EAGLE KNIGHTS: SMALL SIZE ROBOCUP SOCCER TEAM

Luis A. Martinez-Gomez[*]
arrobalf@yahoo.com

Misael Soto[*]
dasot533@hotmail.com

Ismael Diakite[*]
argos_77587@yahoo.com

Guillermo Rodriguez[*]
garsoltero@yahoo.com

David Sotelo[*]
deigal@hotmail.com

Ernesto Torres[*]
dargolad@gmail.com

Octavio Ponce[*]
octaviopm@hotmail.com

Alfredo Weitzenfeld[*]
alfredo@itam.mx

*Instituto Tecnológico Autónomo de México (ITAM) – Computer Engineering Department
Río Hondo #1, San Angel Tizapán, CP 01000
Mexico City, Mexico

ABSTRACT

In this paper we present the design and implementation of our Small Size League RoboCup Soccer Team – Eagle Knights. We explain the three main components of our architecture: Vision System, AI System and Robots. Each element is an independent entity and therefore the explanation focus on the improvements made to our 3rd generation of robots and the way these changes interact with the rest of the elements in the team architecture.

KEYWORDS: Small-size, robocup, autonomous, vision, architecture.

1 INTRODUCTION

RoboCup [Kitano 1995] is an international joint project to promote AI, robotics and related field. In the Small Size League, two teams of five robots up to 18 cm in diameter play soccer on a 4 by 5.4 m carpeted soccer field.

We have a classic architecture of a team in the *Small Size League* (SSL) with four main elements: the vision system, the AI system, five robots and the referee box.

The vision system digitally processes two video signals from the cameras mounted on top of the field. It computes the position of the ball and robots on the field, including orientation of the robots in its own team transmitting the information back to the AI system.

The AI system receives the information and makes strategic decisions. The actions of the team are based in a set of roles (goalkeeper, defense, forward) that exhibit behaviors according to the current state of the game. To avoid collision with robots of the opposite team potential fields are used [Khatib 1986]. The decisions are converted to commands that are sent back to the robots via a wireless link. The robots execute these commands and produce mechanical actions as ordered by the AI system. This cycle is repeated 60 times per second. Finally the referee can communicate additional decisions (infraction, goal scored, start of the game, etc.) sending a set of predefined commands to the AI system through a serial link. Figure 1 shows a schematic of the architecture.

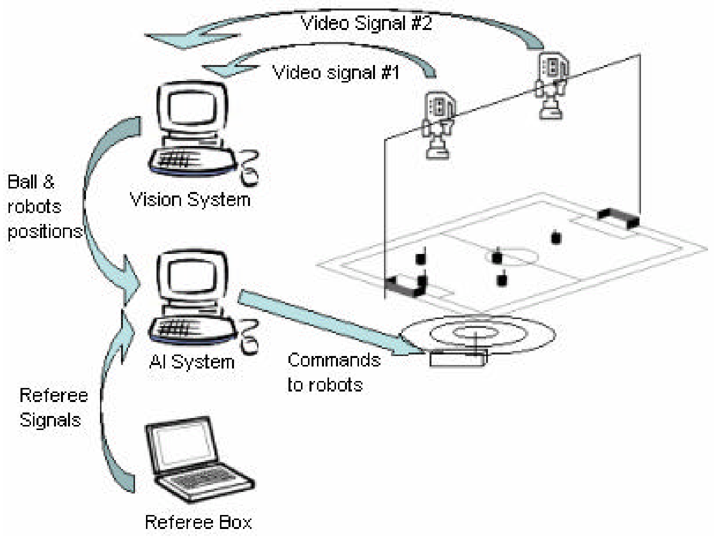


Fig 1. Typical architecture of a SSL team.

In the next sections we describe the improvements in each component of the architecture.

Acknowledgements: This work has been supported by collaboration project NSF-CONACYT 42440, UC MEXUS-CONACYT, LAFMI-CONACYT and "Asociación Mexicana de Cultura, A.C.".

2 VISION SYSTEM

The vision system is the only source of feedback in the whole architecture, if the data returned by the vision system is inaccurate or incorrect the overall performance of the team will be severely affected. That's why the vision system should be robust enough to compensate for any possible errors.

The main object characteristics used by the vision system are the colors defined in the rules of the SSL [Verret 1995]. The ball is a standard orange golf ball. The robots of one team must have on top of them a 50 mm in diameter blue colored circle while the other team must have a yellow patch.

The main tasks of the vision system are:

1. Capture video from the cameras mounted on top of the field in real time.
2. Recognize the set of colors assigned in the rules to the objects of interest in the field (robots and ball).
3. Identify and compute the orientation and position of the robots in the team.
4. Compute the position of the robots of the opposite team.
5. Transmit the information back to the AI system.
6. Adapt to different light conditions (color calibration procedure).

The system has several modules; each module is a functional block with a specific task. Figure 2 shows the vision system architecture.

CAPTURE MODULE. We use two off the shelf cameras with an IEEE 1394 link. The frame capture is done with MS DirectShow that allows us to configure the resolution of the image, the space color and the frame rate. By default we capture RGB images with a 720x480 resolution at 30 fps.

PREPROCESSING MODULE. The preprocessing module is used to improve the quality of the image, such as brightness, contrast, gamma, etc. Although this module is crucial for a proper operation its relevance has been reduced to a minimum thanks to the color calibration procedure.

OBJECT CALIBRATION MODULE. This module is a tool to establish the thresholds of each component according to the space color for every object of interest (robots and ball). The calibration is done in HSV color space where selected thresholds are more robust to changes in field lighting and color changes. The HSV thresholds are then transformed to RGB values to improve segmentation speed and avoid costly color space translation processing. This procedure is the most important improvement to our previous vision system because it allows to accurately discriminate the colors we are interested in. With this change the errors in the localization of the robots has been reduced significantly.

SEGMENTATION MODULE. Assigns each pixel of the images into object classes. The module consist of two segmenters, each one using the thresholds values assigned to the camera for every object of interest. The RGB thresholds are mapped to an array where a logical AND is computed to segment 32 colors at the same time.

BLOB BUILDER MODULE. Connects the segmented pixels into blobs. Before reaching this module the image is composed of separate pixels; when a blob is constructed useful information is computed such as the area, centroid, bounding box, etc. An RLE and four-neighbor search are computed. A joint list of blobs for the two cameras is generated for each color.

ACTIVATION/DEACTIVATION MODULE. Enables or disables the use of a particular robot. Sometimes a team can play with less robots so this information is useful to avoid unnecessary searching processes.

RECOGNITION MODULE. Selects the regions that adjust better to the objects searched. It has selection criteria for every kind of object. For the ball we select the orange blob that is nearest to an area of 85 pixels (with an image resolution of 720x480). For the robots of the opposite team the selection criteria consists in selecting the blobs of the corresponding color of the central patch with an area nearest to 115 pixels (the area of the patch is bigger than the ball). The number of blobs selected is determined in the Activation/Deactivation module. For our team the procedure is similar to the one used for the robots of the opposite team, but additional to the central patch, a search for extra patches is necessary. The extra patches are employed for identification and orientation computation.

GEOMETRIC CALIBRATION MODULE. This module computes the internal and external parameters of the cameras using the Tsai method [Tsai 1987] in OpenCV [OpenCV 2005]. This parameters are used to correct the distortion produced by the lenses of the camera.

LOCALIZATION MODULE. Computes the position of all objects in the field. It uses the camera parameters obtained in the Geometric Calibration module to undistort the image. Also computes the orientation of robots in the team. This computation is done using a set of user-defined points. Each point represent a well known landmark of the field (corners, midline point, etc), when a point appears in both camera images their coordinates are used to match them and discard duplicate pixels data.

GRAPHIC DISPLAY MODULE. It's responsible for displaying video images in the screen and for generating basic drawing functions such as lines, circles, etc. in the video image.

TRANSMISSION MODULE. A UDP network link is setup for the communication between the vision system and the AI system. The module builds a structure appropriate for data transmission. In practice the vision system can perform communication with two or more hosts allowing for distributed AI system processing if necessary.

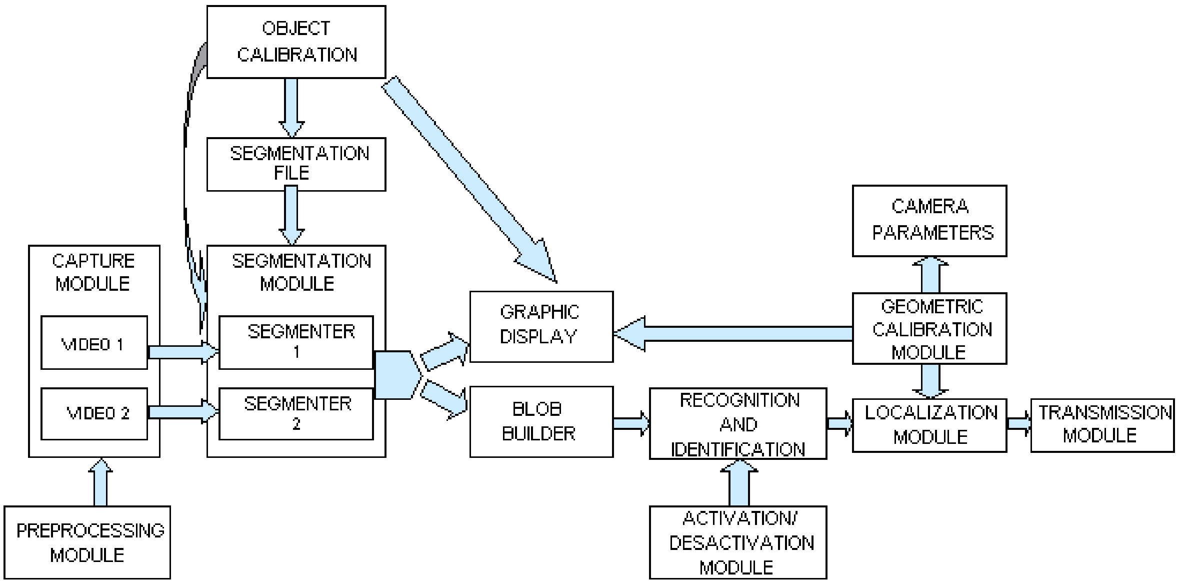
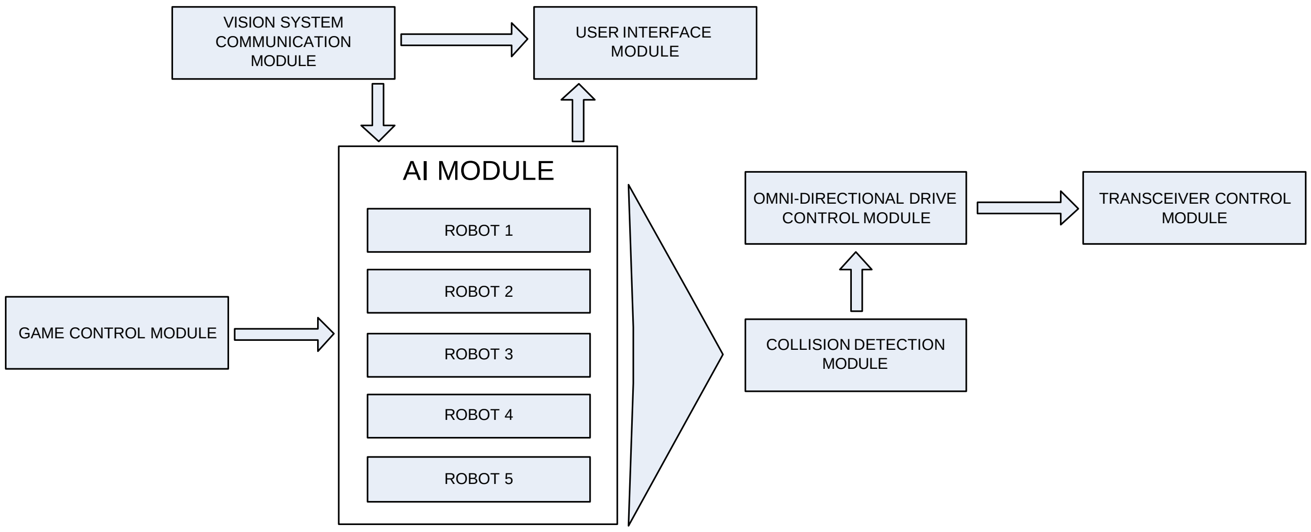


Fig 2. Vision System Modules.



VISION SYSTEM COMMUNICATION MODULE
USER INTERFACE MODULE
AI MODULE
ROBOT 1
ROBOT 2
ROBOT 3
ROBOT 4
ROBOT 5
GAME CONTROL MODULE
OMNI-DIRECTIONAL DRIVE CONTROL MODULE
TRANSCEIVER CONTROL MODULE
COLLISION DETECTION MODULE

Fig 3. Artificial Intelligence System Modules.

3 AI SYSTEM

The AI System or High Level Control Application is formed by seven modules named: Artificial Intelligence, Collision Detection, Transceiver Communication, Omni-directional Drive Control, User Interface, Vision's System Communication and Game Control. The system has a main thread that loops all the time and calls function in each of the different modules. This system is designed in a way that the user can test each module separately. It doesn't need all modules connected and working to make simple tests. The system also has a dynamics simulator in order to test system functions initially in the computer, including collision detection, AI and robot control. Figure 3 shows the interaction between the AI modules.

MAIN THREAD. The Control System has a function that loops all the time that we call the main thread. In the iteration we call each of the functions of the different modules to finally get the commands sent to each one of the robots. The main thread first checks for the vision system's information to know where the robots are. Then we have to check the game state we are in (game state is controlled by the referee). With the robot's coordinates and the game state we call the AI function that returns the desired position to move for each robot and the actions to take. Each robot has to avoid collisions so the future positions are sent to a potential field function that returns the moving vector that avoids the obstacles. With the moving vector we have to calculate the speed for each one of the four wheels of the robot. This task is done by the drive control function. Finally we build the packets for all the robots with the speed and the actions and we send them with the help of the transceiver.

AI MODULE. This module receives the robot's positions, the ball position, the robot's angles, the game state, the robot's roles, the strategy and the direction we're shooting. With all this information the system calculates the future position and the actions for each robot. We can have different strategies that give us the chance to change the game play for each team or situation. We can program and integrate a new strategy in a very easy way thanks to the simple design of this module.

We have three roles named: Goalkeeper, Defense and Forward. The task of the goalie is avoiding at all cost receiving a goal. His behavior is simple and effective; at all times we compute the maximum blocking area we can accomplish with the robot inside its own defense area. The defense behaviors are a combination of maximizing the obstruction of a possible shot to the goal and blocking the passing strategy of the opposite team. To avoid crowding a field area, each robot has a moveable area of influence where it can act with complete freedom, if two influence areas intersect then one of the robots does not invade the other one.

The forward task is to score goals; the behavior of this role is always to go for the ball when the robot passes the field midline looking for a direct shooting line to the opposite goal.

COLLISION DETECTION MODULE. This module receives the movement vector, robot's positions, the ball position, and the robot's angles and returns the new movement vector that assures that there will be no collisions on the movement. The vector is the outcome of a potential field that is the combination of multiple potential fields, each field is locally assigned according to its position on the field, the object and the robot behavior.

TRANSCEIVER COMMUNICATION MODULE. This module receives the speed for each one of the robot's motors and the actions to take. This module builds the packets that we will send using our transceiver. It also makes sure the communication is active at all times.

OMNI-DIRECTIONAL DRIVE CONTROL MODULE. This module receives the movement vector and returns the speed for each one of the four robot's motors. Since we have four omni-directional wheels we need this module to know the speed on each motor so that we move on the desired direction.

USER INTERFACE MODULE. This module is constantly displaying all the information we need for each robot. We have the coordinates, the angle, the motor's speeds, the desired positions, the id, the actions, the game state, the referee's commands and the game positions for each robot. The coordinates, angles, desired positions and actions are displayed graphically in a GUI system programmed using OpenGL [OpenGL 2005].

VISION SYSTEM COMMUNICATION MODULE. In this module we receive the packets sent by the vision system containing what we call the scenario that has the robot's coordinates, the ball coordinate and the robot's angles.

GAME CONTROL MODULE. This module receives all the referee commands through the serial interface and returns the game state of the game. This module is based on the SSL rules.

4 ROBOTS

We design and built five omni-directional robots. Each robot has five motors Faulhaber [MicroMo 2005] 2224006SR with gearheads 30.7:1 (four for the wheels and one for the dribbler), a low resistance solenoid, a DSP, a transceiver, one printed circuit boards and two batteries. Figure 4 shows the Eagle Knights robots.

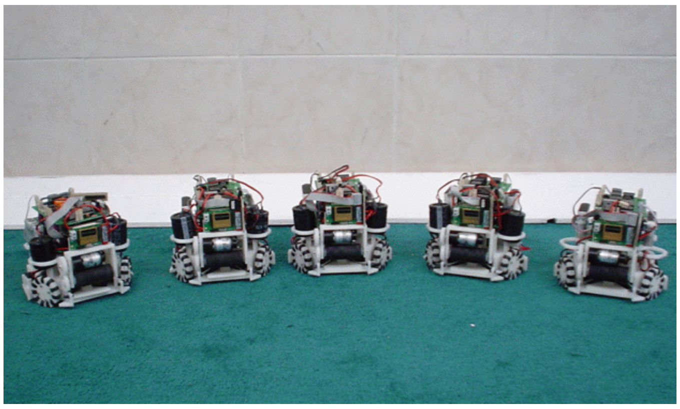


Fig 4. Eagle Knights Robots.

The robot receives commands from the AI system and executes them. To accomplish that each robot has the next functional elements:

ROBOT ID. Each robot incorporates an identification circuit manually setup with a dipswitch making it easy to modify its id if necessary any time during the match.

DSP. The robot micro controller is the Texas Instruments TMS320LF2812 fixed-point single chip DSP (Digital Signal Processor). This device offers low power and high-performance processing capabilities, optimized for digital motor and motion control. The DSP consists of six major blocks of logic: (1) External program and data memory, (2) Analog Interface, (3) I/O Interface, (4) Expansion interface, (5) JTAG Interface, and (6) Parallel Port JTAG Controller Interface, while we currently use only three of these:

- External program and data memory. The RAM module is used in debugging the software with the Parallel Port JTAG Controller Interface.
- I/O Interface. The interface contains different kinds of pins:

    Capture units: used for capturing the rising pulses generated by the motor encoders which can be used to measure speed and direction of the moving motor.

    PWM outputs: these pins have an associated compare unit. A periodic value is established to determine the size of the PWM, and the compare value is used to change the duty cycle.

    Standard I/O: used to read and write values for transceiver communication, motor, kicker and dribbler control.

MOTOR CONTROL. The motor encoders generate a number of square pulses for each completed turn. Each pulse is captured using the DSP and the *feedback* speed is computed. *Feedback* speed along with *received* speed from the transceiver is used as inputs to the PID algorithm in calculating an adjusted PWM signal sent back to the motor. Figure 5 illustrates the procedure.

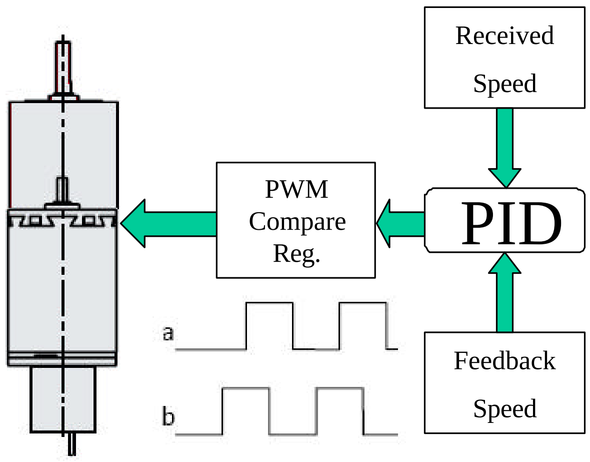


Fig. 5 Motor control.

WIRELESS COMMUNICATION. Wireless communication is controlled by two Radiometrix RPC-914/869-64 transceivers with radio frequency at either 914MHz or 869MHz. The transceiver module is a self-contained plug-in radio incorporating a 64kbit/s packet controller with a parallel port interface.

Data is transferred between the RPC and the host (either DSP or PC) four bits at a time using a fully asynchronous protocol. The nibbles are always sent in pairs to form a byte, having the Least Significant Nibble (bits 0 to 3) transferred first, followed by the Most Significant Nibble (bits 4 to 7). Two pairs of handshake lines REQUEST & ACCEPT, control the flow of data in each direction.

Each data packet has 26 bytes, 1 preamble byte (sync byte and an error check sum) and 25 bytes for the 5 robots (5 bytes for each robot). The first byte (control) specify the status of kicker and dribbler (activated / deactivated), and the other four specify the desired speed for each of the four motors in the robot.

KICKER CONTROL SYSTEM. Small Size soccer robots use different kicking designs to push the ball towards the opposite's team goal. We use a push type solenoid that kicks the ball.

Solenoid kicker system needs a high power supply. For size restrictions robots have only two 7.4V/2100mA batteries, equivalent to 31 Watts of power. With this amount of power we obtain less than the solenoid requires for a minimum performance.

The main idea in power elevation is to store energy, then discharge it when solenoid is activated. To solve this power problem we implemented a four-layer system described below:

- Oscillation generation. Voltage transformers need an AC signal but robots batteries are DC so, in order to accomplish a voltage transformation, we need an oscillating voltage source. Oscillating voltage can be obtained with the DSP. We can generate a 3.3V PWM signal, 50% duty cycle signal at 100KHz.
- Voltage transformation. The oscillating voltage obtained from the DSP is a low power signal, to increase signal power we use an H transistor bridge: L298N. The two direction control bits in L298N are fed from the oscillating signal, one is the inverse of the other. Enable bit is controlled by the DSP to avoid unnecessary voltage transformation. The input power signal in L298N is obtained from the two serial connected 7.4V/2100mA DC batteries. It represents an input 15.8V signal, so the output in L298N is a 29.6V peak to peak, 50%duty cycle at 100KHz signal. L298N output signal feeds a 24/120V voltage transformer. The voltage in the output transformer signal is a 148V peak to peak, 50% duty cycle at 100KHz signal.
- Charge accumulation. The charge in a capacitor is the number of electrons on the two plates. This involves the difference in the quantity of electrons and the unit of quantity is the coulomb.

$$Q = CE$$

Where:

Q = Coulombs.

C = Capacitance in Farads

E = Volts

We use two 2200mF capacitors. Before charging them, the input signal needs to be rectified, where we use a full wave rectifier diode bridge across a DB106.

The full charge capacitor time is 8 seconds approximately. It means solenoid can be activated every 8 seconds.

- Discharge and solenoid activation. If robot has a ball and it is in score position, then the robot needs to kick the ball. An infrared sensor system in the bottom of the robot senses if the robot has the ball, an output sensor system bit indicates this. DSP sends a high-level output bit when the robot is in score position. To discharge the capacitors into the solenoid we need to assure that the robot has the ball too. The Discharge layer uses both the DSP kick bit and the infrared ball detector output bit to discharge the capacitors. Because the capacitors charge level is very high, we need to discharge it using a power mosfet. We selected the NTE 2388. A PWM signal is sent to the MOSFET to control the flow of current through it and thus controlling the intensity of the kick.

Finally the robots were manufactured using a CNC ABC plastic machine at ITAM´s facilities. The circuits were tested successfully and PCBs were ordered. Robot integration took only a week after all parts were ready.

5 CONCLUSIONS

We presented the design and implementation of the SSL Eagle Knights software and hardware modules. The functional blocks of the software systems (Vision and AI) have been explained in detail. We also described the robot's hardware with detailed information of the kicker control system.

Our team has been the first Latinamerican team to be among the top 3 of an official RoboCup Open international competition (USOpen). We have publicly released our Vision System and documentation of our electronics and DSP software to promote the participation of others teams in this initiative.

Our future goals are to better integrate the hardware and software system in order to accomplish a passing strategy using the Djikstra [Cormen 1990] shortest route algorithm where each robot is seen as network node where the task is to move the ball to the opposite goal in a near-optimal trajectory. This approach implies a reconfigurable network because the opposite team will be constantly changing their position in the field so our passing decisions will be valid only at the specific moment they are taken and the subsequent decision tree will be computed when the moment comes.

A successful implementation of this strategy needs robot hardware capable of kicking the ball at different speeds and at a precise direction, together with the ability of effectively receiving a pass. The hardware and software presented here is the first step to reach this goal.

More information can be found in http://robotica.itam.mx/.

REFERENCES

[Cormen 1990] Cormen, T. H.; Leiserson C. E.; & Rivest R. L. (1990) Introduction to Algorithms. MIT Press

[Khatib 1986] Khatib, O.: Real-time Obstacle Avoidance for Manipulators and Mobile Robots.The International Journal of Robotics Research 5 (1986) 90–98

[Kitano 1995] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In Proceedings of the IJCAI-95 Workshop on Entertainment and AI/ALife, 1995.

[MicroMo 2005] MicroMo Product Catalog. http://www.faulhaber-group.com/n41799/n.html

[OpenCV 2005] Open Computer Vision Library. http://www.intel.com/research/mrl/research/opencv/

[OpenGL 2005] OpenGL Open Graphics Library. http://www.opengl.org/

[Tsai 1987] Tsai, R.Y. A versatile camera calibration technique for high accuracy 3D machine vision using off-the-shell TV cameras and lenses. IEEE Journal of robotics and Automation, 1987

[Verret 2005] Verret, Ball, Kiat Ng. Laws of the F180 League Release 3.00a. http://www.itee.uq.edu.au/~wyeth/F180%20Rules/index.htm.