CONTENTS

Chapter 1 Camera & field calibration

Requirements and Installation

1.1 Operating System

SSL-Vision is a Linux application.

1.2 Software Requirements

- git
- g++
- QT >= 4.3 with opengl and networking support
- cmake
- Eigen3
- Google protocol buffers (protoc)
- OpenGL
- GLU
- libdc1394 Version >= 2.0
- libjpeg
- libpng

1.3 Getting and Compiling SSL-Vision

Using git to get the most recent version:

```
git clone https://github.com/RoboCup-SSL/ssl-vision.git
```

After having the code:

```
make
```

1.4 Starting the Applications

After a successful compilation process, you can find three file under the `./bin` directory:

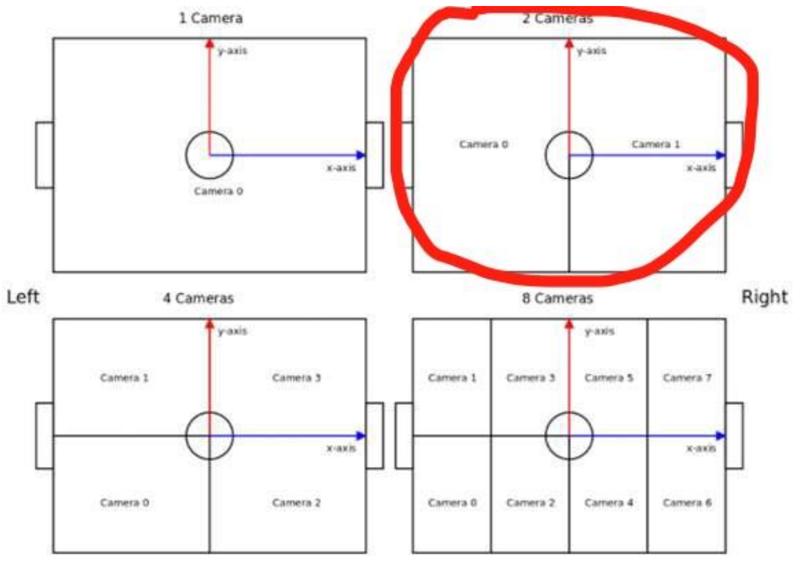
- vision - The SSL-Vision application
- client - A simple sample client
- graphicalClient - A graphical sample client

You must run those application from the project root folder, e.g. bin/vision.

1.5 Field Configuration

1. Coordinate System and Camera Assignment

The cameras and the coordinate system in ssl-vision are set up as shown in the following figure. In our lab we use 2 Cameras.



1.6 Capturing Images

1.Capture Control

Capturing is controlled by the "Start Capture" and "Stop Capture" buttons in the *Camera i/Image Capture/Capture Control* branch.

2.Capture Settings

Capture parameters are in the branch Image *Capture/DC1394/Capture Settings*. Note that internally, SSL-Vision operates in YUV-space. We select the "capture mode" is "yuv422_uyvy".

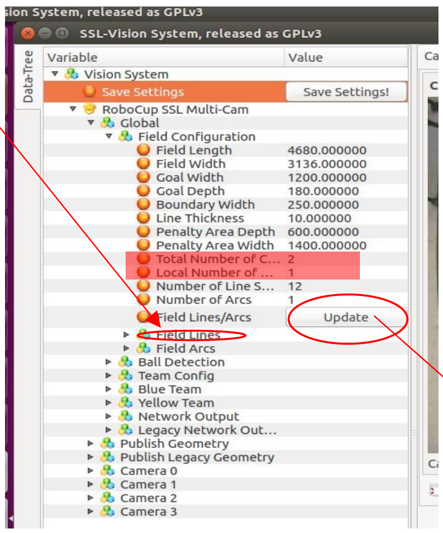
3. Camera Parameters

When you start capturing, you should be able to expand the "Camera Parameters" item, and it should show all the camera's available DCAM parameters. You can adjust all of these parameters in real-time.

1.7 Camera Calibration

1. Update Field Markings

You can adapt the field markings manually by *adding/changing* the Field Length (4680mm), Field Width(3136mm), and Goal Width(1200mm), Goal Depth(180mm),Penalty Area Depth(600) and Penalty Area Width(1400). Those markings are used for line detection. In the laboratory, we use the actual measured value. The value of Total Number of camera is 2 and local number of camera is 1. Field arc/center circle/radius is 375.

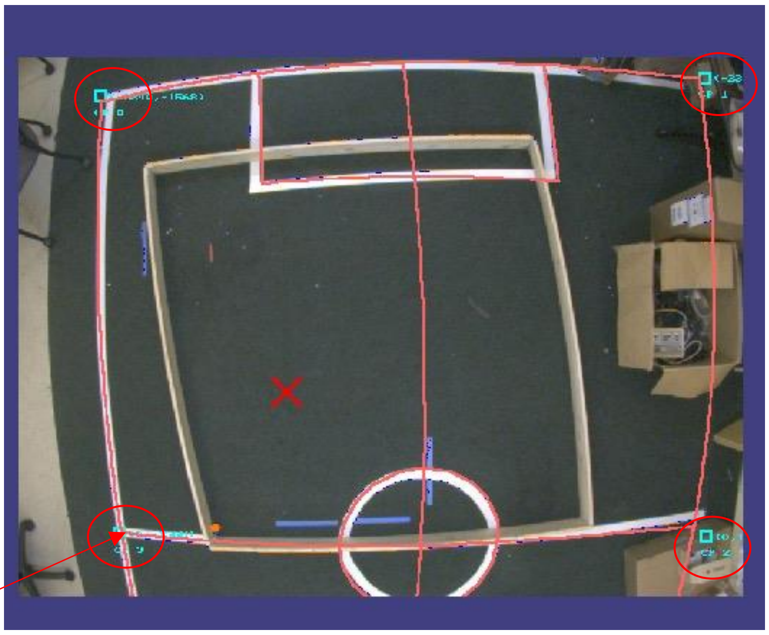


```
sion System, released as GPLv3
SSL-Vision System, released as GPLv3
Data-Tree
Variable                              Value
▼ Vision System
    Save Settings                     Save Settings!
  ▼ RoboCup SSL Multi-Cam
    ▼ Global
      ▼ Field Configuration
          Field Length                4680.000000
          Field Width                 3136.000000
          Goal Width                  1200.000000
          Goal Depth                  180.000000
          Boundary Width              250.000000
          Line Thickness              10.000000
          Penalty Area Depth          600.000000
          Penalty Area Width          1400.000000
          Total Number of C...        2
          Local Number of ...         1
          Number of Line S...         12
          Number of Arcs              1
          Field Lines/Arcs            Update
        ▸ Field Lines
        ▸ Field Arcs
      ▸ Ball Detection
      ▸ Team Config
      ▸ Blue Team
      ▸ Yellow Team
      ▸ Network Output
      ▸ Legacy Network Out...
    ▸ Publish Geometry
    ▸ Publish Legacy Geometry
    ▸ Camera 0
    ▸ Camera 1
    ▸ Camera 2
    ▸ Camera 3
```

When you finished Click Update

2. Update Control Points

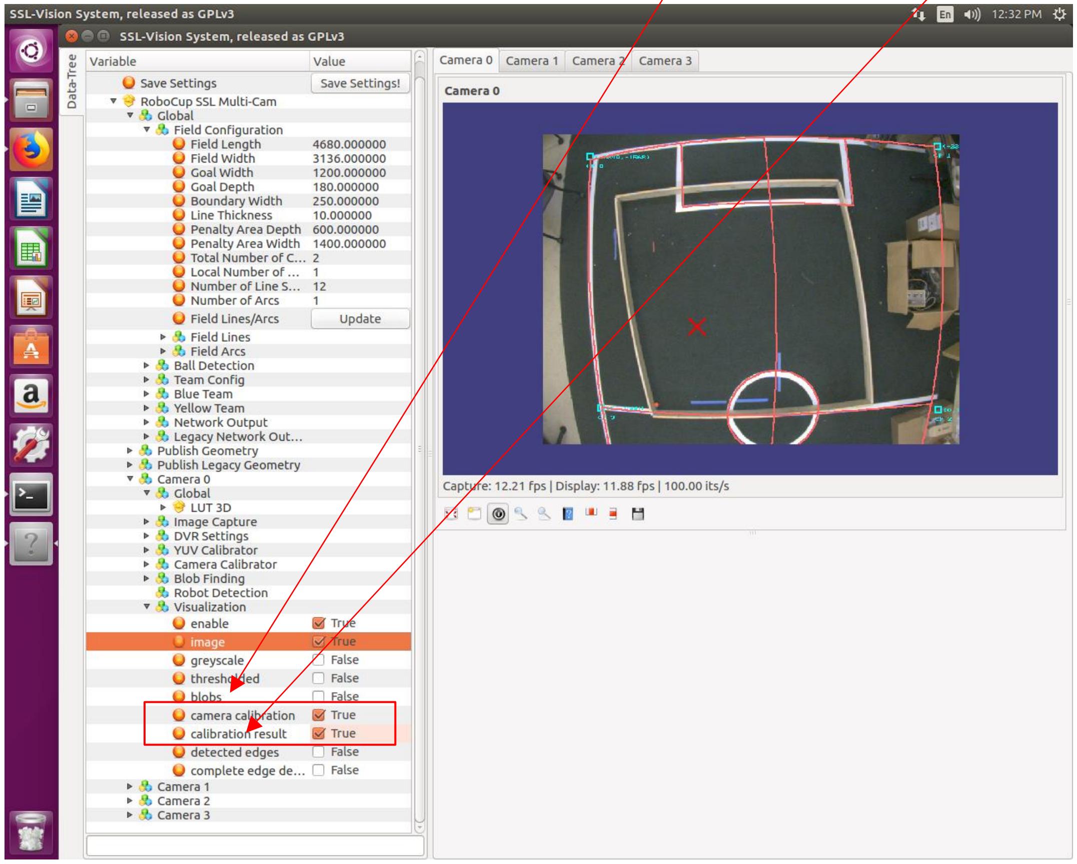
First, choose one of the camera tabs that is backed by a camera. Choose the correct camera id according to the numbering schema, then press "Update control points". This will update the expected field coordinates of the four control points. You can check and update them under "*Camera i / Camera Calibrator / Calibration Parameters / Control point j*", if required. To move the control points around in the image, make sure that the "*camera calibration*" option under visualizations is checked.



Control point

3.Complete Step-by-Step Calibration

1. Enable the camera visualization and check the "camera calibration" and "calibration result" options.



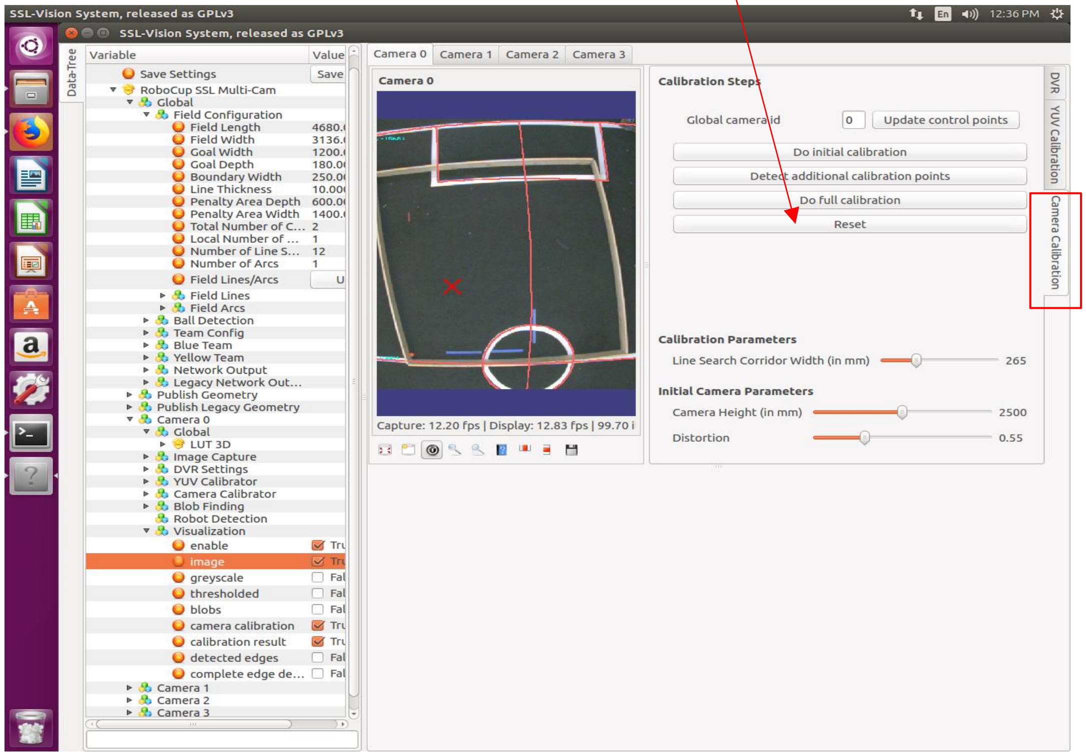
SSL-Vision System, released as GPLv3 — 12:32 PM

```
Variable                          Value
  Save Settings                   Save Settings!
▼ RoboCup SSL Multi-Cam
  ▼ Global
    ▼ Field Configuration
        Field Length              4680.000000
        Field Width               3136.000000
        Goal Width                1200.000000
        Goal Depth                180.000000
        Boundary Width            250.000000
        Line Thickness            10.000000
        Penalty Area Depth        600.000000
        Penalty Area Width        1400.000000
        Total Number of C…        2
        Local Number of …         1
        Number of Line S…         12
        Number of Arcs            1
        Field Lines/Arcs          Update
      ▶ Field Lines
      ▶ Field Arcs
    ▶ Ball Detection
    ▶ Team Config
    ▶ Blue Team
    ▶ Yellow Team
    ▶ Network Output
    ▶ Legacy Network Out…
  ▶ Publish Geometry
  ▶ Publish Legacy Geometry
  ▼ Camera 0
    ▼ Global
      ▶ LUT 3D
    ▶ Image Capture
    ▶ DVR Settings
    ▶ YUV Calibrator
    ▶ Camera Calibrator
    ▶ Blob Finding
      Robot Detection
    ▼ Visualization
        enable                    ☑ True
        image                     ☑ True
        greyscale                 ☐ False
        thresholded               ☐ False
        blobs                     ☐ False
        camera calibration        ☑ True
        calibration result        ☑ True
        detected edges            ☐ False
        complete edge de…         ☐ False
  ▶ Camera 1
  ▶ Camera 2
  ▶ Camera 3
```

Camera 0 | Camera 1 | Camera 2 | Camera 3

Capture: 12.21 fps | Display: 11.88 fps | 100.00 its/s

2. Select the camera calibration tab on the right and click on "Reset".



SSL-Vision System, released as GPLv3 — 12:36 PM

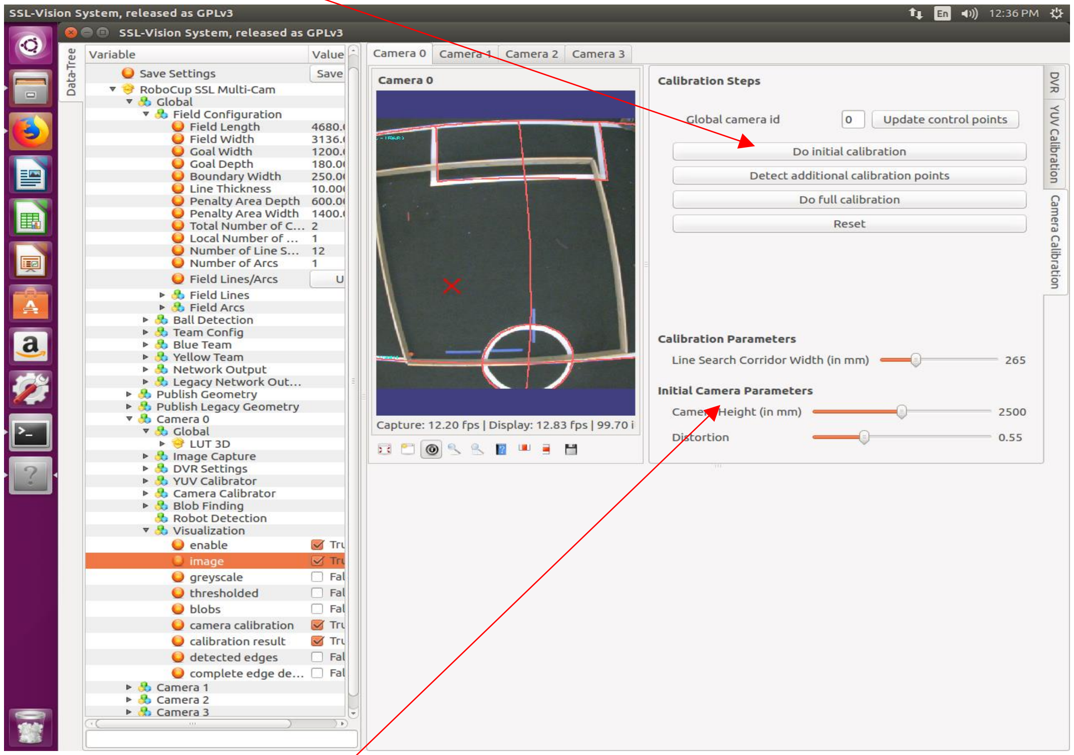
```
Variable                          Value
  Save Settings                   Save
▼ RoboCup SSL Multi-Cam
  ▼ Global
    ▼ Field Configuration
        Field Length              4680.(
        Field Width               3136.(
        Goal Width                1200.(
        Goal Depth                180.0(
        Boundary Width            250.0(
        Line Thickness            10.00(
        Penalty Area Depth        600.0(
        Penalty Area Width        1400.(
        Total Number of C…        2
        Local Number of …         1
        Number of Line S…         12
        Number of Arcs            1
        Field Lines/Arcs          U
      ▶ Field Lines
      ▶ Field Arcs
    ▶ Ball Detection
    ▶ Team Config
    ▶ Blue Team
    ▶ Yellow Team
    ▶ Network Output
    ▶ Legacy Network Out…
  ▶ Publish Geometry
  ▶ Publish Legacy Geometry
  ▼ Camera 0
    ▼ Global
      ▶ LUT 3D
    ▶ Image Capture
    ▶ DVR Settings
    ▶ YUV Calibrator
    ▶ Camera Calibrator
    ▶ Blob Finding
      Robot Detection
    ▼ Visualization
        enable                    ☑ Tru
        image                     ☑ Tru
        greyscale                 ☐ Fal
        thresholded               ☐ Fal
        blobs                     ☐ Fal
        camera calibration        ☑ Tru
        calibration result        ☑ Tru
        detected edges            ☐ Fal
        complete edge de…         ☐ Fal
  ▶ Camera 1
  ▶ Camera 2
  ▶ Camera 3
```

Camera 0 | Camera 1 | Camera 2 | Camera 3

Capture: 12.20 fps | Display: 12.83 fps | 99.70 i

```
Calibration Steps
  Global camera id   0   [Update control points]
  [Do initial calibration]
  [Detect additional calibration points]
  [Do full calibration]
  [Reset]

Calibration Parameters
  Line Search Corridor Width (in mm)   265

Initial Camera Parameters
  Camera Height (in mm)   2500
  Distortion              0.55
```

3. Drag the control points to the locations in the image that correspond to the specified world locations. In the screenshot below, the control points include the center of the field and the intersection of the defense arc and the goal line.
4. Click on "Do Initial Calibration".



SSL-Vision System, released as GPLv3

Variable	Value
Save Settings	Save
RoboCup SSL Multi-Cam	
Global	
Field Configuration	
Field Length	4680.(
Field Width	3136.(
Goal Width	1200.(
Goal Depth	180.0(
Boundary Width	250.0(
Line Thickness	10.00(
Penalty Area Depth	600.0(
Penalty Area Width	1400.(
Total Number of C…	2
Local Number of …	1
Number of Line S…	12
Number of Arcs	1
Field Lines/Arcs	U
Field Lines	
Field Arcs	
Ball Detection	
Team Config	
Blue Team	
Yellow Team	
Network Output	
Legacy Network Out…	
Publish Geometry	
Publish Legacy Geometry	
Camera 0	
Global	
LUT 3D	
Image Capture	
DVR Settings	
YUV Calibrator	
Camera Calibrator	
Blob Finding	
Robot Detection	
Visualization	
enable	✓ Tru
image	✓ Tru
greyscale	Fal
thresholded	Fal
blobs	Fal
camera calibration	✓ Tru
calibration result	✓ Tru
detected edges	Fal
complete edge de…	Fal
Camera 1	
Camera 2	
Camera 3	

Camera 0 | Camera 1 | Camera 2 | Camera 3

Camera 0

Capture: 12.20 fps | Display: 12.83 fps | 99.70 i

Calibration Steps

Global camera id 0 Update control points

Do initial calibration

Detect additional calibration points

Do full calibration

Reset

Calibration Parameters

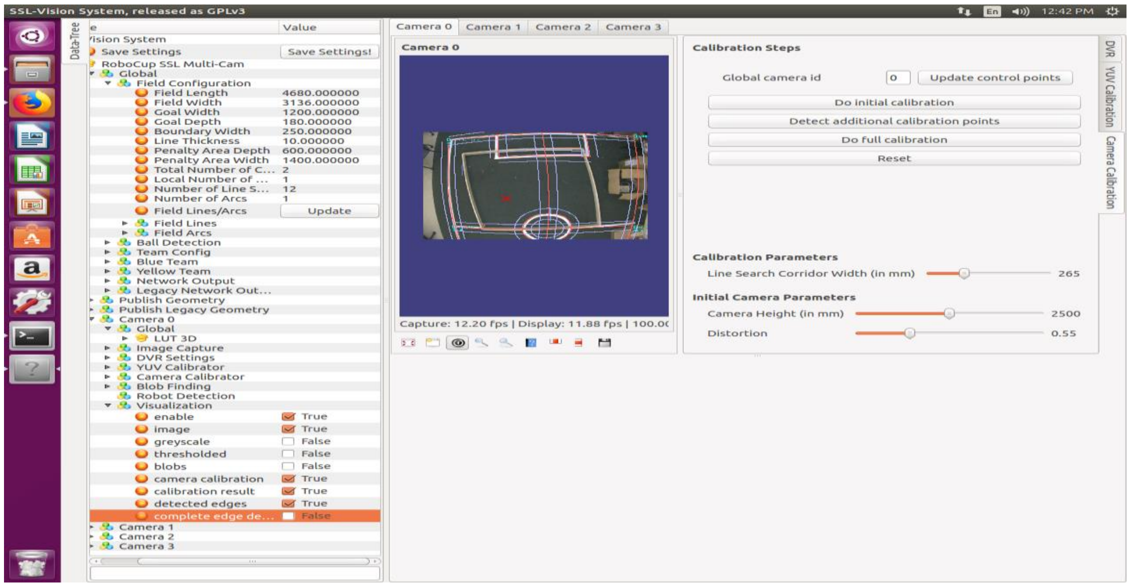
Line Search Corridor Width (in mm) 265

Initial Camera Parameters

Camera Height (in mm) 2500

Distortion 0.55

5. Drag the camera height (2590mm) and camera distortion markers till the calibration result



SSL-Vision System, released as GPLv3

Variable	Value
Vision System	
Save Settings	Save Settings!
RoboCup SSL Multi-Cam	
Global	
Field Configuration	
Field Length	4680.000000
Field Width	3136.000000
Goal Width	1200.000000
Goal Depth	180.000000
Boundary Width	250.000000
Line Thickness	10.000000
Penalty Area Depth	600.000000
Penalty Area Width	1400.000000
Total Number of C…	2
Local Number of …	1
Number of Line S…	12
Number of Arcs	1
Field Lines/Arcs	Update
Field Lines	
Field Arcs	
Ball Detection	
Team Config	
Blue Team	
Yellow Team	
Network Output	
Legacy Network Out…	
Publish Geometry	
Publish Legacy Geometry	
Camera 0	
Global	
LUT 3D	
Image Capture	
DVR Settings	
YUV Calibrator	
Camera Calibrator	
Blob Finding	
Robot Detection	
Visualization	
enable	✓ True
image	✓ True
greyscale	False
thresholded	False
blobs	False
camera calibration	✓ True
calibration result	✓ True
detected edges	✓ True
complete edge de…	False
Camera 1	
Camera 2	
Camera 3	

Camera 0 | Camera 1 | Camera 2 | Camera 3

Camera 0

Capture: 12.20 fps | Display: 11.88 fps | 100.0(

Calibration Steps

Global camera id 0 Update control points

Do initial calibration

Detect additional calibration points

Do full calibration

Reset

Calibration Parameters

Line Search Corridor Width (in mm) 265

Initial Camera Parameters

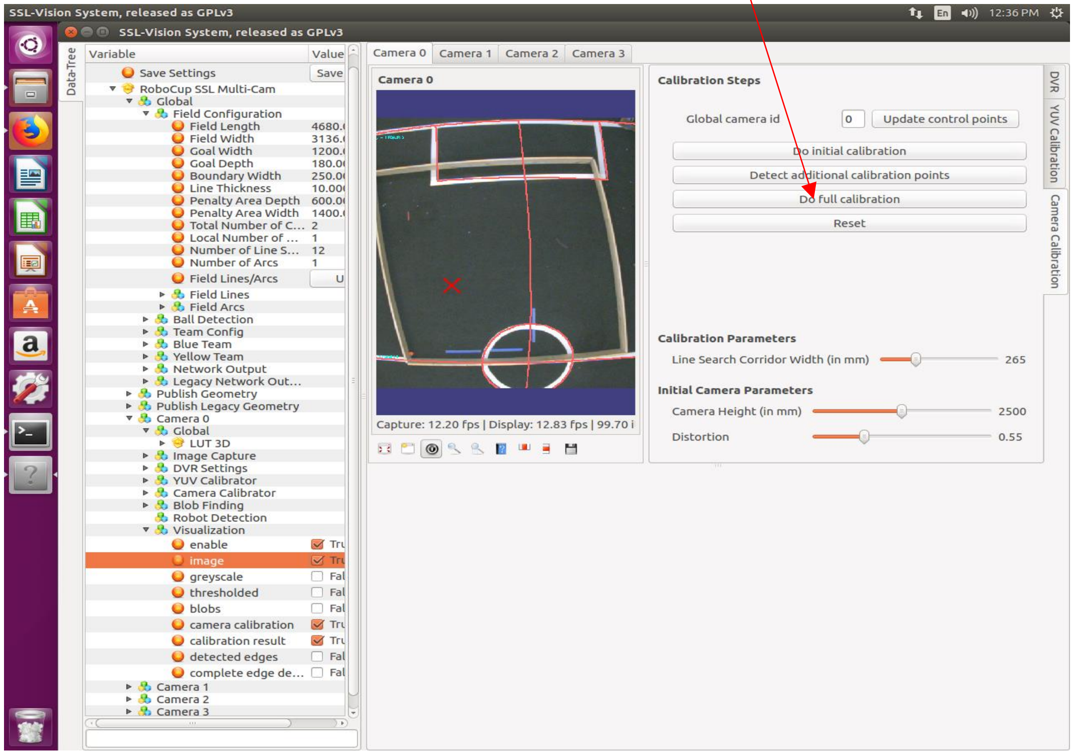
Camera Height (in mm) 2500

Distortion 0.55

roughly approximates the actual field lines. Click on "Do Initial Calibration" again to update the initial calibration guess. Repeat to refine if required.

6. Under the visualizations, turn on "Detected edges" and drag the "Line Search Corridor Width" slider on the calibration tab to increase the search corridors till they include the field markings. Click on "Detect additional points".

7. The detected edges will be displayed. Now click on "Do full calibration" to calibrate using all detected points.



SSL-Vision System, released as GPLv3

Variable	Value
Save Settings	Save
RoboCup SSL Multi-Cam	
Global	
Field Configuration	
Field Length	4680.(
Field Width	3136.(
Goal Width	1200.(
Goal Depth	180.0(
Boundary Width	250.0(
Line Thickness	10.00(
Penalty Area Depth	600.0(
Penalty Area Width	1400.(
Total Number of C…	2
Local Number of …	1
Number of Line S…	12
Number of Arcs	1
Field Lines/Arcs	U
Field Lines	
Field Arcs	
Ball Detection	
Team Config	
Blue Team	
Yellow Team	
Network Output	
Legacy Network Out…	
Publish Geometry	
Publish Legacy Geometry	
Camera 0	
Global	
LUT 3D	
Image Capture	
DVR Settings	
YUV Calibrator	
Camera Calibrator	
Blob Finding	
Robot Detection	
Visualization	
enable	✓ Tru
image	✓ Tru
greyscale	Fal
thresholded	Fal
blobs	Fal
camera calibration	✓ Tru
calibration result	✓ Tru
detected edges	Fal
complete edge de…	Fal
Camera 1	
Camera 2	
Camera 3	

Camera 0 | Camera 1 | Camera 2 | Camera 3

Camera 0

Capture: 12.20 fps | Display: 12.83 fps | 99.70 i

Calibration Steps

Global camera id 0 Update control points

Do initial calibration

Detect additional calibration points

Do full calibration

Reset

Calibration Parameters

Line Search Corridor Width (in mm) 265

Initial Camera Parameters

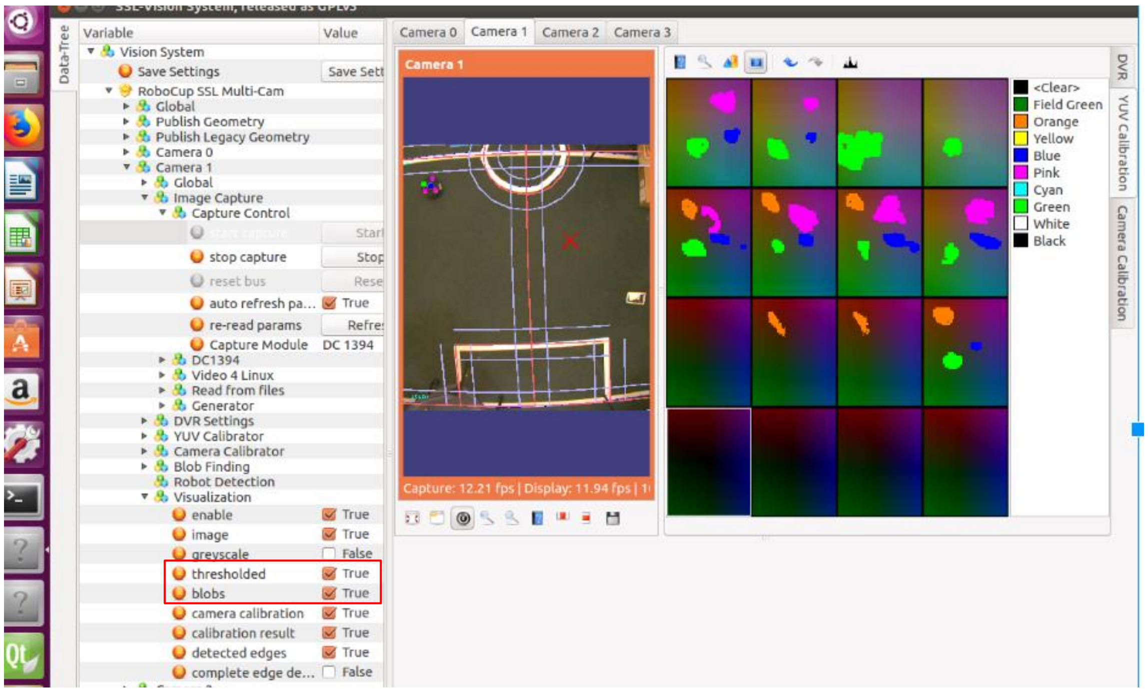
Camera Height (in mm) 2500

Distortion 0.55

8. If necessary, repeat the last 2 steps with a smaller corridor width to refine the calibration result. The final result should match will with the image, as shown below.

1.8 COLOR SEGMENTATION CONFIGURATION

1.Color Thresholding



```
Variable                          Value
▼ Vision System
    Save Settings                 Save Sett
  ▼ RoboCup SSL Multi-Cam
    ▸ Global
    ▸ Publish Geometry
    ▸ Publish Legacy Geometry
    ▸ Camera 0
    ▼ Camera 1
      ▸ Global
      ▼ Image Capture
        ▼ Capture Control
              start capture       Start
              stop capture        Stop
              reset bus           Rese
              auto refresh pa...  ☑ True
              re-read params      Refre
              Capture Module      DC 1394
          ▸ DC1394
          ▸ Video 4 Linux
          ▸ Read from files
          ▸ Generator
      ▸ DVR Settings
      ▸ YUV Calibrator
      ▸ Camera Calibrator
      ▸ Blob Finding
        Robot Detection
      ▼ Visualization
            enable                ☑ True
            image                 ☑ True
            greyscale             ☐ False
            thresholded           ☑ True
            blobs                 ☑ True
            camera calibration    ☑ True
            calibration result    ☑ True
            detected edges        ☑ True
            complete edge de...   ☐ False
```

To check the thresholding results of your LUT, you might want to enable the thresholded visualization of the video stream by checking "thresholded" in the data-tree's visualization options.

To get an idea of where to define LUT color mappings, you should collect color-samples from the video input stream. To do so, make sure visualizations are enabled by setting both enable and image to true in the Camera i/Visualization branch of the data-tree. You should now see the live video stream in the GUI. To collect color samples, simply left click onto the pixel which you would like to sample in the video window. Note, that it is possible to zoom (mousewheel) and pan (right mouse-button) the image (use "space" to reset to normal view). Color samples will be marked by an "x" in the YUV LUT display. Alternatively, to sampling individual pixels, you can also sample all pixels in the image by selecting the video widget and pressing "i", effectively providing you with a YUV histogram. To clear the color sampler, press "c" or use the "Clear Sampler" button above the YUV LUT.

To perform this conversion, SSL-Vision requires that the user defines a mapping from the continuous YUV color-space (http://en.wikipedia.org/wiki/YUV) into the discrete RoboCup color-labels. This mapping is defined in a 3D-Lookup-Table (LUT) which is visualized in the "YUV Calibration"-tab on the right side of the GUI. To visualize the 3D LUT in the 2D interface, the color-cube is essentially "sliced" among the 'Y'-component axis into multiple images, each representing the 'U' and 'V' components of that particular slice. To scroll through the multiple slices, it is possible to use the mouse wheel, or the 'a' and 'z' keys on the keyboard.

To edit the mapping in the LUT, you simply have to select a desired RoboCup color label on the right, (e.g. "Pink" or "Field Green", ...) and then draw directly into the LUT by holding your left mouse button. To speed up the mapping of entire regions, simply draw an outline with the left mouse-button and then flood-fill that particular area with the right mouse-button. The LUT features "Undo/Redo" buttons in case you make a mistake during editing. To erase a mapping, either select the Clear color on the right, or

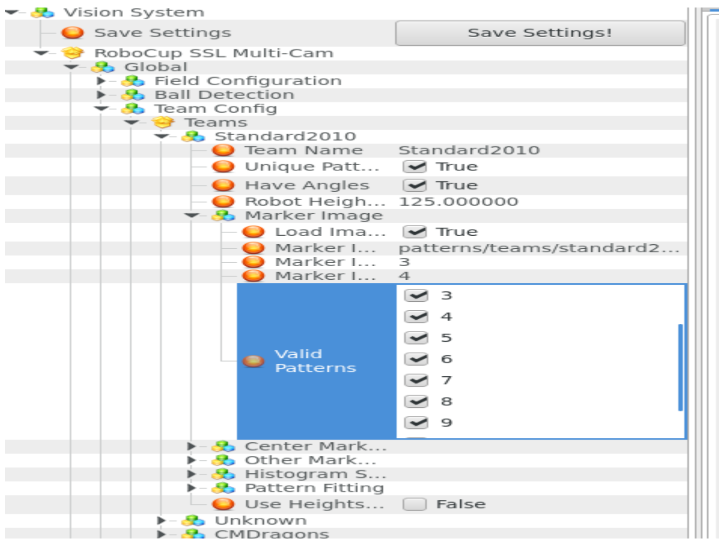
- as a shortcut - simply hold the "Shift"-key while drawing. To draw with a 9-pixel wide stroke, use the "Ctrl" modifier.

The LUT data is automatically saved when the application is quit normally by clicking the close button (not by pressing Ctrl-C in the console!).

2.Blob Detection

The next step in the image processing pipeline is "Blob Finding" which will identify connected components of the thresholded image and compute their bounding boxes and centroids. The only parameter for this process is the "min_blob_area" value in the data-tree (under *Camera i/Blob* Finding) which acts as a global filter, discarding any blobs with fewer pixels than "min_blob_area". To view the blob-finding results in real-time, enable "blobs" in the *Camera i/Visualization* branch.

1.9 Team Configuration and Detection



```
Vision System
  Save Settings                 Save Settings!
  RoboCup SSL Multi-Cam
    Global
      Field Configuration
      Ball Detection
      Team Config
        Teams
          Standard2010
            Team Name           Standard2010
            Unique Patt...      True
            Have Angles         True
            Robot Heigh...      125.000000
            Marker Image
              Load Ima...       True
              Marker I...       patterns/teams/standard2...
              Marker I...       3
              Marker I...       4
              Valid Patterns    3
                                4
                                5
                                6
                                7
                                8
                                9
            Center Mark...
            Other Mark...
            Histogram S...
            Pattern Fitting
            Use Heights...      False
          Unknown
          CMDragons
```

1. Defining the Team's Pattern Image

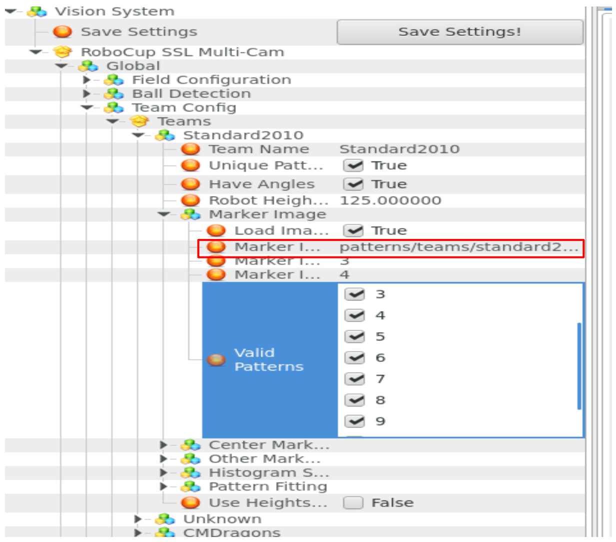
A team's patterns are defined through a .png file in the directory *ssl-vision/patterns*. For official competitions, the patterns are standardized. The standard patterns are defined in *patterns/teams/standard2010.png*. This team image contains a grid of individual robot covers, and it is parsed in row-major order, with increasing robot IDs, starting at 0.

2. Creating the Team in the Data-tree

Open the branch Global/Team Config/ in the Data-Tree. Next, click on Add Team.... This should create a new entry in the Global/Team Config/Teams branch labeled New Team n.

3. Using the Pattern Image

To make use of the png file, make sure to check "Unique Patterns" and “Have Angle” in the team's config branch. Then open the branch "Marker Image", make sure "Load Marker Image" is checked, and enter the path to the .png file as seen from the ssl-vision main directory. For example: *patterns/teams/CMDragons.png.* Next, make sure that the number of robot rows and columns are correct. Finally, uncheck any of the robot IDs which are not being used in your robot definition file, in the "Valid Patterns" field.

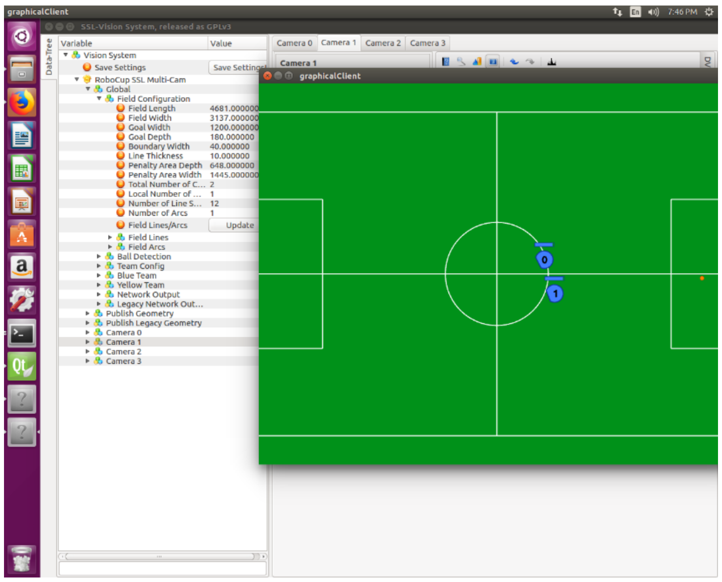


```
Vision System
  Save Settings                Save Settings!
  RoboCup SSL Multi-Cam
    Global
      Field Configuration
      Ball Detection
      Team Config
        Teams
          Standard2010
            Team Name          Standard2010
            Unique Patt...     ✓ True
            Have Angles        ✓ True
            Robot Heigh...     125.000000
            Marker Image
              Load Ima...      ✓ True
              Marker I...      patterns/teams/standard2...
              Marker I...      3
              Marker I...      4
              Valid Patterns   ✓ 3
                               ✓ 4
                               ✓ 5
                               ✓ 6
                               ✓ 7
                               ✓ 8
                               ✓ 9
            Center Mark...
            Other Mark...
            Histogram S...
            Pattern Fitting
            Use Heights...     False
          Unknown
          CMDragons
```

5. Selecting Your Team

Once you have fully configured your team, you can select it under the Global/Blue Team and Global/Yellow Team branches.

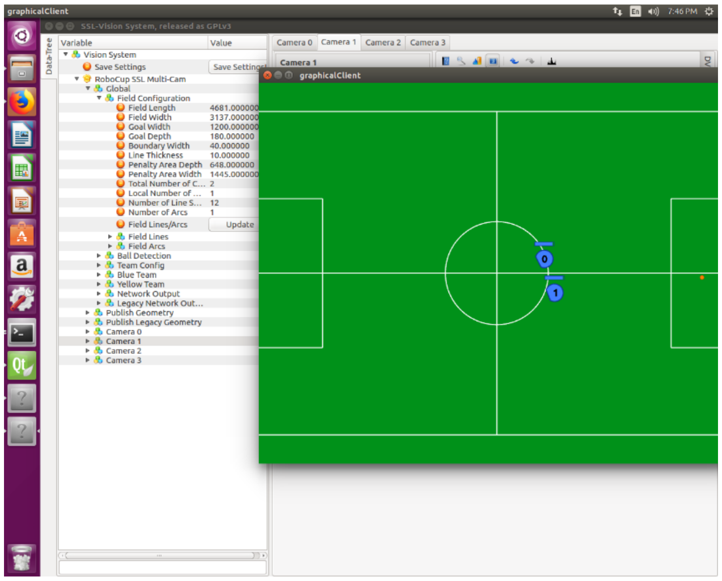
The following image shows a successful setting about the camera calibration, you can see the robot on the graphicalclient.



```
Variable                        Value
Vision System
  Save Settings                 Save Settings
  RoboCup SSL Multi-Cam
    Global
      Field Configuration
        Field Length            4681.000000
        Field Width             3137.000000
        Goal Width              1200.000000
        Goal Depth              180.000000
        Boundary Width          40.000000
        Line Thickness          10.000000
        Penalty Area Depth      648.000000
        Penalty Area Width      1445.000000
        Total Number of C...    2
        Local Number of ...     1
        Number of Line S...     12
        Number of Arcs          1
        Field Lines/Arcs        Update
        Field Lines
        Field Arcs
      Ball Detection
      Team Config
      Blue Team
      Yellow Team
      Network Output
      Legacy Network Out...
    Publish Geometry
    Publish Legacy Geometry
    Camera 0
    Camera 1
    Camera 2
    Camera 3
```

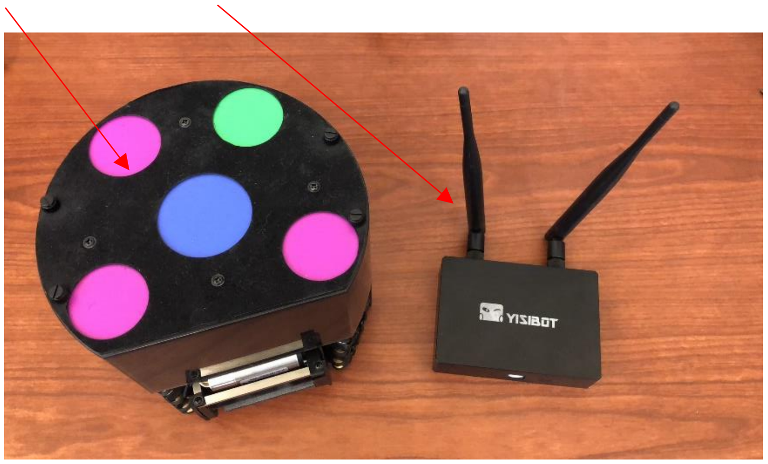
Chapter 2 How to run system on real world

When you have a successful setting about the camera calibration, you can see the robot on the graphicalclient.



Variable	Value
Vision System	
Save Settings	Save Settings
RoboCup SSL Multi-Cam	
Global	
Field Configuration	
Field Length	4681.000000
Field Width	3137.000000
Goal Width	1200.000000
Goal Depth	180.000000
Boundary Width	40.000000
Line Thickness	10.000000
Penalty Area Depth	648.000000
Penalty Area Width	1445.000000
Total Number of C...	2
Local Number of ...	1
Number of Line S...	12
Number of Arcs	1
Field Lines/Arcs	Update
Field Lines	
Field Arcs	
Ball Detection	
Team Config	
Blue Team	
Yellow Team	
Network Output	
Legacy Network Out...	
Publish Geometry	
Publish Legacy Geometry	
Camera 0	
Camera 1	
Camera 2	
Camera 3	

We need the robot and transmitter.



2.1 How to Install Robobulls2 (USF team Robobulls Robocup Software)

The required libraries and tools to run the game can find on the GitHub (https://github.com/mllofriu/robobulls2)

2.2 Getting Started--Running The Project

Open the RoboBulls.pro project in Qt Creator(4.5.0) and compile. There are some issues to notice here, before you actually build the project, make sure you updated your QT creator and QT to the latest version. The last test is based on QT creator 4.5.0 and QT 5.10.0, make sure you set the QT version in your QT creator by going into Tools->Options->Build & Run->Kits. After running the installation script, there should be no compilation issues (Tested on Ubuntu 14.04 only)

2.3List of important modified files

SIMULATED.H (INCLUDE/CONFIG/SIMULATED.H)

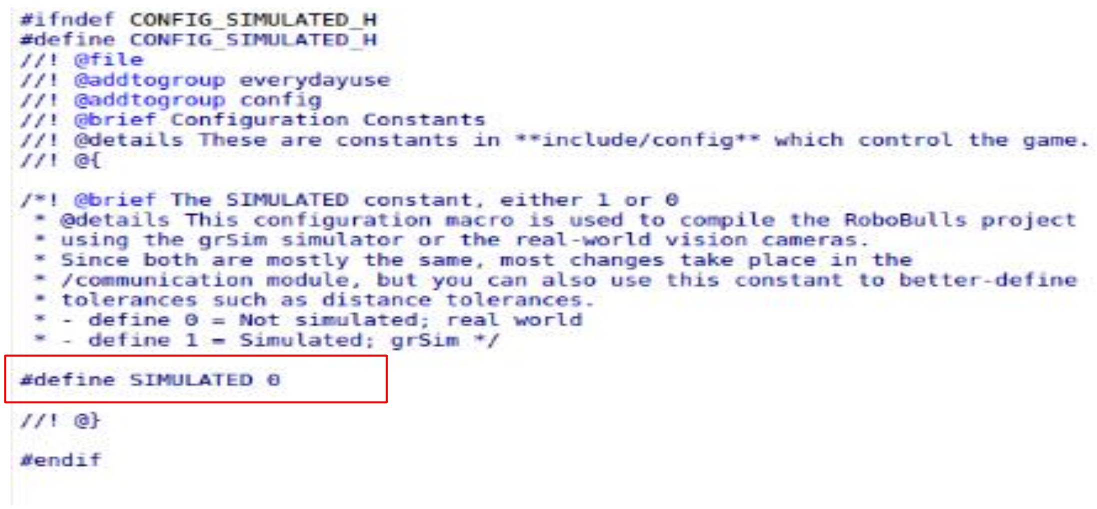
yisirobcomm.cpp (communication/yisirobocomm.cpp)

TEAM.H (INCLUDE/CONFIG/TEAM.H)

COMMUNICATION.H (INCLUDE/CONFIG/COMMUNICATION.H)

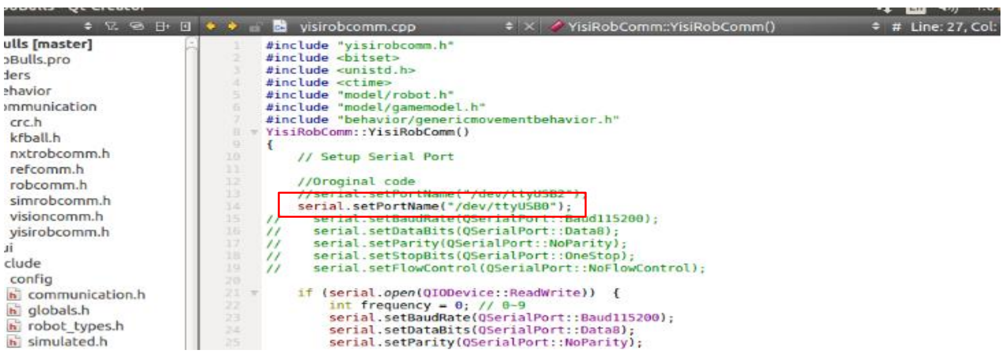
2.4 Running the project on real world

First, for running the yisirobot in real world, simply change the **SIMULATED** into 0(include*/ config / simulated.h*). Also make sure you connected the transmitter to your computer, the two lights flashing means that they are not successfully configured, they will turn into always lighting if they get configured.

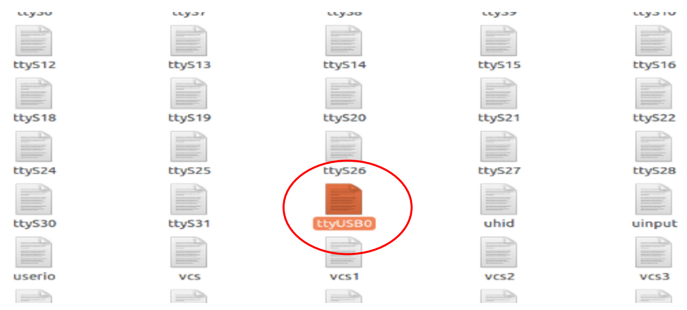


```
#ifndef CONFIG_SIMULATED_H
#define CONFIG_SIMULATED_H
//! @file
//! @addtogroup everydayuse
//! @addtogroup config
//! @brief Configuration Constants
//! @details These are constants in **include/config** which control the game.
//! @{

/*! @brief The SIMULATED constant, either 1 or 0
 * @details This configuration macro is used to compile the RoboBulls project
 * using the grSim simulator or the real-world vision cameras.
 * Since both are mostly the same, most changes take place in the
 * /communication module, but you can also use this constant to better-define
 * tolerances such as distance tolerances.
 *  - define 0 = Not simulated; real world
 *  - define 1 = Simulated; grSim */

#define SIMULATED 0

//! @}

#endif
```

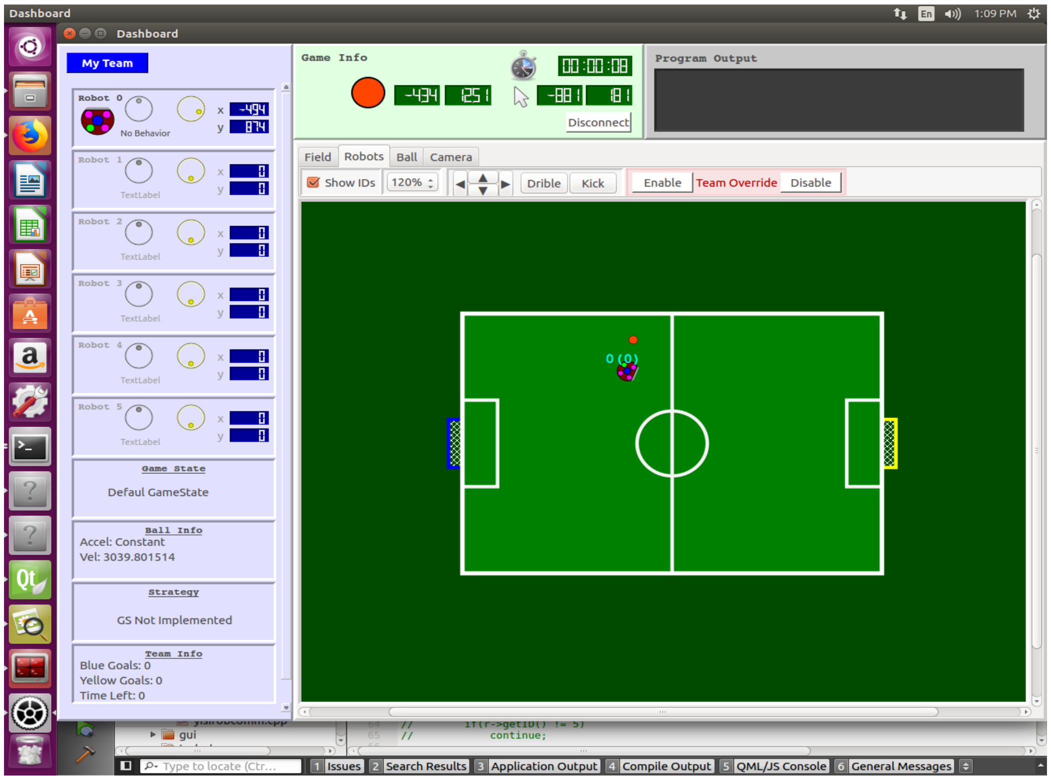
Second, connect the transmitter to the USB, and we should modified the yisirobcomm.cpp (communication/yisirobocomm.cpp). For connecting the transmitter, we should setup serial port and the serial.setPortName() is same as the dev file. You can find the file under the computer/dev/



```
#include "yisirobcomm.h"
#include <bitset>
#include <unistd.h>
#include <ctime>
#include "model/robot.h"
#include "model/gamemodel.h"
#include "behavior/genericmovementbehavior.h"
YisiRobComm::YisiRobComm()
{
    // Setup Serial Port

    //Oroginal code
    //serial.setPortName("/dev/ttyUSB2");
    serial.setPortName("/dev/ttyUSB0");
//    serial.setBaudRate(QSerialPort::Baud115200);
//    serial.setDataBits(QSerialPort::Data8);
//    serial.setParity(QSerialPort::NoParity);
//    serial.setStopBits(QSerialPort::OneStop);
//    serial.setFlowControl(QSerialPort::NoFlowControl);

    if (serial.open(QIODevice::ReadWrite))  {
        int frequency = 0; // 0-9
        serial.setBaudRate(QSerialPort::Baud115200);
        serial.setDataBits(QSerialPort::Data8);
        serial.setParity(QSerialPort::NoParity);
```



After You finished the setup, click run button. You can see the Dashboard like the following image.



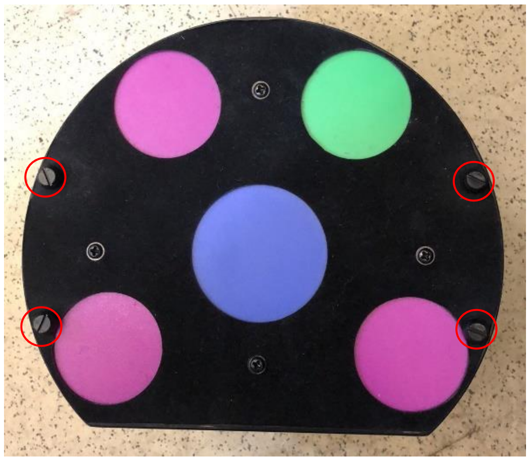
When the transmitter is not configured, the two LEDs will blink alternately. When config succeed, the two LEDs will always light. And when you are sending the transmitpackets, one of the LEDs will blink one time after 15 packets has been sent.

Note: If you want to re-config the frequency, you must unplug the transmitter and plug it to computer again.

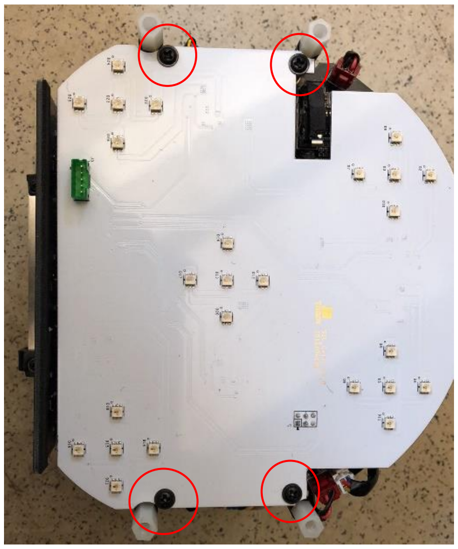
2.5 SET WORKING MODE

Before turning on the power, we should set working mode:

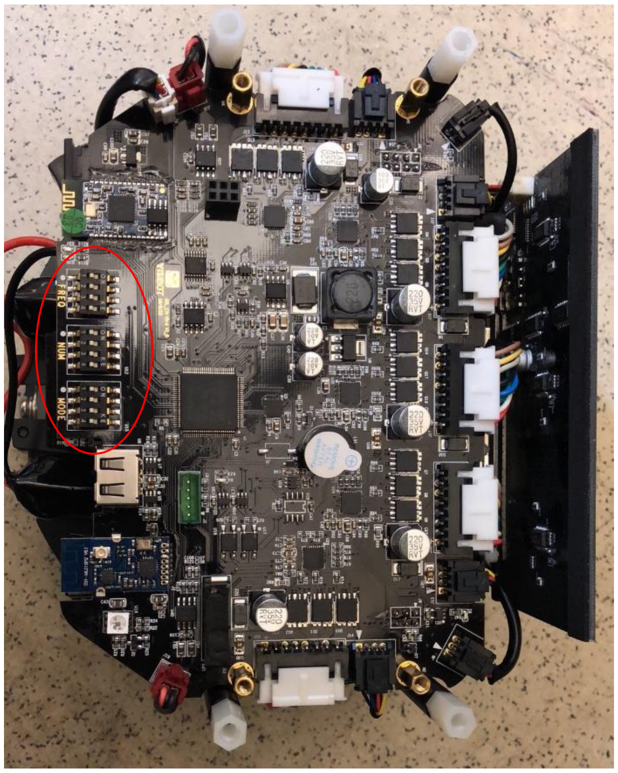
1. remove the robot's shell.



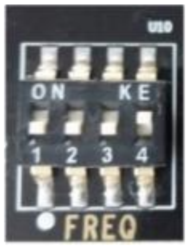
2. remove the top broad.



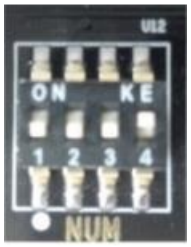
3.We can see the broad for setting working mode.



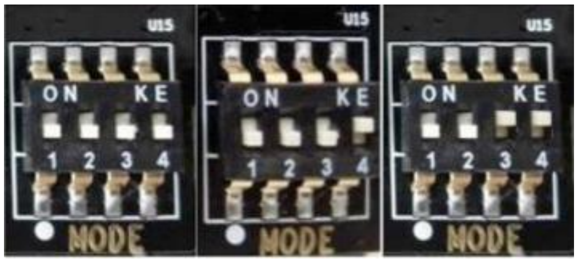
Set frequency point (frequency encoder(FERO)): the robot with the same team should be set as the same frequency. Dial up is 1, On the contrary is 0, So 1000 (8), 0100 (4), 0010 (2), 0001 (1)



Set robot number (robot number encoder(NUM)): The setting of robot number is related to binary. Example the robot ID is 1:

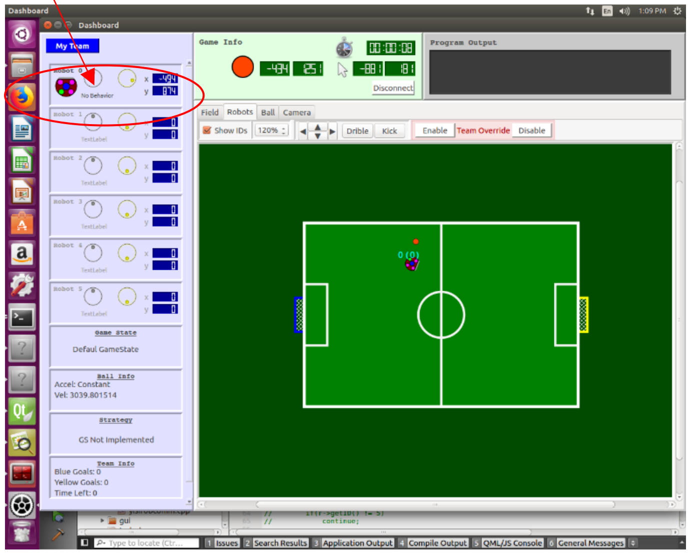


Set operation mode (mode encoder (MODE)): The right is the self-checking mode. The middle is the handle mode. The left is the game mode. We should change it to the game mode.



When your config succeeds, you can control the robot on the keyboard.

Click the robot.



References

1. RoboCup Small Size League: SSL Web Site. http://small-size.informatik.uni- bremen.de (2009)