WORKSHEET

U11.2 Intrinsics and calibration

BugBotLab

Vision · University · about 35 min

Name ________ Class ________ Date ________

What this lesson is about

Focal length in pixels, the principal point, and measuring them from known geometry.

- The intrinsic matrix
- Focal length in pixels, not millimetres
- What calibration actually measures
- Distortion
- Calibrating from what you have
- Why one point is not enough

Questions

7 marks in all

1. A 3.6 mm lens sits on a sensor 4.8 mm wide that produces images 640 pixels wide. What is the focal length in pixels? [1 mark]

2. The 320 pixel wide BugBot image, with fx = 92.4, is downsampled to 160 pixels wide. What is fx for the small image, in pixels? [1 mark]

3. What does a camera calibration actually compute? [1 mark]

- ☐ **A** A least squares fit of the intrinsics, distortion and target poses that minimises the reprojection error
- ☐ **B** The focal length printed on the lens, converted to pixels
- ☐ **C** The average colour of a chessboard under the room's lighting
- ☐ **D** The exact centre of the image in pixels

4. Every calibration image shows the chessboard square on to the camera. What goes wrong? [1 mark]

- ☐ **A** The focal length and the target's distance trade off against each other, so the fit is degenerate
- ☐ **B** The corners cannot be found to sub-pixel accuracy
- ☐ **C** The principal point comes out at zero
- ☐ **D** Nothing, square on is the most accurate view

5. Four tags give tan(bearing) and the observed u - 160. The model u - 160 = f tan(bearing) is a line through the origin, fitted by least squares. What does this print? [1 mark]

```
tans = [0.0, 30 / 70, -40 / 50, 40 / 30]
offsets = [0.0, 40.0, -74.0, 123.0]
f = sum(t * o for t, o in zip(tans, offsets)) / sum(t * t for t in tans)
residuals = [round(o - f * t, 1) for t, o in zip(tans, offsets)]
print(round(f, 1))
print(residuals)
```

6. Why does a tag straight ahead of the robot carry no information about f? [1 mark]

- ☐ **A** Its tan(bearing) is zero, so u - cx is zero whatever f is
- ☐ **B** A tag straight ahead is too close to measure
- ☐ **C** It lies on the principal point, which is unknown
- ☐ **D** Its range is too large for a good fit

7. A calibration reports a root mean square reprojection error of 1.4 pixels. What should you conclude? [1 mark]

- ☐ **A** Something is wrong, such as blurred images, a target that was not flat, or too few angles
- ☐ **B** It is a good calibration
- ☐ **C** The camera needs a longer focal length
- ☐ **D** The residual is irrelevant once the intrinsics are known

The task: measure the focal length

Four tags are on the mat at (100, 130), (130, 110), (60, 90) and (140, 70). The robot stands at (100, 40) facing along +y. Print `focal length:`, fitted in pixels, and `field of view:`, the full horizontal field of view in degrees that your focal length implies.

```
from bugbot import *
import math
connect()

ROBOT = (100.0, 40.0)
TAGS = {1: (100.0, 130.0), 2: (130.0, 110.0), 3: (60.0, 90.0), 4: (140.0, 70.0)}
set_cv("apriltag")
wait(0.3)
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/u11-2-intrinsics/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Print the residual of your fit: for each tag, the difference between the column you observed and the column your fitted `f` predicts. Is it under a pixel?
2. Leave out tag 1, whose bearing is zero. Does the fit change? Explain why that point carried no information about `f`.

3. Fit the principal point as well, by allowing an intercept in the straight line. What do you get, and is the difference from 160 meaningful given your residuals?