Gradient-Based Planar Rectification under Circular Constraints

Filippo Balzarini[a], Paolo Ginefra[a], Martina Missana[a]

[a] *Politecnico di Milano*

June, 2025

1. Introduction

Single-view planar rectification is a fundamental task in computer vision that involves correcting perspective distortion in images. Given a perspective image of a planar surface, the objective is to recover a fronto-parallel view, as if the camera were oriented orthogonally to the plane. This transformation is essential for numerous applications such as architectural measurement, document digitization, augmented reality, and automated visual inspection.
Accurate rectification facilitates precise metric measurements from a single image, simplifies feature extraction, and serves as a key preprocessing step for high-level vision tasks such as object recognition, 3D reconstruction, and camera pose estimation. The core challenge lies in computing the correct homography from image features alone, without access to explicit 3D information.

When circular features are present in the scene, they offer strong geometric constraints that can be exploited for rectification. However, existing methods leveraging circular constraints—such as those based on the projection of circular points—often suffer from numerical instability and sensitivity to noise.

In our investigation, we explored multiple approaches to circle-based planar rectification, including algebraic intersection techniques and homotopy continuation methods. Homotopy continuation, in particular, offers a principled way to compute all solutions to a polynomial system by tracking solution paths as a parameter is varied. Despite its theoretical appeal, we found that this method cannot be properly applied to the analyzed context.

In contrast, we demonstrate that a gradient descent-based solution for the rectification problem yields significantly better results. Our approach achieves higher robustness, faster convergence, and greater numerical stability by directly minimizing a function derived from circular constraints. This makes it well-suited for real-world settings where noise and limited resolution are common. Empirical results confirm that gradient descent consistently outperforms homotopy continuation and other algebraic solvers in accuracy and reliability.

2. Problem Formulation

Given an input image $\mathcal{I}$, we aim to compute a *rectifying homography* $H \in \mathbb{R}^{3\times 3}$ such that the projective distortion induced by the camera perspective is removed, restoring a frontal-parallel view of the plane up to a similarity transformation.

The rectification is guided by *prior knowledge* in the form of three non-degenerate real ellipses, $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, which are known to be *images of Euclidean circles* in the scene.

Let each conic $\mathcal{C}_i$ be represented by a symmetric 3×3 matrix C_i (up to scale) in the projective plane, as defined in Equation 1

$$\mathcal{C}_i = \left\{ \mathbf{x} \in \mathbb{P}^2 \,\middle|\, \mathbf{x}^\top C_i \mathbf{x} = 0 \right\}, \quad i = 1, 2, 3. \tag{1}$$

Equation 2 encodes the additional assumption that the conics C_i are real, non-degenerate ellipses.

$$\det(C_i) > 0, \quad i = 1, 2, 3. \tag{2}$$

Let H be the homography such that, for a point $\mathbf{x} \in \mathbb{P}^2$ in the image, the rectified coordinate is $\mathbf{x}' = H\mathbf{x}$. The homography H must satisfy the constraint that it maps the observed conics C_i to circles in the rectified space. This translates to the constraint represented in Equation 3

$$H^{-\top} C_i H^{-1} \in \mathcal{S}_{\text{circ}}, \quad i = 1, 2, 3, \tag{3}$$

where $\mathcal{S}_{\text{circ}}$ is the set of symmetric matrices representing circular conics.

To quantify the accuracy of the estimated homography H a performance index $\mathcal{J}$ is needed. A *performance index* is a scalar-valued function used to assess the quality of an estimated model with respect to some desired geometric or statistical property. In the context of planar rectification, the performance index could be defined as a non-negative function as stated by Equation 4.

$$\mathcal{J}(H) : \mathbb{R}^{3\times 3} \to \mathbb{R}_{\geq 0}, \tag{4}$$

This measures the extent to which a given homography H satisfies the Euclidean constraints expected in the rectified image.

Since, in real scenarios, the true rectifying homography H^* is not known a priori, our choice should rely on a performance index which is not computed by direct comparison with H^*.

Objective. Estimate the homography $H \in \mathbb{R}^{3\times 3}$ such that H maps the conics C_1, C_2, C_3 onto circular conics in the rectified (Euclidean) domain.

Solution Outline. The estimation of the rectifying homography H proceeds through the following conceptual steps:

1. **Conic Detection.** Identify and represent the three given conics $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ in the image domain as symmetric 3×3 matrices C_i, under the assumption that they are non-degenerate ellipses corresponding to Euclidean circles in the scene.
2. **Homography Estimation.** Determine a homography matrix $H \in \mathbb{R}^{3\times 3}$ such that each transformed conic $H^{-\top} C_i H^{-1}$ is mapped onto a circular conic in the rectified space, thereby satisfying the constraints defined in Equation 3.
3. **Performance Evaluation.** Evaluate the quality of the estimated homography H using a performance index $\mathcal{J}(H)$ that reflects how well the Euclidean properties of the scene are restored in the rectified image.

3. Background

Approaches to planar rectification using circular constraints can be broadly divided into two main categories:
The first category involves estimating the image of the circular points, which enables the recovery of the dual conic of circular points $\mathbf{C}^*_\infty$. These methods have been extensively studied in the computer vision literature and remain among the most widely used techniques for rectifying images under circular constraints.
The second category comprises methods that directly estimate the rectifying homography H, without first computing the image of the circular points I^* and J^*.

In this section, we explore the main techniques, grouped according to these two categories.

3.1. Planar Rectification

In projective geometry, metric rectification of a plane consists of recovering the Euclidean structure from a projective image. A key invariant in this process is the *dual conic of circular points*, introduced in Definition 1.

Definition 1 (Dual Conic of Circular Points). *The* dual conic of circular points*, denoted as* $\mathbf{C}^*_\infty$*, is a fundamental geometric entity in projective geometry used for metric rectification. It represents the set of lines that are tangent to all circles in the Euclidean plane and encodes both the orthogonality and isotropy conditions intrinsic to Euclidean geometry.*

In homogeneous coordinates, $\mathbf{C}^*_\infty$ *is defined as:*

$$\mathbf{C}^*_\infty = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The conic $\mathbf{C}^*_\infty$ is dual to the circular points at infinity I and J, defined as Definition 2.

Definition 2 (Circular Points). *The* circular points *are two special points at infinity in the projective plane that lie on every circle in Euclidean space. In the projective space they are defined in homogeneous coordinates as:*

$$I = \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix}, \quad J = \begin{bmatrix} 1 \\ -i \\ 0 \end{bmatrix}$$

where i is the imaginary unit. These points are invariant under Euclidean transformations and are fundamental in imposing circular constraints for metric rectification.

Under a projective transformation $\mathbf{H}$ (mapping the Euclidean plane to the image), the image of the dual conic becomes:

$$\mathbf{C}^{*'}_\infty = \mathbf{H}^{-\top}\, \mathbf{C}^*_\infty\, \mathbf{H}^{-1}.$$

Recovering this conic in the image allows one to compute a metric rectifying homography $\mathbf{H}$.

3.1.1. Recovering $\mathbf{C}^{*'}_\infty$

If the images of the circular points I and J can be directly located or estimated in the image (for example through the image of known circles), $\mathbf{C}^{*'}_\infty$ can be constructed explicitly.

Let $\mathbf{i}$ and $\mathbf{j}$ be the image points corresponding to I and J. Then, the dual conic can be recovered as the symmetric matrix representing the conic that passes through these two complex conjugate points:

$$\mathbf{C}^{*'}_\infty = \mathbf{I}\mathbf{J}^\top + \mathbf{J}\mathbf{I}^\top.$$

This conic will be rank-2 and real-valued, and it represents the locus of all points satisfying the Euclidean circular constraint in the image.

This method is especially powerful when a true circle in the scene is visible in the image; the image of such a circle is a conic that intersects the vanishing line of the scene in two points, precisely the images of I and J.

3.1.2. Recovering the Rectifying Homography
Given the conic $\mathbf{C}_\infty^{*'}$, the goal is to find a homography $\mathbf{H}$ such that:

$$\mathbf{C}_\infty^{*'} = \mathbf{H}^{-\top}\,\mathbf{C}_\infty^{*}\,\mathbf{H}^{-1}.$$

This is a conic normalization problem. One approach is:

1. Perform Singular Value Decomposition (SVD) of $\mathbf{C}_\infty^{*'}$:
$$\mathbf{C}_\infty^{*'} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^\top,$$
where $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, 0)$ contains the singular values.
2. Construct the homography matrix H as:
$$\mathbf{H} = \begin{bmatrix} \frac{1}{\sqrt{\sigma_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{\sigma_2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{U}^\top$$

3.2. Homotopy continuation
Homotopy continuation is a method for solving problems, not necessarily polynomial, by solving a sequence of easier problems [6]. The underlying approach is to:

1. put the problem we are solving into a family of problems depending on parameters
2. solve the problem for some appropriate points in the parameter space
3. track the solutions as the parameter varies until the original problem we want to solve is reached

The original problem we want to solve can be formally defined as follows, involving a finite set of parameters p and n unknown variables x:

Definition 3 (Original Problem). *Let $p \in \mathbb{C}^m$ be a vector of parameters and $x \in \mathbb{C}^n$ a vector of variables. The original problem consists in solving a system of n polynomial equations in n variables, parameterized by p, defined as:*

$$\mathcal{F}_p(x) = \begin{cases} f_p^1(x) = 0 \\ \vdots \\ f_p^n(x) = 0 \end{cases}$$
$$\text{with } f_p^i \in \mathbb{C}[p, x],\ i = 1, \ldots, n.$$

Here, $\mathbb{C}$ denotes the field of complex numbers, and $\mathbb{C}[p, x]$ is the ring of multivariate polynomials with complex coefficients in the parameters $p = (p_1, \ldots, p_m)$ and variables $x = (x_1, \ldots, x_n)$.

The function $\mathcal{F}_p(x) \colon \mathbb{C}^n \to \mathbb{C}^n$ is a vector-valued function defined by:

$$\mathcal{F}_p(x) = \begin{bmatrix} f_p^1(x) \\ \vdots \\ f_p^n(x) \end{bmatrix},$$

where each $f_p^i(x) \in \mathbb{C}[p, x]$. The goal is to compute all solutions $x \in \mathbb{C}^n$ such that $\mathcal{F}_p(x) = 0$.

Definition 4 (Linear segment homotopy). *Given two points p_0, p_1 in the base space $\mathbb{B}$, a linear segment homotopy is a family of systems of the form:*

$$\mathcal{H}(t) = (1-t)\mathcal{F}_{p_0} + t\mathcal{F}_{p_1} \tag{5}$$

If p_0 and p_1 are chosen in such a way that the system $\mathcal{H}(t)$ has a solution for all $t \in [0, 1]$, then we say that the homotopy is *regular* and it describes a path in the base space $\mathbb{B}$ from p_0 to p_1. [3]

Since γF_p for $\gamma \in \mathbb{C}\setminus\{0\}$ has the same solutions of F_p we can also define a homotopy as written in Equation 6.

$$\mathcal{H}'(t) = (1-t)\gamma_1\mathcal{F}_{p_0} + t\gamma_2\mathcal{F}_{p_1} \tag{6}$$

3.3. Estimation of the Circular Points
In this section, we explore several approaches for estimating the image of the circular points, assuming three conics corresponding to non-degenerate ellipses are given.

Section 3.3.1 presents the classical symbolic approach based on solving a system of homogeneous polynomial equations. Section 3.3.2 describes a well-known conic intersection method, as discussed in [5]. Finally, Section 3.3.3 introduces the use of homotopy continuation techniques for computing the intersection points.

3.3.1. Symbolic computation of the circular points
A classical approach to recover the image of the circular points involves computing the intersection points of three conics in the projective plane using symbolic algebra. Each conic is represented by a homogeneous quadratic equation in the projective coordinates $(x : y : w)$.

Let the conics $\mathcal{C}_i$ with $i \in \{1, 2, 3\}$ be defined by their algebraic forms as shown in Equation 7

$$\begin{aligned}\mathcal{C}_i(x, y, w) = a_i x^2 + b_i xy + c_i y^2 \\ +\ d_i xw + e_i yw + f_i w^2 = 0\end{aligned} \tag{7}$$

The image of the circular points, denoted I^* and J^*, lies at the intersection of these three conics, assuming that the conics are constructed or selected to encode circular constraints (e.g., from circular objects in the scene).

To compute the intersection points, we solve the system of three quadratic equations shown in Equation 8.

$$\begin{cases} \mathcal{C}_1(x, y, w) = 0 \\ \mathcal{C}_2(x, y, w) = 0 \\ \mathcal{C}_3(x, y, w) = 0 \end{cases} \tag{8}$$

The symbolic solution yields a finite set of candidate points $(x : y : w) \in \mathbb{P}^2$, including both real and complex intersections. Since circular points are complex and lie at infinity, one filters the results to exclude real and finite solutions, thereby isolating I^* and J^*.

In the strong assumption of totally noise-free conics, this method provides a direct and exact way to extract geometric invariants from conic arrangements without relying on numerical approximations or optimization.

3.3.2. Algebraic conic intersection for circular point estimation [5]

Let C_A and C_B be two conics in the projective plane represented by symmetric matrices $A, B \in M_{3\times3}(\mathbb{R})$ as in Equation.3.3.2

$$\begin{aligned} C_A &= p \in \mathbb{P}^2 \mid p^T A p = 0 \\ \mathrm{C}_B &= p \in \mathbb{P}^2 \mid p^T B p = 0 \end{aligned} \tag{9}$$

The intersection $C_A \cap C_B$ generically consists of four points, corresponding to the solutions of the system shown in Equation 10

$$\begin{cases} p^T A p = 0 \\ p^T B p = 0 \end{cases} \tag{10}$$

Direct solution of this system requires solving a quartic polynomial system, which presents significant computational challenges. The method under consideration circumvents this difficulty through exploitation of the linear pencil structure.

Pencil of Conics and Degenerate Members. The linear pencil generated by A and B is defined as:

$$\mathcal{P}(A, B) = \lambda A + \mu B : [\lambda : \mu] \in \mathbb{P}^1 \tag{11}$$

Each member of this pencil passes through the four intersection points of C_A and C_B. The degenerate members of the pencil satisfy:

$$\det(\lambda A + \mu B) = 0 \tag{12}$$

Expanding this determinant using multilinearity yields the homogeneous cubic polynomial:

$$F(\lambda, \mu) = \alpha\lambda^3 + \beta\lambda^2\mu + \gamma\lambda\mu^2 + \delta\mu^3 = 0 \tag{13}$$

The coefficients are computed using the multilinearity of the determinant function. If $A = [A_1 \mid A_2 \mid A_3]$ and $B = [B_1 \mid B_2 \mid B_3]$ represent column decompositions, then:

$$\begin{aligned} \alpha &= [A_1, A_2, A_3] \\ \beta &= [A_1, A_2, B_3] + [A_1, B_2, A_3] + [B_1, A_2, A_3] \\ \gamma &= [A_1, B_2, B_3] + [B_1, A_2, B_3] + [B_1, B_2, A_3] \\ \delta &= [B_1, B_2, B_3] \end{aligned}$$

where $[u, v, w]$ denotes the determinant of the matrix with columns u, v, and w.

Decomposition of Degenerate Conics. For a solution (λ_0, μ_0) of the cubic equation, the matrix $C = \lambda_0 A + \mu_0 B$ represents a degenerate conic. When C has rank 2, it represents the union of two distinct lines with homogeneous coordinates $g, h \in \mathbb{R}^3$, expressible as:

$$C = gh^T + hg^T \tag{14}$$

The decomposition algorithm exploits the fundamental relationship:

$$C^* = -(g \times h)(g \times h)^T \tag{15}$$

where C denotes the adjugate matrix of C. Since C is a rank-1 matrix of the form pp^T with $p = g \times h$, the intersection point can be extracted as:

$$p_i = \frac{C_i}{\sqrt{C_{ii}}}$$

for any index i where $C^*_{ii} \neq 0$. The line coordinates are subsequently recovered through the rank-1 decomposition:

$$C \pm M_p = 2gh^T \text{ or } 2hg^T \tag{16}$$

where M_p is the skew-symmetric matrix satisfying $M_p v = p \times v$ for any vector v.

Line-Conic Intersection Algorithm. Given a line l and conic C_A, the intersection points are determined by constructing the dual conic matrix:

$$B = M_l^T A M_l \tag{17}$$

where M_l is the skew-symmetric matrix corresponding to l. The quadratic form $m^T Bm$ vanishes precisely for lines m passing through an intersection point. The intersection points are extracted by identifying a parameter α such that $B + \alpha M_l$ has rank 1. This parameter is determined as shown in Equation 18

$$\alpha = \frac{1}{\tau}\sqrt{-\det\begin{pmatrix} B_{11} & B_{12} \\ B_{12} & B_{22} \end{pmatrix}} \tag{18}$$

where τ represents a non-zero coordinate of l. The resulting rank-1 matrix yields the intersection coordinates through standard matrix decomposition techniques.

Complete Two-Conic Intersection Algorithm. The complete algorithm for intersecting two conics C_A and C_B synthesizes the previously described components into a systematic computational procedure. The method proceeds through three phases:

1. cubic equation solution
2. degenerate conic decomposition
3. line-conic intersection.

Once a solution (λ_0, μ_0) is identified by solving $F(\lambda, \mu) = 0$, the degenerate conic shown in Equation 19 is decomposed into its constituent lines g and h using the adjugate matrix method explained above.

$$C = \lambda_0 A + \mu_0 B \tag{19}$$

Then each line is intersected with one of the original conics C_A through the construction of the dual conic, producing two intersection points per line and thus recovering the four intersection points of the original conic pair.

These solutions involve the image points I^* and J^*, which correspond to complex circular points and can be isolated by removing the real intersections.

3.3.3. Homotopy Continuation for Circular Point Estimation

Homotopy continuation (HC) offers a powerful alternative to classical symbolic approaches for estimating the image of the circular points. Once the estimation problem is formulated as a system of polynomial equations—such as those derived from the intersection of three conics—HC can be employed to numerically track solution paths from a known start system to the target system of interest.

Unlike symbolic methods, which often involve computationally expensive algebraic manipulations and exact solving, HC enables the exploration of solution spaces in a scalable and robust manner. This flexibility allows for a broader range of formulations: essentially any valid system encoding the circular constraints can be used as input, without the need to construct a symbolically tractable representation.

In this work, we leverage HC to compute the intersection points of conics by solving the corresponding polynomial system numerically, thereby recovering the complex circular points at infinity I^* and J^* without relying on symbolic elimination.

3.4. Gradient Descent [7]

Gradient Descent is a widely used optimization algorithm, aimed at minimizing a cost or objective function. Given a function $f(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the parameters to optimize, the algorithm updates $\boldsymbol{\theta}$ iteratively by moving in the direction opposite to the gradient of f with respect to $\boldsymbol{\theta}$. The update rule is expressed as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla f(\boldsymbol{\theta}_t),$$

where η is the learning rate, a small positive constant that determines the step size, and $\nabla f(\boldsymbol{\theta}_t)$ is the gradient of f evaluated at $\boldsymbol{\theta}_t$. This gradient is a vector containing the partial derivatives of f with respect to each component of $\boldsymbol{\theta}$:

$$\nabla f(\boldsymbol{\theta}) = \left[\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \ldots, \frac{\partial f}{\partial \theta_n}\right]^T.$$

By repeating this process, the algorithm reduces the function value $f(\boldsymbol{\theta})$ over time, ideally converging to a minimum point. In the case of convex functions, gradient descent is guaranteed to find the global minimum. However, for non-convex functions, it may settle at a local minimum or a saddle point instead.

Although simple and effective, standard gradient descent can suffer from slow convergence and sensitivity to the choice of learning rate, especially in high-dimensional and sparse data settings.

Adam Optimizer [4]

To address these limitations, more advanced optimization algorithms such as **Adam** (Adaptive Moment Estimation) have been developed. Adam combines the benefits of two other popular extensions of gradient descent: *Momentum* and *RMSProp*.

Adam maintains two moving averages at each iteration: the first moment (mean of gradients) and the second moment (uncentered variance of gradients). The update rules are given by:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\nabla f(\boldsymbol{\theta}_t),$$
$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\left(\nabla f(\boldsymbol{\theta}_t)\right)^2,$$

where $\beta_1, \beta_2 \in [0,1)$ are decay rates for the moving averages, and operations are applied element-wise. To correct bias introduced during initialization, bias-corrected estimates are computed:

$$\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t},$$
$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t}.$$

The parameters are then updated as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon},$$

where ϵ is a small constant to avoid division by zero.

Advantages of Adam

- Straightforward to implement
- Little memory requirements with respect to second-order methods, while maintaining good performance.
- Hyper-parameters have intuitive interpretation and typically require little tuning, usually $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

3.5. Loss

Since multiple loss functions were investigated to assess the quality of the rectification, the following section provides the theoretical background for each of them. In addition, every loss is classified using these discriminative features

- **Needs the true homography matrix:** [Yes/No]
- **Invariance under similarities:** [Yes/No]
- **Intrinsic or Extrinsic:** Does the loss quantify the quality of the rectification matrix itself, or does it measure the accuracy of specific features resulting from the rectification?

3.5.1. Angle Distortion Loss

Property	Value
Needs the true homography matrix	Yes
Invariance under similarities	Yes
Extrinsic/Intrinsic	E

Table 1: Summary of angle distorsion loss properties

This loss function is designed to measure distortion in the geometry of lines under the homography application.

This loss captures how the computed homography distorts the angle between originally orthogonal lines in homogeneous coordinates. The computation takes as input two lines, l and m, that are known to be orthogonal in the original image. Define the distance metric matrix C as:

$$C = H_{\text{true}} \cdot H_{\text{comp}}^{-1} \cdot C_\infty^* \cdot (H_{\text{comp}}^{-1})^\top \cdot H_{\text{true}}^\top$$

Define a distance metric (inner product) using C:

$$\langle l, m \rangle_C = l^\top C m$$

In this formulation, the orthogonal lines are first transformed by H_{true} to obtain their images in the target view. They are then rectified using H_{comp}^{-1}, allowing the measurement of the angle between them.

Then, the loss is computed as the angle (in degrees) between l and m under the metric C:

$$\text{Loss} = \cos^{-1}\left(\frac{\langle l, m \rangle_C}{\sqrt{\langle l, l \rangle_C \cdot \langle m, m \rangle_C}}\right) \cdot \frac{180}{\pi}$$

A lower loss value generally indicates a more accurate rectification. This angle-based loss function offers a geometrically interpretable way to assess the accuracy of a computed homography.

3.5.2. Frobenius Norm Loss

Property	Value
Needs the true homography matrix	No
Invariance under similarities	No
Extrinsic/Intrinsic	I

Table 2: Summary of Frobenius Norm Loss loss properties

Definition 5. *The Frobenius norm is a type of matrix norm defined as the square root of the sum of squares of the matrix elements.*

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2} \tag{20}$$

By computing the Frobenius norm of the difference between the computed homography and the true one, we obtain a quantitative measure of the discrepancy between the estimated and ground-truth transformations. Specifically, the error can be expressed as:

$$\epsilon = |H_{\text{true}} - H_{\text{comp}}|_F \tag{21}$$

This metric provides an intuitive sense of accuracy: the smaller the Frobenius norm of the difference, the closer the estimated homography is to the true one in terms of overall matrix elements. The main drawback of this approach is that the Frobenius norm is not invariant under similarity transformations and is sensitive to noise. In particular, when noise is introduced into the matrix, the Frobenius norm tends to emphasize the variance of the noise rather than capturing meaningful differences.

3.5.3. Lines at Infinity Loss

Property	**Value**
Needs the true homography matrix	No
Invariance under similarities	Yes
Extrinsic/Intrinsic	E

Table 3: Summary of the line at infinity loss properties

Definition 6 (Line at Infinity). *The line at infinity is defined as the locus of all points at infinity. In homogeneous coordinates, points at infinity are characterized by a zero in the last coordinate:*

$$w = 0 \tag{22}$$

A generic line in homogeneous coordinates is represented as:

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = 0 \tag{23}$$

For the line at infinity, this becomes:

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = w = 0 \tag{24}$$

Thus, the line at infinity is represented by the vector:

$$l_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{25}$$

To evaluate the quality of the rectification, we compare the line at infinity in the rectified image with that in the ground-truth geometry. The loss is computed by measuring the angular difference between the two lines, providing a geometric measure of how well is the rectification.

This loss has the drawback of taking into account just the affine reconstruction and not the metric one, so it is not suitable in this context.

3.5.4. Reconstruction Error Loss

Property	**Value**
Needs the true homography matrix	Yes
Invariance under similarities	No
Extrinsic/Intrinsic	E

Table 4: Summary of Reconstruction Error loss properties

This loss quantifies the difference in how two homographies (the computed one and the true one) transform a given set of points.

More specifically, given the ground-truth homography H_{true} and an estimated homography H_{comp}, as well as a set of points $\mathbf{p} \in \mathbb{R}^{3\times N}$ in homogeneous coordinates, the loss is computed as follows:

1. Both homographies are applied to the input points:

$$\mathbf{p}_{\text{true}} = H_{\text{true}} \cdot \mathbf{p}, \quad \mathbf{p}_{\text{comp}} = H_{\text{comp}} \cdot \mathbf{p}$$

2. The reconstruction error is defined as the average Euclidean distance between the corresponding transformed points:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{p}_{\text{true},i} - \mathbf{p}_{\text{comp},i} \right\|_2$$

   where $\| \cdot \|_2$ denotes the L^2 norm, and the points are normalized appropriately to account for homogeneous coordinates.

This loss function effectively measures the discrepancy between the two homographies by comparing their action on a set of points, but it has the drawback of not being invariant under similarities.

3.5.5. Circular Points Loss

Property	**Value**
Needs the true homography matrix	No
Invariance under similarities	Yes
Extrinsic/Intrinsic	E

Table 5: Summary of Circular Points loss properties

This loss is designed to quantify the discrepancy between two sets of circular points in homogeneous coordinates (the true ones in the image and the computed ones).

Given these two sets:

- $\mathbf{CP}_{\text{true}} = \left\{ \mathbf{CP}_1^{\text{true}}, \mathbf{CP}_2^{\text{true}} \right\}$
- $\mathbf{CP}_{\text{comp}} = \left\{ \mathbf{CP}_1^{\text{comp}}, \mathbf{CP}_2^{\text{comp}} \right\}$

the loss is computed with the euclidean distances between corresponding normalized points:

$$d_i = \left\| \hat{\mathbf{CP}}_i^{\text{true}} - \hat{\mathbf{CP}}_i^{\text{comp}} \right\|_2 .$$

The maximum of these distances over all points is recorded as the loss. This loss effectively measures the worst-case (maximum) deviation between corresponding circular points after normalization, ensuring robustness to permutations and scale.

3.5.6. Eccentricity Loss

Property	**Value**
Needs the true homography matrix	No
Invariance under similarities	Yes
Extrinsic/Intrinsic	E

Table 6: Summary of Eccentricity loss properties

Definition 7 (Eccentricity). *The eccentricity of a conic section is a non-negative real number that uniquely characterizes its shape. It is calculated as*

$$e = \sqrt{1 - \frac{b^2}{a^2}} \tag{26}$$

where a is the length of the semi-major axis and b is the length of the semi-minor axis.

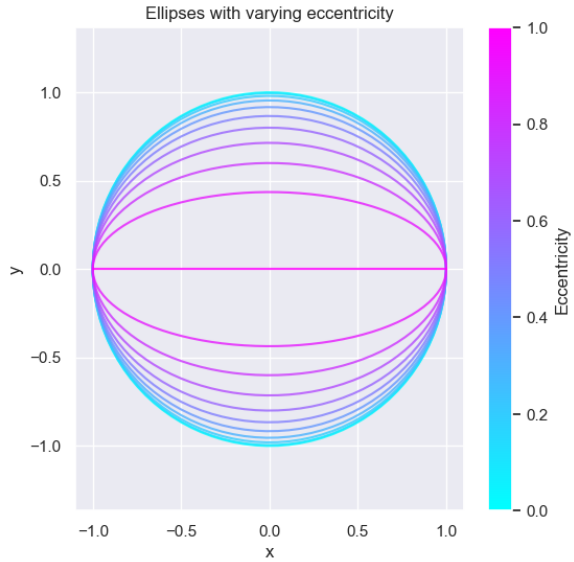
The eccentricity is related to the various conic sections as follows:

Conic section	**Eccentricity** (e)
Circle	0
Ellipse	$\sqrt{1 - \frac{b^2}{a^2}}$
Parabola	1
Hyperbola	$\sqrt{1 + \frac{b^2}{a^2}}$

Table 7: Conic sections and their eccentricities

By computing the eccentricities of the reconstructed conics, it is than possible to quantify how much they deviate from perfect circles.

A key characteristic of this loss is its strong nonlinearity. In the image, ten ellipses are plotted with eccentricities linearly increasing in steps of 0.1. It is evident that, up to an eccentricity of 0.4, the ellipses appear almost indistinguishable from a circle. However, in the same range interval, from 0.4 to 0.8, the difference becomes more pronounced.



3.5.7. Gradient Descent Losses

Property	**Value**
Needs the true homography matrix	No
Invariance under similarities	Yes
Extrinsic/Intrinsic	E

Table 8: Summary of GD losses properties

The loss used in the gradient descent optimization rely on the lenght comparison of the ellipse axes. Given the conic matrix

$$C = \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix},$$

we consider the upper-left 2×2 submatrix

$$Q = \begin{bmatrix} a & \frac{b}{2} \\ \frac{b}{2} & c \end{bmatrix}.$$

The eigenvalues of Q correspond to the *squared reciprocals of the lengths of the principal axes* of the conic, up to scale. That is, if λ_1 and λ_2 are the eigenvalues of Q, then the lengths of the axes are $\frac{1}{\sqrt{\lambda_1}}$ and $\frac{1}{\sqrt{\lambda_2}}$, respectively. In this way it is possible to compare the length of the axes (that are equal for a circle) equally comparing the eigenvalues of Q, thereby eliminating the nonlinear step of taking square roots.

The loss is then defined as

$$\text{Loss} = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2$$

The loss is computed for the three conics and a softmax is used to smoothly aggregate them during optimization.

4. Method

4.1. Algebraic Approach implementation

The Algebraic Rectifier is implemented as a numerical pipeline that adheres closely to the theory of conic intersection via the pencil method explained in Section 3.3.2. Its objective is to recover the images of the circular points by computing the intersection points among three image conics.

The core idea is to exploit pairwise conic intersections to obtain complex solutions that correspond to these fundamental geometric loci. In practice, the intersection of each conic pair is resolved following the algebraic procedure described in Section 3.3.2, including the identification of degenerate conics and their decomposition into intersecting lines.

The algorithm computes three sets of intersection points, one for each pair of the three input conics. The resulting points are aggregated and then post-processed to remove duplicates and normalize their representation. Once the image of the circular points is estimated, the Dual Conic of the Circular Points ($C_\infty^{*'}$) and the rectifying homography are reconstructed as explained in Section 3.1.1 Section 3.1.2.

The pseudo-code of the implementation is represented below.

```
function ComputeIofCP(C_1, C_2, C_3)
    r_1 ← Intersect(C_1, C_2)
    r_2 ← Intersect(C_1, C_3)
    r_3 ← Intersect(C_2, C_3)
    intPt ← Concatenate(r_1, r_2, r_3)
    Normalize(intPt)
    Filter(intPt)
    return int_pt
end function
```

4.2. Homotopy Continuation

To apply the homotopy continuation approach described in 3.3.3 to the rectification problem different techniques have been tried.

The problem of finding the solutions to a system (in this case the circular points) can be approached with different techniques:

- **Direct method:** Standard method, this approach constructs a homotopy $H(x,t) = (1-t)G(x) + tF(x)$ between a known start system $G(x)$ and the target system $F(x)$, and tracks solutions as t goes from 0 to 1. It uses predictor-corrector schemes to follow all solution paths and is suitable for solving polynomial systems. It is robust and guarantees finding all isolated solutions for generic systems.

  Graphically

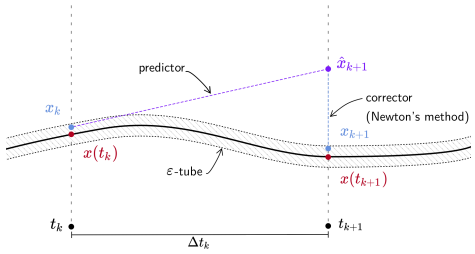


Figure 1: Homotopy Continuation in Julia

- **Optimization method:** This method embeds the homotopy into an optimization framework by minimizing $\|H(x,t)\|^2$ for varying t. It is more flexible, applicable to general nonlinear systems (not just polynomials), and supports noisy or approximate models. It is useful when exact path tracking is unnecessary, but good approximate solutions are desired.

and the problem can be formulated in:

- **Ellipse space:** Representing the ellipses using:
  - a: semi-major axis length
  - b: semi-minor axis length
  - x_0, y_0: center of the ellipse
  - $\cos\theta, \sin\theta$: cosine and sine of the ellipse rotation angle θ (the orientation of the ellipse's axes with respect to the coordinate axes)
- **Conic matrix space:** Representing conics with their matrix, in particular a conic with algebraic form given by

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (27)$$

  is represented as the matrix

$$C = \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix}$$

Since the goal is to rectify real images, where noise is typically present, experiments show that the optimization-based approach in the conic matrix space yields the best results. In the following section the usage of this approach is described.

4.2.1. Homotopy Continuation approach

In order to deal with noise, we aim to find the point z that best fits the system of equations without necessarily being an exact solution.

Consider a single conic represented by the symmetric matrix $C \in \mathbb{R}^{3\times 3}$. We seek a point $z^* \in \mathbb{C}^3$ such that:

$$z^* = \arg\min_{z\in\mathbb{C}^3} \|z^T C z\|_2^2.$$

The squared L_2 norm of a generic complex vector $t = [t_1, t_2, t_3]^T$, where $t_i = a_i + ib_i$, $a_i, b_i \in \mathbb{R}$, is defined as:

$$\|t\|_2^2 = \sum_{i=1}^{3} |t_i|^2 = \sum_{i=1}^{3} (a_i^2 + b_i^2).$$

Since we work in homogeneous coordinates, a constraint must be added to handle the scale ambiguity of the vector z. We choose the constraint

$$g(z) = \|z\|_2^2 - 1 = 0.$$

The problem becomes:

$$z^* = \arg\min_{z\in\mathbb{C}^3} \|z^T C z\|_2^2 \quad \text{s.t.} \quad g(z) = 0.$$

Unfortunately, the `HomotopyContinuation.jl` package does not allow access to the real and imaginary parts of complex variables directly. Therefore, we convert the problem into a real optimization problem.

Express the complex vector z as $z = x+iy$ with $x, y \in \mathbb{R}^3$. The objective function can be rewritten as:

$$\begin{aligned}\|z^T C z\|_2^2 &= \|(x+iy)^T C (x+iy)\|_2^2 \\ &= \|x^T C x - y^T C y + i(y^T C x + x^T C y)\|_2^2 \\ &= \|x^T C x - y^T C y\|_2^2 + \|2x^T C y\|_2^2.\end{aligned}$$

The constraint becomes:

$$\begin{aligned}g(z) &= \|z\|_2^2 - 1 \\ &= \|x+iy\|_2^2 - 1 \\ &= \|x\|_2^2 + \|y\|_2^2 - 1 = 0.\end{aligned}$$

Extending this to three conics C_1, C_2, C_3, the problem becomes:

$$\begin{aligned}z^* = \arg\min_{x,y\in\mathbb{R}^3} \quad & \sum_{i=1}^{3} \left\| x^T C_i x - y^T C_i y \right\|^2 \\ & + \sum_{i=1}^{3} \left\| 2x^T C_i y \right\|^2 \\ \text{s. t} \quad & g(z) = 0.\end{aligned}$$

To solve this constrained optimization problem, we use the method of Lagrange multipliers. The Lagrangian function is defined as:

$$\begin{aligned}\mathcal{L}(x, y, \lambda) = \sum_{i=1}^{3} \left(\left\| x^T C_i x - y^T C_i y \right\|_2^2 + \left\| 2x^T C_i y \right\|_2^2 \right) \\ + \lambda g(z)\end{aligned}$$

where λ is the Lagrange multiplier. This function combines the objective and the constraint by penalizing any violation of $g(z) = 0$.

The optimal solution is found by solving the system:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x} = 0, \\ \frac{\partial \mathcal{L}}{\partial y} = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda} = 0. \end{cases}$$

This system is polynomial and can be solved using the `HomotopyContinuation.jl` package [2].

Note that the solutions obtained are stationary points of the Lagrangian and thus require evaluation to identify the global minimum.

4.3. Gradient Descent

The rectification problem is addressed in this method using a gradient descent approach based on the Adam optimization algorithm, as detailed in Section 3.4. Since the rectified image is obtained by applying the inverse of the true homography, this approach focuses on estimating the inverse homography matrix. In addition, it tries to estimate it directly, without relying on the recovery of the circular points as done in the other methods. The update rule applying Adam to this problem becomes

$$\mathbf{H}_{\text{inv}} \leftarrow \mathbf{H}_{\text{inv}} - \frac{\alpha\, \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

where $\hat{m}_t$ and $\hat{v}_t$ are computed as in Section 3.4. The loss is set to the one detailed in Section 3.5.7.

To compute the gradients, automatic differentiation was employed instead of symbolic or numerical methods, enabling more efficient gradient evaluation, particularly in complex optimization scenarios such as this.

5. Experiments

5.1. Experiment Setup

To test and validate the usage of the various rectification methods implemented for homography estimation and image rectification, a simulator has been developed. The overall pipeline of the simulator is the following:

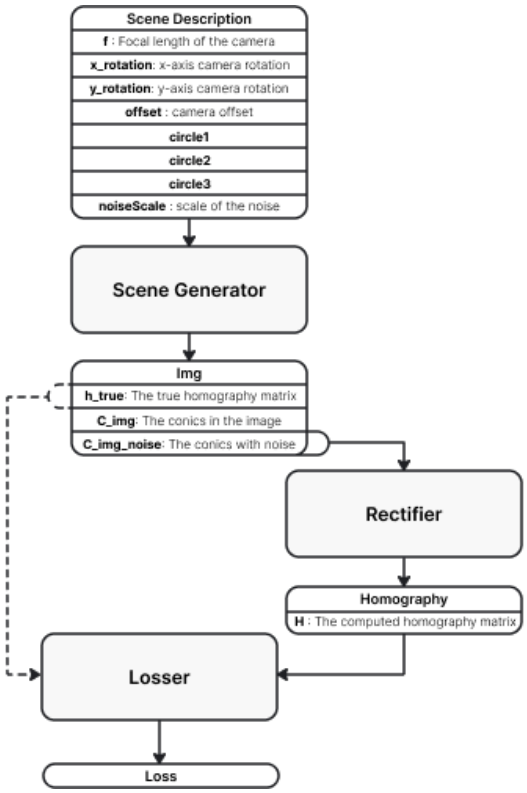


Figure 2: Experiment pipeline

5.1.1. Scene Generation

Given the description of a plane containing three circles and its position in space, the Scene Generator generates the warped image of the scene (which will then require rectification) and the true homography matrix that represents the ground truth for the rectification step. The scene generator works in the camera reference frame and relies on a scene description that includes intrinsic and extrinsic parameters of the camera, as well as geometric properties of the circles. Specifically, the input parameters are:

- **Focal length** f: the camera's internal calibration
- **Rotation angle** $\boldsymbol{\theta_x}$: plane rotation angle (in degrees) around the X-axis
- **Rotation angle** $\boldsymbol{\theta_y}$: plane rotation angle (in degrees) around the Y-axis
- **Plane offset** $\boldsymbol{o_\pi}$: the plane offset vector, which defines the position of the plane with respect to the camera
- **Three circles**: each characterized by its center (a 2D point on the plane) and its radius
- **Noise scale parameter**: random perturbations to the conic parameters (optional)
- **Distortion parameters** $(\boldsymbol{k_1}, \boldsymbol{k_2}, \boldsymbol{k_3}, \boldsymbol{p_1}, \boldsymbol{p_2})$: to simulate lens distortions including radial and tangential effects (optional)

Using these inputs, the simulator converts the circle definitions into conic sections, computes the homography matrix from the scene configuration, applies the homography to project the circles into the image plane, and optionally applies distortion.

The output consists of the warped conics, which represent the observed (potentially distorted and noisy) image, and the true homography matrix, which serves as the target for evaluating the accuracy of rectification algorithms. This scene generation framework thus provides a robust and configurable way to generate synthetic data for testing conic-based homography estimation methods under various conditions.

5.1.2. Rectifier

In this stage of the pipeline, the rectifying homography is computed from the generated image. Any of the rectification methods outlined in Section 4 can be employed for this purpose.

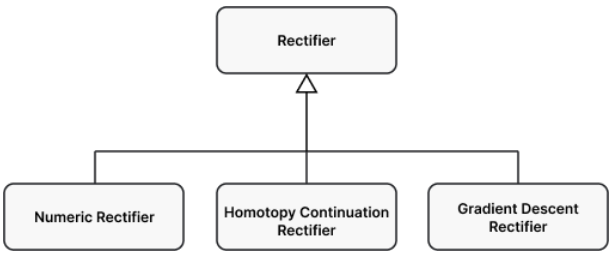


Figure 3: Rectifiers

5.1.3. Losser

This step of the pipeline aims to quantify the quality of the rectification by employing metrics that evaluate the similarity between the rectified image and the original (unwarped) one.

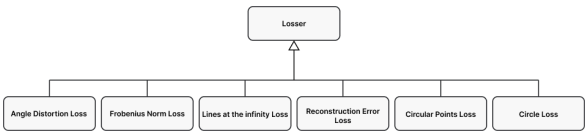


Figure 4: Losses

Although various loss metrics were considered (see Section 3.5), the most meaningful and similarity-invariant ones were the *Circular Points Loss* and the *Eccentricity-based Circle Loss*, which demonstrated superior performance in capturing the geometric accuracy of the reconstruction.

5.1.4. Noise model

To simulate real-world imperfections and evaluate the robustness of the methods, we introduce a controlled Gaussian noise model directly into the conic representations. Noise is applied at the matrix level, perturbing the symmetric 3×3 conic matrix $\mathbf{M}$ associated with each ellipse.

Let $\mathbf{M} \in \mathbb{R}^{3\times3}$ denote the original conic matrix. The perturbed conic $\tilde{\mathbf{M}}$ is computed as:

$$\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{N},$$

where $\mathbf{N} \sim \mathcal{N}(0, \sigma^2)$ is a symmetric noise matrix with entries drawn from a zero-mean Gaussian distribution with standard deviation σ. To ensure that $\tilde{\mathbf{M}}$ remains symmetric (as required for valid conics), the noise matrix is symmetrized as:

$$\mathbf{N} = \frac{1}{2}\left(\mathbf{G} + \mathbf{G}^T\right),$$

where $\mathbf{G} \in \mathbb{R}^{3\times3}$ contains i.i.d. samples from $\mathcal{N}(0, \sigma^2)$.

This model reflects realistic measurement noise, such as uncertainty in ellipse fitting or sensor inaccuracies, while maintaining the algebraic properties of conics. The parameter σ controls the magnitude of the perturbation and can be adjusted to simulate different levels of noise severity.

5.2. Research Questions

In this section, all the major research questions we tried to answer with our experiments are listed and described.

5.2.1. Q1: Is it a good idea to rectify an image using the image of the circular points?

To answer this question, we tested how precisely the circular point must be estimated to achieve an acceptable rectification performance using the algorithm described in 3.1.2.

As a performance measure, we chose the average eccentricity of the rectified ellipses, computed as described in 3.5.6. We defined it to be acceptable when it is less than or equal to 0.3.

To test the robustness, we first chose a base configuration depicted in Figure 5

Using the true homography H_{true} produced by the Scene Generator, the true images of the circular points I^*_{true} and J^*_{true} can be computed as follows:

$$I^*_{true} = H_{true} I$$
$$J^*_{true} = H_{true} J$$

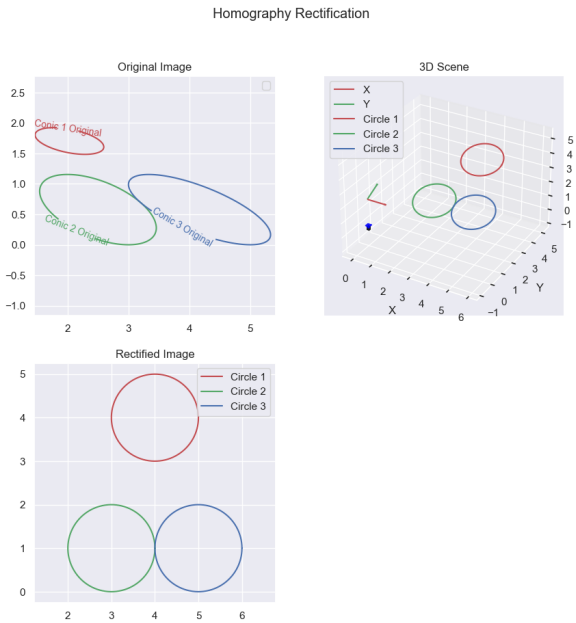


Figure 5: Base Scene for the first research question

Given a perturbation radius δ, the perturbed image of the circular points can be obtained as:

$$I^*_p = I^*_{true} + n_I$$
$$J^*_p = J^*_{true} + n_J$$
$$n_I, n_J \sim \mathcal{N}([0,0,0]^T, \delta^2 I^{3\times3})$$

After the points are perturbed, they need to be normalized to compute the actual distances d_I and d_J to the true ones. As a distance metric, we used the standard Euclidean distance as follows:

$$d_I = ||I^*_{true} - I^*_p / I^*_p[0]||_2$$
$$d_J = ||J^*_{true} - J^*_p / J^*_p[0]||_2$$

And a unified distance d can be computed as the average of d_I and d_J.

We performed a series of experiments varying δ between 10^{-9} to 10 in 500 steps in log_{10} scale, performing 100 perturbations for each δ. The obtained distribution of d_I and d_J can be seen in Figure 6.

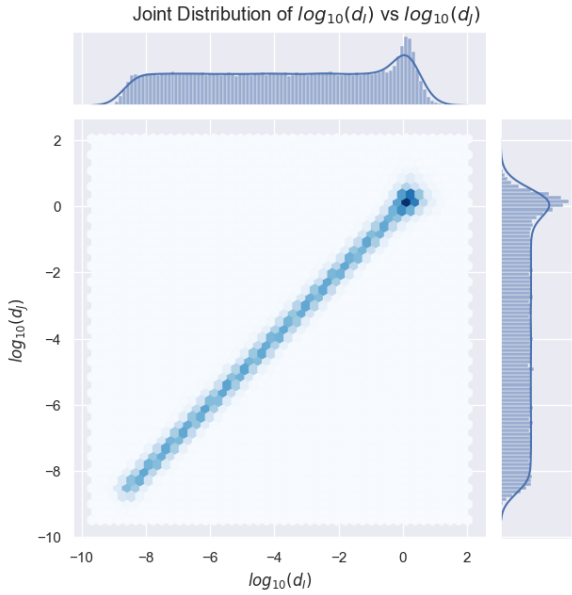


Figure 6: The joint distribution of d_I and d_J for the performed experiments

The marginal distributions of both the average Eccentricity and d can be found in Figure 7. There is a notable increase in density around $d = 1$, this is likely due to the normalization step performed after the perturbation.

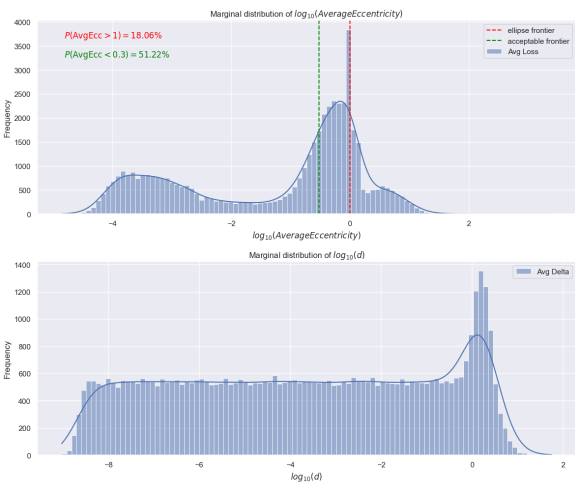


Figure 7: The marginal distributions of the Average Eccentricity and d for the performed experiments. In the upper plot some probabilities are printed. They are computed as $P(event) = \frac{\#\text{samples in which event happens}}{\#\text{samples}}$

The joint distribution of the average Eccentricity and d can be found in Figure 8

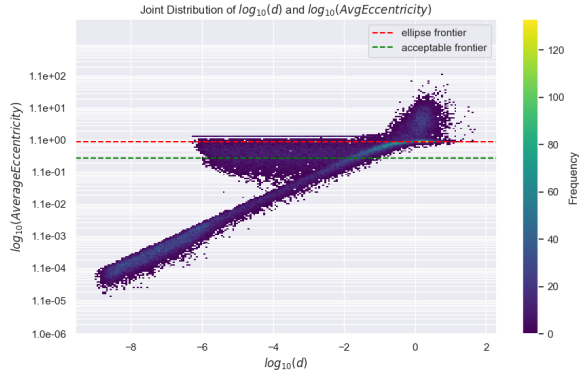


Figure 8: The joint distribution of the average Eccentricity and d

Finally, we estimated, given a distance to the true circular points d, the probability of obtaining an ellipse after the rectification $P_e(d) = P($ and the probability of having an acceptable eccentricity $P_a(d)$. The two probabilities respectively represent:

$$P_e(d') = P(avgEcc < 1 | d = d')$$
$$P_a(d') = P(avgEcc < 0.3 | d = d')$$

To estimate those probabilities, we binned the samples based on the $log_{10}(d)$ to then perform:

$$\overline{P_e}(d') = \frac{\#\text{samples s.t. } d \text{ is in the same bin of } d' \text{ and AvgEcc} < 1}{\#\text{samples s.t. } d \text{ is in the same bin of } d'}$$

$$\overline{P_a}(d') = \frac{\#\text{samples s.t. } d \text{ is in the same bin of } d' \text{ and AvgEcc} < 0.3}{\#\text{samples s.t. } d \text{ is in the same bin of } d'}$$

Furthermore, the confidence intervals on those estimates have been computed using the Wilson score [8]. The result of this analysis can be seen in Figure 9

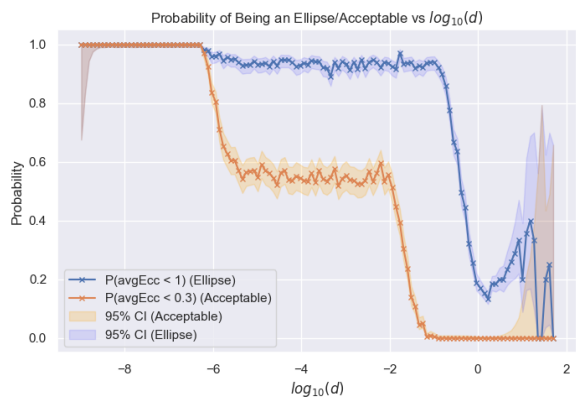


Figure 9: The estimated probabilities of ellipse and acceptable as a function of d

From the plots, it appears that the acceptable probability drops at around $\boldsymbol{d_{max} = 10^{-6}}$, which sets a very stringent lower bound on the precision required for estimating the images of the circular points. This suggests that rectifying an image using an estimation of the image of the circular points is possible but must require an extremely accurate solver for polynomial systems.

5.2.2. *Q2: Which of the methods is the most effective at rectifying images?*

As proven by the previous research question, all methods that estimate the image of the circular points must be very precise, and unfortunately, the Algebraic Approach lacks this feature due to the pairwise intersection approach. When the conics estimation is noisy, a perfect match between the pairwise intersections no longer exists, and thus some interpolation is needed but this far exceeds the d_{max} threshold.

The homotopy continuation can seem promising due to its strong theoretical foundation and extremely high precision for a numerical solver. The real challenge is to find a regular homotopy that applies to this domain. Despite our best efforts, we couldn't find a suitable homotopy. Each of our trials only produced singular paths, meaning paths were the Jacobian becomes singular (and thus non-invertible). This implies that the following of the solution stops before reaching the target system. Despite this, we still managed to rectify some images by selecting the best-fitting solutions between the partially obtained ones, but we couldn't make it either consistent or robust. This happens because, given three randomly selected ellipses, the probability that they share a common intersection is 0, and thus constructing a path in the ellipse space that always allows for an intersection is extremely difficult and most of the time not possible.

Last but not least, the Gradient Descent method shines way above the others, achieving unparalleled performance, effectiveness, consistency, and robustness on a huge variety of both simulated and real tests. This method also doesn't require much hyperparameter tuning since all the tests have been performed with the default values of the parameters.

Some of the rectification results on real images can be seen Appendix A

5.2.3. *Q3: How do the simulator's parameters influence the rectification performance?*

To assess the sensitivity of rectification performance to simulator parameters, we designed a controlled synthetic experiment wherein a scene is generated with three fixed, non-degenerate circles in 3D space.

A hyperparameter optimization framework [1] was employed to systematically vary the camera's rotation angles around the x- and y-axes within bounded intervals while keeping all other simulation parameters fixed. Specifically, we sampled:

- x-axis rotation in the range $[5°, 90°]$,
- y-axis rotation in the range $[5°, 90°]$.

Each sampled configuration was used to generate a synthetic projection of the scene. The gradient-based rectification algorithm then estimated a homography from the noisy conic projections, which was applied to warp the original conics. The objective function maximizes the maximum eccentricity of the resulting conics, as described in Section 3.5.6, to identify the parameter values that lead to the most challenging rectification scenarios—i.e., those that produce the highest deviation from circularity after rectification.

Figure 10 shows the importance of the two parameters explored in the experiment.

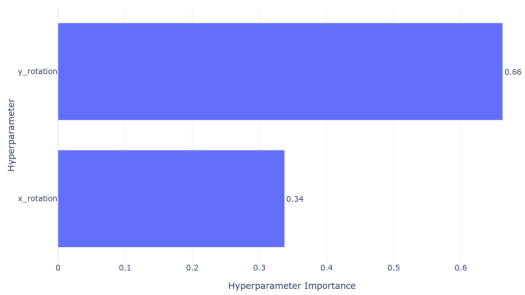


Figure 10: Parameters importance

The results of the experiment are summarized in Figure A.12. From the distribution of outcomes, we observe a clear trend: as the rotation angles increase—particularly along the y-axis—the rectification process becomes progressively more challenging. This is expected, as larger perspective distortions make it harder to accurately estimate the homography that restores circularity in the image plane.

Despite this degradation in performance with increasing angles, the method demonstrates robust behavior. Considering Figure A.11 and excluding a single outlier with a loss close to 0.99, all other trials resulted in a maximum conic eccentricity below 0.18. This indicates that the gradient-based rectifier is generally effective at recovering a satisfying transformation.

6. Conclusion

Our experimental evaluation demonstrates that gradient descent optimization yields accurate rectifi-

cation results, even under realistic noise perturbations. While all tested methods show sensitivity to noise, the gradient-based approach consistently achieves high-precision rectifications. These results highlight the practicality of using optimization-based rectification in scenarios where measurement noise and imperfect geometric features are inevitable.

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, «Optuna: a next-generation hyperparameter optimization framework», in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining (2019).

[2] P. Breiding and S. Timme, «Homotopycontinuation.jl: a package for homotopy continuation in julia», in Mathematical software – icms 2018, edited by J. H. Davenport, M. Kauers, G. Labahn, and J. Urban (2018), pp. 458–465, ISBN: 978-3-319-96418-8.

[3] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars, «Solving polynomial systems via homotopy continuation and monodromy», IMA Journal of Numerical Analysis **39**, 1421–1446, ISSN: 0272-4979 (2018) 10.1093/imanum/dry017.

[4] D. P. Kingma and J. Ba, «Adam: a method for stochastic optimization», arXiv preprint arXiv:1412.6980 (2014).

[5] J. Richter-Gebert, *Perspectives on projective geometry: a guided tour through real and complex geometry* (Springer Berlin Heidelberg, 2011), ISBN: 9783642172861.

[6] A. J. Sommese, C. W. Wampler, et al., *The numerical solution of systems of polynomials arising in engineering and science* (World Scientific, 2005).

[7] Q. Tran-Dinh, «Gradient descent-type methods: background and simple unified convergence analysis», in *Federated learning: theory and practice*, edited by L. M. Nguyen, T. N. Hoang, and P.-Y. Chen (Elsevier, 2024) Chap. 1, pp. 3–28, ISBN: 9780443190377, 10.1016/B978-0-443-19037-7.00006-3.

[8] Wilson, «Probable inference, the law of succession, and statistical inference», Journal of American Statistical Association (1927).

Appendix A. Objective Function Trends and Rotation Parameter Ranking in Optimization Trials

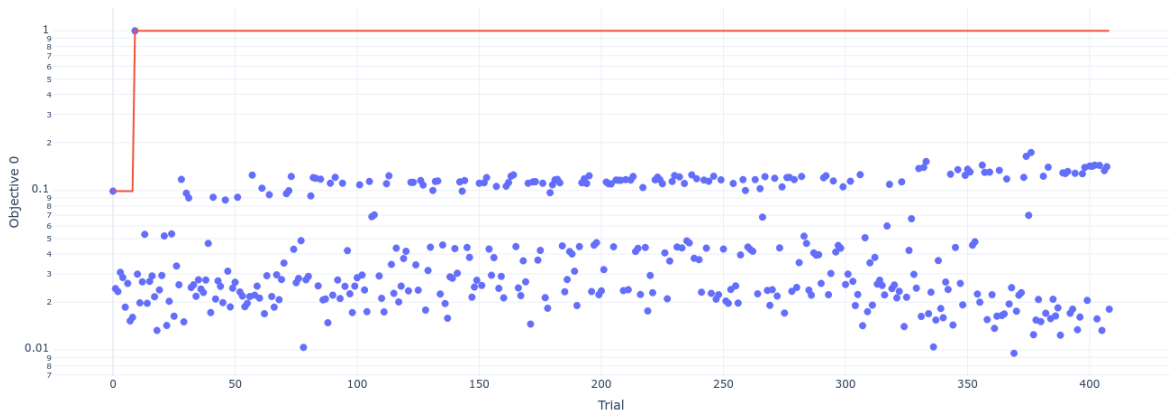


Figure A.11: Trial's objective values

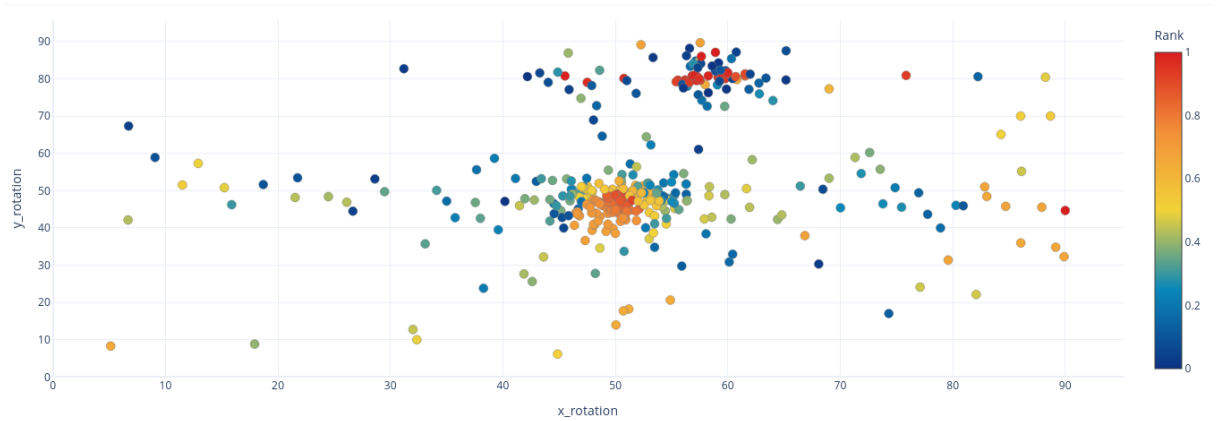
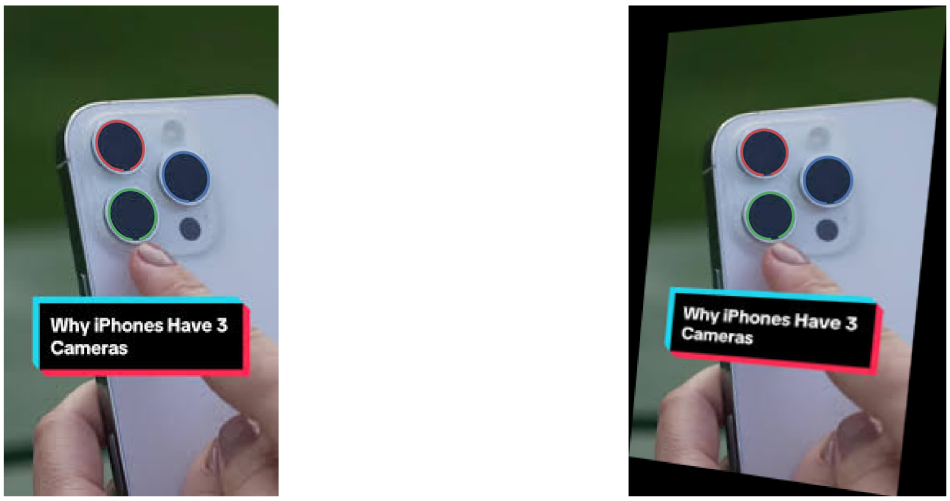


Figure A.12: Rank plot x-rotation vs y-rotation

Appendix B. Visual Results of Planar Rectification With Plotted Conics

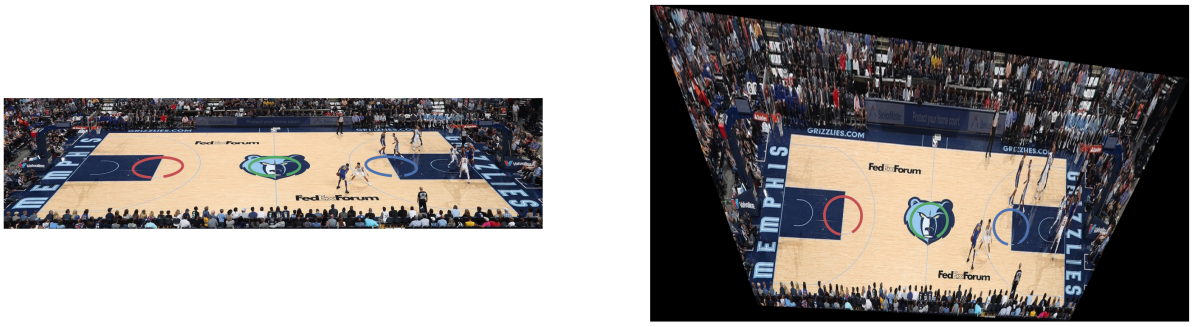
Original and Warped Image

Original and Warped Image

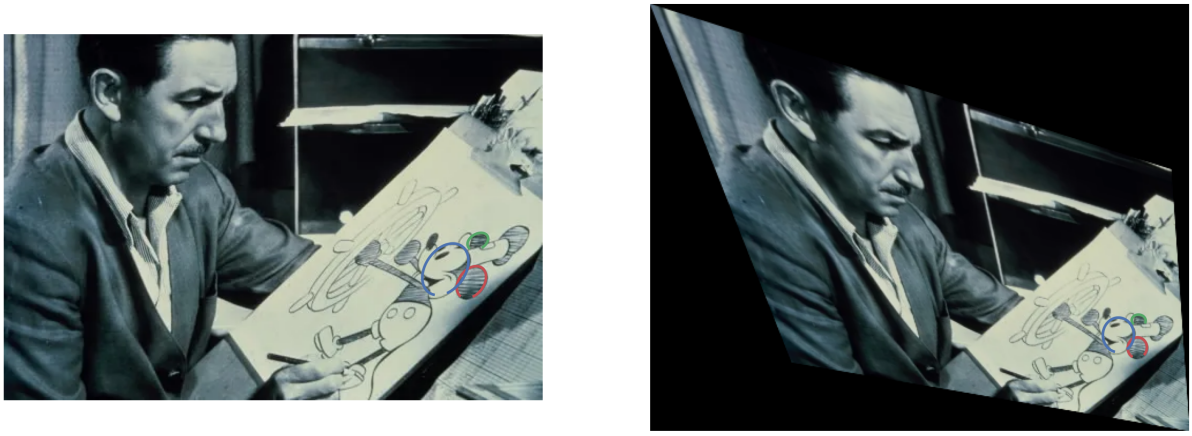


Original and Warped Image

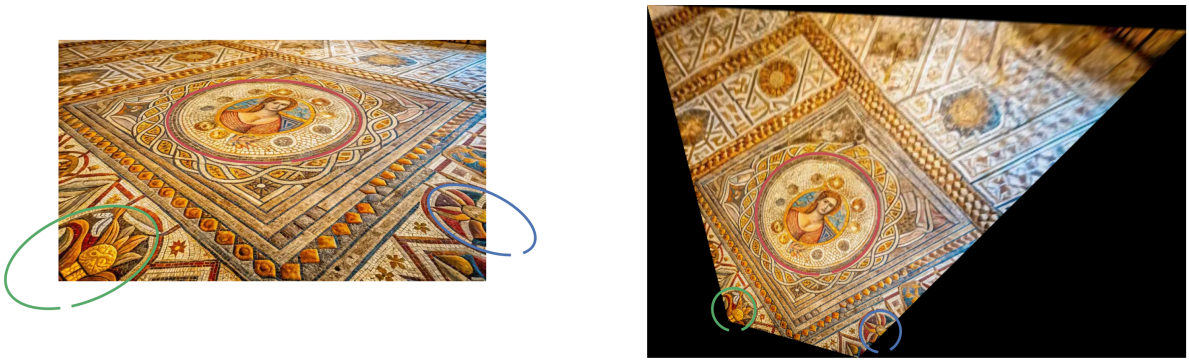
Original and Warped Image



Original and Warped Image

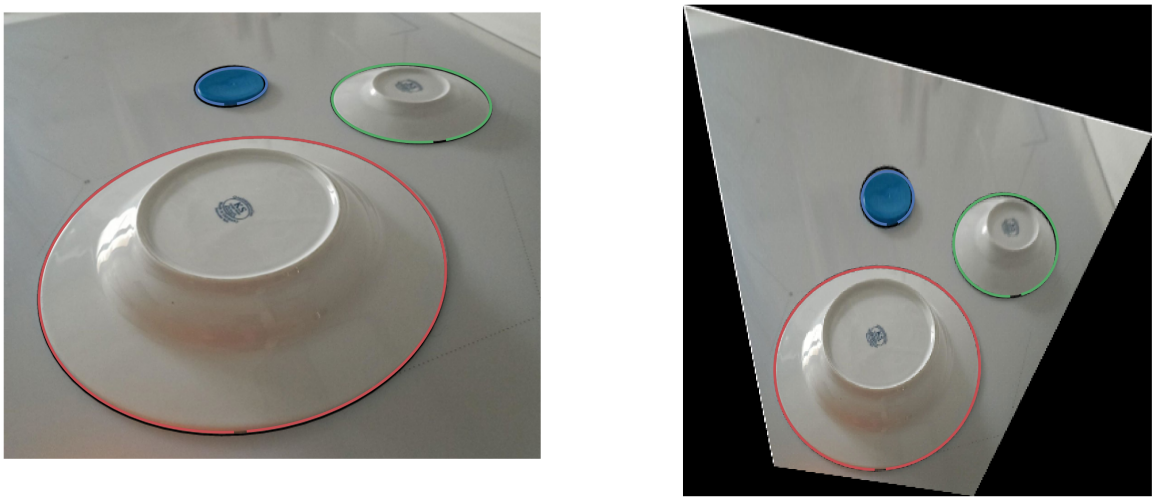


Original and Warped Image

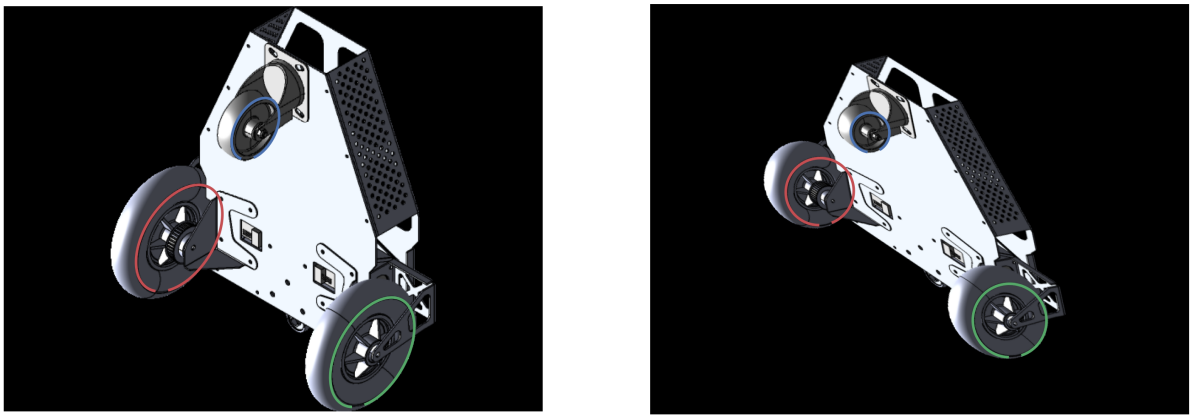


Original and Warped Image

Original and Warped Image



Original and Warped Image



Original and Warped Image

This image was a test to see what happens if the circles impose dependent constraints. The rectifier still makes the ellipses circles but this is not enough for shape reconstruction.