Global cross platform robot soccer development

Petter Reinholdtsen <pere@hungry.com>

The CIIPS Glory robot soccer team consists of five 15 cm tall robots running specially developed operating system and control programs. The system is developed in Germany and Australia, using cross compilers running on Linux and Windows.

In this presentation, you will learn the following:

- What is Robot Soccer?
- Who are CIIPS Glory?
- What is the Eyebot and which OS is it running?
- How do we handle global cross platform development?
- How do robots play soccer?

Robot Soccer

- Since 1996
- MIROSOT ($7.5 \times 7.5 \times 7.5cm^3$)
- RoboCup (Leagues; simulation, small, middle, SONY legged and humanoid)
- Next RoboCup in Melbourne 2000-08-26
  - 73 simulation teams
  - 28 small size teams
  - 31 middle size teams
  - 5 SONY legged teams
  - 3 humanoid teams

CIIPS Glory

- The only robot soccer team in WA
- Located at CIIPS/EE-UWA
- Small size league (2×10 minutes)
  - $274 \times 152.5cm^2$ field (ping-pong table)
  - Max 5 robots on each team
- Hopefully humanoid league
- Local (on-board) and global (overhead) vision
- Eyebot soccer robots
- Previous experience
  - PRICAI'98 (Singapore November 1998)
  - Interact'99 (Melbourne August 1999)

Eyebot Hardware

- Motorola 68332 (25-40 MHz)
- 1 MB RAM and 512 KB Flash ROM
- Color camera (80x60 24bit)
- RS232 38400bps radio (418 or 433 MHz)
- IR distance sensors
- Two wheels with encoders
- Digital and analog compass
- Servo (kicker and camera control)
- Microphone and speaker
- Acceleration sensors

Robot OS (RoBIOS)

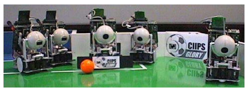
- Features (in addition to HW drivers)
  - Multi-threading
  - OS upgrade using serial connection
  - Hardware description table
  - Up to three programs in ROM (optional autoloading)
  - Simulator
- Complete size < 128 KB
- M68 assembly and C using GNU C cross compiler (v2.95.2)
- GNU C library is available
- Developed in Germany and Australia (CVS)
- Bugzilla bug tracking system (`http://www.mozilla.org/bugs/`).


```
#include <eyebot.h>
#include <stdlib.h>
MotorHandle     leftmotor, leftmotor;
PSDHandle       psd_front, psd_left, psd_right;
VWHandle        vw;
int
main()
{
  LCDMenu("","","","END");
  vw=VWInit(VW_DRIVE,1);

  psd_front = PSDInit(PSD_FRONT);
  psd_left = PSDInit(PSD_LEFT);
  psd_right = PSDInit(PSD_RIGHT);
  PSDStart(psd_front | psd_left | psd_right, TRUE);
  VWStartControl(vw, 7,0.3,7,0.1);

  while (KEYRead() != KEY4)
    {
      if( (PSDGet(psd_front) > 100 && PSDGet(psd_left) > 100
        && PSDGet(psd_right)>100 ))
        VWDriveStraight(vw,0.01,0.2);
      else
        {
          LCDPutString("Backwards \n");
          VWDriveStraight(vw,-0.04,0.2);
          VWDriveWait(vw);

          if((rand()%2)==0)
            {
              VWDriveTurn(vw,1.57,0.6);
              LCDPutString("Turning Left \n");
            }
          else
            {
              VWDriveTurn(vw,-1.57,0.6);
              LCDPutString("Turning Right \n");
            }
          VWDriveWait(vw);
        }
    }/* end while loop */
  return 0;
}
```


Client/server CVS

- Concurrent Versions System
  - Tag source with what, who and why
  - Allow distributed development
- Server on Linux, clients on Linux and Win32
- Nightly build to check consistency
- Auto-generated ChangeLog
- More info on `http://www.sourcegear.com/CVS/`

How do robots play soccer



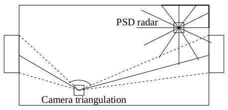
The goal is robot soccer without human interference. The long term goal is to win humanoid football in 2050.

- Sensor input processing
- Self localization, planning and decision making
- Action

Input sensor processing

- Vision
  - Frame rate is everything
  - Overhead camera 25 fps interlaced (video4linux)
  - On-board digital camera 3.7 - 60 fps
  - CPU and bandwidth intensive
  - Recognize objects when light condition changes
- Distance sensors
- Compass
- Dead reckoning

Self localization



We are testing two approaches and wish to combine both for better accuracy.

- PSD radar scanning.
  - Only three sensors (left, front and right)
  - Rotating is slow or inaccurate
- Goal position triangulation.
  - Panning is fast - processing is slow
- Someone might always be in the way...

Planning and decision making

- Prediction based on physics model
- Obstacle avoidance
- Path planning when all the obstacles are moving
- Team play
  - Coordination
  - Role change
  - Offensive or defensive play?

Action

Our goal is autonomous players, capable of playing even when overhead tracking and radio communication break down.

- Communication
  - Standard protocol and conventions
  - Different programs can still communicate
- Drive
  - Need fast and accurate steering.
- Kick
  - Ram the ball instead of servo kicker
  - Rotate to hit with tail

Volunteers

Everyone willing are welcome to join the team. We need at least the following.

- Programmers for robot control, tracking system and HW drivers
- HW design and repair
- Public relations

If so, we would like to hear from you at *ciipsglory@uwa.edu.au*.

Sponsors are also very welcome.

Thank you

`http://ciips.ee.uwa.edu.au/~pere/soccer/`