



8.1 See as numbers

What this lesson is about

What a model gets: the sensor views as lists of numbers.

- The depth grid is 64 numbers
- The level rows
- Same wall, different numbers
- A picture is numbers too

Questions

7 marks in all

1. How many numbers are in the depth grid from `tof_grid()` ? [1 mark]

Answer: 64. Eight rows of eight: 64 distances in centimetres.

2. What does this program print? [1 mark]

```
grid = list(range(64))
level = grid[16:32]
print(len(level), level[0], level[-1])
```

Answer:

16 16 31

`grid[16:32]` takes items 16 up to but not including 32: sixteen items, the third and fourth rows of eight.

3. Which rows of the depth grid look straight ahead, level with the robot? [1 mark]

- ☐ A Rows 2 and 3
- ☐ B Rows 0 and 1
- ☐ C Rows 6 and 7
- ☐ D Rows 4 and 5

Answer: A. Rows 2 and 3 are the level rows, items 16 to 31. The bottom rows look down and hit the mat first.

4. One cell of the depth grid reads 400. What does that mean? [1 mark]

- ☐ A Nothing is in range in that direction
- ☐ B A wall exactly 4 m away
- ☐ C The sensor is broken
- ☐ D Something is touching the robot

Answer: A. 400 is the sensor's way of saying it saw nothing. You see it in open space.

5. The camera image is 320 by 240 pixels, and each pixel is three numbers. How many numbers is one frame? [1 mark]

Answer: 230400. $320 \times 240 = 76,800$ pixels, times 3 for red, green and blue.

6. A model has only seen the depth grid with the robot facing a wall straight on. The robot turns a little. [1 mark]
What happens to the numbers?
- ☐ A Many of them change, and the model has never seen a view like the new one
 - ☐ B None change, because the wall has not moved
 - ☐ C Only the middle column changes
 - ☐ D The model adjusts them automatically

Answer: A. The numbers depend on exactly where the robot is and which way it faces, so collect data from everywhere the robot will be.

7. What is the name for the small set of numbers you choose to hand to a model, which still tell the situations apart? [1 mark]

Answer: features. Choosing good features is most of the work. In this module the features are the 16 level readings.

The task: see as numbers

Print the level rows' nearest and farthest readings as `depth: min <cm>, max <cm>`. Do not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# 64 distances, 8 rows of 8
grid = tof_grid()
print(grid)
```

The hint students can ask for: Take the level rows of the depth grid (rows 2 and 3, 16 numbers) and print `depth: min <cm>, max <cm>`. Do not drive.

A solution

```
from bugbot import *
connect()
grid = tof_grid()
level = grid[16:32]
for row in range(8):
    print(' '.join('%3d' % d for d in grid[row * 8: row * 8 + 8]))
print(f'depth: min {min(level)}, max {max(level)}')
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.