Autonomous Vision Group
Prof. Dr.-Ing. Andreas Geiger

Eberhard Karls
Universität
Tübingen

Computer Vision

Exercise 1 – Image Formation

1 Pen and Paper

1.1 Homogeneous Coordinates

a) Proof that in homogeneous coordinates, the intersection point $\tilde{\mathbf{x}}$ of the two lines $\tilde{\mathbf{l}}_1$ and $\tilde{\mathbf{l}}_2$ is given by $\tilde{\mathbf{x}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2$.

As we discussed in the individual Q/A session, you mainly have two approaches:

Approach 1: We first provide a constructive proof. Let's define $\tilde{\mathbf{l}}_1 = (a_1, a_2, a_3)^\top$ and $\tilde{\mathbf{l}}_2 = (b_1, b_2, b_3)^\top$. We know that the sets of points lying on $\tilde{\mathbf{l}}_1$ or $\tilde{\mathbf{l}}_2$ are given by:

$$\{x, y \mid a_1 x + a_2 y + a_3 = 0\}$$
$$\{x, y \mid b_1 x + b_2 y + b_3 = 0\}$$

As a result, we obtain the system of two linear equations

$$a_1 x + a_2 y + a_3 = 0$$
$$b_1 x + b_2 y + b_3 = 0$$

When solving the first equation for x, we get

$$x = -\frac{a_2}{a_1} y - \frac{a_3}{a_1}$$

Let's insert this in our second equation:

$$-\frac{b_1 a_2}{a_1} y - \frac{a_3 b_1}{a_1} + b_2 y + b_3 = 0$$
$$y \left(b_2 - \frac{b_1 a_2}{a_1} \right) = \frac{a_3 b_1}{a_1} - b_3$$
$$y \left(\frac{b_2 a_1 - b_1 a_2}{a_1} \right) = \frac{a_3 b_1 - b_3 a_1}{a_1}$$
$$y = \frac{a_3 b_1 - b_3 a_1}{b_2 a_1 - b_1 a_2}$$

In a similar fashion, we solve the first equation for y

$$y = -\frac{a_1}{a_2} x - \frac{a_3}{a_2}$$

and insert this in the second equation

$$\begin{aligned} b_1 x - \frac{b_2 a_1}{a_2} x - \frac{b_2 a_3}{a_2} + b_3 &= 0 \\ x\left(b_1 - \frac{b_2 a_1}{a_2}\right) &= \frac{b_2 a_3}{a_2} - b_3 \\ x\left(\frac{b_1 a_2 - b_2 a_1}{a_2}\right) &= \frac{b_2 a_3 - b_3 a_2}{a_2} \\ x &= \frac{b_2 a_3 - b_3 a_2}{b_1 a_2 - b_2 a_1} \\ x &= \frac{b_3 a_2 - b_2 a_3}{b_2 a_1 - b_1 a_2} \end{aligned}$$

We find that the intersection point in heterogenous coordinates is given by

$$\mathbf{x}_{\text{intersect}} = \begin{pmatrix} \frac{b_3 a_2 - b_2 a_3}{b_2 a_1 - b_1 a_2} \\ \frac{a_3 b_1 - b_3 a_1}{b_2 a_1 - b_1 a_2} \end{pmatrix}$$

When we now solve the cross product of the two lines in homogeneous coordinates

$$\tilde{\mathbf{x}}_{\text{intersect}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2 = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix} \sim \begin{pmatrix} \frac{a_2 b_3 - a_3 b_2}{a_1 b_2 - a_2 b_1} \\ \frac{a_3 b_1 - a_1 b_3}{a_1 b_2 - a_2 b_1} \\ 1 \end{pmatrix}$$

Hence, we can see

$$\tilde{\mathbf{x}}_{\text{intersect}} = \begin{pmatrix} \mathbf{x}_{\text{intersect}} \\ 1 \end{pmatrix}$$

Approach 2: Alternatively, we can define the point $\tilde{\mathbf{x}} := \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2$ and then show that $\tilde{\mathbf{x}}$ lies on both lines. To show that a point lies on a line using homogeneous coordinates, we can use the inner product; as defined in the lecture, a line $\tilde{\mathbf{l}}$ in homogeneous coordinates is defined as

$$\{\bar{\mathbf{x}} \mid \tilde{\mathbf{l}}^\top \bar{\mathbf{x}} = 0\}$$

For example, for $\tilde{\mathbf{l}}_1$

$$\begin{aligned} \tilde{\mathbf{l}}_1^\top \tilde{\mathbf{x}} &= \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}^\top \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix} \\ &= a_1 a_2 b_3 - a_1 a_3 b_2 + a_2 a_3 b_1 - a_2 a_1 b_3 + a_3 a_1 b_2 - a_3 a_2 b_1 \\ &= (a_1 a_2 b_3 - a_1 a_2 b_3) + (a_1 a_3 b_2 - a_1 a_3 b_2) + (a_2 a_3 b_1 - a_2 a_3 b_3) \\ &= 0 \end{aligned}$$

And similarly, we can show that $\tilde{\mathbf{l}}_2^\top \tilde{\mathbf{x}} = 0$. As $\tilde{\mathbf{x}}$ lies on both lines, it is the intersection point.

b) Similarly, proof that the line that joins two points $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ is given by $\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2$.

Approach 1: Again we first provide a constructive proof: We write the 2D points in inhomogeneous coordinates

$$\mathbf{x} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

We can write the line between these two points as the set of all $(x, y)^\top$ which satisfy

$$\begin{aligned} y &= x\frac{y_1 - y_0}{x_1 - x_0} + \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} \qquad \Leftrightarrow \\ 0 &= x(y_1 - y_0) + y(x_0 - x_1) + (x_1 y_0 - x_0 y_1) \end{aligned}$$

We can read off the homogeneous line

$$\tilde{\mathbf{l}} = \begin{pmatrix} y_1 - y_0 \\ x_0 - x_1 \\ x_1 y_0 - x_0 y_1 \end{pmatrix}$$

Similarly, when we use the cross product formula, we obtain

$$\tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2 = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} = \begin{pmatrix} y_0 - y_1 \\ x_1 - x_0 \\ x_0 y_1 - y_0 x_1 \end{pmatrix} \sim \begin{pmatrix} y_1 - y_0 \\ x_0 - x_1 \\ x_1 y_0 - x_0 y_1 \end{pmatrix}$$

We follow that

$$\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2$$

Approach 2: As before, the inner product can also be used to verify that $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ both lie on the line $\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2$. For this, let $\tilde{\mathbf{x}}_1 = (\tilde{x}_1^1, \tilde{x}_1^2, \tilde{x}_1^3)^\top$ and $\tilde{\mathbf{x}}_2 = (\tilde{x}_2^1, \tilde{x}_2^2, \tilde{x}_2^3)^\top$. For example, for $\tilde{\mathbf{x}}_1$ we see

$$\begin{aligned} \tilde{\mathbf{l}}^\top \tilde{\mathbf{x}}_1 &= \begin{pmatrix} \tilde{x}_1^2 \tilde{x}_2^3 - \tilde{x}_1^3 \tilde{x}_2^2 \\ \tilde{x}_1^3 \tilde{x}_2^1 - \tilde{x}_1^1 \tilde{x}_2^3 \\ \tilde{x}_1^1 \tilde{x}_2^2 - \tilde{x}_1^2 \tilde{x}_2^1 \end{pmatrix}^\top \begin{pmatrix} \tilde{x}_1^1 \\ \tilde{x}_1^2 \\ \tilde{x}_1^3 \end{pmatrix} \\ &= \tilde{x}_1^2 \tilde{x}_2^3 \tilde{x}_1^1 - \tilde{x}_1^3 \tilde{x}_2^2 \tilde{x}_1^1 + \tilde{x}_1^3 \tilde{x}_2^1 \tilde{x}_1^2 - \tilde{x}_1^1 \tilde{x}_2^3 \tilde{x}_1^2 + \tilde{x}_1^1 \tilde{x}_2^2 \tilde{x}_1^3 - \tilde{x}_1^2 \tilde{x}_2^1 \tilde{x}_1^3 \\ &= (\tilde{x}_1^2 \tilde{x}_2^3 \tilde{x}_1^1 - \tilde{x}_1^2 \tilde{x}_2^3 \tilde{x}_1^1) + (\tilde{x}_1^3 \tilde{x}_2^2 \tilde{x}_1^1 - \tilde{x}_1^3 \tilde{x}_2^2 \tilde{x}_1^1) + (\tilde{x}_1^2 \tilde{x}_2^1 \tilde{x}_1^3 - \tilde{x}_1^2 \tilde{x}_2^1 \tilde{x}_1^3) \\ &= 0 \end{aligned}$$

And similarly, we can show that $\tilde{\mathbf{l}}^\top \tilde{\mathbf{x}}_2 = 0$. As both points $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ lie on the line $\tilde{\mathbf{l}}$, $\tilde{\mathbf{l}}$ is the line going through the two points.

c) You are given the following two lines:

$$\begin{aligned} \mathbf{l}_1 &= \{(x, y)^\top \in \mathbb{R}^3 | x + y + 3 = 0\} \\ \mathbf{l}_2 &= \{(x, y)^\top \in \mathbb{R}^3 | -x - 2y + 7 = 0\} \end{aligned}$$

First, find the intersection point of the two lines by solving the system of linear equations. Next write the lines using homogeneous coordinates and calculate the intersection point using the cross product. Do you obtain the same intersection point?

Rearranging the first equation gives

$$x = -y - 3$$

Inserting this into the second gives

$$\begin{aligned} y + 3 - 2y + 7 &= 0 \\ -y + 10 &= 0 \\ y &= 10 \end{aligned}$$

And we obtain

$$x = -10 - 3 = -13$$
$$y = 10$$

The inhomogeneous intersection point is $(-13, 10)^\top$.

The cross product is given by

$$\begin{bmatrix} 0 & -3 & 1 \\ 3 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{pmatrix} -1 \\ -2 \\ 7 \end{pmatrix} = \begin{pmatrix} 6+7 \\ -3-7 \\ 1-2 \end{pmatrix} = \begin{pmatrix} 13 \\ -10 \\ -1 \end{pmatrix} = \begin{pmatrix} -13 \\ 10 \\ 1 \end{pmatrix}$$

Hence, the homogeneous intersection point is $(-13, 10, 1)^\top$. We can follow that both calculations lead to the same result:

$$\begin{pmatrix} -13 \\ 10 \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \quad \text{with} \quad \mathbf{x} = \begin{pmatrix} -13 \\ 10 \end{pmatrix}$$

d) Write down the line whose normal vector is pointing into the direction $(3, 4)^\top$ and which has a distance of 3 from the origin.

First, it is important to note that you can normalize any non-zero vector $\mathbf{v}$ by dividing it by its norm $||\mathbf{v}||_2$:

$$\begin{aligned} ||\frac{1}{||\mathbf{v}||_2}\mathbf{v}||_2 &= \sqrt{\sum_{i=1}^{N_v} \left(\frac{1}{||\mathbf{v}||_2} v_i\right)^2} \\ &= \sqrt{\frac{1}{||\mathbf{v}||_2^2} \sum_{i=1}^{N_v} v_i^2} \\ &= \sqrt{\frac{1}{||\mathbf{v}||_2^2} ||\mathbf{v}||_2^2} = \sqrt{1} = 1 \end{aligned}$$

Now, first get the norm of the vector: $||(3, 4)^\top||_2 = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$. When dividing by the norm, we obtain the normalized normal vector:

$$\mathbf{n} = \begin{pmatrix} 3/5 \\ 4/5 \end{pmatrix}$$

Hence, we obtain the line as

$$\tilde{\mathbf{l}} = \begin{pmatrix} 3/5 \\ 4/5 \\ 3 \end{pmatrix}$$

e) What distance from the origin and what (normalized) normal vector does the homogeneous line $\tilde{\mathbf{l}} = (2, 5, \frac{\sqrt{29}}{5})^\top$ have?

We now first calculate the norm of the unnormalized normal vector

$$|| \begin{pmatrix} 2 \\ 5 \end{pmatrix} ||_2 = \sqrt{2^2 + 5^2} = \sqrt{29}$$

Now, we divide $\tilde{\mathbf{l}}$ by it

$$\tilde{\mathbf{l}} \sim \frac{1}{\sqrt{29}} \begin{pmatrix} 2 \\ 5 \\ \sqrt{29}/5 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{29} \\ 5/\sqrt{29} \\ 1/5 \end{pmatrix}$$

As a result, we obtain

$$\mathbf{n} = (\frac{2}{\sqrt{29}}, \frac{5}{\sqrt{29}})^\top$$
$$d = \frac{1}{5} = 0.2$$

1.2 Transformations

a) Write down the 2×3 translation matrix which maps $(1, 2)^\top$ onto $(0, 3)^\top$.

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

We can test $\mathbf{T}$:

$$\mathbf{T} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1-1 \\ 2+1 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

b) Let's assume that you are given N 2D correspondence pairs

$$(\mathbf{x}_i, \mathbf{y}_i) = \left(\begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}, \begin{pmatrix} y_1^i \\ y_2^i \end{pmatrix} \right)$$

Find the 2×3 translation matrix mapping $\mathbf{x}_i$ onto $\mathbf{y}_i$ which is optimal in the least square sense.
Hint: Define a cost function as

$$E(\mathbf{T}) = \sum_{i=1}^{N} ||\mathbf{T}\bar{\mathbf{x}}_i - \mathbf{y}_i||_2^2$$

and find the optimal $\mathbf{T}^\star$ which minimizes E:

$$\mathbf{T}^\star = \underset{\mathbf{T}}{\text{argmin}} \; E(\mathbf{T})$$

You can find $\mathbf{T}^\star$ by calculating the Jacobian $\mathbf{J}_E$ of E and setting it to $\mathbf{0}^\top$:

$$\mathbf{J}_E = \left[\frac{\partial E}{\partial t_1}, \ldots, \frac{\partial E}{\partial t_N} \right] \stackrel{!}{=} \mathbf{0}^\top$$

Can you give an intuitive explanation for the equation you derive for $\mathbf{T}^*$?

We first observe

$$T = \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \end{bmatrix}$$

Next:

$$E(\mathbf{T}) = \sum_{i=1}^{N} ||\mathbf{T}\mathbf{x}_i - \mathbf{y}_i||_2^2$$
$$E(t_1, t_2) = \sum_{i=1}^{N} (x_1^i + t_1 - y_1^i)^2 + (x_2^i + t_2 - y_2^i)^2$$

We then calculate the partial derivatives set them equal to 0:

$$\begin{aligned}
\frac{\partial E}{\partial t_1} &= \sum_{i=1}^{N} (x_1^i + t_1 - y_1^i)^2 \\
&= \sum_{i=1}^{N} 2(x_1^i + t_1 - y_1^i) \\
&= \sum_{i=1}^{N} 2(x_1^i - y_1^i) + \sum_{i=1}^{N} 2t_1 \\
&= \sum_{i=1}^{N} 2(x_1^i - y_1^i) + 2Nt_1 \\
&\stackrel{!}{=} 0
\end{aligned}$$

Rearranging gives

$$t_1 = \frac{1}{N} \sum_{i=1}^{N} (y_1^i - x_1^i)$$

And similarly we obtain

$$t_2 = \frac{1}{N} \sum_{i=1}^{N} (y_2^i - x_2^i)$$

We see that the optimal translation is the mean of the individual translations.

c) You are given the following three correspondence pairs:

$$\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -5 \end{pmatrix} \right)$$
$$\left(\begin{pmatrix} 5 \\ 7 \end{pmatrix}, \begin{pmatrix} 7 \\ 6 \end{pmatrix} \right)$$
$$\left(\begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ -4 \end{pmatrix} \right)$$

Using your derived equation, calculate the optimal 2×3 translation matrix $\mathbf{T}^\star$.

$$
\begin{aligned}
\mathbf{t}_1 &= (3, -6)^\top \\
\mathbf{t}_2 &= (2, -1)^\top \\
\mathbf{t}_3 &= (1, -5)^\top \\
\mathbf{t}^\star &= \frac{1}{3}(\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3) = (2, -4)^\top \\
\mathbf{T}^\star &= \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -4 \end{bmatrix}
\end{aligned}
$$

1.3 Camera Projections

a) Calculate the full rank 4×4 projection matrix $\tilde{\mathbf{P}}$ for the following scenario:

- The camera pose consists of a $90°$ rotation around the x axis and translation of $(1, 0, 2)^\top$.
- The focal lengths f_x, f_y are 100.
- The principal point $(c_x, c_y)^\top$ is $(25, 25)$.

The camera pose is given by

$$
\begin{aligned}
\mathbf{R} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha \end{bmatrix} \overset{\alpha \equiv \pi}{=} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\
\mathbf{t} &= \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} \\
\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Our camera matrix is

$$
\mathbf{K} = \begin{bmatrix} 100 & 0 & 25 \\ 0 & 100 & 25 \\ 0 & 0 & 1 \end{bmatrix}
$$

We obtain the projection matrix

$$
\begin{aligned}
\tilde{\mathbf{P}} &= \begin{bmatrix} 100 & 0 & 25 & 0 \\ 0 & 100 & 25 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 100 & 25 & 0 & 150 \\ 0 & 25 & -100 & 50 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

b) For the previously defined projection, find the world point in inhomogeneous coordinates $\mathbf{x}_w$ which corresponds to the projected homogeneous point in screen space $\tilde{\mathbf{x}}_s = (25, 50, 1, 0.25)^\top$.

First, we find the inverse of $\tilde{\mathbf{P}}$ using standard algorithms, e.g., the Gauss–Jordan elimination:

$$\tilde{\mathbf{P}}^{-1} = \begin{bmatrix} 1/100 & 0 & -1/4 & -1 \\ 0 & 0 & 1 & -2 \\ 7 & -1/100 & 1/4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next, we can find the homogeneous world point $\tilde{\mathbf{x}_w}$

$$\begin{aligned} \tilde{\mathbf{x}}_w &= \tilde{\mathbf{P}}^{-1}\tilde{\mathbf{x}}_s \\ &= \begin{bmatrix} 1/100 & 0 & -1/4 & -1 \\ 0 & 0 & 1 & -2 \\ 7 & -1/100 & 1/4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 25 \\ 50 \\ 1 \\ 0.25 \end{pmatrix} \\ &= \begin{pmatrix} -0.25 \\ 0.5 \\ -0.25 \\ 0.25 \end{pmatrix} \sim \begin{pmatrix} -1 \\ 2 \\ -1 \\ 1 \end{pmatrix} \end{aligned}$$

In inhomogeneous coordinates, we write

$$\mathbf{x}_w = \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$$

c) Let's perform our first projection of a geometric shape. We define $\mathcal{C}_0$ as the cube centered at $\mathbf{c}_c = (0, 0, 15)^\top$ with equal side lengths $s = 20$.

i) Project the 8 corners of the cube $\mathcal{C}_0$ to the image plane for the pinhole camera with focal lengths $f_x = f_y = 5$ and the principal point $(c_x, c_y)^\top = (10, 10)^\top$. Draw the projected points and edges of the cube in a coordinate system on a paper.

ii) Let's move the cube further away from the pinhole camera along the z-axis such that the distance of the center of the new cube $\mathcal{C}_1$ is now 20. Further, let's zoom in with our camera such that the focal lengths are $f_x = f_y = 10$ and the principal point remains the same. Draw the projected points and edges of the cube in a coordinate system on a paper.

iii) Let's move it even further away along the z-axis while zooming in. Now, the distance of the center of cube $\mathcal{C}_2$ is 100 and the focal lengths are $f_x = f_y = 90$. Draw the projected points and edges of the cube in a coordinate system on a paper.

iv) Project the 8 corners of the first cube $\mathcal{C}_0$ using an orthographic projection and add the principal point $\mathbf{c}_c = (10, 10)^\top$ onto the obtained pixel coordinates to be in the same coordinate system as before. Draw the projected points and edges of the cube in a coordinate system on a paper.

v) When is the perspective projection most similar to the orthographic projection?

The first camera matrix should be defined as

$$\mathbf{K}_1 = \begin{bmatrix} 5 & 0 & 10 \\ 0 & 5 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

Using homogeneous coordinates, the point in screen space / pixel coordinates are obtained as

$$\tilde{\mathbf{x}}_s = \mathbf{K}_1\tilde{\mathbf{x}}_c$$

Then, the corners of the cube can be read off, e.g.

$$\mathbf{c}_1 = \begin{pmatrix} -10 \\ -10 \\ 5 \end{pmatrix}, \mathbf{c}_2 = \begin{pmatrix} -10 \\ 10 \\ 5 \end{pmatrix}, \ldots$$

which are then projected using matrix multiplication and normalization to obtain the augmented vector, e.g.

$$\mathbf{p}_1 = \mathbf{K}_1 \mathbf{c}_1 = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix} \sim \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Similarly, e.g.

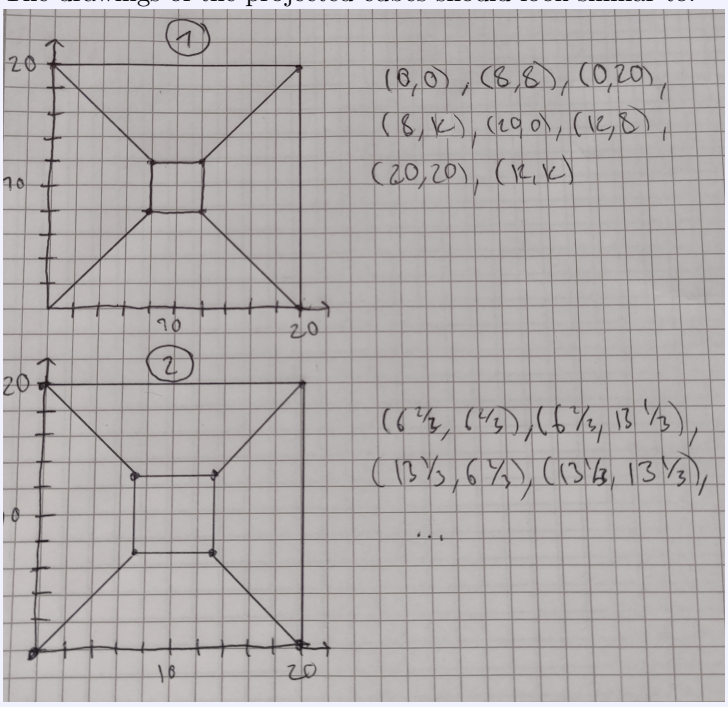
$$\mathbf{p}_2 = \mathbf{K}_1 \mathbf{c}_2 = \begin{pmatrix} 0 \\ 100 \\ 5 \end{pmatrix} \sim \begin{pmatrix} 0 \\ 20 \\ 1 \end{pmatrix}$$

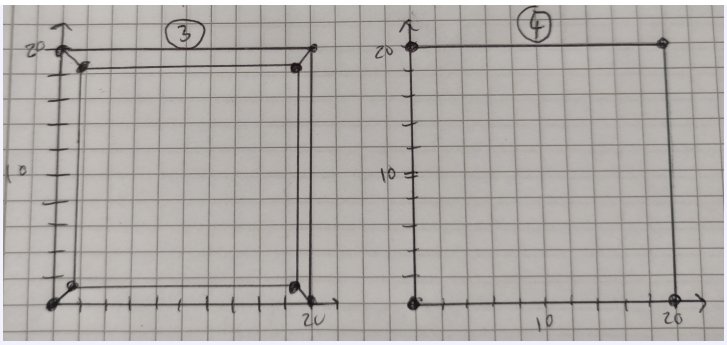
For the cubes $\mathcal{C}_1$ and $\mathcal{C}_2$, define the edges and camera matrices similarly and project them in a similar fashion.

For the orthographic projection, we obtain the screen space coordinates via

$$\mathbf{x}_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}_c + \begin{pmatrix} 10 \\ 10 \end{pmatrix}$$

The drawings of the projected cubes should look similar to:

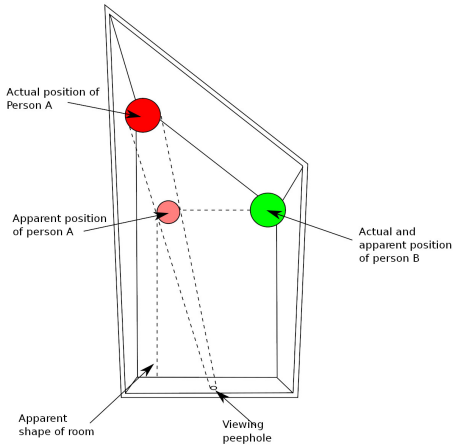




For larger distances and focal lengths, the projection looks more similar to the orthographic projection and the perspective effect vanishes. In the limit, the perspective projection is equal to the orthographic projection.

d) Let's build our own Ames Room (see Fig. 1)! For this, print the pdf document `ames_room.pdf` and follow the instructions given on the document.

(a) Photo of a real-world an Ames Room



(b) Schematic drawing of an Ames Room

Figure 1: **Ames Room.** The ames room creates an optical illusion for the viewer (see Fig. 1a). When looking through the peephole into the room, the room itself appears to be an ordinary cuboid despite its actual distorted shape (see Fig. 1b). As a result, objects in the distorted corner look smaller than they actually are. This is achieved by designing the room in such a way that crucial information for the geometric understanding of the real room are lost when looking through the peephole. It serves as an example of how different 3D scenes (here the perceived cuboid room and the real distorted room) can be projected to the same image when using a perspective projection.

Follow the instructions on the document.

1.4 Photometric Image Formation

a) Write down the thin lens formula and calculate the focus distance z_c in meters for focal length $f = 100mm$ and the distance to the image plane $z_s = 104mm$.

The thin lense formula is

$$\frac{1}{z_s} + \frac{1}{z_c} = \frac{1}{f}$$

We can reformulate this to

$$\begin{aligned} 1 + \frac{z_c}{z_s} &= \frac{z_c}{f} \quad \Leftrightarrow \\ z_c(\frac{1}{z_s} - \frac{1}{f}) &= -1 \quad \Leftrightarrow \\ z_c \frac{f - z_s}{z_s f} &= -1 \quad \Leftrightarrow \\ z_c &= -\frac{z_s f}{f - z_s} \end{aligned}$$

We can now calculate z_c

$$z_c = -\frac{100mm \cdot 104mm}{100mm - 104mm} = 2600mm = 2.6m$$

b) Write the diameter of the circle of confusion c as a function of the focal length f, the image plane distance z_s as well as the distance Δz_s and the f-number N.

We use triangle congruence to read off

$$\begin{aligned} \frac{c}{\Delta z_s} &= \frac{d}{z_s} \quad \Leftrightarrow \\ c &= \frac{d}{z_s} \Delta z_s \end{aligned}$$

We further know

$$\begin{aligned} N &= \frac{f}{d} \quad \Leftrightarrow \\ d &= \frac{N}{f} \end{aligned}$$

We conclude

$$c = \frac{f}{N z_s} \Delta z_s$$

c) Using your derived formula, calculate the diameter of the circle of confusion for the setting $f = 35mm$, N = 1.4, $z_s = 40mm$ when $\Delta z_s = 0.1mm$ as well as when $\Delta z_s = 0.03mm$. Assuming the camera uses a sensor of size $64mm^2$ and a pixel resolution of 400×400 with squared pixels, are the calculated projections sharp or not?

For the first case, we obtain

$$c_1 = \frac{35mm}{1.4 \cdot 40mm} 0.1mm = 0.0625mm$$

For the second case, we obtain

$$c_2 = \frac{35mm}{1.4 \cdot 40mm} 0.03mm = 0.01875mm$$

The projections are sharp if the diameter is smaller than the smallest pixel side length. In the following, we discuss two approaches to calculate this:

Approach 1: One pixel has the surface area

$$a_p = \frac{64mm^2}{400 \cdot 400} = 0.0004mm^2$$

As the pixels are squares, i.e. their height and width are equal, we can obtain a pixel's side length

$$s_p = \sqrt{a_p} = \sqrt{0.0004mm^2} = 0.02mm$$

Approach 2: As the pixels are squared and as the sensor has a squared resolution, we can calculate the sensor's width and height:

$$w_s = h_s = \sqrt{64mm^2} = 8mm$$

We obtain a pixel's side length by dividing it by the height / width pixel resolution:

$$s_p = \frac{w_s}{400} = \frac{h_s}{400} = 0.02mm$$

For both approaches, we obtain the pixel's side length of $0.02mm$. Hence, only the second one is sharp.

2 Coding Exercises

The following exercises will be coding exercises. They are all contained in the jupyter notebook `code/image_formation.ipynb`. The notebook itself is self-contained but you can also use this document as guidance. If you are stuck, you can find *Hints* in the notebook itself which are written upside-down.

See the notebook for the solutions.

2.1 Perspective Projection

a) Implement the function `get_camera_intrinsics` which takes as input the focal lengths f_x, f_y and the principal point $(c_x, c_y)^\top$ and returns a 3×3 camera intrinsics matrix $\mathbf{K}$.

b) Implement the function `get_perspective_projection` which takes as input a 3D point in camera space $\mathbf{x}_c$ and the camera matrix $\mathbf{K}$ and it returns a two-dimensional point in screen space, hence the pixel coordinates $\mathbf{x}_s$ for the 3D point $\mathbf{x}_c$.

c) Implement the function `get_face_color` which maps a normal vector $\mathbf{n}$ and a point light direction vector $\mathbf{r}$ to an RGB color value. We assume the light to be a point light source,i.e., the light source is infinitely small. You should follow the rendering equation discussed in the lecture where you can assume that the surface does not emit light, the BRDF term is always 1, and the incoming light L_{in} is 1 for the point light source direction.

d) Create a visualization where you change both focal lengths similar to the previously given example.

e) Create a visualization where you translate the cube along the y-axis between $[-2, 2]$.

f) Create a visualization showing the Dolly Zoom Effect. For this, linearly change the focal lengths between $[10, 150]$ while you also translate the cube along the z-axis between $[0.9, 5]$.

2.2 Orthographic Projection

a) Implement the function `get_orthographic_projection` which maps a 3D point in camera space $\mathbf{x}_c$ to 2D pixel coordinates $\mathbf{x}_s$ using a orthographic projection.

b) To confirm your analysis, plot a projected cube with center $(0, 0, 150)^\top$ and focal lengths $f_x = f_y = 10000$. When does the perspective projection look most similar to the orthographic projection?

2.3 Panorama Stitching using DLT

a) Implement the function `get_Ai` which takes as input the two homogeneous points $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_i$ and returns the 2×9 matrix $\mathbf{A}_i$ discussed in the lecture. (Note: It's 2×9 and not 3×9 as you can drop the last row (see lecture).)

b) Implement the function `get_A` which uses the `get_Ai` for all N correspondence pairs and concatenates them to return the $2N \times 9$ matrix $\mathbf{A}$.

c) Implement the function `get_homography` which maps the N point correspondences to the homography $\mathbf{H}$ using the Direct Linear Transform (DLT) algorithm.

d) Now, it's your turn: Go out and take two photos of a cool scene. Make sure that you do not change the position of the camera/smartphone, only the angle! Replace the data paths for the images and follow the instructions in the notebook to create your own panorama!