

Exercises in 3D Computer Vision

Exercise 8 **Hierarchy of transformations**

- List all classes of 2D transformations you can recall. Show the containment relation of the different classes in a schematic way.
- Name the invariants of each class of transformations.
- Proof that projective Transformations preserve intersection of lines. Before you start recall what you've learned in a previous exercise about the relation between projective transformations of points and lines. Explain why proofing

$$\mathbf{H}^{-\top} (\mathbf{1} \times \mathbf{l}') \propto \mathbf{Hl} \times \mathbf{Hl}'$$

is sufficient. What does it mean geometrically?

Exercise 9 **Rotation representations**

- Derive the formula for 2D rotations with non homogeneous coordinates.

- Start on a unit circle.
- Recap:

$$\begin{aligned} \sin(a+b) &= \sin(a)\cos(b) + \cos(a)\sin(b) \\ \cos(a+b) &= \cos(a)\cos(b) - \sin(a)\sin(b) \end{aligned}$$

- Derive how to convert the axis angle representation into a matrix representation.
 - Start with a geometric visualization. Project the point to be rotated onto a plane perpendicular to the rotation vector, rotate the point in that plane and rise it again from this plane to its original height.
 - Recap: cross-product properties
 - anticommutative: $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$
 - skew-symmetric matrix: $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b}$
- Use the derivation of b) to determine rotation matrices around the axes of the Euclidean coordinate frame with unit vector basis.

Exercise 10 **2D Image Transformations in MatLab**

In the lecture you have learned how to transform single points in 2D using similarities, affinities, or projectivities. Now, we want to apply these transformations not only to one point but to an entire image.

- a) What problem occurs when dealing with rastered image coordinates? How can it be dealt with?
- b) An image in MatLab is a $n \times m$ matrix with color or intensity values as its entries. The height is the number of rows (n) and the width is the number of columns (m). For image transformations we are at first not interested in the color values but in the coordinates. For applying transformations to the coordinates of an image in MatLab, one has to create a matrix with all coordinates as its entries. Form a matrix in MatLab where each coordinate of an image with size 7×7 pixels is a column with three entries (homogeneous coordinates). Apply your method to an image with 256×256 pixels.
- c) Once you have created the coordinate-matrix you can apply the transformation. How do you get an image from the transformed coordinates?
- d) **(H)** Write a method `imTrans` in MatLab that takes two arguments, an image and a transformation, and returns the transformed image.