

Problem 1



(a) Perspective projection: Parallel lines are not parallel in the image
(b) Orthographic projection: Parallel lines stay parallel in the image

The image to the left is taken normally, while the image to the right is taken from a distance with zoom. On the square boxes one can clearly see the difference. In the perspective projection the edges are converging to the vanishing points, while in the orthographic projection the edges are (approximately) parallel.

Problem 2

Since a parallel projection simply translates the image from the world coordinate system to the virtual cameraplane, we can find the equations by rotating the world coordinate frame such that it is identically oriented as the camera system. In our case, this is a simple rotation around the X-axis by $-\theta$, with positive direction defined as normal for right-handed systems.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & \sin(-\theta) \\ 0 & -\sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

This gives $x = X$ and $y = \cos(\theta)Y - \sin(\theta)Z$. When changing to square pixels, the coordinates will be scaled by some factor α . And since the origin in the virtual cameraplane plane does not need to be aligned with the world origin, we must also add offsets x_0 and y_0 resulting in

$$\begin{aligned} x &= \alpha X + x_0 \\ y &= \alpha(\cos(\theta)Y - \sin(\theta)Z) + y_0 \end{aligned} \quad (2)$$

Problem 3

By our assumption, all vertical edges in the image are vertical in the world. Since Z is constant along vertical edges we get

$$\frac{\partial Z}{\partial y} = 0 \quad (3)$$

From eq. (2) we can find the constraint on horizontal edges. Differentiating along the edge given by t as in the notes gives

$$\begin{aligned} n_x &= -\alpha \sin(\theta) \frac{\partial Z}{\partial t} \\ \frac{\partial Z}{\partial t} &= \frac{-n_x}{\alpha \sin(\theta)} \end{aligned} \quad (4)$$

As for Y , the assumption of planar surfaces means the second derivatives must be zero

$$\begin{aligned} \frac{\partial^2 Z}{\partial x^2} &= 0 \\ \frac{\partial^2 Z}{\partial y^2} &= 0 \\ \frac{\partial^2 Z}{\partial x \partial y} &= 0 \end{aligned} \quad (5)$$

Problem 4

```
# Line 51
Aij [:,:,c] =
    - ny*np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])/8
    + nx*np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])/8;

# Line 65
Aij [:,:,c] =
    0.1*np.array([[0, -1, 0], [0, 2, 0], [0, -1, 0]]);
```

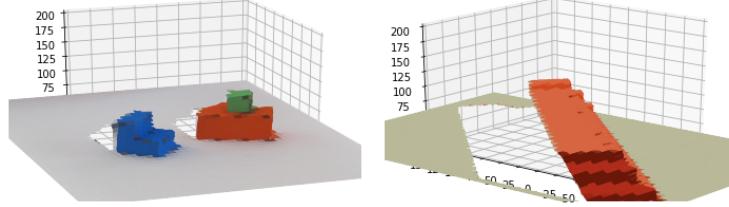
The first kernel corresponds to $-n_y \partial Y / \partial x + n_x \partial Y / \partial y$, while the second corresponds to $\partial^2 Y / \partial y^2$.

Problem 5

Two of the rotated images can be seen in fig. 2.

Problem 6

In fig. 3a you can see how the recovery of image 4 fails. The orange and blue boxes appears to be surfaces in 3D space, and the green cube appears flat in the XZ -plane. The distortion of the blue and orange boxes comes from the models inability to infer the unseen parts of objects, as this would require semantic



(a) Image 2 as seen from a new perspective (b) Image of the steps, as seen from a new perspective

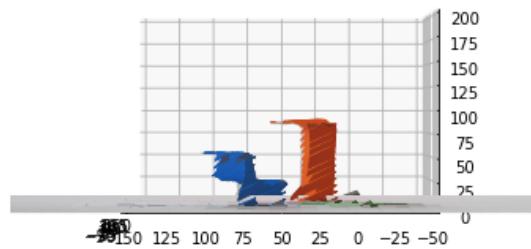
Figure 2: Rotated images rendered back to 3D

knowledge about the world and the object in question. From fig. 3b it seems the model is not detecting all the vertical edges on the green box properly. This causes the top of the green box to be interpreted as ground level as seen in the Y gradient in fig. 3c

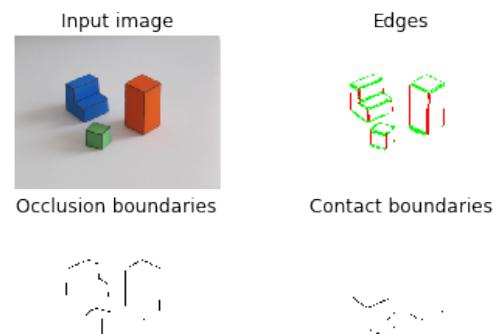
Problem 7

One of the many assumptions we made for the simple vision system was that all surfaces is planar. If we try to use this vision model on something like a cylinder we get incorrect, but hilarious results as the one seen in fig. 4b. The model tried to explain such a shape by modeling it as a planar surface going up and towards the camera.

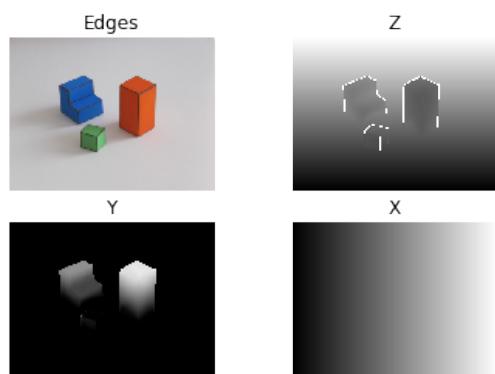
We could approximate the curvature of an edge by considering the normal vectors in a small neighbourhood around each point along the edge. Then perhaps the curvature at a surface point could be estimated by some interpolation of surrounding edges.



(a) Image 4 as seen from a new perspective

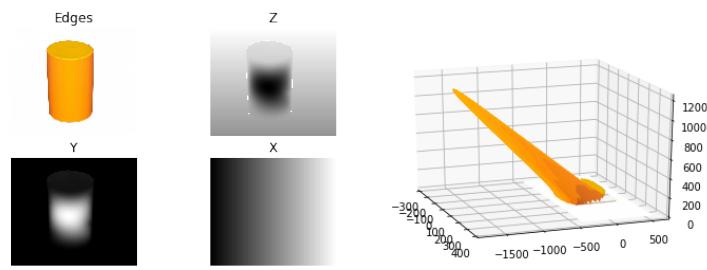


(b) The detected edges in image 4



(c) Coordinate gradients for image 4

Figure 3: Recovered 3D information from image 4



(a) Coordinate gradients for the back to 3D cylinder
(b) Distorted cylinder rendered

Figure 4: Reconstructed cylinder coordinates