

# System Design for an Autonomous Smart Vision System

ANDREAS MAEDER, JIANWEI ZHANG  
University of Hamburg  
Institute TAMS, Department of Computer Science  
Vogt-Koelln-Str. 30, 22527 Hamburg  
GERMANY

*Abstract:* A major problem in the area of service robotics is the limitation of the processing capability within the system. Especially vision-based sensors have high demands on realtime control. This paper presents an autonomous subsystem for FireWire cameras which can be adapted to various tasks, ranging from simple control up to image (pre)processing. The system contains an embedded processor as well as dedicated hardware for signal preprocessing realized on a FPGA prototyping board.

*Key-Words:* System Design, VHDL, Embedded Control, Computer Vision, Service Robotics

## 1 Introduction

Though most of the work in robotics is still in the state of research and only few products have been introduced, all studies predict an exponential growth of the personal and service robotics market in the near future [1]. In contrast to manufacturing robotic systems these robots depend on dynamic sensor information to react to the environment. New modalities and sensor fusion are the keys to dynamic interaction.

While ultrasonic and laser range sensors provide robust data only in predetermined environments, vision technology (in conjunction with other sensors) will be a key factor for service robotics. Image processing is flexible enough to match the needs of different tasks in service robotics.

## 2 Problem

On autonomous robotic platforms, like the one we use for research [2, 3], see Figure 1, several camera systems are installed to fulfill different needs of the subsystems:

- An omnidirectional vision system used for localisation and global vision tasks.
- A stereo camera system mounted on a pan-tilt unit performing detailed three-dimensional vision for interaction with users, handling and manipulation of objects with arms and hands.
- Two cameras mounted on the hands to monitor and control grasps.

The major problem of such complex systems is the limited amount of processing capability provided by

the underlying control hardware. To fulfill the demands on realtime control of such a system, fast control loops have to be realized. Furthermore the vision systems of the service robot are only a sensor subsystems, while the main focus of control is on the robotic tasks itself.

Standard hardware, like the PC that currently controls the robot, does not provide enough processing speed to satisfy all algorithmic needs. In most cases even "simple" image processing, performed on the data stream of only one camera, could not be done in real-time.

### 2.1 Workaround

The straightforward solution of integrating more computers on the robotic platform is not feasible due to the limitations of the space available and, more importantly, the power consumption of the processors.

As a workaround to perform robotics with these limited capabilities, the different tasks within the robot are sequentialized in a way that only a few subsystems are active, while most of the hardware is idle. The following list shows the scenarios for the service robot using the different vision systems:

- Exploration, localisation and movement of the robot is based on odometric and laser range data. Only the omnidirectional vision system is used for localisation tasks outside the realtime control.
- The stereo camera system is active when the immobile robot is operating its arms or interacting with the user, e.g. recognition tasks, gesture and gaze detection, etc.

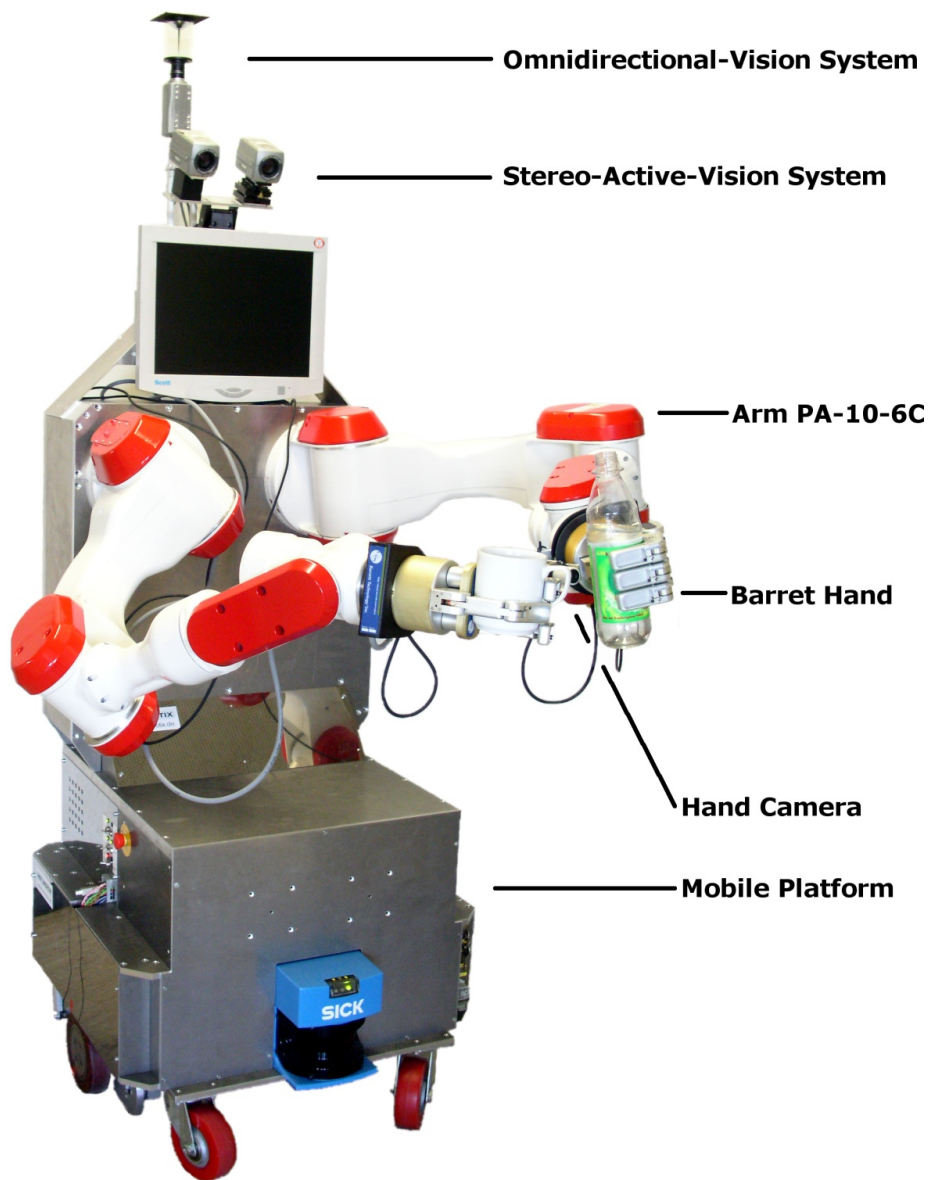


Figure 1: Service robot at our institute

- The hand cameras are needed to grasp objects, guiding the approach of the arm and visually controlling the hand. During this phase, the stereo camera system is offline.

Nevertheless, to keep up with the processing capabilities, control and movement must be slowed down. The long-term goal of completely replacing the functionality of odometric and laser range sensors using the vision systems can only be achieved with continuous image processing.

### 3 Solution

Smart sensors are the key technology for solving this problem. For the vision systems of the robot this re-

quires the cameras to be able to do autonomous image (pre)processing, performing (some of) the different tasks within the processing hierarchy:

1. Image enhancement: normalisation of contrast and color, calibration
2. Preprocessing: filter and morphological operators
3. Feature extraction: segmentation, higher-level operators
4. Image analysis

During the last year, several companies, specialized in computer vision products, introduced so-called smart-cameras [4, 5, 6]. They contain an embedded processor which can do some (limited) preprocessing within the first two levels. The approach presented

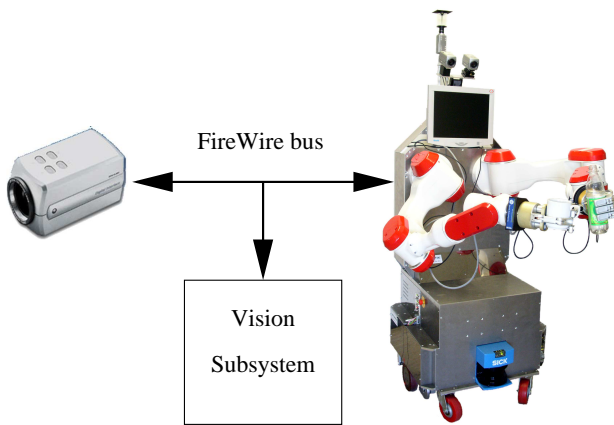


Figure 2: Image processing subsystem

here is much more general, since the image processing should be able to abstract from images and generate feature data with special focus on the application area of service robotics.

### 3.1 Requirements

The system should combine the flexibility of an embedded processor (software-programmed) with the performance of specialized hardware units to meet realtime conditions. The main objective is to offer an experimental prototyping environment, which allows both hardware and software to be used in a flexible manner. The idea is to transfer image processing tasks from the service robot's PC to the camera-based hardware system, beginning with very simple image enhancement up to extended feature extraction, like symmetry [8].

The prototyping character will help to define interfaces to the software components of the service robot and to work out overall requirements. Later on, a less flexible, but therefore more performant vision subsystem should be designed.

## 4 System design

Since the omnidirectional vision camera and the cameras of the stereo vision system are connected to the host computer via the IEEE 1394 bus (aka. FireWire), the image processing subsystem will be integrated as shown in Figure 2.

### 4.1 Prototyping board

To fulfill the requirements, we used a FPGA prototyping board from Altera [9], shown in Figure 3. This

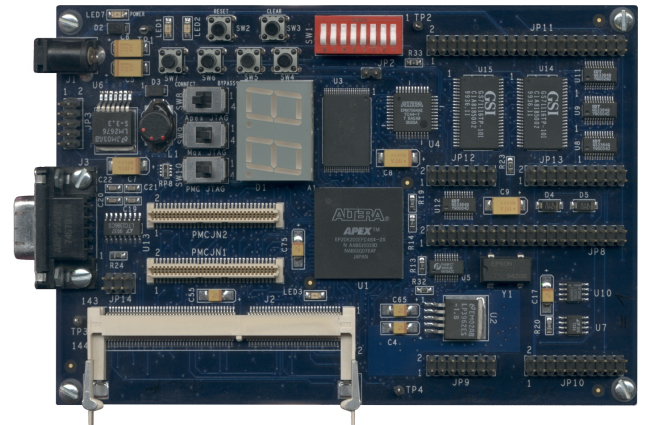


Figure 3: FPGA prototyping board

board could be used in an ideal manner for hardware-software codesign, since both parts could be implemented easily:

**Hardware** is designed using a hardware description language; VHDL [13, 14] in this project. Synthesis and mapping steps [10] generate the data to be downloaded onto the prototyping board.

The board already contains SRAM and flash memory needed for the embedded processor. Several physical interfaces (serial ports, SDRAM, PMC, 3.3V and 5.0V buses) are available to connect to other hardware components.

On-board clock circuitry in conjunction with PLLs inside the FPGA allow clock rates of up to 100 MHz. The FPGA (Apex 20K200EFC484) offers up to 500 000 gates for the design.

A library of parametrizable components enables the system designer to integrate components like memories, FIFOs, arithmetic units or interface circuits into the HDL design. The implementation of these parts is optimized according to the FPGA family.

**Software** is developed for an IP (Intellectual Property) soft core: NIOS [11]. Using Altera SOPC-Builder (System On a Programmable Chip) a custom processor could be synthesized. This system offers great flexibility, parametrizing the NIOS processor core (16/32-bit, hardware multiplication, external alu), the memory layout and mapping (bit width, size, boot address, ...) and the interfaces to user-designed hardware components (parallel IO and interrupts, access to the processor's bus system).

For software development the GnuPro toolkit provides a C cross compiler, which is adapted to the

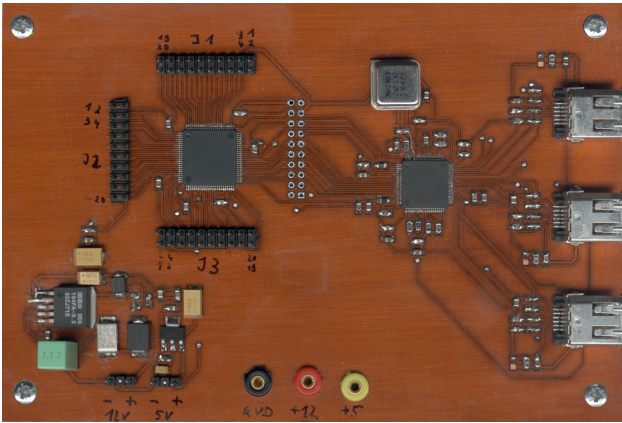


Figure 4: FireWire connection board

options of the processor. Through a serial interface console IO and debugging could be handled.

## 4.2 FireWire board

The cameras transfer their images at 400 Mbps isochronous data rate, either as  $640 \times 480 @ 30$  fps for the stereo cameras or as  $1280 \times 960 @ 7.5$  fps for the omnidirectional camera, both in YUV 4:2:2 format [12]. The FPGA prototype board is not suitable to interact with the FireWire bus at these data rates (with differential signals). Therefore an extra board with standard commercial circuits has been designed, see Figure 4. The two chips (Texas Instruments TSB12LV01 / TSB41AB3 [15, 16]) implement the physical and the link layer of the IEEE 1394 protocol. A memory-mapped interface for configuration and data transfer connects the link layer chip to processors or, like in our case, to custom hardware.

## 4.3 Design flow

The design flow follows the idea of embedded system design: time-critical parts are implemented as hardware units to ensure the necessary data throughput, while control-driven algorithmic parts are implemented in software computing on the processor. Examples for data-driven, hardware-implemented parts of the design are:

- The interface to the link layer circuit of the FireWire board, where control and response have to be processed within one clock cycle of the board, running at 30 MHz. Therefore a finite-state machine connects both boards sharing the same clock with different (rising/falling) clock edges.
- Data processing parts of the design running at full speed, like digital filters.

- The data transfer on the FireWire bus is done within isochronous cycles delivering the image data in burst packages. Synchronisation between the hardware unit is done via “enable” or “data ready” signals. Several internal FIFOs buffer data in between these units, to keep up with the data while subsequent components are still busy.

According to the abstraction levels within image processing, system development was done in a bottom-up manner:

1. The first step was synchronisation with the isochronous FireWire data stream.
  2. Then asynchronous communication for simple control tasks and transfer of preprocessed data was added.
  3. Generation of isochronous data streams, needed to transfer preprocessed data to the host PC, completes the integration into the FireWire protocol.
- Based on the abilities above, specific feature extraction and image processing will be added.

Exploring the design space, the FPGA system can be used as an architectural workbench evaluating design alternatives and specifying requirements of the overall system.

## 5 Results

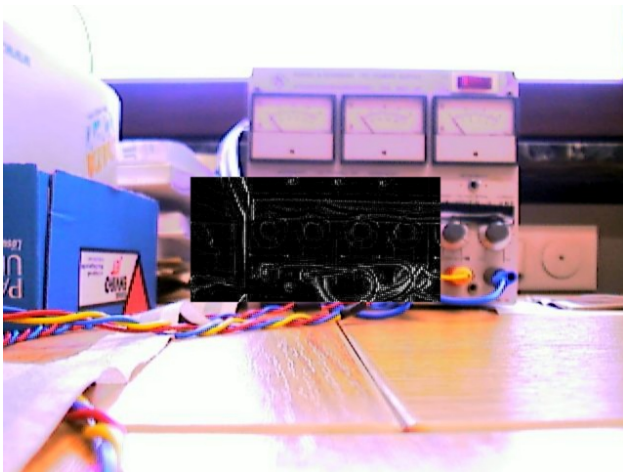
As stated in the previous section “design flow” several tasks of the vision system for the service robot could be implemented.

### 5.1 Automatic focus

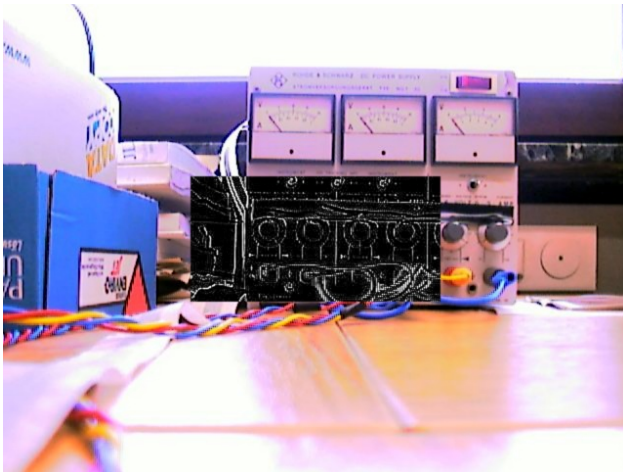
The cameras of the active stereo vision system (Sony DFW-VL500 [7]) have a good optical system with  $12 \times$  zoom and focus adjustment, but the focus has to be adjusted manually. A sample software implementation of an automatic focus already causes a processor load of about 30% on the robot. Since this “basic load” only for image enhancement is too high, no adjustment is done and a fixed focus adjusted to the arms length is used now. So for practical reasons all efforts concerning camera calibration are voided, since the optics already lack fine tuning.

An autonomous automatic focus is the first application for the vision subsystem board, enhancing the quality of the images. This task belongs to the simple class of control applications, where the subsystem listens to the isochronous data on the FireWire bus, computes some control values for the camera and sends the values as asynchronous FireWire commands.

In a first step a software prototype has been programmed to validate the algorithm and to deliver



Sum: 255 960 — Defocused: -10 cm



Sum: 465 479 — Focus optimized

Figure 5: Focus region with Laplace filtered image

benchmark results, a hardware implementation followed. The algorithm itself computes a Laplace-filtered image of the central region and uses the sum of this edge image as a measure for the focus quality. The sample images from Figure 5 illustrates this, showing the region used for focus control, together with its computed Laplace image and the resulting sum value.

Maximizing this value yields the best focus, therefore a two-stage algorithm controls the camera: *Global optimization* has been implemented as an interval search. Afterwards, in a second control stage, *local optimization* adapts the focus to changing scenes. Dynamic threshold techniques control the transition between these operation modes.

In the final implementation, the arithmetic pipeline containing a  $3 \times 3$  neighbourhood generation (implemented with on-chip delay lines), the Laplace filter itself (adder tree implementation) and the fi-

nal sum computation are implemented in hardware. The focus algorithm itself can be transferred without changes and runs on the embedded processor core.

## 5.2 Programmable digital filter

Extending the architecture in a way that it can also send an isochronous FireWire data stream enabled the implementation of a programmable digital filter. The prototype version is based on the previous design, using only  $3 \times 3$  windows. Due to the hardware restrictions only simple fixed point arithmetic (division as shift) has been implemented to date.

Further projects will develop appropriate pipeline units.

## 5.3 Omnidirectional to panoramic conversion

The conversion of the high-resolution omnidirectional data into panoramic images, needed in different localisation algorithms of the service robot, is the actual (ongoing) work. The vision subsystem should convert the video stream in realtime. Since the images are quite large (2.4 MB) the extra memory, implementing input and output image buffer, was added to the system using the SRAM connector of the prototype board.

The computation itself is implemented completely in hardware, containing specialized data paths for memory addressing and pixel interpolation, controlled by several control FSMs (Finite State Machines).

## 6 Conclusion

This paper presented a flexible and extendible vision subsystem to be connected to the FireWire cameras used on our service robot. Using both programmable hardware as an FPGA in conjunction with an embedded processor makes this system an ideal choice for an experimental workbench investigating system architectures.

A prototype has been successfully implemented, providing a simple but efficient automatic focus for the existing vision system. Current activities deal with the conversion of omnidirectional into panoramic video data and architectures for digital filters.

### References:

- [1] *Robotics Market*, RoboNexus Conference and Expo, San Jose, CA, Oct. 6-9 2005 — URL [www.robonex.com/roboticsmarket.htm](http://www.robonex.com/roboticsmarket.htm)

- [2] D. Westhoff et.al., *A flexible framework for task-oriented programming of service robots*, in Robotik 2004, VDI-Berichte (ISBN 3-18-091841-1), Munich, Germany, June 2004.
- [3] J. Zhang et.al., *A flexible software architecture for multi-modal service robots*, submitted for IROS 2006, Beijing, China, 9-15 Oct.
- [4] Basler Vision Technologies, *Intelligent Camera eXcite*, Ahrensburg, Germany – URL [www.baslerweb.com](http://www.baslerweb.com)
- [5] Matrox Imaging, *Iris P-Series / E-Series*, Dorval, Canada – URL [www.matrox.com/imaging/](http://www.matrox.com/imaging/)
- [6] Sony Electronics Inc., *XCI-SX1*, San Diego, CA – URL [news.sel.sony.com/corporateinfo/](http://news.sel.sony.com/corporateinfo/)
- [7] Sony Electronics Inc., *Sony DFW-V500, DFW-VL500 Technical Manual*, San Diego, CA – URL [news.sel.sony.com/corporateinfo/](http://news.sel.sony.com/corporateinfo/)
- [8] D. Westhoff, K. Huebner, J. Zhang, *Robust Illumination-Invariant Features by Quantitative Bilateral Symmetry Detection*, in Proc. IEEE International Conference on Information Acquisition (ICIA 2005), Hong Kong, 2005.
- [9] Altera Corporation, *Nios Embedded Processor Development Board – Online Documentation*, San Jose, CA – URL [www.altera.com/literature/](http://www.altera.com/literature/)
- [10] Altera Corporation, *Quartus II Development Software Handbook v5.1*, San Jose, CA – URL [www.altera.com/literature/](http://www.altera.com/literature/)
- [11] Altera Corporation, *Nios Processor – Online Documentation*, San Jose, CA – URL [www.altera.com/literature/](http://www.altera.com/literature/)
- [12] *1394-based Digital Camera Specification*, 1394 Trade Association, Southlake, TX – URL [www.1394ta.org/Technology/Specifications/](http://www.1394ta.org/Technology/Specifications/)
- [13] *Standard 1076-1993, IEEE Standard VHDL Language Reference Manual*, Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1993.
- [14] *Standard 1076-2002, IEEE Standard VHDL Language Reference Manual*, Institute of Electrical and Electronics Engineers, Inc., New York, NY, 2002.
- [15] Texas Instruments Inc., *TSB12LV01B IEEE 1394-1995 High-Speed Serial-Bus Link-Layer Controller, Data Manual*, Dallas, TX.
- [16] Texas Instruments Inc., *TSB41AB3 IEEE 1394a-2000 Three-Port Cable Transceiver/Arbiter*, Dallas, TX.