power supply, autopilot and a sturdy aluminum frame to support these components the image processing system is built. With 105 mm  $\times$  65 mm  $\times$  100 mm in length, width, and height, respectively, and a mass of only 430 g, the size and capabilities of the new image processing system are currently unique.

After setting up the image processing system and implementing the operating system, the image processing algorithms for detecting the landing point marker and computing its position and attitude has been implemented and tested successfully. Functionality of all components of the new image processing system has been verified.

Being a highly flexible and powerful signal processing computer, the developed system can be reconfigured relatively easily, for example new interfaces could be implemented to support other sensors. Because heavy parallel computing is supported, not only image data can be processed efficiently but rather any type of digital data. A new area of research at the Institute of Aerospace Systems is flight control using neural networks. Using the vast amount of logic cells, these nets can be implemented and run in parallel, resulting in computing capabilities previously unavailable in this class of small aircraft.

#### Acknowledgment

The author would like to thank Marco Buschmann of the Institute of Aerospace Systems for designing the printed circuit board of the image processing system.

#### References

- [1] Schulz, H.-W., Buschmann, M., Kordes, T., Krüger, L., Winkler, S., and Vörsmann, P., "The Autonomous Micro and Mini UAVs of the CAROLO-Family," *Proceedings of the AIAA Infotech@Aerospace*, AIAA, New York, NY, 2005, AIAA Paper, AIAA-2005-7092.
- [2] Quigley, M., Goodrich, M. A., Griffiths, S., Eldredge, A., and Beard, R., "Target Acquisition, Localization, and Surveillance using a Fixed-Wing Mini-UAV and Gimballed Camera," *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 2600–2605.
- [3] Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M., "Vision-Guided Flight Stability and Control for Micro Air Vehicles," *Proceedings IEEE International Conference on Intelligent Robots and Systems*, IEEE, Vol. 3, 2002, pp. 2134–2140.

- [4] Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M., "Towards Flight Autonomy: Vision-Based Horizon Detection for Micro Air Vehicles," *2002 Florida Conference on Recent Advances in Robotics*, Miami, FL, May 2002.
- [5] Cornall, T., and Egan, G., "Measuring Horizon Angle from Video on a Small Unmanned Air Vehicle," *2nd International Conference on Autonomous Robots and Agents*, Palmerston North, New Zealand, Dec. 13–15, 2004.
- [6] Cornall, T., and Egan, G., "Calculating Attitude from Horizon Vision," *Proceedings of the Eleventh Australian International Aerospace Congress*, Melbourne, Australia, March 2005.
- [7] Petruszka, A., and Stentz, A., "Stereo Vision Automatic Landing of VTOL UAVS," *Proceedings of Association of Unmanned Vehicle Systems International*, 1996, pp. 245–263.
- [8] Yang, Z. F., and Tsai, W. H., "Using Parallel Line Information for Vision-based Landmark Location Estimation and an Application to Automatic Helicopter Landing," *Robotics and Computer-Integrated Manufacturing*, Vol. 14, No. 4, 1998, pp. 297–306.  
doi: [10.1016/S0736-5845\(98\)00007-6](https://doi.org/10.1016/S0736-5845(98)00007-6)
- [9] Shakernia, O., Ma, Y., Koo, T. J., Hespanha, J., and Sastry, S., "Vision Guided Landing of an Unmanned Air Vehicle," *IEEE Conference on Decision and Control*, IEEE, Phoenix, AZ, December 1999, pp. 4143–4148.
- [10] Sharp, C. S., Shakernia, O., and Sastry, S. S., "A Vision System for Landing an Unmanned Aerial Vehicle," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Republic of Korea, 2001, pp. 1720–1727.
- [11] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Vision-based Autonomous Landing of an Unmanned Aerial Vehicle," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, 2002, pp. 2799–2804.
- [12] Thielecke, F., Dittrich, J., and Bernatz, A., "ARTIS: Ein VTOL UAV Demonstrator," Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. (ed.): *Jahrbuch Deutscher Luft- und Raumfahrtkongress 2004*, Vols 1 and 2, Deutsche Gesellschaft für Luft- und Raumfahrt, Bonn, Germany, 2004.
- [13] Schulz, H.-W., and Vörsmann, P., "Towards Vision-Based Autonomous Landing for Small UAVs: Hardware and Control Algorithms," *Proceedings of the AIAA Infotech@Aerospace*, AIAA, New York, NY, 2007, AIAA Paper AIAA-2007-2851.
- [14] Schulz, H.-W., Buschmann, M., Krüger, L., Winkler, S., and Vörsmann, P., "Towards Vision-Based Autonomous Landing for Small UAVs – First Experimental Results of the Vision System," *Journal of Aerospace Computing, Information, and Communication*, Vol. 4, No. 5, 2007, pp. 785–797.  
doi: [10.2514/1.26789](https://doi.org/10.2514/1.26789)
- [15] Schulz, H.-W., Buschmann, M., Krüger, L., Winkler, S., and Vörsmann, P., "Vision-Based Autonomous Landing for Small UAVs - First Experimental Results," *Proceedings of the AIAA Infotech@Aerospace*, AIAA, New York, NY, 2005, AIAA Paper AIAA-2005-6980.

Tim Howard  
Guest Editor