WF Wolves KidSize Team Description
RoboCup 2010

Prof. Dr. R. Gerndt, Stefan Krupop, Jan Carstensen,
Sönke Michalik, Sören Michalik, and
Richard Benedix

Ostfalia Hochschule für angewandte WIssenschaften, Department of Computer
Sciences,
Am Exer 2, 38302 Wolfenbüttel, Germany
`{r.gerndt,s.krupop,j.carstensen`
`soe.michalik,so.michalik,r.benedix`
`}@ostfalia.de`
`http://robocup.ostfalia.de`

Abstract. This is the Team Description of the WF Wolves, the RoboCup Team of the University of Applied Sciences Wolfenbüttel, for the RoboCup 2009. The Team has been in the Mixed Reality since 2007. Last year, we won the Championship. After bringing many improvements to the Mixed Reality we are currently looking forward in competing in the Humanoid KidSize League as well. Our team uses a modified RoboBuilder-Huno.

1 Introduction

Our team has been founded at the beginning of 2007 at this University of Applied Sciences. It participated in the RoboCup 2007 in Atlanta, where it gained the fourth place in the Mixed Reality Competition. At the RoboCup 2008, our team has won the Championship in the Mixed Reality. Our Team provided many developments to the Mixed Reality, such as the Battery Charger 2008, Programming Adapter, EcoBe!2008 Firmware and various hardware improvements of the Ecobe!2008. Since 2008, we are working to set up our humanoid team. We are using a software which allows us to use the Agents from other leagues such as the Mixed Reality with the humanoids. In 2009 we participated in the Humanoid League.

2 About the Team

This team currently has twelve student team members from different faculties such as computer science, electrical engineering, mechanical engineering and law. It is organized independently but supported by the faculties.
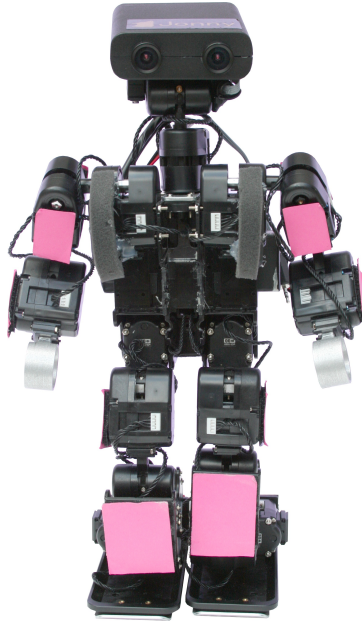
3   Mechanical and Electrical Design

Fig. 1. WF Wolves 2009

The basic set of the Robot-Builder was enhanced by stronger servos, it was equipped with a main processing unit with 2x 2Megapixel cameras and a new servo controller board with IMU support (3-axis acceleration and gyro sensors).

The main proccesing unit works with an Analog Devices Blackfin BF-561 Dual-Core DSP Processor with each core running at 500 MHz . The module has additional 64MB SDRam and 8MB Flash for the demanding image processing algorithm and the artificial intelligence running on an uClinux operating system.

The board features an micro-sd slot and a USB2.0 device port for external usb devices and uses UART to communicate with the servo controller board. It is also equippes with 802.11g WLAN for remote debugging and communication with the other robots an the field.

The DSP processor architecture features two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bit shifter UART for processing the images. The Dual 12-Channel DMA is used to transfer the camera-data into the SD-Ram and internal L1 Cache with multiple Transfer-Buffers. with the multiple DMA-Channels the image data can be transferred simultaneously from both camera ports.

The IMU-Board is based on an AT91SAM7X256 ARM7 microcontroller. The controller has a direct connection to the IMU sensor and controls the servo

motors. It runs at 50MHz, so we are able to calculate the demanding kalman filter data and adjust the movements simultaneously.

There is a pan/tilt unit added to the top. We reduced the size of the feet to make the size fit the rules. Some parts of the leg assembly are completely redesigned and CNC-manufactured replacements for old plastic parts.

The power for the robot servos and the processing unit is supplied by a LiPo accumulator. The Robot has 18 Degrees of Freedom total. Each Leg has five Degrees of Freedom, each arm has three and two are used to move the head (pan/tilt).

4   Perception

4.1   Stereo Image Processing

Our perception aim is to locate objects as well as an amount of marks on the field. Because we use a stereo vision, we are able to calculate the distance to the object very accurately. By scanning the picture with an speed-optimized Scanline-algorithm with color-segmentation, we can detect or estimate the colored object positions and distances in few cycles. Using this information we can only process these preprocessed segments of the image, using windowing. The algorithm is able to detect just objects on the field to reduce errors caused by false detected colors surrounding the field. We are using the YUV image format, that supports a detection of colors with different light intensities. The Stereo processing algorithm uses both camera frames to calculate the image disparity. the disparity is used to estimate robots distance to the object.
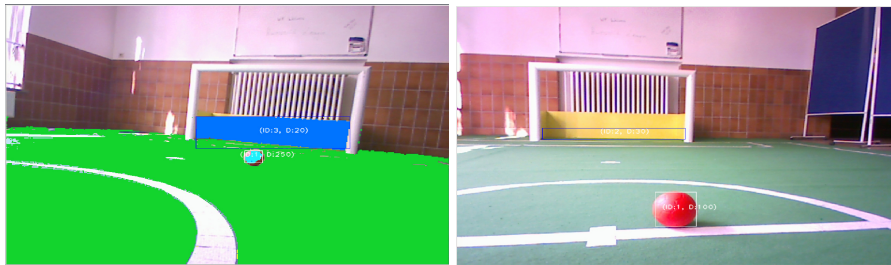

Fig. 2. Color Segmentation and detected Objects

Beside the detection of objects we are using an algorithm which detects the lines on the field. Lines are represented by an array of coordinates of detected line-points. These points can be matched with an particle filter to estimate the robots position on the field. Using the found object positions and distances, the robot will try to determine its position more precisely. Using triangulation and trilateration we are able to get an amount of possible locations for each state. Taking the last position and getting the movement data, the Robot will update to the most probable position.

4.2   IMU supported Movements

We are using a control-algorithm to adjust the movements of the robot. The algorithm prevent the robot of falling and is nesesary to walk stable at higher speed.

5   Robot Control

We are using a framework which allows us to use the same Agent we are using in the Mixed Reality League for the humanoid League. We have made some major changes to the Robot Control since Graz 2009. We are currently using uClinux as an embedded Operating System for the robots. Since we decided to let go the PDAs used for high level control in the previous version of the robots in order to save weight and to improve walking abilities, we needed a new platform for the robot system. Because the vision system running on the dual-core Blackfin module did not completely utilize the DSP processors resources, we decided to move the processes previously running on the PDA into that system. To achieve this, we decided to use an embedded Linux as the operating system: uClinux, for which a Blackfin port existed. The first step was to make the downloaded version of the distribution compile and run on our custom Head processor board. After the Linux did successfully boot for the first time, we successively added drivers for the hardware used on the processor board. These include drivers for a secondary serial port, Micro-SD cards, the USB controller and both cameras of the vision system. While for most hardware only minor changes to the supplied drivers were necessary, the camera driver supported only a single camera. One of our achievements was the enhancement of the driver to support capturing images from to cameras at the same time and adding support for our particular camera module. The final adapted version of the uClinux image now running on our robots provides a complete and easy to use environment for writing the robot control and vision systems, which now can be developed as normal Linux programs thanks to using the same driver architectures on our embedded system as they would be used on a pc (e.g. Video4Linux for capturing images from the cameras). By using an embedded version of Linux as our base system, we can use many well-known tools from the open source scene and have excellent debugging support in our toolchain. Besides the new OS we have a localization which gives us the possibility to save information we gathered in the perception.