WORKSHEET

U11.1 The pinhole camera

BugBotLab

Vision · University · about 30 min

Name _____ Class _____ Date _____

What this lesson is about

One pixel is one ray. What that buys you, and what it does not.

- The model
- What a pixel is worth
- The tangent matters
- A bearing is not a position
- Getting the depth back

Questions

7 marks in all

1. The BugBot's camera is 320 pixels wide with a 120 degree field of view. What does this print? [1 mark]

```
import math
f = (320 / 2) / math.tan(math.radians(120 / 2))
bearing = math.degrees(math.atan((240 - 160) / f))
print(round(f, 1), round(bearing, 1))
```

2. Which line works out the focal length in pixels correctly? [1 mark]

- ☐ **A** f = 160 / math.tan(math.radians(60))
- ☐ **B** f = 160 / math.tan(60)
- ☐ **C** f = 160 * math.tan(math.radians(60))
- ☐ **D** f = 320 / math.tan(math.radians(120))

3. A blob's centre is at column 200. What is its bearing from straight ahead, in degrees to 1 decimal place, using f = 92.4? [1 mark]

4. The linear guess bearing = (u - 160) x 0.375 degrees is used instead of the pinhole formula. How many degrees wrong is it at column 200? Give 1 decimal place. [1 mark]

5. Why can a single pixel not tell you where an object is? [1 mark]

- ☐ **A** Depth appears only as a divisor, so every point along a ray lands on the same pixel
- ☐ **B** The image is too low resolution to locate the object
- ☐ **C** The principal point is not exactly at the centre
- ☐ **D** Lens distortion moves the pixel

6. Which of these add the second constraint needed to get depth from a camera? [1 mark]

Tick every answer that is true.

- ☐ **A** Knowing the object's real size
- ☐ **B** Knowing the object is on a flat floor, with the camera's height and tilt
- ☐ **C** A second camera a known distance away
- ☐ **D** The same camera at two times, with the motion between them known
- ☐ **E** Averaging the bearing over many frames
- ☐ **F** A higher resolution sensor

7. A tag is seen at column 220 and the detector reports it 50 cm away. What does this print? [1 mark]

```
import math
F = 92.4
bearing = math.degrees(math.atan((220 - 160) / F))
print(round(bearing, 1), round(50 * math.sin(math.radians(bearing)), 1), round(50 *
math.cos(math.radians(bearing)), 1))
```

The task: turn a pixel into a bearing

Tag 7 is on the mat and the robot is standing still, facing along +y. Print `bearing:`, the angle from straight ahead to the tag in degrees, and `across:`, how far to the right of the robot's nose line the tag actually is in centimetres.

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

F = 92.4
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/u11-1-the-pinhole-camera/

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

Challenges

1. Work out how far wrong the linear approximation would be for this tag, in centimetres of sideways error.
2. One pixel of column error is how many centimetres of sideways error at this range? At double the range?
3. The detector reports `cy` as well. Given the camera sits 5 cm above the mat, what would the row of a thing standing on the floor tell you, and what would it not? (The simulated tag detector always reports the middle row, 120, so work this one on paper, or try it with a ball and the blob detector, whose `cy` does move.)