6.869 Advances in Computer Vision

Problem Set 1

Odin Aleksander Severinsen

Problem 1 *Perspective and orthographic projections* (1 point)

Figure 1: *Perspective projection.*

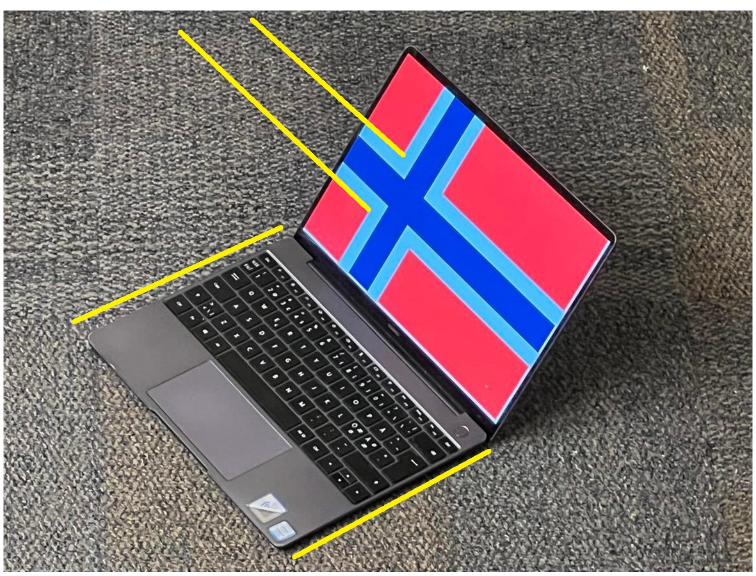

Figure 2: *Orthographic projection.*

There are multiple giveaways in the two picture for their respective type of projection. In Figure 1 we see that lines in the floor carpet, in addition the Norwegian flag, are not parallel, and have a vanishing point towards the top of the image. Perhaps most strikingly, the laptop's left side is shorter than the right side. In Figure 2, however, the left and right side of the laptop are both roughly the same side. Both the base and the screen of the laptop are proper rectangles, while in Figure 1 they take on a much more trapezoidal shape.

Problem 2 *Orthographic projection equations* (1 point)

We have the equation

$$\boldsymbol{x} = \alpha \cdot \mathbf{P} \cdot \mathbf{R}_x(\theta)\boldsymbol{X} + \boldsymbol{x}_0. \tag{1}$$

The rotation matrix is given as

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

An orthographic projection drops the Z coordinate, so the resulting matrix is simply given as

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Multiplying out gives

$$\begin{aligned}
\begin{bmatrix} x \\ y \end{bmatrix} &= \alpha \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} && (2) \\
&= \alpha \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y\cos(\theta) - Z\sin(\theta) \\ Z\cos(\theta) + Y\sin(\theta) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} && (3) \\
&= \alpha \cdot \begin{bmatrix} X \\ Y\cos(\theta) - Z\sin(\theta) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} && (4) \\
\implies \quad & x = \alpha X + x_0, && (5) \\
& y = \alpha\left(Y\cos(\theta) - Z\sin(\theta)\right) + y_0 && (6)
\end{aligned}$$

which we were supposed to show.

The given constraints give the equations

$$\begin{aligned}
0 &= \alpha \cdot 0 + x_0, && (7) \\
0 &= \alpha \cdot (\cos(\theta) \cdot 0 - \sin(\theta) \cdot 0) + y_0, && (8) \\
3 &= \alpha \cdot 1 + 0, && (9)
\end{aligned}$$

from which we derive that

$$\begin{aligned}
x_0 &= 0, && (10) \\
y_0 &= 0, && (11) \\
\alpha &= 3. && (12)
\end{aligned}$$

Problem 3 *Edge and surface constraints* (1 point)

For Z, we get the similar constraints as for Y,

$$\frac{\partial Z}{\partial \boldsymbol{t}} = \mathbf{0} \qquad \text{along vertical edges,} \tag{13}$$

$$\text{where} \tag{14}$$

$$\boldsymbol{t} \perp \boldsymbol{n}, \tag{15}$$

$$\boldsymbol{n} = \nabla I(x,y) / \left\| \nabla I(x,y) \right\|, \tag{16}$$

$$\frac{\partial Z}{\partial y} = \frac{-1}{\alpha \sin(\theta)} \qquad \text{along horizontal edges.} \tag{17}$$

$$\frac{\partial^2 Z}{\partial x^2} = 0, \tag{18}$$

$$\frac{\partial^2 Z}{\partial y^2} = 0, \qquad \text{along faces (not edges)} \tag{19}$$

$$\frac{\partial^2 Z}{\partial x \partial y} = 0. \tag{20}$$

where $I(x,y)$ is the intensity function of the image I.

Equation (13) follows from the fact that, along vertical edges, we must have that $Z(x,y) = \text{const}$, and so $\partial Z / \partial \boldsymbol{t} = \mathbf{0}$ when $\boldsymbol{t}$ is parallell to the edge. Equation (17) follows from the fact that $Y(x,y) = \text{const}$ for horizontal edges, and so differentiating (6) with respect to y gives the desired result when assuming the edge is horizontal. The remaining equations follow from the fact that all faces are flat, and so we cannot have any curvature.

Problem 4 *Complete the code* (2 points)

```python
# Contact edge: dY/dy = ?
# Requires: a transform matrix
if contact_edges[i, j]:
  Aij[:,:,c] = np.array([
    [0, 0, 0],
    [0, 1, 0],
    [0, 0, 0]
  ])
  b[c] = 0
  update_indices()

# Vertical edge: dY/Dy = ?
# Requires: a transform matrix, alpha
if verticalsum > 0 and groundsum == 0:
  Aij[:,:,c] = np.array([
    [-1, -2, -1],
    [0, 0, 0],
    [1, 2, 1]
  ])/8
  b[c] = 1/np.cos(alpha)
  update_indices()

# dY/dt = 0 (you'll have to express t using other variables)
# Requires: a transform matrix, i, j, theta
if horizontalsum > 0 and groundsum == 0 and verticalsum == 0:
```

```
  Aij[:,:,c] = np.array([
    [-1, -2, -1],
    [0, 0, 0],
    [1, 2, 1]
  ])*np.sin(theta[i,j]) + np.array([
    [-1, 0, 1],
    [-2, 0, 2],
    [-1, 0, 1]
  ])*-np.cos(theta[i,j])
  b[c] = 0
  update_indices()

# laplacian = 0 (weighted by 0.1 to reduce constraint strength)
# Requires: multiple transform matrices
if groundsum == 0:
  Aij[:,:,c] = 0.1*np.array([
    [0, 0, 0],
    [-1, 2, -1],
    [0, 0, 0]
  ])
  b[c]       = 0
  update_indices()

  Aij[:,:,c] = 0.1*np.array([
    [0, -1, 0],
    [0, 2, 0],
    [0, -1, 0]
  ])
  b[c]       = 0
  update_indices()

  Aij[:,:,c] = 0.1*np.array([[ 0., -1.,  1.],
 [ 0.,  1., -1.],
 [ 0.,  0.,  0.]])

  b[c]       = 0
  update_indices()
```

Problem 5 *Run the code* (1 point)

See Figure 3, Figure 4 and Figure 5.

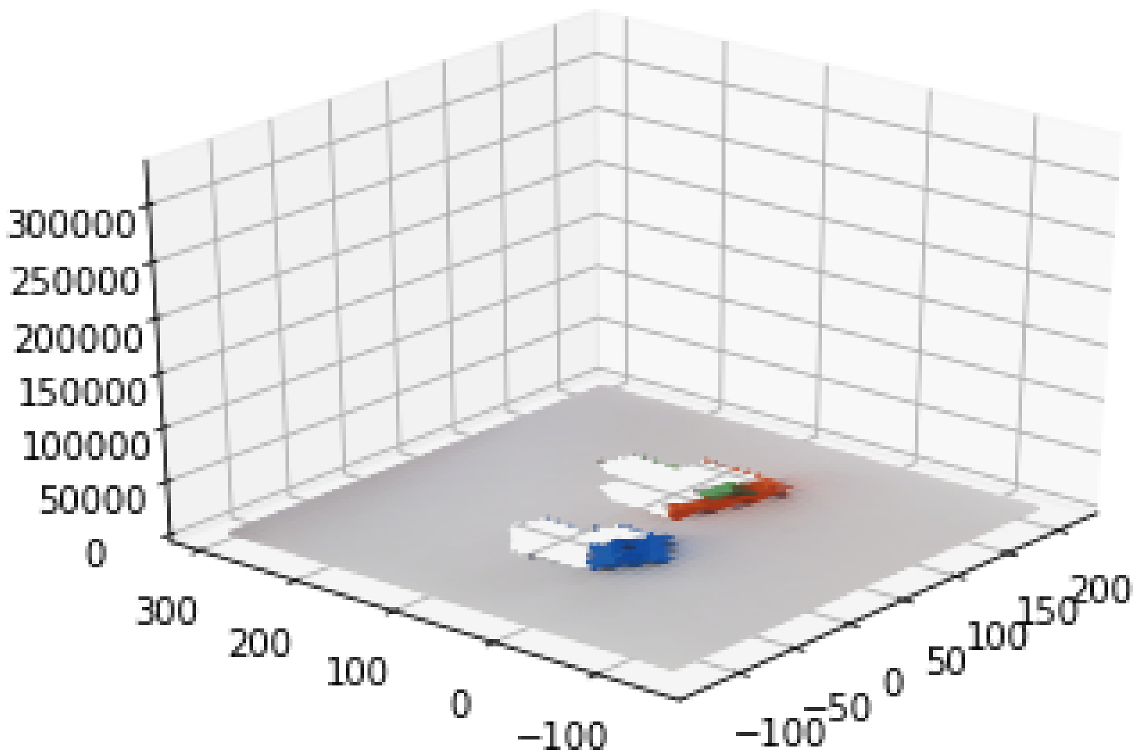

Figure 3: *`img2.png` 3D reconstruction.*

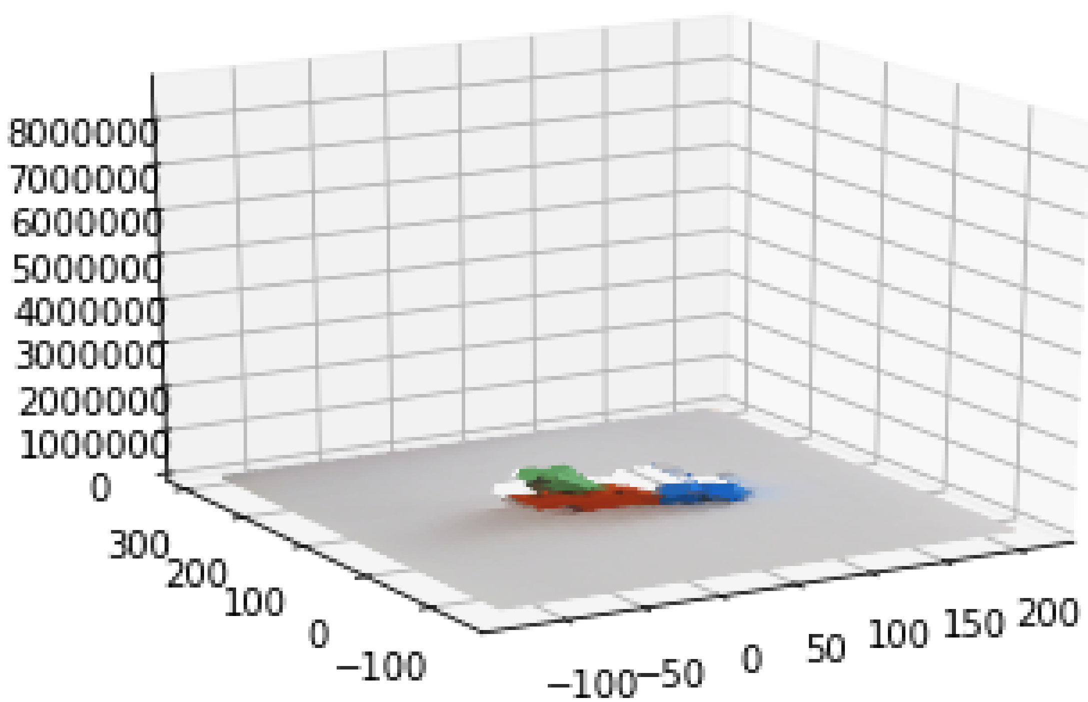

Figure 4: `img3.png` *3D reconstruction.*

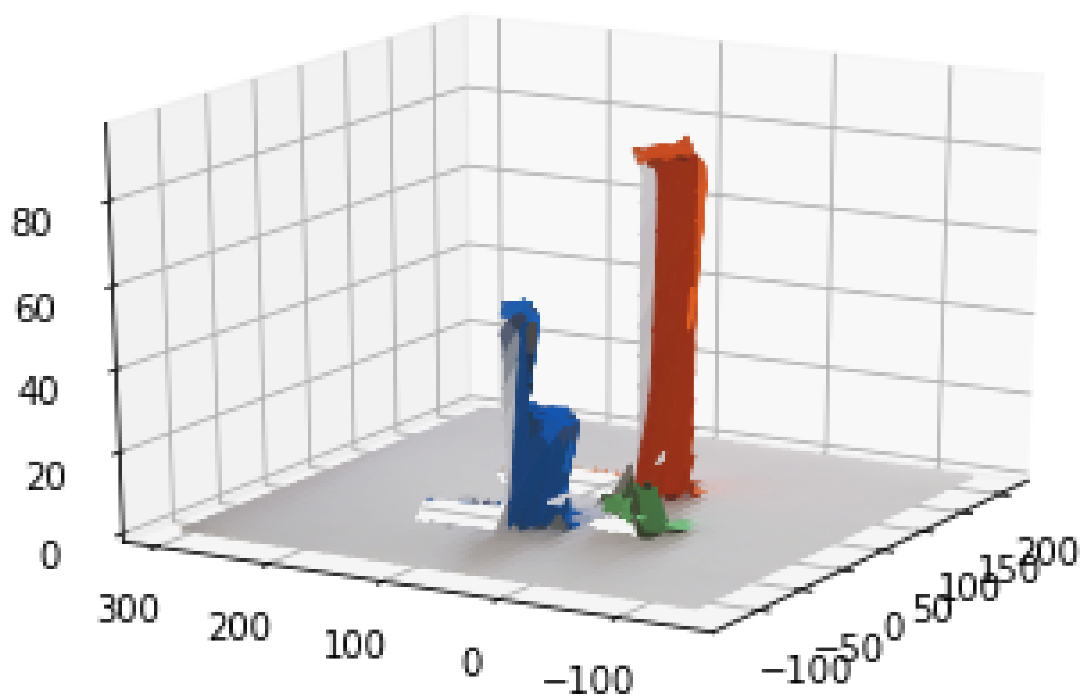

Figure 5: `img4.png` *3D reconstruction.*

Problem 6 *Violating simple world assumptions* (1 point)

Image `img3.png` is chosen as comparison. As can be seen in Figure 6 and Figure 4, the 3D reconstruction catastrophically fails. Comparing `img3.png` with `img1.png` reveals that the only difference is the obscuring of one block by another. Especially considering that `img4.png` does not have occlusion, same as `img1.png`, and also does not have catastrophic failure, this suggests that the occlusion of blocks by other blocks is the reason for causing failure. This is again confirmed by the fact that `img2.png` 3D reconstruction in Figure 3 also has occlusion of blocks and also suffers from catastrophic failure. This occlusion causes violations in our assumptions about the simple 3D world we have constructed as the edges around the blocks no longer simply define the transition to the background, but may be another block that is obscuring the rear block.

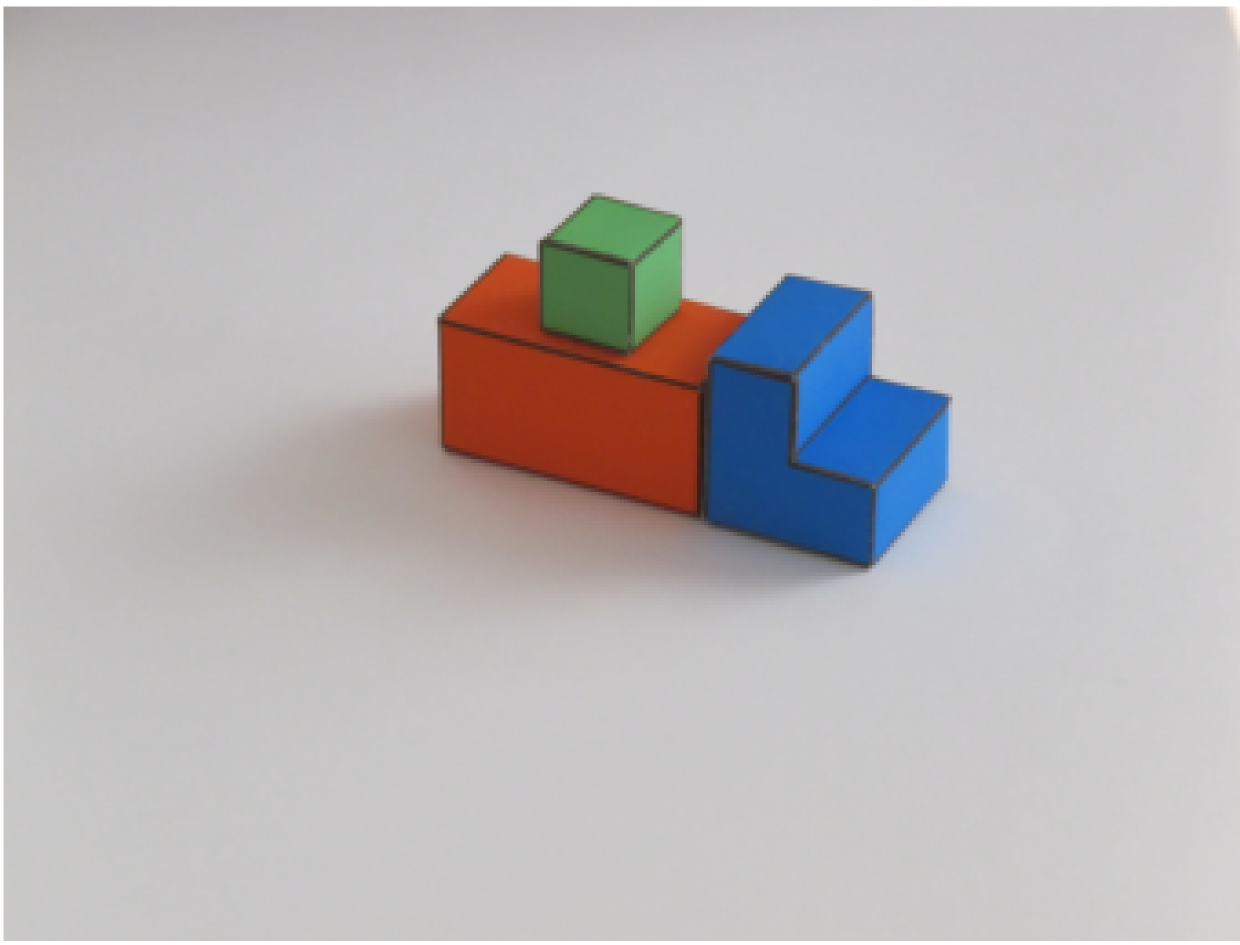

Figure 6: `img3.png`.