31388 Advanced Autonomous Robots exercise 5: Perception - Vision

(rev 6.52)

Introduction

In this exercise you should make a vision based perception system that is able to determine the position of an object (a coloured ball) in robot coordinates. The perception function shall be implemented in a camera server plug-in, that enhances the server functionality. This new functionality should be used later in the course to navigate actively relative to the ball position.

Estimated to take 2 exercise days (about 8 hours).

Aim

When you have finished this exercise you will be able to:

- Utilize camera images to obtain robot navigation data.
- Analyse camera images and extract perception data using openCV.
- Implement the image operations on the robot for real time use.

Evaluation

Some of the results from each part of the exercise must be presented on the homepage, especially:

- Samples of objects seen from robot camera (images and measured scene layout).
- Selected method and calculations used to extract object from image.
- Some results from the C++ plug-in implementation and the source code.

Exercise

The exercise has four parts:

- Camera parameters, to use camera as position measuring device
- Image analysis, to find objects in image (by colour separation)
- Robot software – make a mobotware plug-in and enhance its functionality
- Extracting navigation data from images

Read

Look at http://en.wikipedia.org/wiki/HSV to get a feel for the HSV colour format.

Camera parameters

To be able to use the camera to determine position of objects relative to the robot, the minimum set of parameters are the camera focal length and the camera position on the robot.

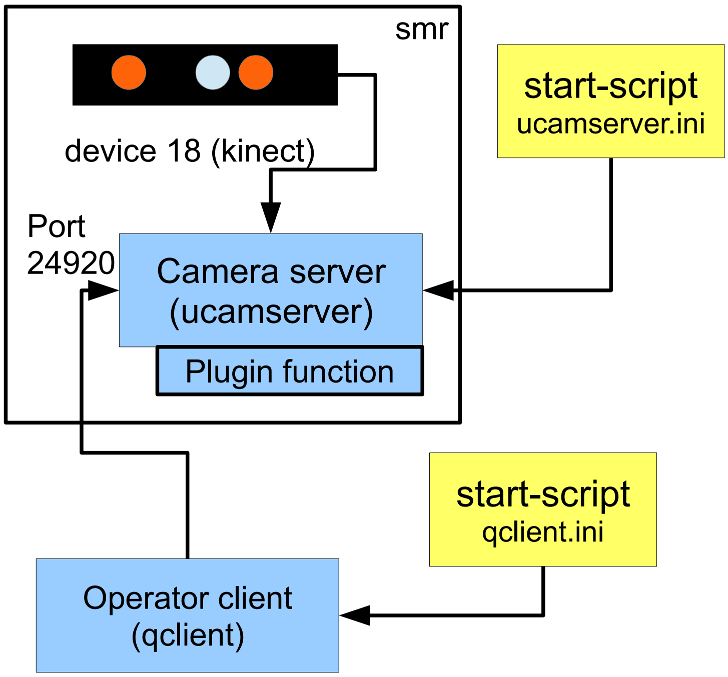

Figure 1:Camera server configuration

Focal length

The focal length, is the distance from the pinhole to the image plane in a pinhole model of the camera, this is the most important camera parameter (in this exercise this distance is called *f* (for focal length), see figure 4.22 in the textbook page 129 for reference).

The focal length is calculated using formula (4.25) in the book page 130, but reduced to a single camera:

$$\frac{x_i}{f} = \frac{x}{z}$$

Where *x* is the width of the object in meters, *z* the distance from the object to the camera, x_i is the width of the object in image pixels and *f* is the focal length (in pixels).

Determine focal length for Kinect color camera

If you do not have a robot available you can use the picture **focuslength.png** from /shome/31388/examples directory. The image is fetched by using the poolset command

poolset img=70 loadpng="/shome/31388/examples/focuslength.png"

the size of an A4-paper is 21.0 x 29.7 cm.

The distance to the paper is 1m.

Use an object of known size and position and measure its size in the image (we here assume that the lens distortion is minimal).

Start the camera server on your SMR as usual (as in figure 1):

```
$ ssh smrX      (log-on to your SMR)
$ cd live
$ ucamserver
```

To see an image from the camera start "qclient":

```
$ cd live
$ qclient
```

```
$ cam connect=smrx:24920
```

Carefull with image update rate to client, as a combination of load on the wireless connection and a time-out in the client may crash the client.

Get image

Open camera

>> cam kinect open

Request images (of full size) by

```
>> cam poolget img=18 fmt=rgb all
```

Place a flat object with good contrast and a known size (vertically, horizontally or both) facing the camera at an exactly known distance of about 1m, measured to within 1 cm, from the camera (lens):

Width of object: x= 0.38 m distance from camera: z= 1 m

When you have the right image, save it:

```
>> cam poolget img=18 savePNG=”focuslength.png”
```

Determine focal length

Now analyse the image to get the pixel size of the object – use e.g. the “`gimp`” (zoom and use measure tool).

Width of object in image: x_i= 204 pixels

Found focal length: f= 537 pixels

Modify the ucamserver.ini file to see the commands executed when the camera server is started, and add the command to set the found camera focal length (in pixels), like:

`camset device=18 focallength=798` (or whatever focal length you found)

Focal length is inserted assuming an image resolution is 640x480 pixels, if your image was 320x240, then you must multiply the found focal length by 2 before inserting. When retrieving focal length (>>cam camget device=XX focallength), this length is adjusted to current image size.

- Save focal length calculation and the test image on the group web-page.

Camera position on the robot

This part can only be done if a robot is available.

The camera position on the robot is needed to be able to convert from camera coordinates to robot coordinates.

The camera position in robot coordinates:

(x_c, y_c, z_c) is the position (x=forward, y=left and z=up) of the camera (at the centre, where the lens connects to the camera house). Where $(0,0,0)$ is the centre between rear wheels at floor level.

Omega, Phi, Kappa (Ω_c, Φ_c, K_c) are the camera rotations around the x, y and z axis. Assuming that the camera is looking forward and is upright, then the only relevant rotation is Φ_c around the y-axis (horizontal forward upright is $(0,0,0)$, and Φ_c is positive when the camera is looking down).

Place the robot on the floor (on the wheels) and take an image of a flat object on the floor (e.g. a tape line) at the centre-line of the camera (centre of image). Measure the x distance – in robot coordinates – to where the centre-line of the camera (centre of image) meets the floor, and combine with the camera position, to get the Φ_c .

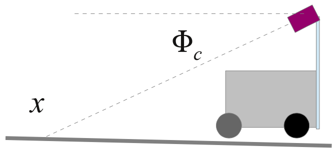

Figure 2: Camera angle

Φ_c = ____0.244____ radians from horizontal.

Insert this position in the camera server ini-file, like this:

```
camset device=18 posX=x.xx posY=y.yy posZ=z.zz rotPhi=tilt
```