Reconstructing a Sphere and the Camera Focal Length from a Single View by Fitting Planes

Erol Ozgur, Mohammad Alkhatib, Youcef Mezouar, and Adrien Bartoli*

Clermont Auvergne INP, Institut Pascal, Clermont-Ferrand, France

*University Hospital, Clermont-Ferrand, France

September 16, 2025

Corresponding Author: Erol Ozgur
erol.ozgur@sigma-clermont.fr

Abstract

We address two problems. First, reconstructing a sphere of a prescribed radius from a single calibrated view of its occluding contour. Second, reconstructing simultaneously a sphere of a prescribed radius and the camera focal length from a single view of the sphere's occluding contour. A sphere's occluding contour generally appears as an ellipse and existing reconstruction methods use ellipse fitting, thus requiring ≥ 5 contour points. The calibrated minimal solution requires 3 points, and a few methods can deal with it. The minimal solution with an unknown focal length requires 4 points, and there exists no method to deal with it. All existing methods share two shortcomings: *(i)* they fail for non-elliptic occluding contours, including parabola and hyperbola, and *(ii)* they use the point-to-ellipse distance, whose computation is not closed-form. On the first problem, we make the observation that the spherically-normalised contour points form a circle in space, which we reconstruct by plane fitting. This handles minimal 3-point and redundant > 3 point fitting, copes with elliptic and non-elliptic contours, and benefits from the simple point-to-plane distance. The reconstructed circle then leads to a one-parameter sphere family from which the actual sphere of prescribed radius is uniquely retrieved. We name our method `SpherO`, where letter 'O' depicts a circle. We robustify `SpherO` using random sampling at the 3-point plane fitting stage. Experimental comparisons show that `SpherO` outperforms the current-best 3-point method. On the second problem, we make the observation that the spherically-normalised contour points generally form a non-circular spatial elliptic curve for wrong camera parameters. The calibration constraint is thus that the spherically-normalised points must be cocircular, which implies coplanarity. The coplanarity constraint allows us to solve the minimal 4-point case. We solve redundant > 4 point case by fitting planes. This simultaneously reconstructs a circle and the camera focal length from a non-circular spatial elliptic curve. Finally the reconstructed circle and the camera focal length allow us to retrieve the sphere of prescribed radius. We name our method `SpherOf`, where letter '`f`' is for the focal length. We robustify `SpherOf` using random sampling at the 4-point coplanarity constraint formation and 3-point plane fitting stages. Experiments show that `SpherOf` has comparable performance to `SpherO`.

Contents

1 Introduction

Sphere reconstruction, equivalent to sphere-based camera localisation, forms a practical tool for applications requiring the pose of sphere-shaped objects. These applications include real-time surgical tool tracking and guidance, and visual robot servoing. Sphere reconstruction, given its radius and its occluding contour in a single view taken by a perspective camera, entails finding the sphere's 3D centre point. We address two sphere reconstruction problems. The first problem is reconstructing a sphere of a prescribed radius from a single calibrated view of its occluding contour. The second problem is reconstructing simultaneously a sphere of a prescribed radius and the unknown camera focal length from a single view of the sphere's occluding contour, given the other intrinsic parameters including the principal point. A sphere's occluding contour is observed as either an ellipse, a parabola or a hyperbola in the perspective image. Existing methods mostly address the first problem. They handle the elliptic case –the most common one– but fail for the other two cases. They approximate the point-to-ellipse distance, which is not closed-form, by its algebraic counterpart.

We propose a novel simple and convex method resolving the state-of-the-art's limitations. We make the observation that the spherically-normalised contour points form a circle in space, which we reconstruct by plane fitting. For the first problem, the reconstructed circle leads to a one-parameter sphere family from which the actual sphere of prescribed radius is uniquely retrieved. We name our method `SpherO`, where letter 'O' depicts a circle. `SpherO` uses mere plane-fitting, making it the most compelling method with respect to existing work. First, `SpherO` handles both minimal 3-point and redundant > 3 point cases seamlessly. It can thus serve as minimal method in the inner loop of RANSAC-like robust methods and as final refinement method given the inlier set. Second, `SpherO` uses a geometric cost function, namely a point-to-plane distance. It entirely avoids the point-to-conic distance required by existing methods, providing a quick and sound way of determining the inlier set within RANSAC. Third, `SpherO` seamlessly deals with the settings where the sphere is seen as an ellipse, a parabola or a hyperbola. Fourth, `SpherO` provides a one-parameter family of spheres when the sphere's radius is unknown and uniquely reconstructs the sphere otherwise. Fifth, `SpherO` does not have artificial degeneracies. In other words, `SpherO` deals with all settings where the sphere's occluding contour is a proper conic. For the second problem, which extends `SpherO` to also reconstruct the focal length, we make the observation that the spherically-normalised contour points generally form a non-circular spatial elliptic curve under wrong calibration parameters. The calibration constraint is thus that the spherically-normalised points must be cocircular, implying coplanarity. We optimise the coplanarity of a non-circular spatial elliptic curve to form a circle by fitting planes. This simultaneously reconstructs a circle and the camera focal length. Finally, the reconstructed circle and camera focal length allow us to retrieve the sphere of prescribed radius. We name our method `SpherOf`, where letter 'f' is for the focal length. `SpherOf` inherits all the advantages of `SpherO`. It handles both minimal 4-point and redundant > 4 point cases seamlessly. It can thus serve as minimal method in the inner loop of RANSAC-like robust methods and as final refinement method given the inlier set.

This paper is an extension of the short conference paper [9], with additional methodological and experimental contributions, as follows:

1. We experimentally show, using a new refinement method, that the robustified `SpherO` solution (`Robust-SpherO`) does not require refinement. Consequently, it

remains as the most accurate and the fastest sphere reconstruction method from a single calibrated view [9].

2. We introduce the second problem, which extends the first problem from [9] by including the focal length as an unknown parameter, for which we propose `SpherOf`. It forms the first method which handles the minimal 4-point case.

The rest of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the modelling and assumptions. Section 4 deals with the first problem and section 5 with the second problem. Section 6 concludes.

2 Related Work

We review existing methods for sphere reconstruction from a single view of a sphere.

2.1 Reconstructing a Sphere from a Calibrated View

There exist only three methods which use 3 points or more [13, 11, 14]. Methods [13] and [11] are not robust; they minimise the squared algebraic point-to-ellipse image distance. Methods in [14] minimise the geometric distance from a cone tangent to the sphere (method `Toth`); it provides a RANSAC-based robust solution specifically relying on the ellipse to form the consensus set (method `Robust-Toth`). It is shown in [14] that `Toth` outperforms the two methods [13, 11] and other existing methods requiring at least 5 points [5, 7, 8], and is thus the state of the art. Table 1 summarises the characteristics of the proposed `SpherO` and `Robust-SpherO` in comparison to `Toth` and `Robust-Toth`. We have that `SpherO` and `Toth` are both minimal and redundant, can handle any conical occluding contours and use a geometric cost. We will see that `SpherO` is simpler to derive and faster. We have that `Robust-SpherO` is the first robust method to handle any conical occluding contours, as `Robust-Toth` is restricted to ellipses.

Table 1: Comparison of $\geq$ 3-point sphere reconstruction methods.

	Handles all image conics	Fits	Geometric costs
`Toth` [14]	✓	cone	secant
`Robust-Toth` [14]	only ellipse	ellipse cone	point-to-ellipse distance* secant
`SpherO`	✓	plane	point-to-plane distance
`Robust-SpherO`	✓	plane	point-to-plane distance

**numerically approximated by algebraic distance*

2.2 Reconstructing a Sphere and the Camera Focal Length

There does not exist methods which handle the minimal 4-point uncalibrated case. Only one method solves analytically the camera focal length from a single view of the sphere's occluding contour given the principal point [6] through conic fitting with 5 or more points. It then proposes to use a calibrated method from [15] to reconstruct the sphere.

Method [6] is not robust and does not provide a complete solution for reconstructing a sphere and the camera focal length.

In contrast, `SpherOf` reconstructs simultaneously both the sphere and the camera focal length. It requires neither conic fitting nor algebraic point-to-conic distance computation. `SpherOf` handles the minimal 4-point and redundant > 4 point cases. We also propose three random sampling-based `Robust-SpherOf` methods.

3 Modelling and Assumptions

We define the two reconstruction problems and then present the modelling and the assumptions made to address the problems.

3.1 Problem Statements

Calibrated sphere reconstruction. We reconstruct a sphere of a prescribed radius from a single calibrated view of its occluding contour.

Uncalibrated sphere reconstruction. We reconstruct simultaneously a sphere of a prescribed radius and the unknown camera focal length from a single view of the sphere's occluding contour, given the other intrinsic parameters including the principal point.

3.2 Modelling

We introduce the notation and theoretical properties on these two problems.

3.2.1 Notation

We denote the set of real numbers as $\mathbb{R}$ and the set of natural numbers as $\mathbb{N}$. We express all 3-space coordinates in the standard pinhole camera coordinate frame. We denote the intrinsic matrix as $\mathbf{K} \in \mathbb{R}^{3\times3}$, the optical centre as $\mathbf{O} \in \mathbb{R}^3$, the unit direction vector of the principal axis as $\underline{\mathbf{z}} \in \mathbb{R}^{3\times1}$, the principal point as $\mathbf{p}_o \in \mathbb{R}^2$ in pixels, and the focal length as $f \in \mathbb{R}^+$ in pixels. We denote the sphere's centre as $\mathbf{C} \in \mathbb{R}^3$ and the sphere's radius as $R \in \mathbb{R}^+$. We denote the points on the sphere's contour generator as $\mathbf{P}_i \in \mathbb{R}^3$ for $i = 1, \ldots, m \in \mathbb{N}$. We denote their corresponding points on the sphere's occluding contour in the image as $\mathbf{p}_i \in \mathbb{R}^2$ in pixels. They are related to each other proportionally as $\bar{\mathbf{p}}_i \propto \mathbf{K}\mathbf{P}_i$ where $\bar{\mathbf{p}}_i = \left[\mathbf{p}_i^\top, 1\right]^\top$ is the homogeneous coordinates of an occluding contour point. The backprojection of the occluding contour point forms a ray that starts from the optical centre $\mathbf{O}$ and passes through its corresponding point $\mathbf{P}_i$ on the sphere's contour generator. We define $\mathtt{stk}(a, b) = [a^\top, b^\top]^\top$ as the stacking function and $\mathcal{P} : \mathbb{R}^3 \to \mathbb{R}^2$ with $\mathcal{P}(X, Y, Z) = (X, Y)/Z$ as the canonical perspective projection function.

3.2.2 Theoretical Properties

Spherical normalisation places an image point on the camera-centred unit sphere. We build the theoretical properties of calibrated and uncalibrated sphere reconstruction problems based on our geometric observations on the outputs of the spherical normalisation function:

$$\eta(\,\mathbf{x}\,) \;=\; \mathbf{x}/\|\mathbf{x}\| \tag{1}$$

where $\mathbf{x}$ is a 3-space vector. We provide two geometric observations. One for the correct and the other for incorrect camera parameters. These observations also reveal the constraints of the problems.

Correct camera parameters. Considering that the camera parameters are correct, we formalise the following proposition on a sphere's occluding contour and illustrate it in figure 1.

Proposition 1. *A sphere's spherically-normalised occluding contour is a sphero-circle.*

A sphero-conic is a curve formed by intersecting a quadric cone with the unit sphere whose centre is the vertex of the cone. If the cone is circular then the sphero-conic is a circle [2] also known as 'sphero-circle'. Using this result with the camera centre taken as vertex and the sphere's projection lines taken as circular cone proves proposition 1.

Incorrect camera parameters. Considering that the camera parameters are wrong, we formalise the following general configuration on a sphere's occluding contour.

General configuration 1. *Incorrect camera calibration parameters make a sphere's spherically-normalised occluding contour a non-circular sphero-ellipse.*

An incorrect calibration translates the image plane in 3D. Explicitly, an incorrect principal point translates the image plane along x and y axes of the camera's coordinate frame. An incorrect focal length translates the image plane along the optical axis. The occluding contour observed on the camera's sensor is supposed on the translated image plane. Subsequently, the occluding contour and the optical center form a new cone. This cone generally turns out to be non-circular. It follows that the intersection of a unit sphere with a non-circular cone, whose vertex is the unit sphere's centre, is a non-circular sphero-ellipse [1]. Next, we give possible degenerate configurations.

Degenerate configuration 1. *Incorrect camera calibration parameters make a sphere's spherically-normalised occluding contour a sphero-circle in the following cases:*

Case 1. The image plane translates parallel to the direction showing the sphere's centre.

Case 2. Symmetric configurations of Case 1 with respect to the optical axis.

Constraints. We know from proposition 1 that the correct camera parameters yield a sphero-circle upon spherical normalisation. Thus the spherically-normalised points must be cocircular. This constraint implies coplanarity, a weaker constraint which can be considered a relaxation of cocircularity in 3-space. We exploit the coplanarity constraint in the solutions of the calibrated and uncalibrated sphere reconstruction problems.

3.3 Assumptions

We make the following assumptions on the two reconstruction problems.

On calibrated sphere reconstruction. We consider that camera's intrinsic parameters are known.

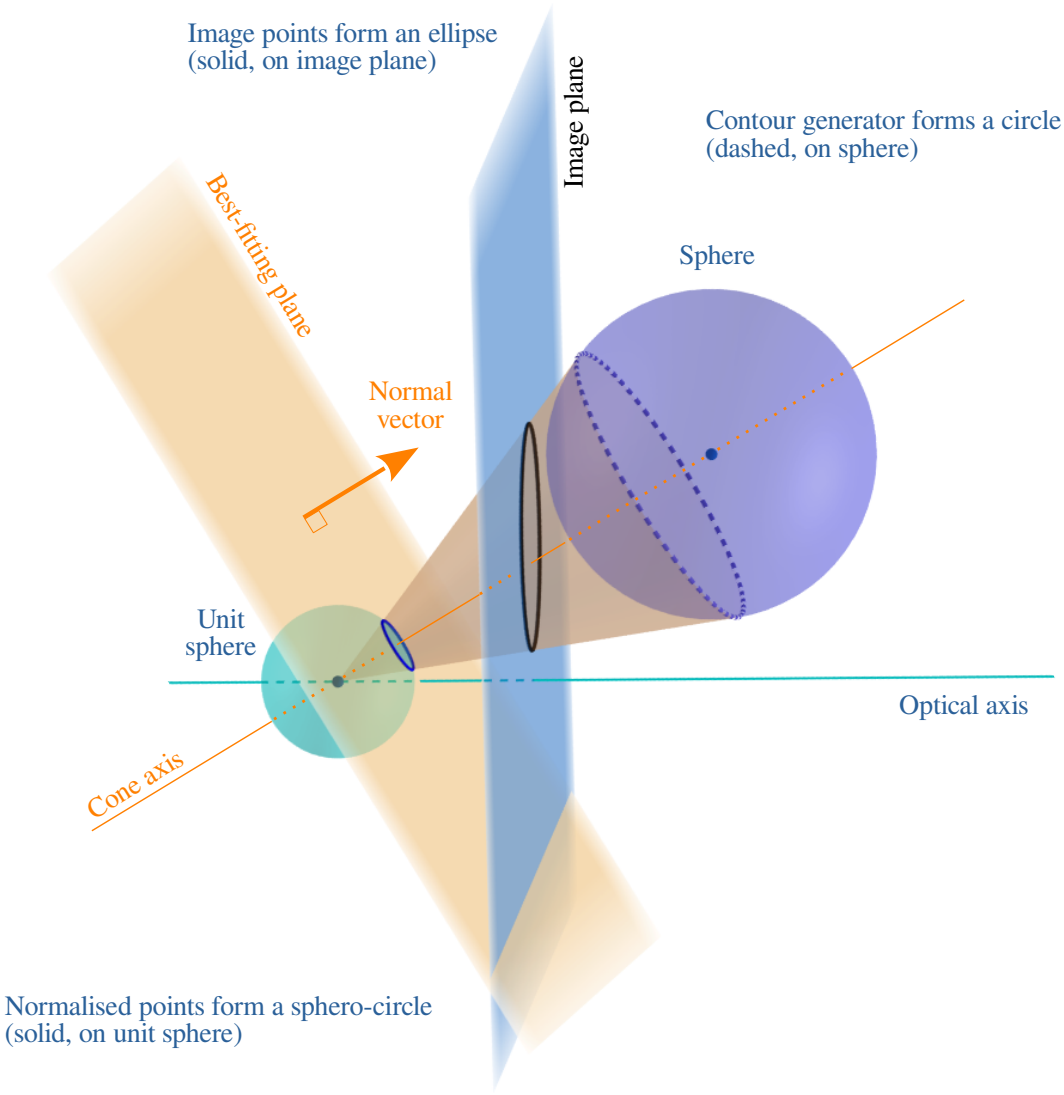

Figure 1: The geometric observation subtending the calibrated sphere reconstruction. Tilting the camera's image plane towards the cone axis leads to the sphere's occluding contour (solid, on image plane) becoming a parabola and eventually a hyperbola. Importantly, the contour generator (dashed, on sphere) and normalised points (solid, on unit sphere) remain invariant, hence sphero-circles, in all projection cases.

On uncalibrated sphere reconstruction. We discuss three possible uncalibrated reconstruction setups: (*i*) reconstructing the focal length given the principal point, (*ii*) reconstructing the principal point given the focal length, and (*iii*) reconstructing the focal length and the principal point.

In the first setup (*i*), knowing the principal point avoids most of the degenerate configurations of setup (*iii*). This is because the image plane misplacement can only be along the optical axis due to an incorrect focal length. Under these circumstances, degenerate configurations can only occur if the sphere's centre is on the optical axis. This corresponds to *Case 1* in degenerate configuration 1, and *Case 2* becomes *Case 1*. Otherwise, there remains only one possible sphero-circle formation between the sphero pole and the sphero great circle of the unit sphere yielding the correct focal length.

In the second setup (*ii*), knowing the focal length fixes the depth of the image plane. The image plane misplacement can thus be along the x and y axes of the camera's coordinate frame due to an incorrect principal point. Under these circumstances, there exist two possible sphero-circle formations on the unit sphere symmetrical to the optical axis. Therefore, there are two solutions to the principal point. Choosing the solution closest to the image centre should usually yield the correct principal point. If the sphere's centre is on the optical axis, then there exist one sphero-circle formation which yields the correct principal point.

In the third setup (*iii*), the image plane misplacement can be along the x, y and z axes of the camera's coordinate frame. Under these circumstances, degenerate configurations can appear for both *Case 1* and *Case 2* of degenerate configuration 1. Furthermore, the effects of the principal point and the focal length on the sphero-circle formation are coupled. This is because varying the principal point alone can slide the normalised points on the unit sphere in a very similar way to varying the focal length alone would do. However, the other way around is not necessarily true. This setup thus has important marked singularities.

In practice, the focal length is the most important parameter and the only one that cannot be set to a default value; many self-calibration algorithms estimate only the focal length (*e.g.*, [10, 4]). We therefore focus on setup (*i*): reconstructing the focal length given the principal point. It addresses many applications where the principal point is known or can be approximated with the image centre. Following this discussion, we assume that the focal length is shared on the x and y image axes (in other words, the pixels are square), that the focal length is the only unknown intrinsic parameter, and that the view is in general configuration (in particular, that the sphere's centre is not on the optical axis).

4 Reconstructing a Sphere from a Calibrated View

We first introduce `SpherO`'s robust solution as a robust initialising method and then a refinement method for the reconstruction of a sphere from a calibrated view.

4.1 Robust Initialisation

`SpherO` reconstructs the sphere's centre from a single calibrated view. `SpherO` has three steps. First, following proposition 1, we spherically normalise the sphere's occluding contour to form a sphero-circle as illustrated in figure 1. Second, we reconstruct the sphero-circle's support plane, centre and radius. This induces a one-parameter sphere family. Third, we retrieve the actual sphere of prescribed radius. The use of spherical normalisation, following proposition 1, allows one to remove the distorting effects of the camera's image plane by converting the image points into a 3D circle on the camera-centered unit sphere. In the following, we give `SpherO`'s three steps and its robustification `Robust-SpherO`.

4.1.1 Spherical Normalisation of the Occluding Contour

Spherical normalisation is the first step of `SpherO`. We spherically normalise the homogeneous occluding contour points using equation (1) with the given intrinsic matrix $\mathbf{K}$ as follows:

$$\mathbf{q}_i \;=\; \eta(\,\mathbf{K}^{-1}\bar{\mathbf{p}}_i\,) \tag{2}$$

which form a sphero-circle.

4.1.2 Sphero-circle Reconstruction by Plane Fitting

Sphero-circle reconstruction is the second step of `SpherO`. We give a solution using plane fitting. We first give the general case for $m \geqslant 3$ points. We then give an equivalent but lower computation cost version for $m = 3$, which is meant to be used in RANSAC's inner loop. We finally give closed-forms for the sphero-circle's centre and radius.

General plane fitting. We define the best-fitting plane as the minimizer of the sum of the squared orthogonal distances between the spherically-normalised points and the plane. Let $\mathbf{q}_0$ be the average of the points. We use a simple singular value decomposition of the $3 \times m$ matrix containing the centred points $\mathbf{q}_i - \mathbf{q}_0$, whose least singular vector gives the plane's unit normal vector $\underline{\mathbf{n}} \in S^2$ [3]. The plane's signed distance to origin is $d_\Pi \;=\; \mathbf{q}_0^\top \underline{\mathbf{n}}$ and the plane's coordinate vector is $\Pi = [\,\mathtt{sign}(d_\Pi)\,\underline{\mathbf{n}}^\top,\, -|d_\Pi|\,]^\top$, where $\mathtt{sign}(d_\Pi)$ ensures that the normal vector points toward the sphere.

Minimal 3-point plane recovery. Given 3 points $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$, the plane normal is:

$$\underline{\mathbf{n}} \;=\; \eta\,(\,(\mathbf{q}_3 - \mathbf{q}_1) \times (\mathbf{q}_2 - \mathbf{q}_1)\,)\,. \tag{3}$$

The plane's signed distance to the origin is then $d_\Pi \;=\; \mathbf{q}_i^\top \underline{\mathbf{n}}$ where $i \in \{1, 2, 3\}$ is any of the 3 points. Since any set of 3 points lies perfectly on a plane, the 3 points' projections onto the plane's normal yield the same distance d_Π. Finally, the plane coordinates Π are formed as above for $m \geqslant 3$.

Sphero-circle reconstruction. The best-fitting plane's normal $\underline{\mathbf{n}}$ yields the sphero-circle's support plane normal. We recall that the intersection of a plane orthogonal to the axis of a circular cone is a circle whose centre is the intersection point with the cone's axis. This implies that the sphero-circle's support plane is perpendicular to the cone's axis. It follows that the sphero-circle's centre $\mathbf{c} \;=\; d_\Pi\,\underline{\mathbf{n}}$ is the closest point of its support plane

to the origin. We thus proceed to compute the sphero-circle's radius as follows. First, we form a right triangle $\measuredangle(\mathbf{O}\,\mathbf{c}\,\mathbf{q}_i)$ using the camera centre $\mathbf{O}$, the sphero-circle's centre $\mathbf{c}$ and any normalised point $\mathbf{q}_i$ of the sphero-circle. The length of hypotenuse $\| \mathbf{O} - \mathbf{q}_i \|$ is 1 since $\mathbf{q}_i$ is spherically normalised. Second, we apply Pythagoras's theorem to calculate the sphero-circle's radius $r = \sqrt{1 - d_{\Pi}^2}$. We finally write the sphero-circle's coordinate vector as $\odot = [\,\underline{\mathbf{n}}^\top,\ \mathbf{c}^\top,\ r\,]^\top$.

4.1.3 Sphere Family Reconstruction and Sphere Localisation

We use the sphero-circle to reconstruct a one-parameter sphere family containing all spheres whose occluding contour matches the observed one. We parameterise this family by the sphere radius κ. If the reconstruction radius R is not prescribed, this family represents all potential reconstruction solutions; if it is prescribed, then `SpherO`'s third step uses it to localise the uniquely reconstructed sphere from the family, by simply setting $\kappa = R$.

We represent a sphere by a 4-vector in the affine space $\mathbb{R}^4$ as $\mathbf{s}(\kappa) = [\,\mathbf{C}(\kappa)^\top,\ \kappa\,]^\top$, with centre $\mathbf{C}(\kappa) \in \mathbb{R}^3$ and radius $\kappa \in \mathbb{R}^+$. The centre $\mathbf{C}(\kappa)$ must be on the ray which starts from the camera centre $\mathbf{O}$. It thus passes through the sphero-circle's centre $\mathbf{c}$ and is also aligned with the sphero-circle's support plane normal, leading to:

$$\mathbf{C}(\kappa) = d(\kappa)\ \underline{\mathbf{n}}, \tag{4}$$

where $d(\kappa)$ encodes the depth of the sphere's centre. This depth may be found by forming two similar right triangles. The first triangle $\measuredangle(\mathbf{O}\,\mathbf{c}\,\mathbf{q}_i)$ is the same one as explained in the sphero-circle reconstruction. The second triangle is $\measuredangle(\mathbf{O}\,\mathbf{P}_i\,\mathbf{C})$. It follows that:

$$d(\kappa) = \kappa / r. \tag{5}$$

Finally, when the sphere radius is prescribed as R, the reconstructed sphere is $R\,[\underline{\mathbf{n}}^\top / r\ ,\ 1]^\top$.

4.1.4 Robust Solution from Random Sampling

We perform sampling-based robust sphere reconstruction using `SpherO`'s minimal plane-fitting solution. We give `Robust-SpherO` in algorithm 1. The inputs are the sphere's

Algorithm 1 `Robust-SpherO`

Input: Sphere's occluding contour pixels $\{\,\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^2,\ i = \{1, \ldots, m\},\ m \geqslant 3\,\}$, sphere's radius $R \in \mathbb{R}^+$ in metric units, camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, and RANSAC threshold $\tau_{pix} \in \mathbb{R}^+$ in pixels (*e.g.*, 1 pixel)

Output: Sphere $\mathbf{s} \in \mathbb{R}^4$

1: τ = `Threshold Normalisation`$(\,\tau_{pix},\,\mathbf{K}\,)$ // equation (6)
2: $\{\,\mathbf{q}_i\,\}$ = `Spherical Normalisation`$(\{\,\mathbf{p}_i\,\},\,\mathbf{K}\,)$ // equation (2)
3: Π = `RANSAC Plane Fit`$(\{\,\mathbf{q}_i\,\},\,\tau\,)$ // section 4.1.2
4: $\odot$ = `Sphero-Circle Reconstruction`$(\,\Pi\,)$ // section 4.1.2
5: $\mathbf{s}$ = `Sphere Recovery`$(\,\odot,\,R\,)$ // section 4.1.3

occluding contour pixels, the sphere's prescribed radius, the camera intrinsics and the RANSAC threshold. The output is the sphere of prescribed radius. Line 1 transforms the RANSAC threshold distance from pixel units to a distance in the normalised image plane as follows:

$$\tau = \tau_{pix} / \mathtt{max}(f_x, f_y), \tag{6}$$

where f_x and f_y are the focal lengths in pixels from the camera intrinsics. The RANSAC threshold τ_{pix} can be considered as the maximum reprojection error of a reconstructed sphere solution. Intuitively, one can choose it as equal to or greater than the camera's intrinsic calibration accuracy (*i.e.*, reprojection error in pixels). Line 2 spherically normalises the occluding contour. Line 3 uses RANSAC on the normalised occluding contour points. The algorithm iterates using the 3-point plane recovery formula (3) on random 3-point samples. Each plane uses a threshold scaled by its depth $\tau_{\Pi} = |d_{\Pi}|\,\tau$ to form its consensus set. The algorithm terminates with a best-fitting plane on the largest consensus set. Line 4 reconstructs the sphero-circle from the best-fitting plane. Line 5 retrieves the sphere of prescribed radius from the sphere family obtained through the sphero-circle.

4.2 Refinement

Section 4.1 provides an initialisation. It works from m image points $\mathbf{p}_1, \ldots, \mathbf{p}_m$ on the occluding contour, which certainly contain noise and may contain blunders. Our objective is to refine the initial solution. The proposed method is inspired from the conic fitting method [12].

4.2.1 Assumptions

We assume that the noise is Gaussian IID on the point to contour distance. Hence, the maximum likelihood solution is given by minimising the sum of squared distances. As the points may include blunders, we replace the square by an M-estimator ρ, though the method is generally applicable with any function ρ on the distances. Details on the M-estimator theory can be found from [16].

4.2.2 Principle

The sphere's occluding contour in the image is a conic, generally an ellipse. This creates a major difficulty, that the point to conic distance does not have an analytic expression, as it involves solving a quartic expression with up to four roots. This was apparently overlooked in previous work, were the algebraic approximation is used without a clear mention to it not being the true geometric distance.

Following [12], the proposed method introduces latent variables to cope with this difficulty. In essence, an unknown point is estimated for each data point and constrained to be on the conic and closest to the data point. A latent variable represents each of these unknown points. Geometrically, we know that the occluding contour is a space circle, which is the intersection of the sphere with some plane. We thus parameterise this circle by a canonical circle, centred at the origin and with unit radius. The unknown points are located on this circle and parameterised by an angle α_j, which is the latent variable.

4.2.3 Method

Occluding contour generation. Let $\mathbf{o} = \mathtt{stk}(\cos\alpha, \sin\alpha)$ be a point on the canonical circle. Point $\mathbf{o}$ is embedded in 3-space by a scaled-rigid transformation, as:

$$\mathbf{P} = s\bar{\mathbf{R}}\mathbf{o} + \mathbf{T}, \tag{7}$$

where $s \in \mathbb{R}$ is the circle radius of the contour generator, $\bar{\mathbf{R}}$ is an SS_{23} matrix, *i.e.*, the first two columns of a 3D rotation matrix, and $\mathbf{T}$ is a translation given the circle centre

in 3-space. The point is then projected to the image as $\mathbf{K}(s\bar{\mathbf{R}}\mathbf{o} + \mathbf{T})$ in homogeneous coordinates. This is exactly a homography, as it can be rewritten as:

$$\mathbf{H}\,\mathtt{stk}(\mathbf{o}, 1) \quad \text{with} \quad \mathbf{H} = \mathbf{K}[s\bar{\mathbf{R}}\,\mathbf{T}]. \tag{8}$$

In this homography however, as $\mathbf{K}$ is known, only 6 DOF remain, not 8. The projection in cartesian coordinates, which we require to compute the distance for the cost function, is given by:

$$\mathcal{P}\left(\mathbf{K}(s\bar{\mathbf{R}}\mathbf{o} + \mathbf{T})\right). \tag{9}$$

Problem formulation. Let $\mathbf{A} = s\bar{\mathbf{R}}$, hence it must satisfy $\mathbf{A}^\top\mathbf{A} \propto \mathbf{I}_{2\times 2}$. We thus formulate the problem as:

$$\min_{\substack{\mathbf{A},\mathbf{T}\\ \alpha_1,\ldots,\alpha_m}} \sum_{j=1}^{m} \rho\left(\| \mathbf{p}_j - \mathcal{P}\left(\mathbf{K}\left(\mathbf{A}\,\mathtt{stk}(\cos\alpha_i, \sin\alpha_i) + \mathbf{T}\right)\right)\|\right) \quad \text{s.t.} \quad \mathbf{A}^\top\mathbf{A} \propto \mathbf{I}_{2\times 2}. \tag{10}$$

Dealing with the orthogonality constraints may be inconvenient. We further reduce the set of matrices which $\bar{\mathbf{R}}$ belongs to. Indeed, let $\mathbf{R} = \mathbf{R}_X\mathbf{R}_Y\mathbf{R}_Z$ be the Euler angle parameterisation of the base rotation corresponding to $\bar{\mathbf{R}}$, we can fix the in-plane rotation origin by setting $\mathbf{R}_Z = \mathbf{I}$, hence $\theta_Z = 0$. We thus have:

$$\mathbf{A} = s\bar{\mathbf{R}} = s\mathbf{R}_X\mathbf{R}_Y\,\mathtt{stk}(\mathbf{I}, \mathbf{0}^\top) = \begin{bmatrix} sc_2 & 0 \\ ss_1s_2 & sc_1 \\ -sc_1s_2 & ss_1 \end{bmatrix}, \tag{11}$$

where $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. Matrix $\mathbf{A}$ has 5 non-zero entries but 3 DOF. There is unfortunately no simple parameterisation with 3 parameters. In order to avoid explicit constraints, we thus use the 2 Euler angles, giving:

$$\min_{\substack{s,\theta_1,\theta_2,\mathbf{T}\\ \alpha_1,\ldots,\alpha_m}} \sum_{j=1}^{m} \rho\left(\| \mathbf{p}_j - \mathcal{P}\left(\mathbf{K}\left(\mathbf{A}\,\mathtt{stk}(\cos\alpha_i, \sin\alpha_i) + \mathbf{T}\right)\right)\|\right). \tag{12}$$

This is minimised by gradient descent straightforwardly.

4.3 Experimental Results

We first compare the robust initialising methods and discuss the results. We then compare the best initialising method with the refinement method and discuss the results.

4.3.1 Comparison of Robust Initialisation Methods

We compare `Robust-SpherO` to the state of the art `Robust-Toth`, *i.e.*, *3pFit + Direct3pFit* from [14], through synthetic and real data experiments.

4.3.1.1 Synthetic Data Experiments

We test the methods' robustness against elliptic and non-elliptic occluding contours. We used the intrinsics of a calibrated camera throughout the synthetic data experiments. The focal length was $f_x = f_y = 1174$ pixels, the principal point was $x_0 = 1028.4$ and $y_0 = 673.4$ pixels and the skew was 0. For a fair comparison of the methods, they

both used the same random samples from the sphere's occluding contour and the same RANSAC threshold τ_{pix}, chosen equal to the image noise level.

The graphs show the mean and standard deviation of the error, which is the Euclidean distance to the ground-truth sphere centres against the varying noise level, number of correct and erroneous points, occlusion level, depth and occluding contour type. We repeated the experiments 1000 times for each value of a varying parameter.

4.3.1.1.1 Robustness with Elliptic Occluding Contours

We formed the elliptic occluding contours. The sphere radius was $R = 0.5$ metres and the sphere centre was generated randomly in each trial as $\mathbf{C} = [x \sim \mathcal{N}(\mu, \sigma^2),\ y \sim \mathcal{N}(\mu, \sigma^2),\ z \sim \mathcal{N}(\mu_z, \sigma_z^2)]^\top$ with $\mu = 0$ and $\sigma^2 = 2$ and $\mu_z = 5$ and $\sigma_z^2 = 1$ metres.

Accuracy versus image noise and number of correct points. We varied the white Gaussian noise level from 0 to 10 pixels with step size of 1 pixel. Experiments were performed with 100 correct points (figure 2 - left). We then varied the number of correct points from 10 points to 100 points with step size of 10 points. Experiments were performed with 2 pixels white Gaussian noise (figure 2 - right). `Robust-SpherO` outperforms `Robust-Toth` in both set of experiments, with errors consistently twice as low.

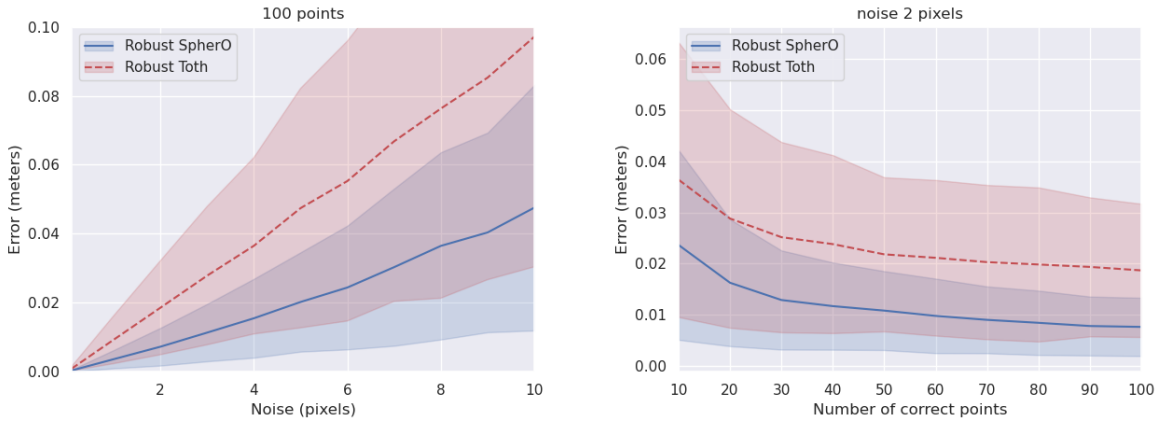

Figure 2: Mean and standard deviation of errors versus image noise levels (left) and number of correct points (right). Correct points belong to the sphere's occluding contour.

Accuracy versus number of erroneous points. We varied the erroneous point rate from 5% to 75% with step size of 5% and using 100 points in each trial. Figure 3 presents the results with the following fixed parameters: 1 pixel white Gaussian noise (left) and 2 pixels white Gaussian noise (right). `Robust-SpherO` outperforms `Robust-Toth`, though a breakpoint appears at 50% and 35% of erroneous point rates, respectively.

Accuracy versus occlusion. We varied the occlusion level from 10% to 70% with step size of 10% over 100 contour points. Figure 4 presents the results with the following fixed parameters: 1 pixel white Gaussian noise and 10% erroneous points (left); 2 pixels white Gaussian noise and 20% erroneous points (right). `Robust-SpherO` outperforms `Robust-Toth`, though a breakpoint appears at 50% and 40% of occlusions, respectively.

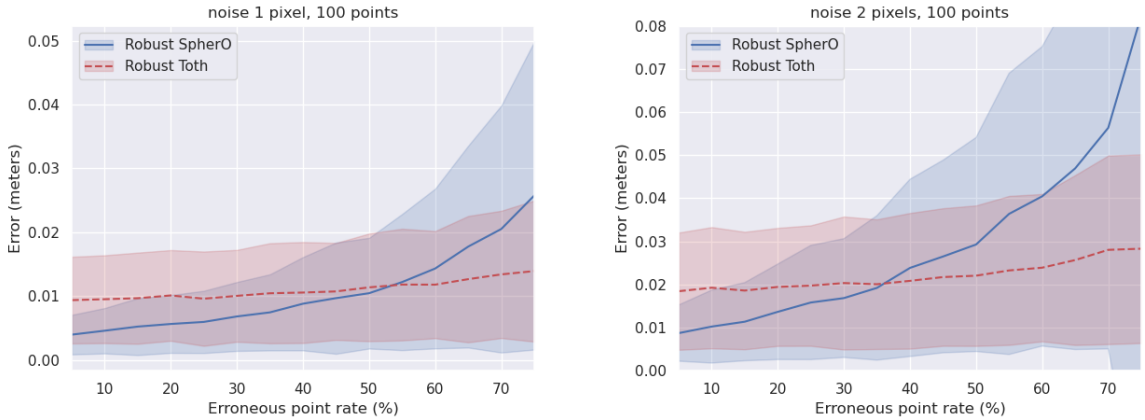

Figure 3: Mean and standard deviation of errors versus erroneous point rates with 1 pixel noise (left) and with 2 pixels noise (right). Erroneous points do not belong to the sphere's occluding contour.

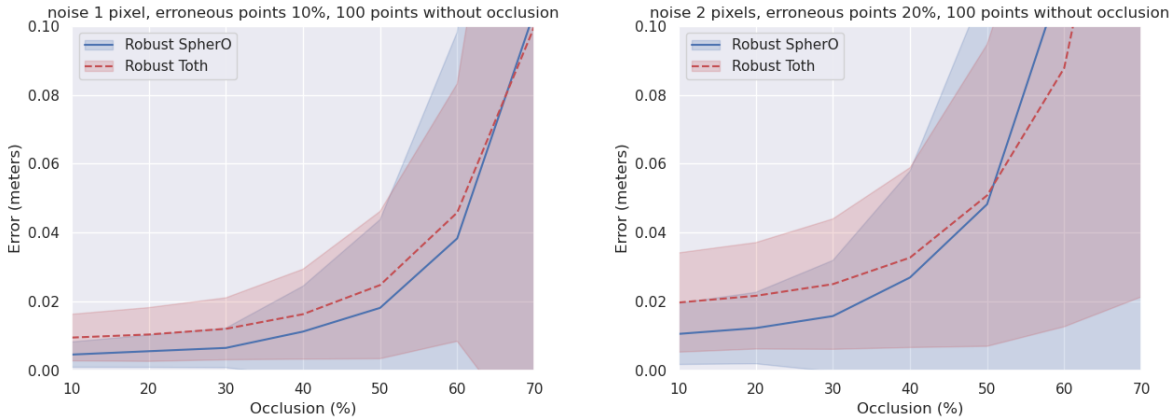

Figure 4: Mean and standard deviation of errors versus occlusion with 1 pixel noise and 10% erroneous points (left) and with 2 pixels noise and 20% erroneous points (right).

Accuracy versus depth. We varied the sphere's depth along the camera's optical axis from 1 to 10 metres with a step size of 1 metre and using 100 points in each trial with 1 pixel and 2 pixels white Gaussian noise, respectively. The error substantially increases beyond 4 metres for both methods. Figure 5 shows that `Robust-SpherO` outperforms `Robust-Toth` with errors consistently at least twice as low.

Synthesis. `Robust-SpherO` is almost always substantially more accurate than its contender `Robust-Toth`. Nonetheless, there exist extreme cases, where the number of erroneous points or the level of occlusion are very large, for which the performance order reverts. `Robust-SpherO` uses 3D points to form a plane's consensus set while `Robust-Toth` uses 2D points to form an ellipse's consensus set. Both methods use the same RANSAC threshold, which is an in-plane distance. `Robust-SpherO` thus includes more points in its consensus set than `Robust-Toth` because some out-of-plane 3D points remain within the range of the RANSAC threshold although they would be out of range if projected onto the consensus plane. The higher the percentage of erroneous points, the higher the

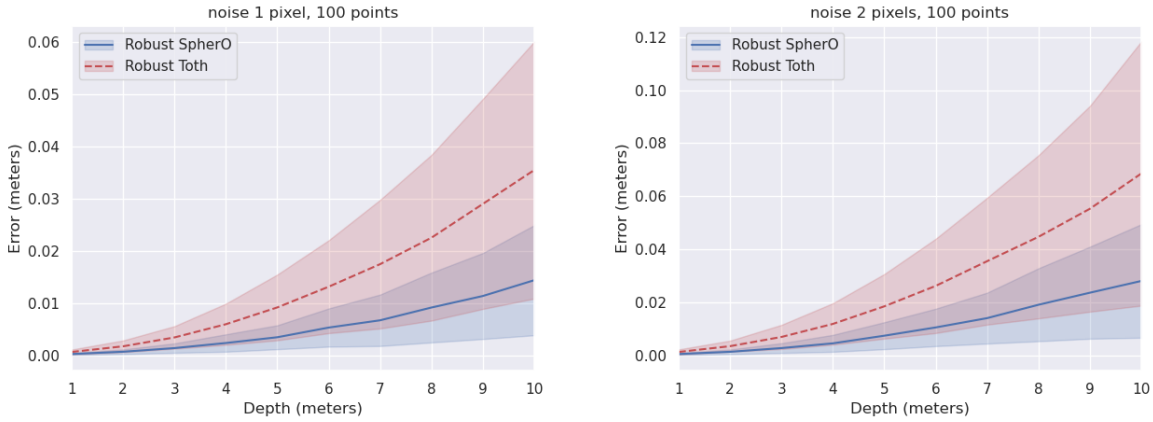

Figure 5: Mean and standard deviation of errors versus depth with 1 pixel noise (left) and with 2 pixels noise (right).

probability of `Robust-SpherO` to include a higher percentage of erroneous points in its consensus set. Yet, `Robust-SpherO` (*i.e.*, plane fitting) reconstructs stably up to a large rate of erroneous points. These extreme cases are however unlikely to happen in real scenarios.

4.3.1.1.2 Robustness with Non-elliptic Occluding Contours

`Robust-Toth` is not designed to handle the parabola (*i.e.*, $\mathbf{C}^\top \underline{\mathbf{z}} = R$) nor the hyperbola (*i.e.*, $\mathbf{C}^\top \underline{\mathbf{z}} < R$), only the ellipse. Subsequently, a parabola yields an `Inf` through division by zero in equation (21) in [14] by a projection parallel to the normalised image plane for a point which otherwise would yield an endpoint of the major axis of an ellipse. However, `Robust-Toth` rarely breaks down in practice for a parabola. This is because a mild noise on a sample of the contour points can easily shift the fit to an ellipse well aligned with the contour points. On the other hand, in a hyperbola case, an endpoint of the major axis of an ellipse is projected to the opposite side of the image from the backside of the camera in equation (21) in [14]. This yields a flipped and scaled ellipse away from the contour points. Subsequently, `Robust-Toth` produces wrong results with a hyperbola. We formed a parabolic occluding contour by placing the sphere centre at $\mathbf{C} = [1.2, 0, 1]^\top$ with radius $R = 1$ in metres. We formed a hyperbolic occluding contour by placing the sphere centre at $\mathbf{C} = [0, -1.2, 0.8]^\top$ again with radius $R = 1$ in metres. We made 10 trials on each occluding contour with 1 pixel image noise and 5% erroneous points. In each trial, the erroneous points were randomly chosen over 100 points which were also randomly chosen from the occluding contour. Figure 6 presents the results. `Robust-SpherO` substantially outperforms `Robust-Toth`.

4.3.1.2 Real Data Experiments

4.3.1.2.1 Relative Accuracy Evaluation

We evaluate the relative accuracy of the compared methods by individually reconstructing multiple spheres whose centre distances are known. We place a target sphere with known radius R in contact with at least three auxiliary spheres whose radii R_i are also known. This yields the ground-truth distances between the target sphere's centre $\mathbf{C}$ and

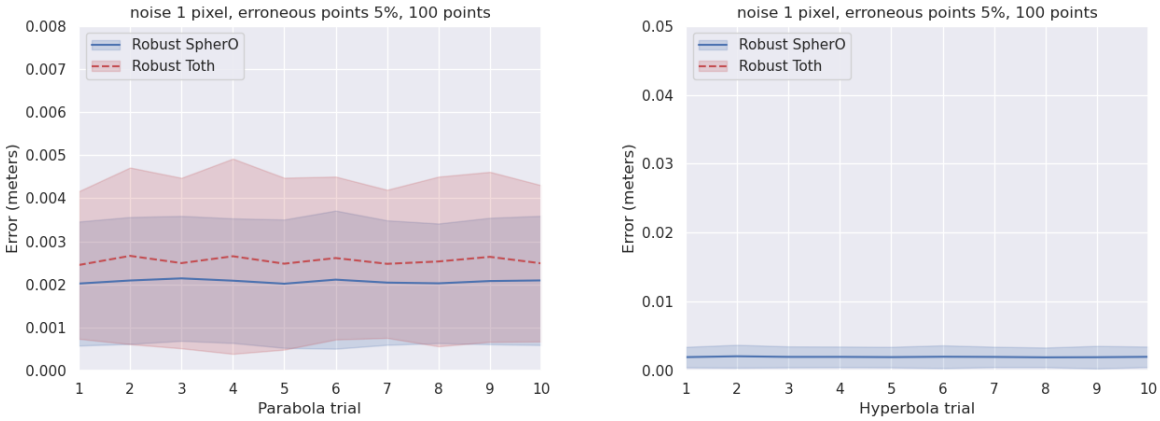

Figure 6: Mean and standard deviation of errors for the parabola (left) and hyperbola (right) occluding contours with 1 pixel noise and 5% erroneous points. For the hyperbola case (right), `Robust-Toth` produced out-of-bound results with a mean of about 0.42 metres.

the auxiliary spheres' centres $\mathbf{C}_i$ as $\| \mathbf{C} - \mathbf{C}_i \| = R + R_i$. Three auxiliary spheres fully constrain the target sphere's centre $\mathbf{C}$, although two solutions exist. We use the compared methods to reconstruct the respective spheres' centres $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{C}}_i$ from which a pairwise error is computed as $e_i = | \| \widehat{\mathbf{C}} - \widehat{\mathbf{C}}_i \| - (R + R_i) |$ for each method. Figure 7 reports the results with 3 auxiliary spheres. The relative error is computed as $\|\mathbf{e}\| = \|[e_1, e_2, e_3]^\top\|$. The target sphere's radius was $8.5\,cm$ (football) and the auxiliary spheres' radii were $3.25\,cm$ (tennis ball), $3.5\,cm$ (green ball) and $4.75\,cm$ (rainbow ball), respectively. The target sphere's centre was about $40\,cm$ away from the camera. We set the RANSAC threshold $\tau_{pix} = 0.5$ pixels which was about twice of the intrinsic calibration accuracy of the camera. We repeated the experiment 1000 times with both methods. `Robust-SpherO` yielded $4.76\,mm$ mean error with $0.01\,mm$ std. `Robust-Toth` yielded $6.77\,mm$ mean error with $1.9\,mm$ std. We also evaluate the relative accuracy of the compared methods

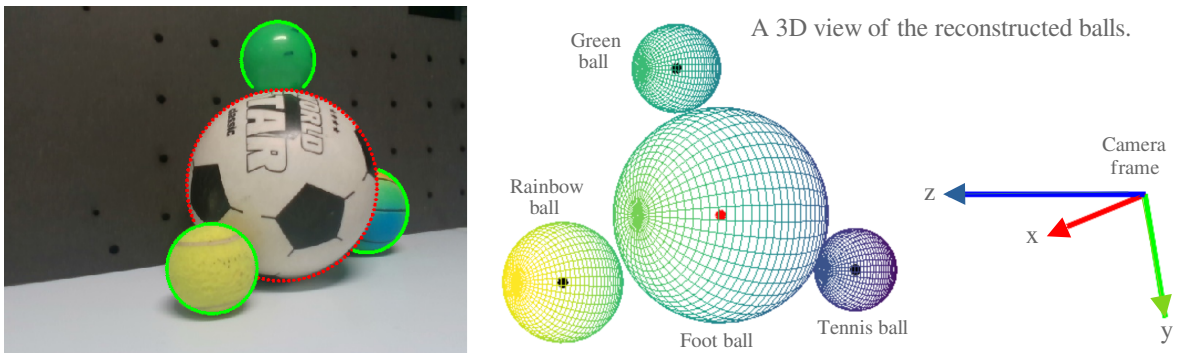

Figure 7: The input image with the spheres' occluding contours (left). A 3D view from behind the balls reconstructed by `Robust-SpherO` (right). The rainbow ball was mostly occluded by the football. We used 100 contour points for each ball.

at different depths. Figure 8 shows the relative reconstruction errors in millimetres of the football in contact with a tennis ball, a small basket ball and a yellow ball. The balls are placed at 1 metre, 2 metres and 3 metres away from the camera, respectively.

Robust-SpherO outperformed Robust-Toth with 5.9 mm, 84.6 mm and 231.1 mm mean errors versus 6.9 mm, 91 mm and 232.2 mm at 1 metre, 2 metres and 3 metres, respectively. We observe that the reconstruction accuracy mostly depends on the quality of the segmented occluding contours. The yellow ball is mostly occluded and segmenting its occluding contours at 2 and 3 metres was significantly difficult. This yielded the worst relative errors for both methods.

	Robust-SpherO	Robust-Toth
$e_{tennisball}$	2.0 ± 1.4	2.0 ± 1.4
$e_{basketball}$	5.6 ± 0.0	6.6 ± 1.0
$e_{yellowball}$	0.3 ± 0.0	0.3 ± 0.0

	Robust-SpherO	Robust-Toth
$e_{tennisball}$	3.5 ± 1.5	3.5 ± 1.5
$e_{basketball}$	0.7 ± 0.0	2.9 ± 2.6
$e_{yellowball}$	84.6 ± 0.0	90.9 ± 12

	Robust-SpherO	Robust-Toth
$e_{tennisball}$	5.2 ± 0.2	5.2 ± 0.2
$e_{basketball}$	4.5 ± 0.0	23 ± 15
$e_{yellowball}$	231 ± 0.0	231 ± 0.4

Figure 8: Mean and standard deviation of relative reconstruction errors (mm) of the football with respect to auxiliary balls. The experiments are repeated 1000 times.

4.3.1.2.2 Absolute Accuracy Evaluation

We used an Intel RealSense 3D camera to measure 3D surface points of the spheres from which the spheres' centres $\mathbf{C}_i$ were computed. We used the methods to reconstruct the sphere centres $\widehat{\mathbf{C}_i}$. We then computed an absolute error $e_i = \| \widehat{\mathbf{C}_i} - \mathbf{C}_i \|$ per sphere. We used the image shown in figure 7 for the experiment, which was taken by the depth camera. We repeated the experiment 1000 times with both methods. Table 2 lists the mean and std of the errors in millimetres with respect to the 3D camera measured centres. We observe that the relative accuracy, as explained in section 4.3.1.2.1, computed from

Table 2: Reconstruction error of the four balls (mean $\pm$ std in mm).

	Football	Tennis ball	Green ball	Rainbow ball
Robust-Toth [14]	3.3 ± 0.0	4.1 ± 1.9	8.2 ± 1.6	15.2 ± 2.5
Robust-SpherO	3.3 ± 0.0	1.9 ± 0.0	4.2 ± 0.0	7.4 ± 0.0

the spheres' centres measured by the 3D camera was 11.4 mm. However, we also observe that the reconstructed spheres are close to their absolute true locations. Robust-SpherO outperforms Robust-Toth by a large margin, having a lower absolute error, both on average and in standard deviation.

In some of the 1000 repetition experiments, the standard deviation values computed from the reconstruction errors of Robust-SpherO remain 0 because in every trial the consensus set included all the occluding contour points of a ball for the given RANSAC threshold.

4.3.1.3 Runtimes

The mean and standard deviation of the runtimes combining the synthetic and real data experiments for Robust-SpherO and Robust-Toth were $7.5 \pm 0.7\,ms$ and $13 \pm 0.5\,ms$,

respectively. `Robust-SpherO` thus brings an overall speed up factor of approximately 1.5 compared to `Robust-Toth`.

4.3.2 Comparison with the Refinement Method

We compared `Robust-SpherO` and its refinement. We used the refining method presented in section 4.2, which was initialised with `Robust-SpherO`. We repeated the synthetic data experiments of section 4.3.1. Table 3 lists the over all mean and std of the errors in millimetres versus the set of experiments. These include noise level, number of correct points, number of erroneous points, occlusion level, and depth experiments. We observe

Table 3: Overall sphere centre reconstruction errors (mm) versus experiments.

	Noise	Correct points	Erroneous points	Occlusion	Depth
`Robust-SpherO`	16.5 ± 12	11.7 ± 5	10.7 ± 6	27.6 ± 37	5.4 ± 4
`Refinement`	17 ± 12	12 ± 5	55 ± 42	29 ± 40	5.5 ± 4

that the difference between `Robust-SpherO` and its refinement is lower than a millimetre in noise level, number of correct points and depth level experiments. The difference between `Robust-SpherO` and its refinement in occlusion level experiments is about a few millimetres, while it is significantly larger in varying number of erroneous points experiments. `Robust-SpherO` outperforms its refinement. We conclude that `Robust-SpherO` does not require refinement, and remains as the most accurate, skillful and fastest sphere reconstruction method.

5 Reconstructing a Sphere and the Camera Focal Length

We propose `SpherOf`. It reconstructs a sphere's centre and the unknown camera focal length. `SpherOf` has three steps. First, following general configuration 1, we spherically normalise the sphere's occluding contour which would generally form a non-circular sphero-ellipse for an incorrect focal length. Second, we reconstruct the camera focal length by solving the coplanarity constraint on the normalised points. This simultaneously reconstructs a sphero-circle (*i.e.*, support plane, centre and radius) which induces a one-parameter sphere family. Third, we retrieve the actual sphere of prescribed radius. The use of spherical normalisation, following general configuration 1, allows one to distinguish the correct focal length from an incorrect one. In the following, we give its first and second steps, and its robustifications `Robust-SpherOfs`. Its third step is the same as section 4.1.3.

5.1 Spherical Normalisation of the Occluding Contour

Spherical normalisation is the first step of `SpherOf`. We spherically normalise the homogeneous occluding contour points using equation (1) with an intrinsic matrix $\widehat{\mathbf{K}}$ written from a focal length estimate $\hat{f}$ and the given principal point $\mathbf{p}_o$, as follows:

$$\widehat{\mathbf{q}}_i \;=\; \eta(\,\widehat{\mathbf{K}}^{-1}\bar{\mathbf{p}}_i\,). \tag{13}$$

This would generally form a non-circular sphero-ellipse for an incorrect focal length. In other words, the spherically-normalised points would generally be non-coplanar.

5.2 Focal Length Reconstruction

Focal length reconstruction is the second step of `SpherOf`. In the following, we present its minimal 4-point case solution and redundant $m \geqslant 4$ point case least-squares solution.

5.2.1 A Polynomial Minimal 4-Point Solution to the Focal Length

As 3 points are always coplanar and cocircular, the minimal case is for 4 points. We formulate the coplanarity constraint using any four spherically-normalised points as follows:

$$\det\left(\begin{bmatrix}\widehat{\mathbf{q}}_1 & \widehat{\mathbf{q}}_2 & \widehat{\mathbf{q}}_3 & \widehat{\mathbf{q}}_4 \\ 1 & 1 & 1 & 1\end{bmatrix}\right) = 0. \tag{14}$$

This coplanarity constraint, expressed by the determinant, will hereafter be referred to as ζ. Geometrically, this implies that the four points form a zero-volume parallelepiped. Let $[x,\, y]^\top = \mathbf{p} - \mathbf{p}_o$ be the coordinates of an image point with the known principal point undone. We then factor equation (14) as:

$$\det\left(\mathbf{N}\,\mathbf{D}\right) = 0, \tag{15}$$

where:

$$\mathbf{N} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ \hat{f} & \hat{f} & \hat{f} & \hat{f} \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \mathtt{diag}(1/\alpha_1, 1/\alpha_2, 1/\alpha_3, 1/\alpha_4), \tag{16}$$

with an $\alpha_j = \sqrt{x_j^2 + y_j^2 + \hat{f}^2}$ defining a point's norm. The determinant constraint can be formulated on the first matrix factor $\mathbf{N}$ only, giving a single equation on $\hat{f}$. Furthermore, we have the following:

$$\det(\mathbf{N}) = \hat{f}\det(\mathbf{L}), \tag{17}$$

with:

$$\mathbf{L} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix}. \tag{18}$$

Note that $\mathbf{L}$ contains the points' norms α_j which are functions of $\hat{f}^2$. Therefore, $\mathbf{L}$ is also a function of $\hat{f}^2$. The coplanarity constraint ζ thus reduces to $\zeta(\,\hat{f}^2\,) = \det(\mathbf{L}(\,\hat{f}^2\,)) = 0$. Let $d_j^2 = x_j^2 + y_j^2$ be the square distance of the image point to the origin and a_j the signed area of the triangle whose vertices are the remaining 3 points when point j is excluded from a minimal 4-point set. By ordering the points along the occluding contour, all signed areas are positive, hence $a_j > 0$. The equation can then be written as a combination of areas weighted by some function of the distance:

$$\zeta(\,\hat{f}^2\,) = a_1\sqrt{d_1^2 + \hat{f}^2} + a_3\sqrt{d_3^2 + \hat{f}^2} - a_2\sqrt{d_2^2 + \hat{f}^2} - a_4\sqrt{d_4^2 + \hat{f}^2} = 0. \tag{19}$$

This is a radical equation, given as the sum of a convex (the square root) and a concave (the opposite square root) functions, which unfortunately cannot be solved in closed-form. We thus search for a polynomial whose roots would contain the radical's solution. We define $c_j = a_j \alpha_j$ and rewrite (19) as:

$$\zeta(\,\hat{f}^2\,) \;=\; c_1 - c_2 + c_3 - c_4 = 0. \tag{20}$$

In order to obtain a polynomial, we need to cancel all square roots appearing in each of the c_j terms by squaring them. We rewrite the equation as:

$$c_1 + c_3 = c_2 + c_4. \tag{21}$$

We use a method involving three squaring rounds. The first squaring round and rearranging gives the following:

$$\begin{aligned} c_1^2 + c_3^2 + 2c_1c_3 &= c_2^2 + c_4^2 + 2c_2c_4 \\ 2(c_1c_3 - c_2c_4) &= -c_1^2 + c_2^2 - c_3^2 + c_4^2. \end{aligned} \tag{22}$$

The second squaring round and rearranging gives the following:

$$\begin{aligned} 4\left(c_1^2c_3^2 + c_2^2c_4^2 - 2c_1c_2c_3c_4\right) &= \left(-c_1^2 + c_2^2 - c_3^2 + c_4^2\right)^2 \\ -8c_1c_2c_3c_4 &= \left(-c_1^2 + c_2^2 - c_3^2 + c_4^2\right)^2 - 4\left(c_1^2c_3^2 + c_2^2c_4^2\right). \end{aligned} \tag{23}$$

The third squaring round finally gives the sought polynomial:

$$\begin{aligned} 64c_1^2c_2^2c_3^2c_4^2 = \left(-c_1^2 + c_2^2 - c_3^2 + c_4^2\right)^4 + 16\left(c_1^2c_3^2 + c_2^2c_4^2\right)^2 \\ -8\left(-c_1^2 + c_2^2 - c_3^2 + c_4^2\right)^2\left(c_1^2c_3^2 + c_2^2c_4^2\right). \end{aligned} \tag{24}$$

The total degree in $\hat{f}^2$ is 4, so this equation is a quartic. This is intuitive as the original $\hat{f}^2$ is encapsulated in a square root; in other words, each c_i^2 is linear in $\hat{f}^2$. Squaring to obtain the quartic introduces spurious solutions. Specifically, we know that the solution is part of the roots, so the quartic has at least 2 real roots (as complex roots come in pairs, real roots too for an even degree polynomial). Overall, we thus have either 2 or 4 real roots. We solve for these roots and keep the solution which best satisfies the original radical equation (19).

5.2.2 Least-Squares Solution to the Focal Length by Fitting Planes

We solve for $\hat{f}$ by optimizing coplanarity of a non-circular sphero-ellipse's $m \;\geqslant\; 4$ points by forming a sphero-circle. This thus reconstructs simultaneously a focal length estimate $\hat{f}$ and a sphero-circle (*i.e.*, $\odot \;=\; [\,\underline{\mathbf{n}}^\top,\; \mathbf{c}^\top,\; r\,]^\top$). We follow two steps.

First, we compute the coplanarity through the best-fitting plane of m points. To do so, we calculate the singular values $\{\,\lambda_1,\, \lambda_2,\, \lambda_3\,\}$ of a $3 \times m$ matrix containing the centered normalised points $\widehat{\mathbf{q}}_i \;-\; \widehat{\mathbf{q}}_0$, where point $\widehat{\mathbf{q}}_0$ is the centroid. The singular values $\{\,\lambda_1,\, \lambda_2,\, \lambda_3\,\}$ are in descending order. The first two singular values λ_1 and λ_2 encode the cocircularity. The third singular value λ_3 quantifies directly the coplanarity. Explicitly, λ_3 is the sum of squared distances of the points to the best-fitting plane.

Second, we optimize an initial focal length estimate $\hat{f}$ by minimizing the following coplanarity cost:

$$\hat{f} \;=\; \underset{\hat{f}}{\arg\min}\;\; \lambda_3. \tag{25}$$

We note that λ_3 is a function of $\{\,\widehat{\mathbf{q}}_i\,\}$ which are functions of $\hat{f}$. Importantly, λ_3 is a continuous function of the matrix entries, from Weyl's inequality. There is no closed-form solution of the singular values in terms of the focal length. Hence we use gradient descent for a numerical minimisation of cost (25). There are three solutions for $\hat{f} \in \{0,\, f^*,\, \infty\}$. Focal length $\hat{f} = 0$ is a singularity where all the normalised points converge to the sphero great circle. Focal length $\hat{f} = \infty$ is another singularity where all the normalised points converge to the sphero pole. Focal length $\hat{f} = f^*$ is the correct solution and lies between bounds f_{lower} and f_{upper} which are usually known. We can bound the solution between $f_{lower} \leqslant \hat{f} \leqslant f_{upper}$ to find a good estimate. The right-singular vector associated to the minimized λ_3 yields the sphero-circle's support plane normal vector $\underline{\mathbf{n}}$. The sphero-circle's support plane depth is $d_{\Pi} = \widehat{\mathbf{q}}_0^{\top}\underline{\mathbf{n}}$. It then follows section 4.1 to reconstruct the sphere's center.

5.3 Robust Solutions from Random Sampling

We perform sampling-based robust sphere and camera focal length reconstruction. We present three robust methods. First, `Robust-SpherOf-ScanN` scans all the focal lengths and applies a 3-point plane RANSAC per focal length. Second, `Robust-SpherOf-Scan1` scans all the focal lengths and uses only one random 3-point plane sample per focal length. Third, `Robust-SpherOf-Quartic` applies 4-point focal length RANSAC and inside 3-point plane RANSACs. All the methods have the same inputs and outputs. The inputs are the sphere's occluding contour pixels, the sphere's prescribed radius, the camera's principle point and the focal length bounds, and the coplanarity consensus threshold. The outputs are the sphere of prescribed radius and the focal length estimate.

5.3.1 `Robust-SpherOf-ScanN`

We present `Robust-SpherOf-ScanN` in algorithm 2. Line 1 scans the focal lengths within

Algorithm 2 `Robust-SpherOf-ScanN`

Input: Sphere's occluding contour pixels $\{\,\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^2,\ i = \{1, \ldots, m\},\ m \geqslant 4\,\}$, sphere's radius $R \in \mathbb{R}^+$ in metric units, and the camera's principal point $\mathbf{p}_o$ and the focal length bounds $[f_{lower},\, f_{upper}]$, consensus threshold $\tau_{pix} \in \mathbb{R}^+$ in pixels.

Output: Sphere $\mathbf{s} \in \mathbb{R}^4$ and focal length estimate $\hat{f}$.

1: **for** $\hat{f}_o$ **from** f_{lower} **to** f_{upper} **with a step of** 1 pixel
2: $\quad\{\,\widehat{\mathbf{q}}_i\,\} = \texttt{Spherical Normalisation}(\{\,\mathbf{p}_i\,\},\, (\hat{f}_o,\, \mathbf{p}_o)\,)$
3: $\quad\tau = \texttt{Threshold Normalisation}(\,\tau_{pix},\, \hat{f}_o\,)$
4: $\quad\mathcal{C} = \texttt{RANSAC Coplanarity Consensus Set}(\,\{\,\widehat{\mathbf{q}}_i\,\},\, \tau\,)$ where $\mathcal{C} \subseteq \{\,\widehat{\mathbf{q}}_i\,\}$
5: $\quad$ 📦 $= \texttt{Stack Into Memory}(\,\mathcal{C},\, \hat{f}_o\,)$
6: **end for**
7: $\{\,\mathcal{C},\, \hat{f}_o\,\} = \texttt{Max Cardinality Consensus Set}($ 📦 $)$ // robust initialization for (25)
8: $\{\,\odot,\, \hat{f}\,\} = \texttt{Refine Coplanarity}(\,\mathcal{C},\, \hat{f}_o\,)$ // minimize (25)
9: $\mathbf{s} = \texttt{Sphere Recovery}(\,\odot,\, R\,)$ // section 4.1.3

the bounds with an incremental step of 1 pixel. Line 2 spherically normalizes the occluding contour using the chosen focal length. Line 3 transforms the coplanarity consensus threshold from pixel units to a distance in the normalized image plane as $\tau = \tau_{pix}/\hat{f}_o$. Line 4 applies RANSAC on samples of 3 points to form the best coplanarity consensus

set for the chosen focal length. RANSAC iterates on each sample of 3 points to define *(i)* a plane Π, *(ii)* a consensus threshold scaled by the plane's depth $\tau_{\Pi} = |d_{\Pi}|\,\tau$, and *(iii)* a consensus set using *(i)* and *(ii)*. Line 5 stacks the coplanarity consensus set and its focal length into the memory. Line 6 ends the scanning of the focal lengths. Line 7 retrieves the largest consensus set and its corresponding focal length. This yields a robust initial guess for the refinement. Line 8 refines the robust initial guess over the largest consensus set by minimizing (25). This yields a focal length estimate and a sphero-circle. Line 9 retrieves the sphere of prescribed radius.

5.3.2 `Robust-SpherOf-Scan1`

We present `Robust-SpherOf-Scan1` in algorithm 3. Line 1 initializes the focal length

Algorithm 3 `Robust-SpherOf-Scan1`

Input: Sphere's occluding contour pixels $\{\,\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^2,\ i = \{1, \dots, m\},\ m \geqslant 4\,\}$, sphere's radius $R \in \mathbb{R}^+$ in metric units, and the camera's principal point $\mathbf{p}_o$ and the focal length bounds $[f_{lower},\ f_{upper}]$, consensus threshold $\tau_{pix} \in \mathbb{R}^+$ in pixels.

Output: Sphere $\mathbf{s} \in \mathbb{R}^4$ and focal length estimate $\hat{f}$.

1: $\hat{f} = 0,\ iter = 0,\ N = 3$ // initialize the variables
2: **while** $(\,\hat{f} < f_{lower}$ **or** $\hat{f} > f_{upper}\,)$ **and** $(iter < N)$
3: **for** $\hat{f}_o$ **from** f_{lower} **to** f_{upper} **with a step of** 1 pixel
4: $\{\,\widehat{\mathbf{q}}_i\,\}$ = `Spherical Normalisation`$(\{\,\mathbf{p}_i\,\}, (\hat{f}_o, \mathbf{p}_o))$
5: $\{\,\widehat{\mathbf{q}}_1, \widehat{\mathbf{q}}_2, \widehat{\mathbf{q}}_3\,\}$ = `Randomly Select 3 Points`$(\{\,\widehat{\mathbf{q}}_i\,\})$ // one sample of 3 points
6: Π = `Minimal Plane Solution`$(\{\,\widehat{\mathbf{q}}_1, \widehat{\mathbf{q}}_2, \widehat{\mathbf{q}}_3\,\})$
7: τ = `Threshold Normalisation`$(\tau_{pix}, \hat{f}_o)$
8: $\mathcal{C}$ = `Coplanarity Consensus Set`$(\Pi, \{\,\widehat{\mathbf{q}}_i\,\}, \tau)$ where $\mathcal{C} \subseteq \{\,\widehat{\mathbf{q}}_i\,\}$
9: 📦 = `Stack Into Memory`$(\mathcal{C}, \hat{f}_o)$
10: **end for**
11: $\{\,\mathcal{C}, \hat{f}_o\,\}$ = `Max Cardinality Consensus Set`$(📦)$ // robust initialization for (25)
12: $\{\,\odot, \hat{f}\,\}$ = `Refine Coplanarity`$(\mathcal{C}, \hat{f}_o)$ // minimize (25)
13: $iter = iter + 1$
14: **end while**
15: $\mathbf{s}$ = `Sphere Recovery`$(\odot, R)$ // section 4.1.3

estimate and the loop parameters. Line 2 checks if the focal length estimate is within the bounds. If not, then the focal length estimation is repeated maximum N times. Line 3 scans the focal lengths within the bounds with an incremental step of 1 pixel. Line 4 spherically normalizes the occluding contour using the chosen focal length. Line 5 selects randomly one sample of 3 points from the spherically-normalized points. This is the minimum number required to form a plane. Line 6 builds a plane from these 3 points. Line 7 transforms the coplanarity consensus threshold from pixel units to a distance in the normalized image plane as $\tau = \tau_{pix}/\hat{f}_o$. Line 8 forms the coplanarity consensus set with respect to plane Π using a threshold scaled by its depth $\tau_{\Pi} = |d_{\Pi}|\,\tau$. Line 9 stacks the coplanarity consensus set and its focal length into the memory. Line 10 ends the scanning of the focal lengths. Line 11 retrieves the largest consensus set and its corresponding focal length. This yields a robust initial guess within k-pixels neighborhood of the ground-truth focal length. For instance, if there exists 50% erroneous points in the occluding contour, then $k = 35$ pixels. Line 12 refines the robust initial guess of the largest consensus set

by minimizing (25). This yields a focal length estimate and a sphero-circle. Line 13 increments the while loop counter. Line 14 ends the focal length estimation while loop. Line 15 retrieves the sphere of prescribed radius.

5.3.3 `Robust-SpherOf-Quartic`

We present `Robust-SpherOf-Quartic` in algorithm 4. Line 1 initializes the RANSAC

Algorithm 4 `Robust-SpherOf-Quartic`

Input: Sphere's occluding contour pixels $\{\, \mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^2,\ i = \{1, \ldots, m\},\ m \geqslant 4 \,\}$, sphere's radius $R \in \mathbb{R}^+$ in metric units, and the camera's principal point $\mathbf{p}_o$ and the focal length bounds $[f_{lower},\ f_{upper}]$, consensus threshold $\tau_{pix} \in \mathbb{R}^+$ in pixels.

Output: Sphere $\mathbf{s} \in \mathbb{R}^4$ and focal length estimate $\hat{f}$.

1: $N = \infty,\ iter = 0,\ \epsilon_{worst} = 0.99,\ p = 0.99,\ s = 4$ // RANSAC parameters
2: **while**$(\, N > iter \,)$
3: $\quad \{\, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4 \,\} =$ `Randomly Select 4 Points`$(\{\, \mathbf{p}_i \,\})$ // one sample of 4 points
4: $\quad \hat{f}_o =$ `Focal from Quartic`$(\{\, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4 \,\}, \mathbf{p}_o, [f_{lower}, f_{upper}])$
5: $\quad \{\, \widehat{\mathbf{q}}_i \,\} =$ `Spherical Normalisation`$(\{\, \mathbf{p}_i \,\}, (\hat{f}_o, \mathbf{p}_o))$
6: $\quad \tau =$ `Threshold Normalisation`$(\, \tau_{pix}, \hat{f}_o \,)$
7: $\quad \mathcal{C} =$ `RANSAC Coplanarity Consensus Set`$(\, \{\, \widehat{\mathbf{q}}_i \,\}, \tau \,)$ where $\mathcal{C} \subseteq \{\, \widehat{\mathbf{q}}_i \,\}$
8: $\quad$ 📦 $=$ `Stack Into Memory`$(\mathcal{C}, \hat{f}_o)$
9: $\quad \{\, N, \epsilon_{worst} \,\} =$ `Update Max Iter`$(\, |\mathcal{C}|, \epsilon_{worst}, p, s \,)$
10: $\quad iter = iter + 1$
11: **end while**
12: $\{\, \mathcal{C}, \hat{f}_o \,\} =$ `Max Cardinality Consensus Set`$(\,$📦$\,)$ // robust initialization for (25)
13: $\{\, \odot, \hat{f} \,\} =$ `Refine Coplanarity`$(\mathcal{C}, \hat{f}_o)$ // minimize (25)
14: $\mathbf{s} =$ `Sphere Recovery`$(\, \odot, R \,)$ // section 4.1.3

parameters. N is the number of maximum iterations. ϵ_{worst} is the ratio of erroneous points. p is the probability of success. s is the minimum number of points required to solve for the focal length. Line 2 starts the RANSAC loop. Line 3 selects randomly 4 points from the image points. Line 4 computes the focal length from the roots of the quartic polynomial which remains within the bounds. Line 5 spherically normalizes the occluding contour using the computed focal length. Line 6 transforms the coplanarity consensus threshold from pixel units to a distance in the normalized image plane as $\tau = \tau_{pix} / \hat{f}_o$. Line 7 applies RANSAC on samples of 3 points to form the best coplanarity consensus set for the computed focal length. RANSAC iterates on each sample of 3 points to define *(i)* a plane Π, *(ii)* a consensus threshold scaled by the plane's depth $\tau_{\Pi} = |d_{\Pi}|\, \tau$, and *(iii)* a consensus set using *(i)* and *(ii)*. Line 8 stacks the coplanarity consensus set and its focal length into the memory. Line 9 updates the number of maximum iterations. Line 10 increments the RANSAC loop counter. Line 11 ends the RANSAC loop. Line 12 retrieves the largest consensus set and its corresponding focal length. This yields a robust initial guess for the refinement. Line 13 refines the robust initial guess over the largest consensus set by minimizing (25). This yields a focal length estimate and a sphero-circle. Line 14 retrieves the sphere of prescribed radius.

5.4 Experimental Results

We compare the proposed three uncalibrated robust methods, `Robust-SpherOf-ScanN`, `Robust-SpherOf-Scan1` and `Robust-SpherOf-Quartic`, through synthetic and real data experiments. In the sequel, we abbreviate them as `ScanN`, `Scan1` and `Quartic` for the sake of simplicity. In synthetic data experiments, we also included the proposed calibrated method `Robust-SpherO` for comparison purposes.

5.4.1 Synthetic Data Experiments

We test the methods' robustness against elliptic and non-elliptic occluding contours. The ground-truth focal length f^* and the principal point $\mathbf{p}_o$ are generated randomly in each and every reconstruction trial. The ground-truth focal length is generated between the bounds $f_{lower} \leqslant f^* \leqslant f_{upper}$ where $f_{lower} = 200$ pixels and $f_{upper} = 2000$ pixels. The principal point is generated as $\mathbf{p}_o = [\, x_o \sim \mathcal{N}(\mu_{x_o}, \sigma^2)\,,\ y_o \sim \mathcal{N}(\mu_{y_o}, \sigma^2)\,]^\top$ with $\mu_{x_o} = 1000$ and $\mu_{y_o} = 600$ and $\sigma^2 = 50$ pixels. RANSAC threshold τ_{pix} is the same for all the methods and set as for all the experiments $\tau_{pix} = 1$ pixel.

The graphs show the mean and standard deviation of the error. The errors are the Euclidean distance to the ground-truth sphere centres and the difference from the ground-truth focal lengths in terms of percentages. The graphs present the results against the varying noise level, number of correct and erroneous points, occlusion level, depth and occluding contour type. We repeated the experiments 1000 times for each value of a varying parameter.

5.4.1.1 Robustness with Elliptic Occluding Contours

We formed the elliptic occluding contours. The sphere radius was $R \sim \mathcal{N}(\mu_R, \sigma_R^2)$ with $\mu_R = 0.075$ and $\sigma_R^2 = 0.025$ metres. The sphere centre was generated randomly in each trial as $\mathbf{C} = [x \sim \mathcal{N}(\mu, \sigma^2)\,,\ y \sim \mathcal{N}(\mu, \sigma^2)\,,\ z \sim \mathcal{N}(\mu_z, \sigma_z^2)\,]^\top$ with $\mu = 0$ and $\sigma^2 = 0.3$ and $\mu_z = 4R$ and $\sigma_z^2 = 2R$ metres.

Accuracy versus image noise. We varied the white Gaussian noise level from 1 to 10 pixels with step size of 1 pixel. Experiments were performed with 100 correct points. Figure 9 shows that the three uncalibrated robust methods perform similarly well.

Accuracy versus number of correct points. We varied the number of correct points from 10 points to 100 points with step size of 10 points. Experiments were performed with 2 pixels white Gaussian noise. Figure 10 shows that the three uncalibrated robust methods perform similarly well.

Accuracy versus number of erroneous points. We varied the erroneous point rate from 5% to 75% with step size of 5% and using 100 points in each trial. Experiments were performed with 2 pixels white Gaussian noise. Figure 11 shows that `Quartic` outperforms `Scan1` which outperforms `ScanN`. The accuracy of all the uncalibrated methods substantially degrades after 40% of erroneous points.

Accuracy versus occlusion. We varied the occlusion level from 10% to 70% with step size of 10% over 100 contour points. Figure 12 presents the results with 1 pixel white Gaussian noise and 10% erroneous points. `Quartic` outperforms `Scan1` and `ScanN`. A breakpoint appears for all uncalibrated methods at 30% of occlusion.

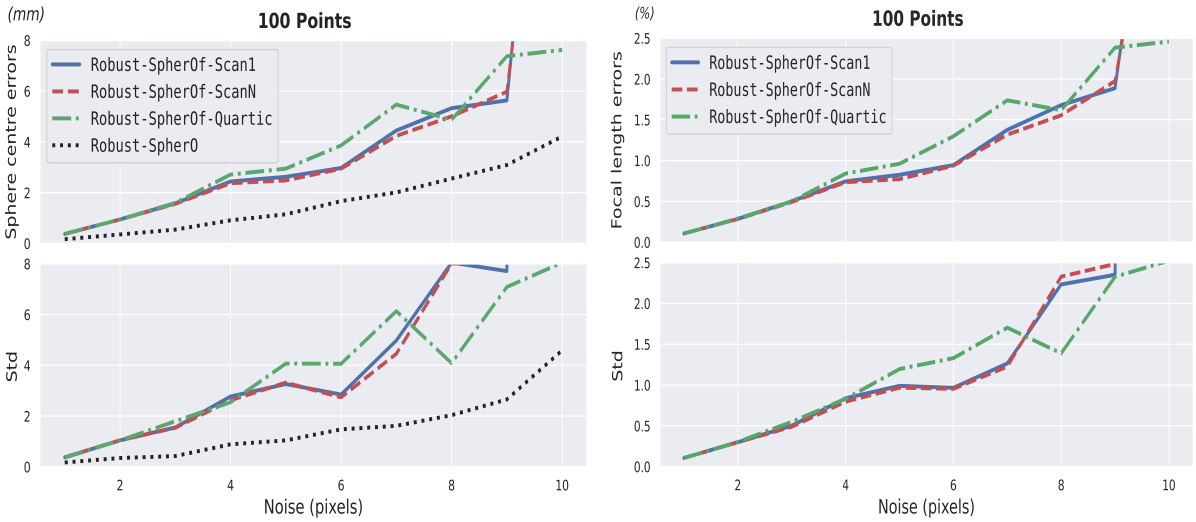

Figure 9: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus image noise levels.

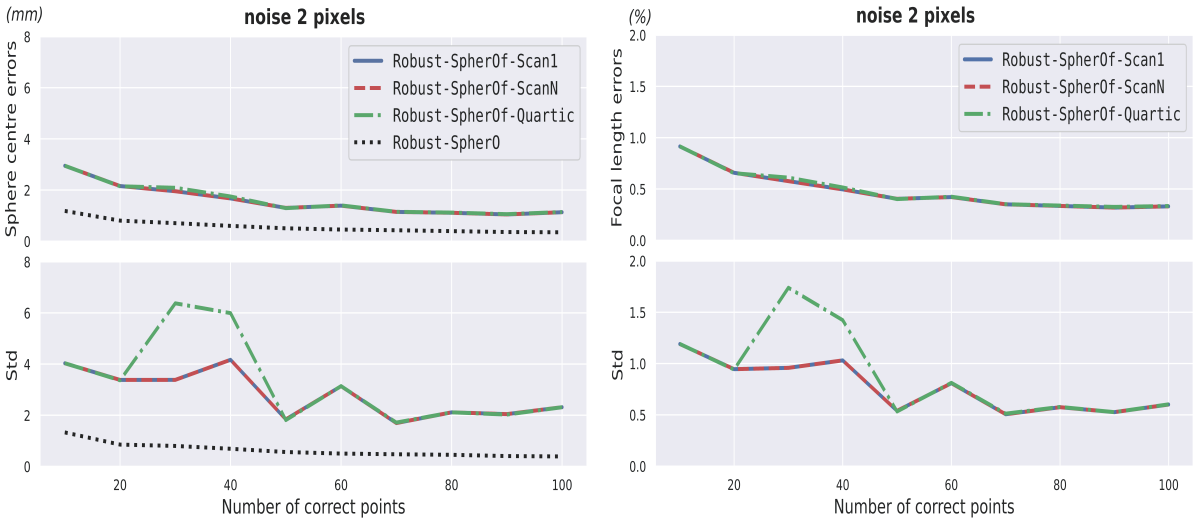

Figure 10: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus number of correct points. Correct points belong to the occluding contour.

Accuracy versus depth. We varied the sphere's depth along the camera's optical axis from $3R$ to $10R$ metres with a step size of R metres and using 100 points in each trial with 2 pixels white Gaussian noise. Figure 13 shows that the three uncalibrated methods perform similarly well up to $5R$. After this point they perform worse.

5.4.1.2 Robustness with Non-elliptic Occluding Contours

`SpherOf` can also handle parabolic (*i.e.*, $\mathbf{C}^\top \underline{\mathbf{z}} = R$) and hyperbolic (*i.e.*, $\mathbf{C}^\top \underline{\mathbf{z}} < R$) occluding contours, since it inherits this skill from `SpherO`. We tested `Scan1`, `ScanN` and `Quartic` on the parabolic and hyperbolic occluding contours defined as in section 4.3.1.1.2. We formed a parabolic occluding contour by placing the sphere centre

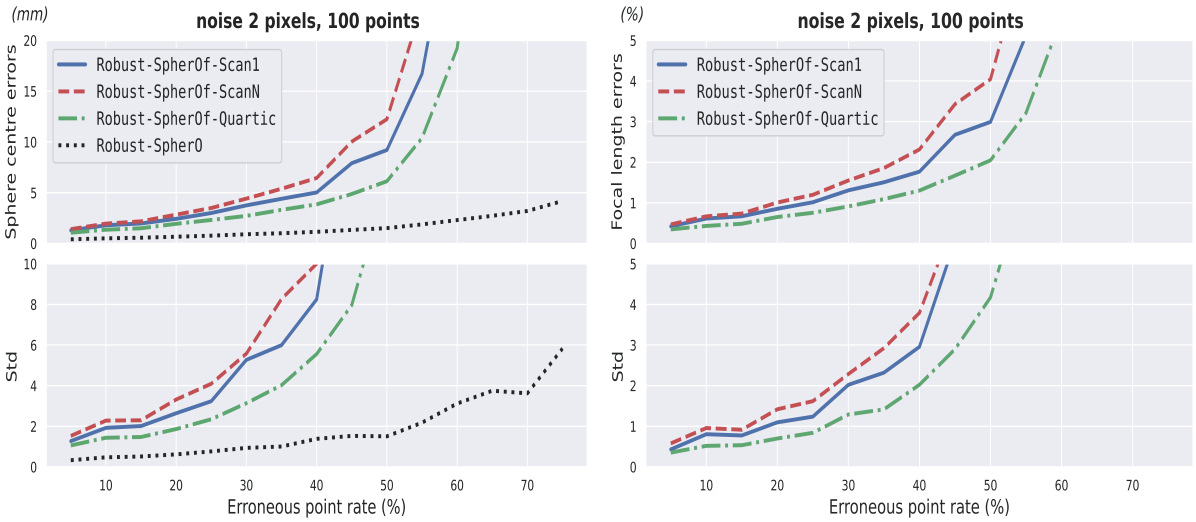

Figure 11: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus erroneous point rates with 2 pixels noise. Erroneous points do not belong to the occluding contour.

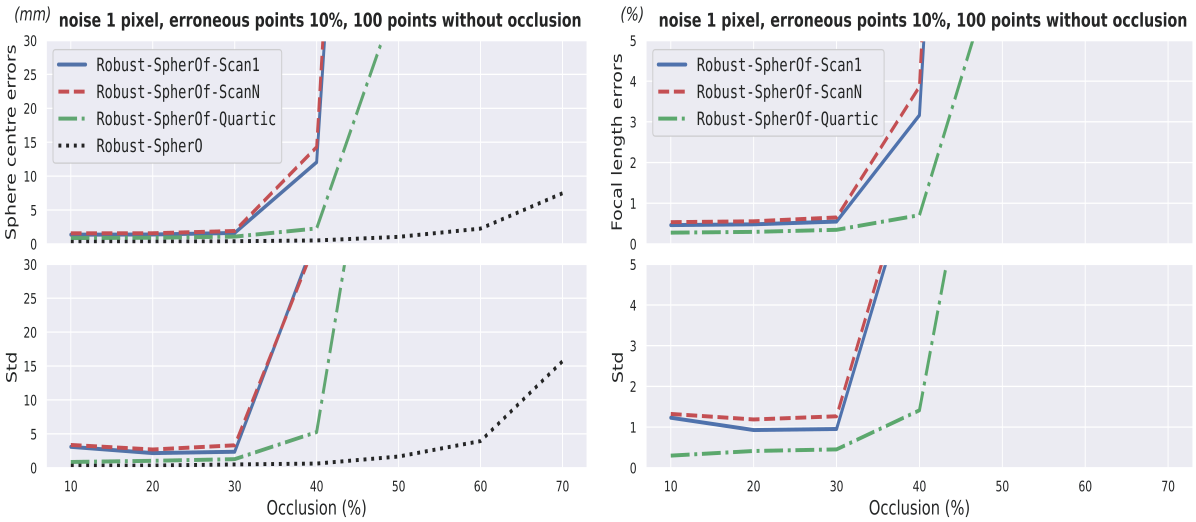

Figure 12: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus occlusion with 1 pixel noise and 10% erroneous points.

at $\mathbf{C} = [1.2, 0, 1]^\top$ with radius $R = 1$ in metres. We formed a hyperbolic occluding contour by placing the sphere centre at $\mathbf{C} = [0, -1.2, 0.8]^\top$ again with radius $R = 1$ in metres. We made 10 trials on each occluding contour with 1 pixel image noise and 5% erroneous points. In each trial, the erroneous points were randomly chosen over 100 points which were also randomly chosen from the occluding contour. Figures 14 and 15 present the results. Three uncalibrated robust methods perform similarly well. `ScanN` slightly outperforms `Scan1` and `Quartic`.

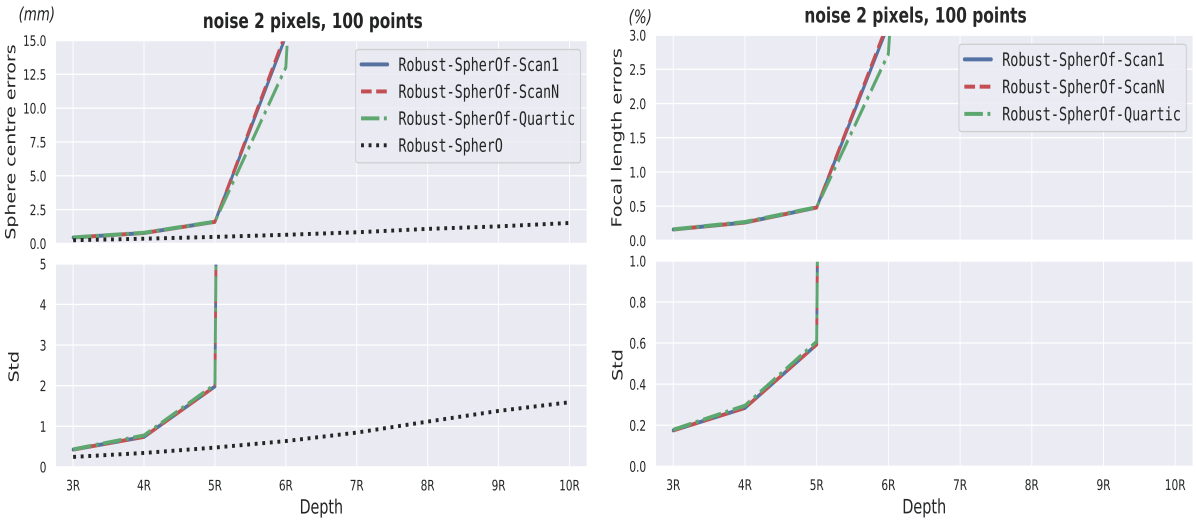

Figure 13: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus depth with 2 pixels noise.

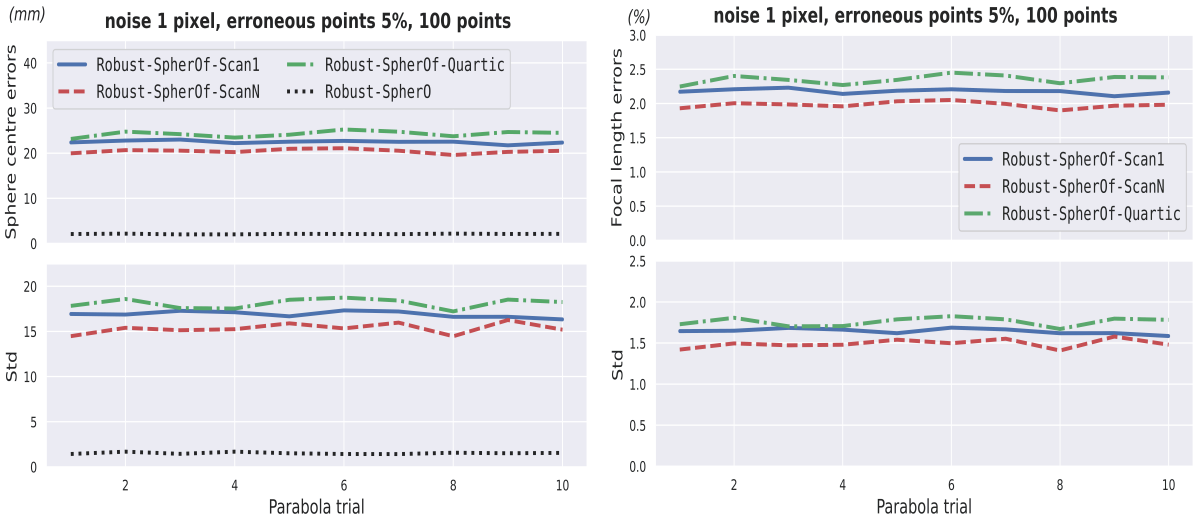

Figure 14: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus parabolic occluding contours with 1 pixel noise and 5% erroneous points.

5.4.1.3 Discussion on Calibrated and Uncalibrated Methods

The calibrated robust method `Robust-SpherO` consistently outperforms the uncalibrated methods in all experiments. `Robust-SpherO` is also stabler. In elliptic occluding contour experiments, both the calibrated and uncalibrated robust methods have comparable accuracies, if *(i)* the noise level is about 2 pixels, *(ii)* the erroneous point rate is lower than 10%, *(iii)* the occlusion level is lower than 30%, and *(iv)* the depth is lower than $5R$.

5.4.2 Real Data Experiments

We compared `Scan1`, `ScanN` and `Quartic` using the occluding contours of three balls shown in figure 16. The image resolution is 1920×1080 pixels. The ground-truth focal

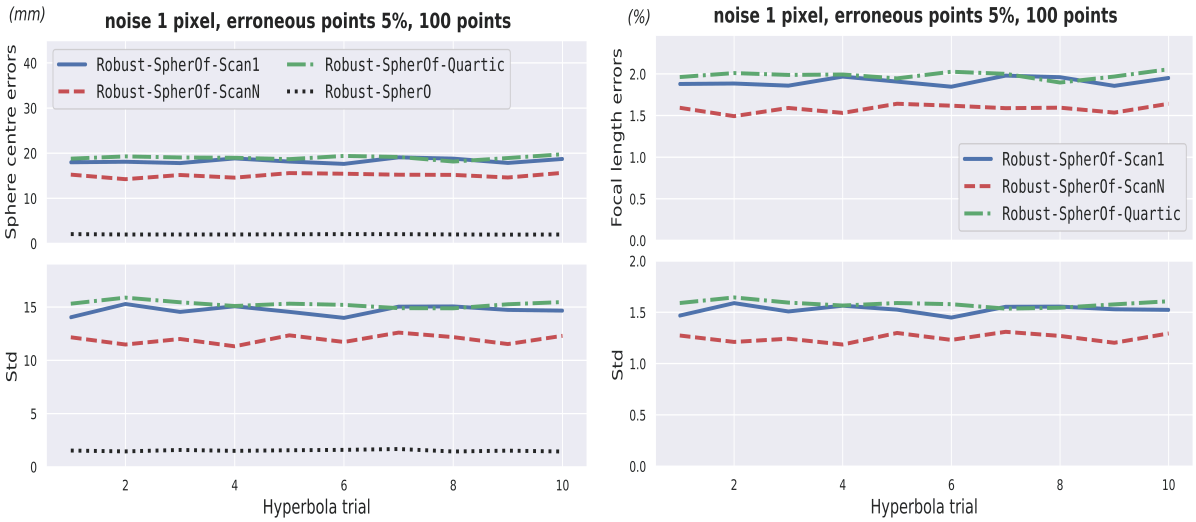

Figure 15: Mean and standard deviation of sphere centre (left, in mm) and focal length (right, in %) errors versus hyperbolic occluding contours with 1 pixel noise and 5% erroneous points.

length is $f^* = 1364.6$ pixels. We use bounds $f_{lower} = 200$ pixels and $f_{upper} = 2000$ pixels. The given principal point is $\mathbf{p}_o = [979.227, 536.237]^\top$ pixels. RANSAC threshold is set as $\tau_{pix} = 0.1$ pixels. The balls are approximately $30\,cm$ away from the camera. The football's radius is $8.5\,cm$. The green ball's radius is $4\,cm$. The yellow ball's radius is $3.5\,cm$. We used `Robust-SpherO` to compute the balls' centres and considered them as the ground-truths. Fastest runtimes are denoted as 1x in tables below.

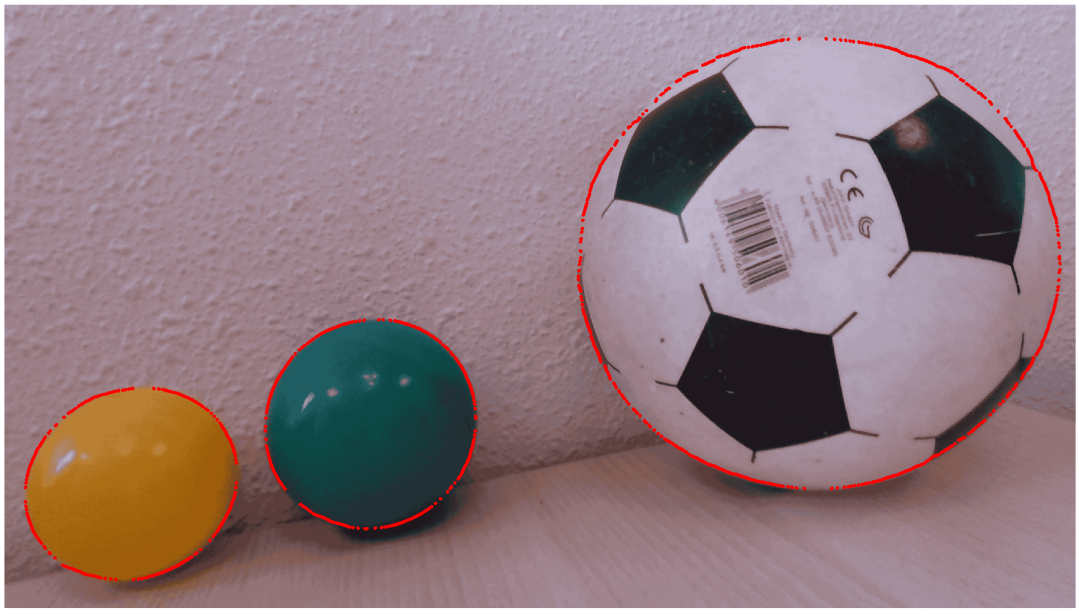

Figure 16: The balls with their segmented occluding contours.

Table 4 presents the reconstruction errors of the three methods using the football's occluding contour as well as the runtimes. Table 5 presents the reconstruction errors of the three methods using the green ball's occluding contour as well as the runtimes. Table 6 presents the reconstruction errors of the three methods using the yellow ball's

Table 4: Reconstruction errors using football's occluding contour and runtimes.

	Focal length	Sphere centre	Runtime
`Robust-SpherOf-Scan1`	3.15 %	7 mm	1x
`Robust-SpherOf-ScanN`	0.8 %	2.1 mm	531x
`Robust-SpherOf-Quartic`	1 %	2.6 mm	935x

Table 5: Reconstruction errors using green ball's occluding contour and runtimes.

	Focal length	Sphere centre	Runtime
`Robust-SpherOf-Scan1`	2.8 %	7.8 mm	1x
`Robust-SpherOf-ScanN`	0.5 %	1.13 mm	73x
`Robust-SpherOf-Quartic`	1.6 %	5.1 mm	1.6x

occluding contour as well as the runtimes.

Table 6: Reconstruction errors using yellow ball's occluding contour and runtimes.

	Focal length	Sphere centre	Runtime
`Robust-SpherOf-Scan1`	2.5 %	6.6 mm	1.2x
`Robust-SpherOf-ScanN`	1.9 %	5.5 mm	40x
`Robust-SpherOf-Quartic`	1.6 %	4.2 mm	1x

6 Conclusion

We have proposed two novel methods `SpherO` and `SpherOf` to reconstruct a sphere from a single view of its occluding contour. `SpherO` reconstructs a sphere of a prescribed radius from a single calibrated view of its occluding contour. It is accurate, simple and fast. Its robustified version `Robust-SpherO` outperforms the state of the art. In addition, `SpherO` provides a closed-form, convex solution for the image conic from the sphere's occluding contour points, minimising a geometric error criterion. `SpherOf` reconstructs simultaneously a sphere of a prescribed radius and the unknown camera focal length from a single view of the sphere's occluding contour given the principal point. `SpherOf` inherits all the advantages of `SpherO`. It is the first minimal method which can reconstruct a sphere and the camera focal length from 4 points. Its robustified versions `Robust-SpherOf`s form the state of the art. We also highlight that `SpherOf` can be directly extended to a multi-view method. More precisely, it can allow one to locate an uncalibrated camera (without knowing the focal length) thanks to a sphere. As the scale of an uncalibrated reconstruction is arbitrary, it is indistinguishable from the sphere's radius. We can thus reconstruct with a simple two-step algorithm: *(i)* randomly choose a radius $R > 0$, and *(ii)* for $i = 1$ to n (the number of images), use `SpherOf` to reconstruct the camera pose i and the focal length i. As future work, we shall *(i)* address the reconstruction of a sphere from multiple uncalibrated views of its occluding contour, *(ii)* study the reconstruction

of different geometric objects such as the ellipsoid and the circle, and *(iii)* propose a theoretical framework to study the degenerate cases in both problems.

Acknowledgment

This work is funded by project ANR JCJC - IMMORTALLS.

Data Availability Statement

No datasets were generated or analysed during the current study.

References

[1] Marcel Berger. Geometry II. *Springer-Verlag, Berlin Heidelberg. (pp. 312)*, 1987.

[2] M. Chasles and C. Graves. *Two geometrical memoirs on the general properties of cones of the second degree, and on the spherical conics*. Dublin : For Grant and Bolton, 1841.

[3] O. Edlund. *Some Notes on Least Squares, QR-factorization, SVD and Fitting*. Lecture notes on Numerical Analysis, Luleå University of Technology, 2013.

[4] P. Gurdjos, A. Bartoli, and P. Sturm. Is dual linear self-calibration artificially ambiguous? In *IEEE International Conference on Computer Vision ICCV*, 2009.

[5] R. Halir and J. Flusser. Numerically stable direct least squares fitting of ellipses. In *WSCG98*, 1998.

[6] K. Han, Kwan-Yee Kenneth Wong, and Xiao Tan. Single view 3d reconstruction under an uncalibrated camera and an unknown mirror sphere. *Fourth International Conference on 3D Vision (3DV)*, pages 408–416, 2016.

[7] C. Lu, S. Xia, M. Shao, and Y. Fu. Arc-support line segments revisited: An efficient high-quality ellipse detection. *IEEE Transactions on Image Processing*, 29:768–781, 2020.

[8] C. Meng, Z. Li, X. Bai, and F. Zhou. Arc adjacency matrix-based fast ellipse detection. *IEEE Transactions on Image Processing*, 29:4406–4420, 2020.

[9] Erol Ozgur, Mohammad Alkhatib, Youcef Mezouar, and Adrien Bartoli. Reconstructing spheres by fitting planes. In *British Machine Vision Conference BMVC*, 2024.

[10] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32:7–25, 1999.

[11] K. Shi, X. Li, and H. et al. Xu. Sphere localization from a minimal number of points in a single image. In *Proceedings of the international conference on advances in computer technology, information science and communications–Volume 1: CTISC, INSTICC. SciTePress*, pages 65–70, 2019.

[12] Peter Sturm and Pau Gargallo. Conic fitting using the geometric distance. In *8th Asian Conference on Computer Vision (ACCV)*, pages 784–795, 2007.

[13] J. Sun, H. He, and D. Zeng. Global calibration of multiple cameras based on sphere targets. *Sensors*, 16, 2016.

[14] T. Toth and L. Hajder. A minimal solution for image-based sphere estimation. *International Journal of Computer Vision*, 131:1428–1447, 2023.

[15] Kwan-Yee K. Wong, Dirk Schnieders, and Shuda Li. Recovering light directions and camera poses from a single sphere. In *European Conference on Computer Vision (ECCV)*, pages 631–642, 2008.

[16] Zhengyou Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing (IVC)*, 15:59–76, 1997.