A Real-time Image Recognition System for Tiny Autonomous Mobile Robots

Stefan Mahlknecht
Vienna Univ. of Technology
Institute of Computer Technology
mahlknecht@ict.tuwien.ac.at

Roland Oberhammer
Vienna Univ. of Technology
Institute of Computer Technology
e9526443@stud3.tuwien.ac.at

Gregor Novak
Vienna Univ. of Technology
gregor@bluetechnix.at

Abstract

Intelligent sensors for mobile robots play an important role in many technical applications. In this paper a real-time image recognition system for a tiny soccer playing robot is presented, capable of detecting objects in real-time at a frame rate of up to 60frames/s. The image recognition module has very low power consumption of less than 250mW and fits into a package of only 35x35mm including a CMOS camera and a low power, high performance signal processor. We propose an object recognition algorithm that is optimized for deeply embedded systems used in energy and performance constrained devices. The algorithm is based on a combination of edge and color detection and uses a fixed model for each object to be recognized. Results of the ball recognition application show that its relative polar coordinates are found within 11ms.

1. Introduction

Image recognition systems are widely used in different industries such as production plants to detect faulty components on a conveyor or as surveillance systems that are capable of detecting intrusion, differentiating people or observing their motion. What all these systems have in common is the use of high performance cameras and powerful computers with few constraints in power consumption, real-time behavior, size or cost. In autonomous mobile systems such as the soccer playing robot application presented in this paper, the framework is very different.

Fully autonomous robots are an active area of research in academia and industry. Some examples of very advanced robots such as the humanoid robot Asimo [1] from Honda or wheel robots such as the latest mars robot Spirit [2] employed in the mars mission of 2004 have all camera systems integrated to build up a model of the environment and to navigate through it. The problematic of building a view of the world is fundamental to all autonomous systems. Indeed several other sensors are available, infrared, ultra sonic, etc., but none of them is as powerful as a camera system.

State of the art robots such as the Asimo or Spirit, are fairly big and move slowly, thus allowing the image recognition system to have relaxed time constraints (in the order of seconds) to make decisions. In addition, these robots have enough battery resources and space available to allow the use of high performance of the shelf computers. Since these conditions do not apply for our robot called Tinyphoon and shown in figure 1, new approaches for hardware and software implementation had to be considered to allow real-time object recognition in the order of a few milliseconds in a highly dynamic environment.

Figure 1. Tinyphoon robot

Tinyphoon [3] is a two-wheel driven robot, which fits into a cube with an edge length of 75mm. It is equipped with two CMOS-cameras, acceleration, gyro (yaw rate) sensors and has its own “intelligence”. For such a small robot, none of the standard industrial sensor solutions can be used. Because Tinyphoon can reach a speed of up to 2.5m/s, a very fast image processing system supporting a high frame rate had to be realized. The goal of the image recognition system was to recognize specific objects such as the ball and the other players and measuring their relative distance and angle.

The rest of the paper is organized as follows: In section 2 related work in the area of object recognition for mobile robots is presented. We concentrate mainly on work done in the RoboCUP League [4] since it represents a very similar application area but where movements are much slower and hence object recognition is not as time

critical as in the smaller MiroSOT league [5] for which the Tinyphoon robot was designed. Still in section two the requirements for the hardware and the software algorithm are defined and appropriate solutions are discussed. Section 3 describes the image recognition algorithm. Section 4 gives details about the realized system architecture, the hardware components chosen and discusses performance issues. Section 5 evaluates the performance in terms of real-time behavior. Section 6 discusses the results and gives details about the distance and angle measurement system.

2. Problem analysis and related work

Many image recognition systems analyze the image data in two to three steps. First edge detection is performed and a respective edge image stored. Optionally a range of colors can be searched in a second step and the respective areas are marked. Third, a model of the object to be found must be present and a search algorithm must be applied to find the object within the frame based on the available model. The model can be a reference image or edge image. It can be a mathematical description of the object or the like. The comparison between the model and the object can be a very computation intensive task that mainly depends on the type of the object and the possible degrees of freedom. In the soccer application the robot should be able to detect an orange colored ball, the boundaries that have two black markings and the other robots that have rectangular blue or yellow colored shapes on the side, depending on their team membership. These represent some basic objects such as a circle or a rectangle with a certain color, distance and angle in space. Based on a robot soccer system of the category MiroSOT as shown in figure 2, it was our ambition to replace the global camera by local ones mounted on the robots. In MiroSOT 3, 5 or 7 radio controlled robots with a size of 75x75x75mm play soccer supervised by a global camera and controlled by a host computer.

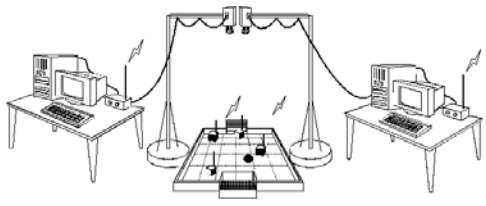

Figure 2. MiroSOT overall system

The camera, mounted above the playground, is a standard industrial CCD camera. The robots are marked at their top with different colors, which the program at the host computer uses to detect the robots positions and angle in space. Details about this type of image recognition system can be found in [14] and [6].

In RoboCUP [15] and [8] middle size league, the robots are much bigger than in MiroSOT and have a diameter of 50cm and a height of 80cm. Each team consists of four robots, which are linked via radio, but only local sensors are allowed. Figure 3 shows a typical RoboCup Robot. These robots are in general equipped with an industrial CCD camera in addition to laser and ultra sonic sensors. There are two different possibilities in mounting the camera. One is to let the camera look in the moving direction of the robot, as shown in figure 3. The other one is to let the camera look to the top, the so-called omni-camera, where a mirror is mounted, which allows a sight of the whole playground, as shown in figure 4. However, a standard industrial camera and the standard PC hardware are much too big and energy wasting and cannot be integrated in a MiroSOT robot. We therefore exploit the option of using small CMOS cameras in combination with state of the art embedded microcontrollers and digital signal processors used for highly demanding applications in mobile multimedia devices. A similar approach using a DSP and a CMOS camera and representing a simple low cost color vision system is described in [7], but this system is much too slow for our application.

Figure 3. Typical RoboCUP robot

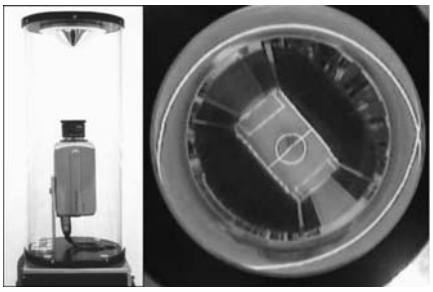

Figure 4. RoboCup camera system

For the RoboCUP league, much work has been done in object recognition where the focus is mainly on identifying a ball, the other robots and their own position on the field. In [8] a ball recognition algorithm is

described that uses simple color-matching algorithms only requiring scanning the image in one pass from the lower left corner to the upper right corner. The disadvantage of a color-only algorithm is its poor robustness due to the strong influence of different lighting conditions. In addition, only the object to be searched can have the color the algorithm is looking for; other patches of the same color not belonging to this object will lead to a failure of the algorithm. However if processing resources are limited, color matching algorithms can be very fast.

Image recognition systems for RoboCUP using an omni-camera as shown in Figure 4 have a view of the whole area of interest, but require much processing to distort the image and have only little resolution available for the single objects due to the broader view of the camera. This results in inaccurate position detection or false detection, which can only be solved by higher resolution cameras requiring even more processing.

Besides the work in [7] most other image recognition systems we have found during literature study use the traditionally approach of a camera, a frame grabber, and an associated computer to interface to the frame grabber and run the algorithm. Timing is also not very critical since these systems are mounted on big robots that move slowly.

The framework for our application is although very different:

- The environment in which the robot acts is highly dynamic. To mention an example, a ball might travel at a speed of 10m/s thus every ms of delay in the image recognition process will make the ball travel 1cm further. One can imagine how hard it is to hit a ball if 10ms – 20 ms are lost due to a poor algorithm or slow hardware.
- The processing power available is limited due to the fact that not only space but also energy is very limited. The robot runs with small batteries providing only 600mAh at 10V, other electronics and mainly the motors will take up most of the available energy hence only little energy is left for image processing.
- The resolution of the camera must be limited. Although we could make use of higher resolution cameras (640x480, 1024x768) the problem is that we would need very fast processing capabilities that stand in contrast to the limited energy, or would reduce the frame rate which does not allow coping with a highly dynamic environment.

For the above reasons, one has to exploit very efficient algorithms as well as an efficient way of implementing these algorithms in hardware and in software depended on the most energy efficient usage at an acceptable flexibility.

The requirements of the image recognition algorithm can be stated as follow:

- The algorithm must take less time than the inverse of the given frame-rate (real-time condition). The goal is to exploit the maximum camera frame rate of 60 frames/s; hence, the upper bound deadline the algorithm has to meet is 16.6ms if no frame is lost.
- It has to be robust against the direction of the light source and against changing brightness levels within ±30% of the optimal illumination.
- Simple objects that may be covered partially should still be recognized. A circle should be detected even if covered by 30-40%
- The algorithm should support different models that allow searching for other objects.
- It should be able to determine the distance and relative angle of objects with a well-known size.

Trying to fit these requirements in energy and space constrained devices, taking account of available hardware and camera modules, leads to the following system requirements of the camera system and processing hardware:

- The camera should be capable of providing the desired 60frames/s at resolutions of 320x240pixels
- It should provide a standardized digital interface such as the ITU656 standard for direct connection to an embedded processor.
- Processing hardware should have very low power consumption, less than 500mW due to the limited battery capacity and desired operating time of at least 30 minutes.
- The entire system including the camera should fit into a small sized package of less than 75x75mm
- In a first order estimate, at least 800MegaMACS (Multiply and Accumulate) operations are required to meat the recognition deadline set to be below 20ms.

There are two basic design options to meet the hardware requirements: The first is a field-programmable gate array (FPGA) based solution in combination with an external microcontroller or medium performance digital signal processor (DSP). The second is a DSP-only based solution where only high performance processors will be taken into consideration for edge and color detection as well as object recognition. The use of an FPGA for the pre-processing of the images would have the advantage that the different pixels could be processed in parallel thus allowing to speed-up the edge detection and the classification of the color detection process. However, as we will see in the performance evaluation at the end of this paper, edge and color classification only account for a small fraction of the whole algorithm. Of course, one could also implement the whole algorithm in hardware but the development is more costly and not as flexible as when moving to a software-centric design model. As can be seen in the datasheets of Altera and Xilinx, two mayor producers of FPGA's a 1Million Gates FPGA such as the Spartan-3 [10] consumes a lot of power compared to state of the art microprocessors shown in the next paragraph. The costs of an FPGA based solution are also higher and therefore we tried to avoid using FPGA's as long as the Performance/Watts/Price ratio could satisfy our needs.

The following embedded microprocessors where evaluated for a FPGA-less solution:

- Analog Devices ADSP-BF533 Blackfin
- Motorola MC9328MXL i.MX
- Texas Instruments TMS320VC5502

The BF533 Blackfin Processor [11] is the latest 16Bit embedded processor from Analog, designed to meet the computational demands and power constraints of embedded audio, video and communications applications. Due to the 0,13um process technology and very low core voltage of 0.8 – 1.2 Volt dependent upon the clock, it requires only 280mW at 600MHz and 1200MMACs (At 25°C room temperature, typical) The Blackfin processor combines a 32-bit RISC instruction set, dual 16-bit multiply accumulate (MAC) DSP functionality and 8-bit video processing commands. A SD-RAM interface and a dedicated parallel port interface supporting the digital video standard ITU656 are supported by the chip.

The Motorola MC9328MXL i.MX media processor [12] is an embedded processor, optimized for mobile multimedia applications in PDA or cellular phone applications. It is based on an ARM920T microprocessor core and has dedicated multimedia acceleration hardware for efficient video processing.

The TMS320VC5502 [13] is a high-performance, fixed-point digital signal processor that is comparable to the Blackfin processor of Analog. In contrast to the Blackfin processor it does not have hardware support for the video interface and far less MMACs/Watt performance.

Company	**Analog Devices** ADSP-BF533 Blackfin	**Motorola** MC9328MXL i.MX	**Texas Instruments** TMS320VC5502
Power	280 mW	1100 mW	200mW
Clock	[1]600 MHz	200 MHz	300 MHz
Pins	160	256	176
MMACs	1200	400	600
Price estimate (1k)	~19 USD	~14 USD	~12 USD
Architecture	16 Bit fixpoint	32 Bit fixpoint	16 Bit fixpoint
Video interfaces	1	1	0
Memory	144 kBytes	144 kBytes	64 kBytes

Table 1: Comparison of Embedded Processors

The highest performance DSP's available in Industry, such as the Tiger Sharc processor (ADSP-TS201) from Analog or the Texas Instruments TMS320C6000, do offer even higher performance than the processors shown in Table 1 but have also much higher power consumption, are more expensive and hence will not be taken into consideration for our implementation.

The processors that support a digital camera interface have an advantage in performance because the CPU does not have to care about low-level video synchronizations and transfers. These interfaces all have DMA (Direct Memory Access) capability, which is the only efficient way of handling the huge amount of data (3 x 320 x 240 x 60fps/s that is about 15Mbytes/s) with no CPU intervention. In this way, the CPU can analyze a frame in the main memory while the camera can grab the next frame and move it to the main memory.

We have decided to use the Blackfin processor from Analog Devices due to its leading Performance/Watts criteria and its small size of only 12x12mm. A FPGA based approach as a comparison to the DSP based approach will be attempted in the future. In a first stage we wanted to test the feasibility on a platform that was software-centric to allow fast modifications of the algorithm.

3. Image recognition algorithm

The algorithm is divided into three steps which are explained in the following. In a first step edge detection and color recognition is performed in one pass by a single software loop and two distinct image frames derived from the original frame are stored in memory. Figure 5 and figure 6 show a sample of the two images directly obtained from a main memory dump.

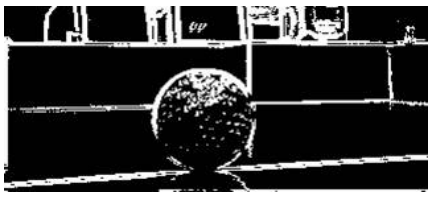

Figure 5. Edge image

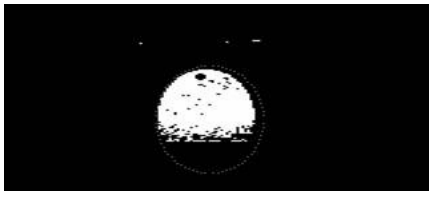

Figure 6. Color marked image

Even though one bit per pixel would be enough for the black and white images, 8 bits are used to mark certain regions for later processing. Most edge detection approaches use local operators (such as the Laplace or Sobl operators) and require a matrix operation for each

[1] Maximum Clock is specified at 750MHz, resulting into 1500MMACS

pixel and hence multiple loops are required. To mention an example of a small 3x3 matrix operator, 9 multiplications and 9 additions are needed for each pixel. The result of the matrix multiplication is an edge image that can be used for further processing. Instead of using local operators we use a simple and efficient method for edge detection that requires only two comparisons for each pixel. This approach consists of comparing the Euclid distance between each pixel and his upper and left neighbour with a certain threshold. To further minimize the loops of the algorithm we combined the edge detection and color detection process in a single loop.

The key of our implementation is that in this first step another operation takes place which we have called "short line detection" or SLD. This allows making a pre-selection of relevant edges that might be part of the objects to be found. This allows us to significantly speed up the recognition process performed in the third step. The short line detection operation marks all relevant pixels that are part of short lines (from 20-50pixels in the case of ball recognition) and stores these lines in a table. With this feature, the search of the objects in the third step is reduced to a local search in the region of the short lines. The advantage can especially be seen in objects such as a ball, which has the characteristic of being always a circle from whatever perspective the image was taken.

In a second step, smoothing of the color patches is performed to eliminate the fraying edges with an "opening" followed by a "closing", a well known approach in image processing. In the third and last step, the actual search of the object takes place. In the case of the ball the short line table is consulted, the first line extracted and in 45 degree angle a search is started in every direction to find another short line which might be part of the circle. If not successful the next short line is evaluated.

4. Implementation

The main components of the implemented object recognition system are depicted in Figure 7.

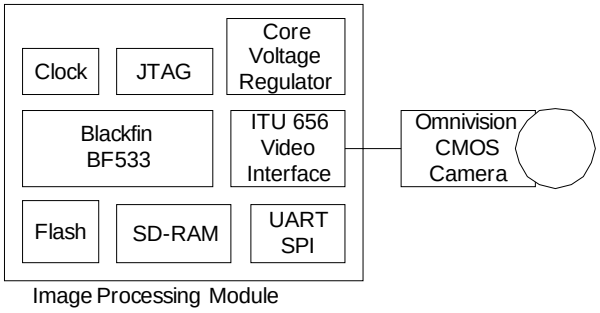

Figure 7. System components

The heart of the image-processing module is the Blackfin processor running at 400MHz (could be clocked at up to 750MHz) and connected to different peripherals such as memory devices and the video camera. Since the internal Level 1 data memory that can be accessed at full speed is by far not big enough to hold an entire image in RGB or YUV format (230Kbytes at a resolution of 320x240), a fast external single chip SD-Ram is attached to the processor. Other mandatory components of the single board processing module are the Flash Memory (16Mbit), the core voltage regulator (0.8-1.2V) and the JTAG interface for debugging and code download. The BF533 possesses a flexible parallel port interface (PPI), which can be configured to support the ITU656 standard by hardware. The UART or SPI ports are used to connect the object recognition module to a fieldbus controller providing a CAN interface to other units on the robot.

The image sensor is a CMOS camera module of Omni Vision that integrates a DSP with digital ITU656 Interface. The module was set to supply the graphic data in the YUV 4:2:2 format. For the edge recognition only the luminance channel is used, for the color recognition only the chrominance channels U and V are used. Figure 8 shows the robot's view of the soccer field. One can recognize the ball, which is a golf ball, the borders – marked by a blue line and the robot itself.

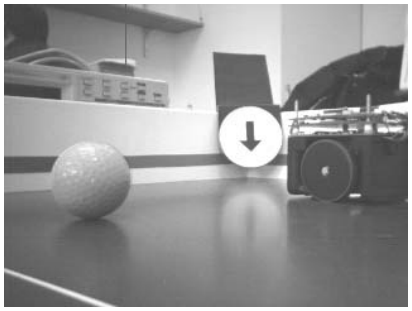

Figure 8. The robot's view

The camera requires only 40mW of power in full operation mode and was designed by Omni Vision for UMTS cellular phones. The resolution can be either VGA or QVGA with 30 or 60 frames per second respectively. For the recognition of simple objects, the standard QVGA resolution (320*240 pixels) is sufficient for the identification of objects with a diameter of less than 4cm up to 100cm of distance. If the objects are further away and their size is smaller than 3cm, the camera mode can be switched by software to source VGA frames where distances of up to 200cm can be covered.

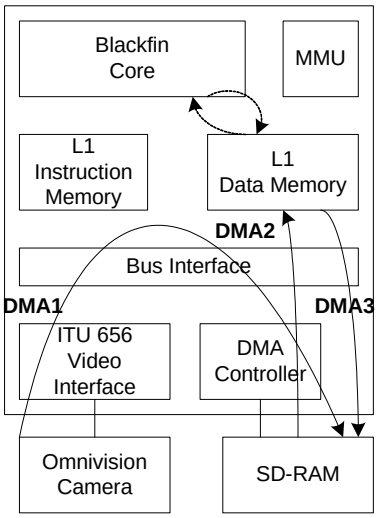

Figure 9. DMA memory transfer

The design challenge of the entire system was to avoid any bottlenecks in the data transfer form camera to the main memory as well as form the main memory to the CPU. We have solved this problem by extensive use of the DMA capabilities of the Blackfin processor that helped not only to speed up the transfers but also to keep the CPU free form performing slow memory accesses. Figure 9 shows the different DMA channels employed by the application. In a first step, the data sourced from the camera is transferred to the main memory. With 15Mbytes/s every 75ns a new byte is available at the video interface. The SD-RAM can be clocked up to 133MHz thus providing an access time of less than 8ns storing two bytes per access. A second DMA channel is transferring one part after another of the video frame into the internal L1 memory that is running at full CPU clock. Unfortunately, only part of the frame fits into the internal memory, therefore we implemented a ring buffer that is continuously filled by the transfers of the DMA channel and emptied by the CPU. A third DMA channel is responsible for writing back the results of the pre-processing such as the black and white edge image and a respective color mapped image. The timing of the different DMA channels can be seen in the next section (figure 10) where a simple performance evaluation is shown.

To increase the efficiency of the implemented edge and color detection, several assembler optimisations were made by using special features of the hardware such as the zero overhead loop counter or by using the dual MAC-accumulate unit efficiently to calculate two Euclid distances in the edge detection process in one single clock cycle. For each edge frame stored in the SD-RAM memory a different color marking code is used to avoid the initialisation of the memory area after each frame. With this approach only every 254 frames the memory must be reinitialized.

5. Evaluation of performance

Real-time behavior in an object recognition system that employs a video camera with a pre-defined frame rate is given when the process of transferring and recognizing an object takes no longer than the inverse of the frame rate. If this cannot be guaranteed, images will be lost, or if stored in RAM the process of recognition will delay at each frame for an additional time. The upper bound delay (T_{ubd}) is hence defined by the frame rate (FR) as follow:

$T_{ub} = \frac{1}{FR}$ where FR = 60 and hence Tub=16.6ms in our implementation. In Figure 10 an overview of the overlapping memory transfers and CPU activity is shown.

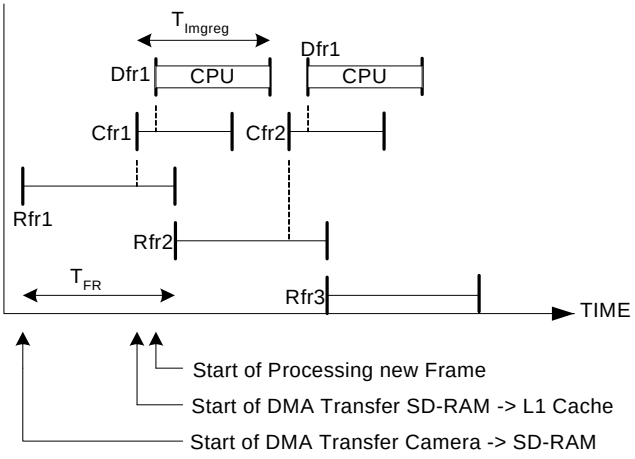

Figure 10: Timing of DMA Transfers

Rfr1, Rfr2 and Rfr3 (RAM frame 1,2,3) are three consecutive DMA transfers from the camera to the main memory. These DMA transfers occupy only 5-10% if the SD-RAM bandwidth, hence other DMA transfers can be overlapped as shown by Cfr1 and Cfr2 (L1 data memory DMA frame). When the internal L1 data memory reaches a certain minimum filling level, the CPU starts to process a new frame as soon as it is ready. Cfr1 and Cfr2 are bi-directional DMA transfers occupying the bus in a multiplexed fashion. T_{imgreg} is the Time of the image recognition process and should be always less than the upper bound delay T_{ub}. TFR is the time it takes to transfer a single frame. Since the camera is continuously sourcing data the transfer time equals to the frame rate.

In order to evaluate the image recognition performance we make the following assumptions:

- The robot is searching for two different types of objects (defined by two different models) in every frame.

- In the actual environment where the tests were performed, at most only one object of every type is available and can be captured by a single frame.

It is very difficult to extract the longest processing time path from the algorithm by counting the loops and estimating the memory accesses, therefore we preferred the more empirical approach of measuring the processing time based on an available set of scenarios. We measured the time it takes the CPU from the first access to the dataset of a new frame to the end of processing a frame where found objects and their distances and relative angles where ready for transmission over one of the serial interfaces of the Blackfin processor. After applying 20 different samples to the algorithm a mean processing time of 11ms was measured. The minimum and maximum processing time did just differ by ±15%. The total image recognition time (T_{imgreg}) is in any case less than the upper bound time of 16.6ms, which confirmed that real time was achievable under the given circumstances.

6. Results and future research

With the implementation of this mobile image recognition system we tried to show how a whole system design approach optimizing all components of the system, from using hardware efficiently to DMA based memory transfers and the algorithm can lead to an implementation that is not only very power efficient but also capable of real time. At our best knowledge there is no other available solution in industry requiring only 250mW (Camera + DSP@400MHz) and capable of recognizing simple objects in real-time with 60frames/s.

One interesting result is the fact that most of the processing time is not wasted for the edge or color detection process but for the later analysis and search of a given object. While the edge and color detection is performed in 1,5ms, the actual object recognition based on two different objects did take more that 9ms even though the short line detection did speed up the search process significantly.

The disadvantage of the actual implementation is the fixed camera, which reduces the view of the robot to an angle of 55degree in the direction of the actual movement of the robot. To allow the tracking of moving objects while the robot itself also moves, a pivoted camera system will be developed. The camera will be mounted on a rotating head, which allows the robot to look around 360degrees independently from the orientation of the robot.

In a next step, we want to equip the image recognition system of the robot with a tiny stereo camera system to allow the robot to recognize 3-D objects (ball, other robots, boundary and goal) and their relative distances to the camera system. Thus, it will not be necessary anymore to solely rely on color patches; instead, more of the object detection can be done in the space of edges, because available in stereo.

Finally, in order to solve the problem of adapting source code for each object, a meta-language for describing the objects should be developed. Such a higher-level description language permits to join geometrical simple objects of basic shapes together and allow the robot to search for targets that are more complex. We intend to use XML for this task.

7. References

[1] Information about Hondas ASIMO robot can be found at: *http://world.honda.com/ASI*

[2] Information about Spirit can be found at: *http://marsrovers.jpl.nasa.gov/home/index.html*

[3] The Homepage of the Tinyphoon robot can be found at: *www.tinyphoon.com.*

[4] Information on the MiroSot league can be found at: *http://www.fira.net/soccer/mirosot/overview.html*

[5] Official RoboCUP Homepage: *www.roboocup.org*

[6] M. Biribauer: Ein Bildverarbeitungssystem zur Navigation mobiler Roboter, VUT, March 2002, diploma work (German).

[7] A. Rowe, C. Rosenberg and I. Nourbakhsh: *A Simple Low Cost Color Vision System*, Tech sketch paper presented at CVPR 2001.

[8] M. Jamzad, B.S. Sadjad, V.S. Mirrokni, M. Kazemi, H. Chitsaz, a. Heydarnoori, M.T. Hajiaghai, and E. Chiniforoosha: *A Fast Vision System for Middle Size Robots in Robocup, Robocup 2001*: Robot Soccer world cup V, Springer Verlag , pp. 71-80, 2001.

[9] R. Sargent and B. Bailey and C. Witty and A. Wright: Dynamic Object Capture Using Fast Vision Tracking, AI Magazine, vol. 18, no. 1, 1997.

[10] Xilinx Spartan-3 FPGA found at: www.xilinx.com

[11] Blackfin BF53x found at: www.analog.com/blackfin

[12]Motorola MC9328MXL: http://ewww.motorola.com/

[13]Texas Instruments TSM320C5502 found at: www.ti.com/dsp

[14]G. Novak: Multi Agent Systems - Robot Soccer, VUT, April 2002, PhD theses.

[15]E. Schulenburg: Selbstlokalisierung im Roboter-Fußball unter Verwendung einer omnidirektionalen Kamera, Freiburg 2003 Diploma Work, found under: http://www.informatik.uni-freiburg.de/~robocup/publications.htm (German)