# 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. Here is the original image, I call it “a piece of paper on the floor”:

A piece of paper on a tile floor

AI-generated content may be incorrect.

Figure 1 - a piece of paper on the floor. Lior Kotlar, 2025. Modern Realism.

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 the 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:**

* – 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 described above.
* – Applies the rectification using OpenCV's warpPerspective.

Here is the image after applied with projective rectification. I call it “a piece of paper on the floor – projectively rectified”:

A piece of paper on the floor

AI-generated content may be incorrect.

Figure 2 - a piece of paper on the floor – projectively rectified. Modern Projectively Rectified Realism.   
Lior Kotlar, 2025. Guggenheim Museum for contemporary arts.

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:

* – 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.

Here is the image after applied with metric rectification. I call it “a piece of paper on the floor – metricly rectified”:

A white square on a grey tile floor

AI-generated content may be incorrect.

Figure 3 - a piece of paper on the floor – projectively rectified. Modern Metricly Rectified Realism.  
Lior Kotlar, 2025. Louvre, Paris.

Task 3:

Theoretical background

In projective geometry, all different conic sections can be represented using a general second-degree equation with this form:

This equation can describe any conic. Which conic it is depends on the discriminant:

If : Ellipse/Circle, if : Parabola, If : Hyperbola.

A circle in 3D space, when viewed from different angles, looks like it experiences projective transformation. From certain viewpoints, the image plane can intersect the in such a way that the image of the circle appears not as a circle or ellipse, but as a hyperbola.

This typically happens when the camera is positioned sharply off-angle, such that part of the circle goes behind the image plane.

The images I worked on were two images of a cup:

1. Image 1 – cup plane is completely in front of the image plane:  
   The camera is placed such that the plane of the circular rim is entirely in front of the camera but is not fully visible.

A white trash can with a yellow stain

AI-generated content may be incorrect.

1. Image 2 – cup plane only partly in front of the image plane:  
   The cup is tilted away from the camera such that part of the circle lies behind the image plane.

A group of people sitting at a desk

AI-generated content may be incorrect.

According to projective geometry, Image 2 is expected to include a hyperbolic cone section.

To verify whether the observed shape in the image is a hyperbola, you implemented a method to:

1. The user selects points manually along the visible rim of the cup (at least 6).
2. Fit a conic of the form .
3. Solve for coefficients using SVD.
4. Check the discriminant to classify the conic.

The results are that in image 1 there is no hyperbola, and in image 2 there is a hyperbola.