# Exercise 1

Student Name: Lior Kotlar, Username: lior.kotlar, ID: 316219963

Task 1:

Task 2:

Your code performs image rectification in two steps:

1. Affine Rectification – Removes the projective distortion using two pairs of parallel lines.
2. Metric Rectification – Removes affine distortions to recover angles and scales using perpendicular lines and the circular points concept.

This process transforms a projectively distorted image into one that looks like an image taken from the front without an angle.

Theoretical background

Affine Rectification:

In projective geometry:

* Under perspective projection, parallel lines appear to meet at a vanishing point.
* Different sets of parallel lines meet at different vanishing points.
* The line joining two vanishing points of orthogonal directions is called the horizon line.

The goal of affine rectification is to transform the image so that this horizon becomes the line at infinity. This step removes the projective distortion but keeps affine distortions (like shear and scale).

This is done using a homography that maps the estimated horizon line to ​. If the estimated horizon line has coordinates , then:

This matrix transforms the image such that the horizon is now at infinity.

**Code Breakdown:**

* – Lets the user draw lines interactively by clicking two points.
* – Collects two sets of parallel lines (which gives two vanishing points).
* – Converts two image points to a line in homogeneous coordinates.
* – Uses two vanishing points ​ and ​ to compute the horizon line .
* – Constructs the affine rectification matrix as above.
* – Applies the rectification using OpenCV's warpPerspective.

Metric Rectification:

After affine rectification, lines that were parallel in the scene are now parallel in the image. But angles and lengths are still distorted.

To fix this, we need metric rectification, which requires restoring orthogonality using lines that are perpendicular in the real world.

**Circular Points**

In projective geometry, all orthogonality and metric properties are contained within the absolute conic equation, which lies on the line at infinity. There are the two circular points that are included in the absolute conic:

Their inner product is zero:

A constraint is derived from that :

where is a symmetric matrix representing the image of the absolute conic (IAC), and ​ are perpendicular lines in the image. You build a system of equations from multiple orthogonal line pairs and solve for the elements of . From , you recover the rectifying transformation using Cholesky decomposition.

Code Breakdown:

* – Used again to collect two pairs of perpendicular lines after affine rectification.
* :

1. Constructs a system where each row comes from the constraint .
2. Solves for using to rebuild the conic matrix:
3. Computes the rectification matrix such that using Cholesky.

* – Applies the metric rectification to the affinely rectified image.