



A Minimal Solution for Image-Based Sphere Estimation

Tekla Tóth¹ · Levente Hajder¹

Received: 11 February 2022 / Accepted: 25 January 2023 / Published online: 2 March 2023
© The Author(s) 2023

Abstract

We propose a novel minimal solver for sphere fitting via its 2D central projection, *i.e.*, a special ellipse. The input of the presented algorithm consists of contour points detected in a camera image. General ellipse fitting problems require five contour points. However, taking advantage of the isotropic spherical target, three points are enough to define the tangent cone parameters of the sphere. This yields the sought ellipse parameters. Similarly, the sphere center can be estimated from the cone if the radius is known. These proposed geometric methods are rapid, numerically stable, and easy to implement. Experimental results—on synthetic, photorealistic, and real images—showcase the superiority of the proposed solutions to the state-of-the-art methods. A real-world LiDAR-camera calibration application justifies the utility of the sphere-based approach resulting in an error below a few centimeters.

Keywords Ellipse fitting · Sphere fitting · Minimal solver · Image-based reconstruction

1 Introduction

Target-based sensor calibration is a fundamental topic in 3D computer vision. The efficiency of these methods depends on the accuracy and robustness of the object detection algorithms. This paper focuses on spherical targets for robust fitting and extrinsic camera calibration, as they are easy to identify in camera images owing to the rotation symmetry. In the image plane, the perspective projection of a sphere is an ellipse; thus, its contour detection becomes an essential step for sphere recognition.

Consequently, we propose sample-minimizing solutions for the special ellipse and the sphere parameter estimation. The methods take advantage of the spherical projection since the shape of the ellipse is restricted by the sphere and the intrinsic camera parameters. These approaches require only three contour points as opposed to the general five-point problem of ellipse fitting.

Spherical calibration. The industry often utilizes spherical targets for calibrating sensors with different modalities (Ruijl et al. 2001; Klüing and Meli 2005). They are exploited in the field of optics (Xiong et al. 2021; Liu et al. 2017). Depth cameras or laser scanners, like LiDAR, generate three-dimensional pointclouds where spherical objects are readily distinguished. As this shape is represented by the coordinates of the center in space and the radius, the search space has four variables that can be accurately estimated. If the radius of the sphere is known, the degrees of freedom become three. This paper focuses only on estimation of extrinsic parameters even if intrinsic ones can also be computed using at least three spherical objects (Lu and Payandeh 2010).

Most of the related papers do not consider the geometric advantage of the setup. Numerous related publications advise a contrastive background and target object during real-world tests. For instance, Sun et al. (2016) use white spheres in front of a black board. Guan et al. (2015) apply the sphere as the only light source.

Robust Detectors. We strive to find a robust method without restrictions during measurements to improve our sphere-based calibration pipeline (Tóth et al. 2020). This previous work approximates the spherical projection with a circle and detects the ellipse contour with Random Sampling Consensus (RANSAC) algorithm (Fischler and Bolles 1981). Recently, RANSAC and its variants (Zuliani et al. 2005; Chum and Matas 2005; Barath and Matas 2021) have been

Communicated by Federica Arrigoni.

✉ Tekla Tóth
teklatoth@inf.elte.hu

Levente Hajder
hajder@inf.elte.hu
http://cv.inf.elte.hu/

¹ Geometric Computer Vision Group, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

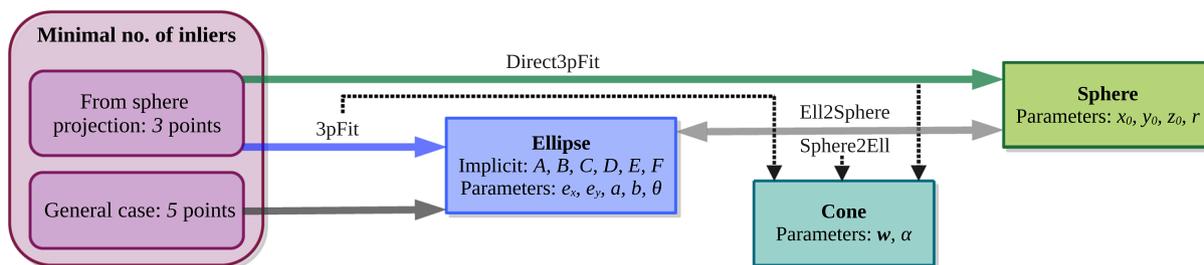


Fig. 1 The proposed algorithms of the paper. The novel techniques *3pFit* (Sect. 3.1) and *Direct3pFit* (Sect. 3.3) require only 3 input points and produce the ellipse or the sphere parameters. In spite of that, the classical methods assuming general the case use 5 points. We proposed the algorithms *Ell2Sphere* (Sect. 3.2) and *Sphere2Ell* (Sect. 2.2) for

ellipse-sphere conversions as well. This allows both the comparison under any method and the measurement of Ground-Truth (GT) ellipse parameters from the sphere. As a sub-step, the cone parameters are also estimated

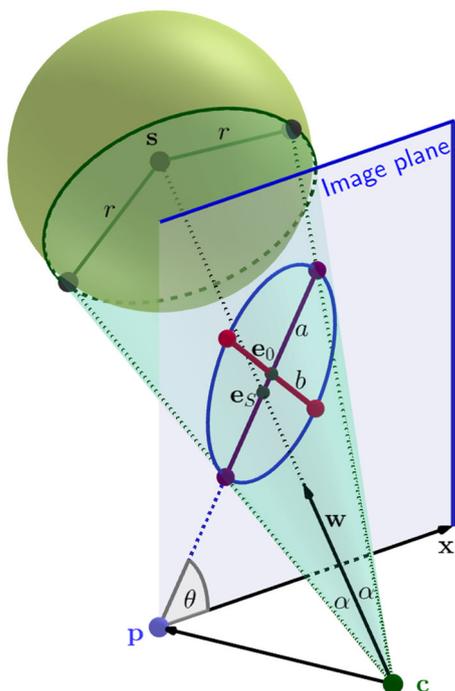


Fig. 2 The geometric setup of the problem. The sphere is parameterized with the coordinates of its center s and its radius r . The tangential cone about this sphere is described with the apex at the camera position c , the direction w , and the opening angle α . The principal point p and the coordinate axis x lie on the image plane. w intersects the image plane at e_s , which is the foci of the ellipse. The ellipse is the result of the cone-image plane intersection defined by the semi-major and the semi-minor axes a and b , the center coordinates e_0 , and the rotation angle θ

widely used for robust model and multi-model estimation. These methods require both minimal and overdetermined solvers. The minimal method runs in each iteration on a randomly selected set of points and calculates the parameters from minimal input. The overdetermined solver estimates the model parameters from a larger sample set. The time complexity of RANSAC exponentially depends on the minimal sample size (see Table 1). Therefore a reliable ellipse edge

Table 1 The proposed and the state-of-the-art spherical ellipse fitting methods (Sun et al. 2016; Shi et al. 2019) reduce the minimal sample size n from 5 to 3 points

n	Outlier ratio			
	50%	65%	80%	95%
3	3.50×10^1	1.06×10^2	5.73×10^2	$\sim 10^4$
5	1.46×10^2	8.75×10^2	$\sim 10^4$	$\sim 10^7$

This table presents the iteration number required for RANSAC (Fischler and Bolles 1981) with confidence set to 99%. The different rows show the size of the varying input: the general ellipse fitting problem requires five points while the spherical problem needs only three. Columns are varying w.r.t. outlier ratio from 50% to 95%. It is well seen that fitting by RANSAC is significantly faster if the method can take advantage of the spherical scenario

detection for minimal and overdetermined cases is essential for rapid RANSAC-based estimation.

Ellipse Detectors. The main categories of ellipse fitting methods are parameter estimation by ellipse contour points and edge-following algorithms. The classical point-based estimators (Fitzgibbon et al. 1996, 1999; Zhang 1997; Rosin 1993; Prasad et al. 2013; Halir and Flusser 1998) apply e.g. least squares method and analytical or numerical approximation to find the ellipse parameters. The latter approaches (Mai et al. 2007; Liu and Qiao 2009; Lu et al. 2020; Meng et al. 2020) extract arcs from images and groups those belonging to the same ellipse.

Ellipse detectors can utilize the Hough transform (HT) to estimate the general ellipse parameters by a voting scheme (Hough 1962; Nair and Saunders 1996). The time and memory demands of this kind of detector are very high. Moreover, HT is also very sensitive to the quantification of the parameters. Xie and Ji (2002) proposed a rapid HT-based method where only one parameter is considered during the voting, and the other four parameters are estimated based on the symmetry of the ellipses.

Recent methods apply deep learning for ellipse estimation (Liu et al. 2007; Dong et al. 2021). Machine learning is very efficient for e.g. free-form object detection (Cai and Vasconcelos 2018). Nonetheless, other approaches are more accurate when the sought model is describable by a few clear geometric constraints and only a few parameters. Machine learning is not a suitable tool for such problems as the object translation and rotation, the so-called pose, can be poorly retrieved (Jabbar et al. 2019).

Sphere Detectors. For sphere detection in images, point-based methods are commonly used requiring at least three contour points. Shi et al. (2019) proposed a point-based minimal ellipse detector minimizing an algebraic error function, where the center estimation is composed of a particular and a free solution in the null space. In the work of Sun et al. (2016), a flexible and global spherical calibration is introduced. The paper solves the standard problem of spherical projection fitting without outliers as they apply monochromatic background. We propose analytical point-based methods for both sphere and spherical ellipse estimations to reach more accurate results in our calibration pipeline with robust detectors. The methods are compared to both general five-point and minimal three-point methods.

Contribution. This paper presents how to accurately estimate the projected ellipse parameters from contour points for calibrated cameras. Furthermore, we reveal how to evaluate the sphere parameters analytically. It is proven that three contour points of an ellipse are sufficient to estimate the projective line of the sphere centers. Hence, if the radius of the sphere is known, the sphere position can be estimated. We propose a direct method to estimate the ellipse with parametric representation. Finally, it is shown how the sphere parameters can be computed if the contour ellipse is known. We address the problem of both minimal and overdetermined cases. The applied error value is geometric and no numerical optimization is needed contrary to the algebraic error-based state-of-the-art methods (Shi et al. 2019; Sun et al. 2016).

Paper outline. Fig. 1 shows the proposed methods along with the input and output parameters. The paper is structured to present these techniques in the following way.

Section 2 overviews the mathematical background of the problem visualized in Fig. 2. Section 2.1 gives a formal description of the setup. Two constraints corresponding to the elliptical projection are derived in Sect. 2.2.

The proposed algorithms are introduced in Sect. 3: (i) *3pFit* (Sect. 3.1) is for three-point ellipse detection, where the ellipse fitting is considered with parametric representation, (ii) *Ell2Sphere* (Sect. 3.2) converts the ellipse parameters to sphere parameters, and (iii) *Direct3pFit* (Sect. 3.3) gives a direct solution to fit the sphere parameters from a pointset. The entire derivations in the appendices prove the reliability of our methods.

Section 4 discusses our novel algorithms and provides pseudocodes and performance analysis. The validation of the proposed methods in Sect. 5 is also special in the sense that both semi-synthetic and real tests are used here. Images of spheres with ground truth parameters are rendered with Blender. The proposed and state-of-the-art methods are *quantitatively* compared. The methods are *qualitatively* compared on real data as well.

2 Mathematical Background

The aim of this paper is to estimate the parameters of a conic section (Glaeser et al. 2016), *i.e.*, an ellipse (see Lemma 1), in an image which contains the projected contour of a sphere. This section describes the shape and curve representations relevant to the sphere projection problem.

2.1 Preliminaries and Notation

In this subsection, we enumerate the used formulae for the camera model and the main geometric objects: the sphere, the cone, and the ellipse.

The perspective camera. Assume, we have a calibrated perspective camera $[s_u \ s_v]^T$, $[u_0 \ v_0]^T$, and f denote the pixel scaling, the principal point, and the focal length, respectively. Then the intrinsic parameters are as follows:

$$\mathbf{K} = \begin{bmatrix} s_u f & 0 & u_0 \\ 0 & s_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

The pixels of the image can be normalized by the camera parameters which map the points of the image plane into the world frame:

$$\begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u-u_0}{s_u f} \\ \frac{v-v_0}{s_v f} \\ 1 \end{bmatrix}, \quad (2)$$

where $[\hat{u} \ \hat{v} \ 1]^T$ denotes the normalized homogeneous coordinates of the point $[u \ v]^T$ in pixels. Hence, the camera location and the principal point are $\mathbf{c} = [0 \ 0 \ 0]^T$ and $\mathbf{p} = [0 \ 0 \ f = 1]^T$, respectively. The unit vector to any 3D point except for the origin $\mathbf{p}_i \in \mathbb{R}^3$ is $\mathbf{q}_i = \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|}$. For the principal point, $\mathbf{q} = \mathbf{p}$ holds.

Geometry representation. Fig. 2 illustrates the sphere projection in the camera. To find the mathematical connection between any contour points of the conic section and the sphere, we use the following notations and equations.

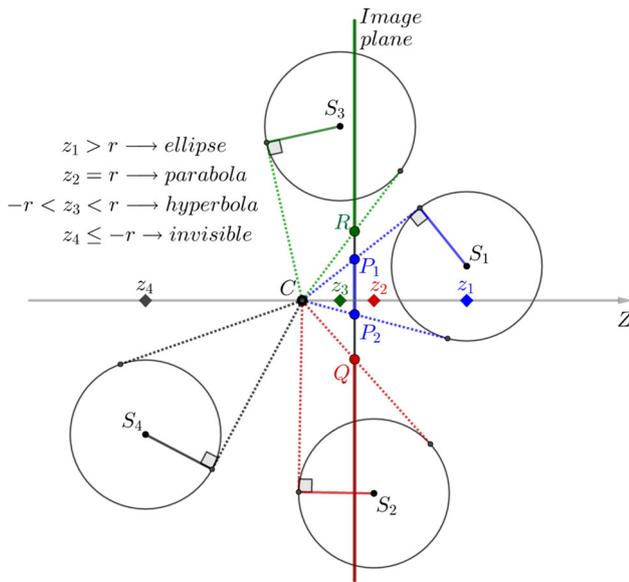


Fig. 3 Visibility of a sphere w.r.t. the depth coordinate z

Let us consider the implicit representation of a sphere as:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0, \tag{3}$$

where $\mathbf{s} = [x_0 \ y_0 \ z_0]^T$ and r are the sphere center and radius, respectively. We assume the radius is known in our approach and the sphere center is an unknown position.

The cone from the camera center, touching the sphere, determines how the sphere becomes a conic section in the image and vice versa. Throughout this paper, the cone is defined by the following two parameters: the unit vector of the cone axis \mathbf{w} and the angle α between the axis and the generators. The apex of the cone is at the camera position \mathbf{c} .

2.2 Ellipse Estimation from a Sphere

If a sphere is projected into the image plane, the equations of the ellipse in both implicit and parametric forms can be deduced from the sphere parameters. Moreover, two invariant properties can be observed.

Implicit representation. In the general case, the implicit equation of a conic section contains six unknown parameters:

$$A\hat{u}^2 + B\hat{u}\hat{v} + C\hat{v}^2 + D\hat{u} + E\hat{v} + F = 0, \tag{4}$$

which is an ellipse if $\Delta = B^2 - 4AC < 0$, i.e., when the conic discriminant is negative. In this special case, it becomes indeed an ellipse (Hajder et al. 2020), as we proved in Lemma 1.

Lemma 1 *If a conic section is the projection of a sphere lying in front of the image plane, the result is an ellipse.*

Proof If a conic section is an ellipse, the conic discriminant is negative:

$$\Delta = B^2 - 4AC < 0. \tag{5}$$

Since the conic section is visible in the image, we introduce a restriction on z_0 which expresses that the whole sphere surface is strictly in front of the camera: $r < z_0$. Otherwise, the sphere would be either behind the image plane and invisible, or touch or contain the camera itself, which is unrealistic in a real-world scenario, as discussed in Fig. 3. The conic discriminant becomes:

$$\begin{aligned} \Delta &= 4x_0^2y_0^2 - 4 \left((y_0^2 + z_0^2 - r^2)(x_0^2 + z_0^2 - r^2) \right) \\ &= -4 \left(z_0^2 - r^2 \right) \left(x_0^2 + y_0^2 + z_0^2 - r^2 \right). \end{aligned} \tag{6}$$

After substituting (6) into the inequality in (5):

$$\left(z_0^2 - r^2 \right) \left(x_0^2 + y_0^2 + z_0^2 - r^2 \right) > 0 \tag{7}$$

The restriction yields $0 < z_0^2 - r^2 \leq x_0^2 + y_0^2 + z_0^2 - r^2$. Hence, (7) holds, that is $\Delta < 0$, and the conic section is indeed an ellipse. \square

The back projection of an image point corresponds to a ray in space. If we use the normalized coordinates defined in (2), the coordinates of the ray can be written as $[x \ y \ z]^T = z [\hat{u} \ \hat{v} \ 1]^T$. The implicit representation of the sphere (3) gives the intersections of a ray and the sphere:

$$(z\hat{u} - x_0)^2 + (z\hat{v} - y_0)^2 + (z - z_0)^2 = r^2. \tag{8}$$

This leads to the following coefficients of (4) as it comes from (A.7) discussed in ‘‘Appendix A’’ and based on Tóth et al. (2020):

$$\begin{aligned} A &= y_0^2 + z_0^2 - r^2, & B &= -2x_0y_0, \\ C &= x_0^2 + z_0^2 - r^2, & D &= -2x_0z_0, \\ E &= -2y_0z_0, & F &= x_0^2 + y_0^2 - r^2. \end{aligned} \tag{9}$$

Three invariant properties can be defined:

$$\begin{aligned} C_1 &: D(BD - 2AE) - E(BE - 2CD) = 0, \\ C_2 &: B(BD - 2AE) - E(ED - 2BF) = 0, \\ C_3 &: B(BE - 2CD) - D(ED - 2BF) = 0. \end{aligned} \tag{10}$$

The equations are linearly dependent, thus, one can be dropped as

$$DC_2 - BC_1 - EC_3 = 0. \tag{11}$$

Parametric representation. Another approach to define an ellipse is a 2D affine transformation describing how an origin-centered unit circle becomes an ellipse specified by a rotation angle, two scaling factors, and a translation vector. Hence, the parametric ellipse is

$$e(\phi) = \mathbf{R}(\theta) \begin{bmatrix} a \cos \phi \\ b \sin \phi \end{bmatrix} + \mathbf{e}_0, \quad \phi \in [0; 2\pi), \quad (12)$$

where $\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ is a 2D rotation matrix calculated from the angle $\theta \in [-\frac{\pi}{2}; \frac{\pi}{2})$, a and b are the major and minor semi-axes, $\mathbf{e}_0 = [e_x \ e_y]^T$ is the center of the ellipse.

The centre of the ellipse \mathbf{e}_0 and the central projection of the sphere center \mathbf{e}_s are not identical, see the spatial view in Fig. 2 or the planar view in Fig. 4. This disparity in the image gives the eccentricity of the projection $\varepsilon = \|\mathbf{e}_s - \mathbf{e}_0\|$ which has to be considered (Matsuoka and Maruyama 2016) when constructing the new method.

To find the parametric representation of an ellipse using the sphere parameters, an important property of a sphere projection is that the line of the major ellipse axis goes through the principal point of the image (Lu and Payandeh 2010). In other words, if the ellipse center \mathbf{e}_0 is known, it yields the value of the rotation angle directly: $\theta = \text{atan2}(\mathbf{e}_y, \mathbf{e}_x)$. In this paper, we call this property the *axial constraint*.

Let us use constraints $\Delta = B^2 - 4AC$ and $\Psi = AE^2 + CD^2 - BDE + F\Delta$ to formulate the equations which converts a general ellipse from implicit form to its parametric one (Hoffmann 1989):

$$\mathbf{e}_0 = \frac{1}{\Delta} \begin{bmatrix} 2CD - BE \\ 2AE - BD \end{bmatrix}, \quad \theta = \arctan \frac{B}{A - C},$$

$$a; b = \frac{\sqrt{2\Psi \left(A + C \pm \sqrt{(A - C)^2 + B^2} \right)}}{-\Delta}. \quad (13)$$

Substituting (9) into (13) with the *axial constraint* gives that the parametric ellipse parameters can be delivered from the sphere parameters directly:

$$\mathbf{e}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \frac{z_0}{z_0^2 - r^2}, \quad a = \frac{r\sqrt{x_0^2 + y_0^2 + z_0^2 - r^2}}{z_0^2 - r^2},$$

$$\theta = \text{atan2}(y_0, x_0), \quad b = \frac{r}{\sqrt{z_0^2 - r^2}}. \quad (14)$$

An important observation describes the connection between the ellipse center and the semi-axes as follows:

$$\frac{a^2}{b^2} = \frac{e_x^2 + e_y^2}{b^2 + 1} + 1. \quad (15)$$

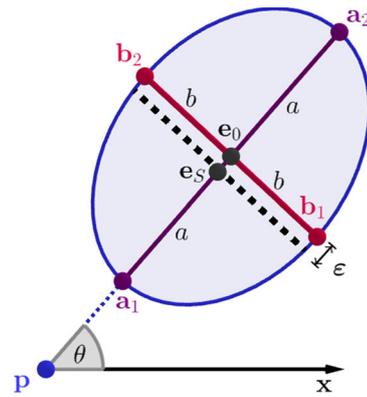


Fig. 4 The *projectional constraint* reveals that if the ellipse center \mathbf{e}_0 and one of the semi-axes are estimated, it determines the other one. *I.e.*, because of (15), either a or b has to be estimated. The principal point \mathbf{p} , the foci \mathbf{e}_s , and the minor axis $(\mathbf{a}_1, \mathbf{e}_0, \mathbf{a}_2)$ are collinear. However, \mathbf{e}_s is linearly independent of the minor axis $(\mathbf{b}_1, \mathbf{e}_0, \mathbf{b}_2)$ if the eccentricity of the ellipse ε is non-zero. The foci, which is the cone axis intersection point of the image, can be expressed straightforwardly if the cone parameters are known. Hence, the end-points of the major semi-axis a can be estimated easier than the minor semi-axis b

This means that an ellipse axis is defined if the two center coordinates and the other axis are known. In conclusion, if \mathbf{e}_0 and either a or b are known, then θ and the remaining semi-axis can be calculated as well. In this paper, this property is called *projectional constraint*. The projected sphere center is the first two coordinates of \mathbf{s} normalized by the third one which is $\mathbf{e}_s = \frac{1}{z_0} [x_0 \ y_0]^T$.

Observation. Both deductions contain two constraints. The implicit form has constraints $\mathcal{C}_1, \mathcal{C}_2$, and \mathcal{C}_3 in (10), but only two of them are independent because of (11). The parametric form has an *axial* and a *projectional constraint*.

In this paper, the formulae in (9) and (14) are referred to as *Sphere2Ell*. This name covers both possible conversions depending on the representation of the required output as we know the parametric-implicit conversion of an ellipse.

3 Proposed Minimal Solvers

Classical methods (Fitzgibbon et al. 1996, 1999; Halir and Flusser 1998; Zhang 1997; Rosin 1993) use five points for ellipse estimations. Based on our special scenario, we can consider the two constraints introduced in the previous section. Hence, the degrees of freedom are reduced from five to three. Therefore, three contour points, denoted by $\mathbf{p}_i = [\hat{u}_i \ \hat{v}_i \ 1]^T$, are enough for estimation the ellipse parameters, where \hat{u}_i and \hat{v}_i are the normalized coordinates of the i -th sample point as in (2), $i \in \{1, 2, 3\}$.

We pursue to apply two approaches here: one is based on the constraints of the implicit representation similar to some state-of-the-art methods (Sun et al. 2016; Shi et al. 2019),

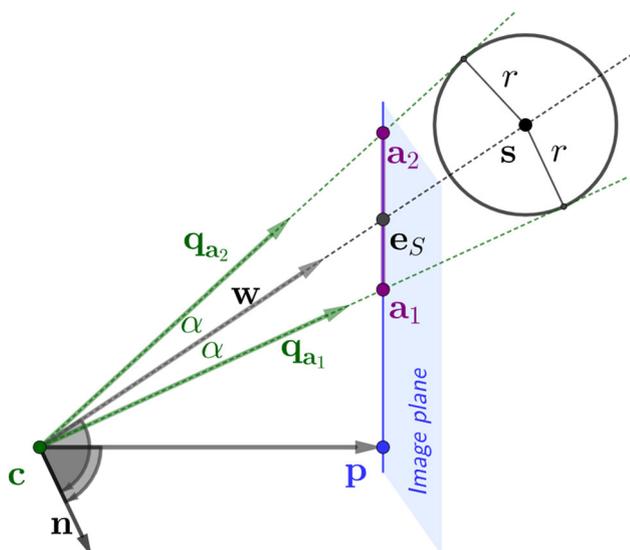


Fig. 5 The *axial constraint* is built on the observation that the camera c , the sphere center s , and the major ellipse axis (a_1, e_s, a_2) are coplanar. The vector n is perpendicular to the cone direction w and viewing direction p . Rotating w around n by α gives vectors q_{a_1} and q_{a_2} that intersect the image plane at a_1 and a_2 , i.e., the end-points of the major ellipse axis

and the other is based on a novel geometric approach. The implicit solution in Appendix B has no closed-form solution and did not overperform the rival methods. The following approach shows that the estimation problem can be solved significantly easier and faster than the previously mentioned algorithm as the solution can be given in closed form, without any numerical approximation applied.

3.1 Geometric Approach

The main idea of the proposed method is that the projection can be defined by the cone directed to the sphere center when the generator of the cone touches the sphere and the apex is at the camera focal point. The intersection of this cone and the image plane gives the ellipse contour in the image. Therefore, the cone parameters can be calculated from a few contour points; then, the ellipse parameters can be retrieved.

The cone parameters. If three points $p_i, i \in \{1, 2, 3\}$ are given on the ellipse contour, the cone can be defined by the following process. Let the unit vectors from c to p_i be denoted by q_i . Then

$$\langle q_1, w \rangle = \langle q_2, w \rangle = \langle q_3, w \rangle = \cos \alpha. \tag{16}$$

Let matrix $Q = [q_1 \ q_2 \ q_3]$ contain the normals and 1_3 be a one 3-vector. The inner products can be written in matrix form after re-ordering as

$$(\cos \alpha)^{-1} w = Q^{-T} 1_3. \tag{17}$$

Hence, the magnitude and the direction of vector $Q^{-T} 1_3$ give the cone parameters $\cos \alpha$ and w .

The end-points of the major axis. If the cone parameters $\cos \alpha$ and w are known, the ellipse parameters can be calculated as well. For this purpose, we estimate two well-defined contour points of the ellipse: the end-points of the major axis denoted by a_1 and a_2 (see Fig. 4). The *axial constraint* makes these points and the semi-major axis a measurable in the image.

Figure 5 illustrates the problem, where the plane contains the points c, p, a_1 , and a_2 . A unit normal of this plane can be defined as $p \times w$ which equals to

$$n = \begin{cases} [-w_y \ w_x \ 0]^T \cdot (w_x^2 + w_y^2)^{-\frac{1}{2}} & \text{for } w \neq p \\ [0 \ 1 \ 0]^T & \text{for } w = p. \end{cases} \tag{18}$$

Thus, the directions from c to a_1 and a_2 can be calculated, if $R(n, \alpha)$ denotes the axis-angle representation of a 3-dimensional rotation describing the rotation around vector n with angle α

$$q_{a_1} = R(n, \alpha) \cdot w, \tag{19}$$

$$q_{a_2} = R(n, -\alpha) \cdot w = R(n, \alpha)^T \cdot w. \tag{20}$$

If q_{a_1} and q_{a_2} are known, the axis end-points are

$$a_1 = \Pi_{\mathcal{I}}(q_{a_1}), \quad a_2 = \Pi_{\mathcal{I}}(q_{a_2}), \tag{21}$$

where the projection operator $\Pi_{\mathcal{I}} : \mathbb{R}^3 \mapsto \mathbb{R}^2$ maps a vector from 3D space to a point in the normalized image plane.

The parametric ellipse. As it was demonstrated, calculating the major axis end-points a_1 and a_2 are straightforward using the *axial constraint*. Moreover, the *projectional constraint* and formula (15) determines b based on e_0 and a . Let us use the constraint $\Omega = e_x^2 + e_y^2 + 1 - a^2$, then

$$e_0 = \frac{a_1 + a_2}{2}, \quad a = \frac{\|a_1 - a_2\|}{2},$$

$$\theta = \text{atan2}(e_y, e_x), \quad b = \sqrt{\frac{-\Omega + \sqrt{\Omega^2 + 4a^2}}{2}}. \tag{22}$$

3.1.1 Overdetermined Case

If we have a large set of points consisting of $n > 3$ ellipse contour points, (17) is modified as

$$(\cos \alpha)^{-1} w = Q^{+T} 1_n, \tag{23}$$

where $Q = [q_1 \dots q_n]$, Q^+ is the Moore-Penrose inverse of matrix Q and 1_n is a one n -vector. This technique minimizes a

geometric error. The remaining steps of the algorithm are the same. We refer to both minimal and overdetermined solutions as *3pFit*.

The main advantage over the state-of-the-art algorithms (Sun et al. 2016; Shi et al. 2019) is the minimization of the geometric error, *i.e.*, the inverse of $\cos \alpha$ in (23) and (17), instead of an algebraic one. $\cos^{-1} \alpha$ equals to the ratio of the focal point—sphere distance and the radius.

3.2 Center from Parametric Ellipse

We describe a sphere center estimator algorithm called *Ell2Sphere* next that allows us to compute the sphere parameters based on the proposed ellipse estimators. The inputs of the method are the ellipse parameters and the radius of the sphere. The main goal is to define the cone parameters and then find the sphere center.

First, we estimate the end-points of the major ellipse axis. Based on the geometric meaning of the parametric ellipse described before (12), the input of function e are the intersection points of an origin-centered unit circle and axis x , *i.e.*, $\phi = \{0, \pi\}$. Hence, the sought points are $\mathbf{a}_1 = e(0)$ and $\mathbf{a}_2 = e(\pi)$.

The next step is to find the direction of the cone axis. We back-project the points \mathbf{a}_1 and \mathbf{a}_2 into the 3D space:

$$\mathbf{q}_{\mathbf{a}_1} = \Pi_S(\mathbf{a}_1), \quad \mathbf{q}_{\mathbf{a}_2} = \Pi_S(\mathbf{a}_2), \quad (24)$$

where the projection operator $\Pi_S : \mathbb{R}^2 \mapsto \mathbb{R}^3$ provides the image ray and maps a point \mathbf{x} from the normalized image plane to a vector into the 3D space as $\Pi_S(\mathbf{x}) = \Pi_{\mathcal{I}}^{-1}(\mathbf{x}) / \|\Pi_{\mathcal{I}}^{-1}(\mathbf{x})\|$. Afterward, the angle between $\mathbf{q}_{\mathbf{a}_1}$ and $\mathbf{q}_{\mathbf{a}_2}$ gives the angle of the cone and the angle bisector of these vectors gives the direction as

$$\alpha = \frac{\arccos\langle \mathbf{q}_{\mathbf{a}_1}, \mathbf{q}_{\mathbf{a}_2} \rangle}{2}, \quad \mathbf{w} = \frac{\mathbf{q}_{\mathbf{a}_1} + \mathbf{q}_{\mathbf{a}_2}}{2}. \quad (25)$$

Since $\sin \alpha = r / \|\mathbf{s}\|$ (see Fig. 5), variables α and r give the distance between the camera focal point and the sphere center, the latter of which is obtained as follows:

$$\mathbf{s} = \frac{r}{\sin \alpha} \mathbf{w}. \quad (26)$$

Based on $\langle \mathbf{q}_{\mathbf{a}_1}, \mathbf{q}_{\mathbf{a}_2} \rangle = \cos 2\alpha$, trigonometrical identities allow us to calculate the sphere center \mathbf{s} without measuring angle α :

$$\mathbf{s} = \frac{\sqrt{2}r}{\sqrt{1 - \langle \mathbf{q}_{\mathbf{a}_1}, \mathbf{q}_{\mathbf{a}_2} \rangle}} \mathbf{w}. \quad (27)$$

3.3 Center from Contour Points

Some ellipse contour points can define ellipse contour points define the sphere center straightforwardly and the ellipse estimation can be omitted from the proposed method. This simplified algorithm called *Direct3pFit* is suitable for both minimal and overdetermined problems.

The method has three main steps: estimation of \mathbf{Q} containing the normal vectors, then the cone parameters, which yields the sphere center. It starts the same way as in Sect. 3.1 until (17) or (23) when the cone parameters are already estimated. Then the sphere center is written like in (26). The optimization step of (27) is modified as $\cos \alpha$ is estimated in (17) and (23) instead of $\cos 2\alpha$:

$$\mathbf{s} = \frac{r}{\sqrt{1 - \cos^2 \alpha}} \mathbf{w}. \quad (28)$$

4 Algorithms

The actual implementations of the algorithms have a large effect on the speed. Therefore, we provide some implementation details and list the algorithmic components that facilitate fast performance. In this section, we list these optimization steps, justify the design choices and present the pseudocode of our methods: *3pFit*, *Ell2Sphere*, and *Direct3pFit* are outlined in Algorithms 1, 2, and 3, respectively. The algorithms contain only high-level descriptions, and the operations on the left have to be performed.

3pFit and *Ell2Sphere* estimate the major axis end-points \mathbf{a}_1 and \mathbf{a}_2 to find the ellipse and cone parameters instead of measuring the minor axis end-points \mathbf{b}_1 and \mathbf{b}_2 , although the problem could be formulated and solved in both ways. Figure 4 illustrates both cases. The minor axis end-points can be calculated as $\mathbf{b}_1 = \Pi_{\mathcal{I}}(\mathbf{R}(\mathbf{w}, \pi/2) \cdot \mathbf{q}_{\mathbf{a}_1})$ and $\mathbf{b}_2 = \Pi_{\mathcal{I}}(\mathbf{R}(\mathbf{w}, \pi/2)^T \cdot \mathbf{q}_{\mathbf{a}_1})$ similar to (21). This means that finding the end-points regarding b instead of a would need extra rotation calculation and cause more numerical error.

In *3pFit* and *Direct3pFit*, the reformulation of $\mathbf{q}_{\mathbf{a}_2}$ (20) comes from the properties of a Lie algebra. Accordingly, we can reuse the already calculated $\mathbf{R}(\mathbf{n}, \alpha)$ in (19) instead of estimating the new matrix which would be more time-consuming than the simple transposition. This simplification was executed also in the equation of \mathbf{b}_2 in the last paragraph as $\mathbf{R}(\mathbf{w}, -\pi/2) = \mathbf{R}(\mathbf{w}, \pi/2)^T$.

Finally, we advise not to measure α in *Ell2Sphere* and *Direct3pFit* directly to calculate \mathbf{s} in (26) because the arccos operation inflicts rounding error in (25). The deductions with $\cos 2\alpha$ in (27) and $\cos \alpha$ in (28) are more accurate and rapid.

Stability analysis in Sect. 5.3 supports these steps. Furthermore, a real-world example in Table 7 presents that

Algorithm 1 3pFit

Require: points $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^2 \mid i = \{1, \dots, n\}, n \geq 3\}$; camera matrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$
Ensure: ellipse parameters: $\mathbf{e}_0 \in \mathbb{R}^2, a \in \mathbb{R}_{>0}, b \in \mathbb{R}_{>0}, \theta \in [-\pi/2; \pi/2]$

```

1:  $\mathbf{Q} := \text{Normalize}(\mathcal{P}, \mathbf{K})$  ▷ Normalization by Eq. (2) & calc. unit vectors
2: if  $n = 3$  then ▷ Cone opening angle & direction from Eq. (17)
3:    $\cos \alpha, \mathbf{w} := \text{ConeFromPoints}(\mathbf{Q}^{-T})$ 
4: else ▷ Cone opening angle & direction from Eq. (23)
5:    $\cos \alpha, \mathbf{w} := \text{ConeFromPoints}(\mathbf{Q}^{+T})$ 
6: end if
7:  $\mathbf{a}_1, \mathbf{a}_2 := \text{MajorAxisFromCone}(\cos \alpha, \mathbf{w})$  ▷ Ellipse major axis end-points from Eqs. (18–21)
8:  $\mathbf{e}_0 := (\mathbf{a}_1 + \mathbf{a}_2)/2$  ▷ Ellipse center from Eq. (22)
9:  $a := \|\mathbf{a}_1 - \mathbf{a}_2\|/2$  ▷ Ellipse major semi-axis from Eq. (22)
10:  $b := \text{ProjectionalConstraint}(\mathbf{e}_0, a)$  ▷ Ellipse minor semi-axis from Eq. (22)
11:  $\theta := \text{AxialConstraint}(\mathbf{e}_0)$  ▷ Ellipse rotation angle from Eq. (22)
12: return  $\mathbf{e}_0, a, b, \theta$ 

```

Table 2 Main differences between the proposed and state-of-the-art three-point methods

Three-point methods	Coeff. matrix size	Matrix operations	\mathcal{P} to \mathbf{s} directly?	Ellipse rep
<i>Proposed</i>	$N \times 3$	NORM + PINV	Yes	$a, b, \mathbf{e}_0, \theta$
<i>L LSB</i> (Sun et al. 2016)	$N \times 3$	NORM + LSQ	Yes	$A - F$
<i>IMM</i> (Shi et al. 2019)	$N \times 6$	PINV + SVD	No	$A - F$

The operations on the coefficient matrices (normalization (NORM); pseudoinverse (PINV); homogeneous linear least squares method (LSQ); singular value decomposition (SVD)) are related to the computational complexity of the methods. The third column itemizes whether the algorithms provide the point-based closed formula of the sphere center \mathbf{s} or whether the ellipse parameters have to be estimated as well. Finally, the last column specifies the estimated ellipse representations from (4) or (12) of every method

the accuracy holds if the sequential execution of *3pFit* and *Direct3pFit* changed to *Direct3pFit* while it is less time-consuming.

Computational complexity. In Table 2, we compared the computational complexity of the three-point algorithms and summarized the principal differences to reveal the innovation of the proposed approach. *IMM* (Shi et al. 2019) computes on a double-sized coefficient matrix contrary to the proposed and *L LSB* (Sun et al. 2016) methods. Besides elementary computational steps, the primary matrix operations of the methods are normalization (NORM), computing the pseudoinverse matrix (PINV), linear least squares method (LSQ), and singular value decomposition (SVD). In terms of complexity, all three methods guarantee linear runtime. Nevertheless, the expected time demand of *IMM* is higher due to the larger matrix size. Besides, the fourth column presents that the sphere estimation in *IMM* is not a direct method from point set \mathcal{P} but contains the estimation of the ellipse as a sub-step. *L LSB* and the proposed *Direct3pFit* define the sphere position \mathbf{s} via the cone parameters more efficiently but with different approaches.

The last column shows whether the algorithm can produce the ellipse parameters of (4) or (12), which is critical in terms of application in a robust detector. The primary motivation of the paper is to find suitable methods for the calibration pipeline of Tóth et al. (2020), which estimates the ellipses using RANSAC. The distances between every point and the ellipse are evaluated within every iteration to find the largest

consensus set. The distance function d of the later experiment in (29) computes the same value. This step requires the geometric parameters a, b, \mathbf{e}_0 and θ estimated only by the proposed *3pFit* method directly. Hence, the output of *IMM* and *L LSB* would be converted from the $A - F$ parameters in every iteration of RANSAC, causing extra computational time.

5 Results

To demonstrate the performance of the sphere and ellipse fitting methods, we simulated several test scenarios in MATLAB, Blender, and made real-world tests. The source code of the proposed methods is available on GitHub as part of a sensor calibration project: <https://github.com/tothteklia/SphereCalib>.

The next paragraphs outline the tested ellipse fitting methods.

LSQDir (Halir and Flusser 1998): This variant of the classical direct ellipse fitting algorithm of Fitzgibbon et al. (1996) provides better numerical stability. *LSQDir* finds the parameters of a general ellipse via eigenvector computation of a small matrix. Therefore it is very fast and non-iterative. The drawback of this method is that it is biased toward finding small ellipses.

HQ (Lu et al. 2020): This is a high-quality, four-stage ellipse detector. It detects arcs from Sobel or Canny edge

Algorithm 2 Ell2Sphere

Require: ellipse parameters: $\mathbf{e}_0 \in \mathbb{R}^2$, $a \in \mathbb{R}_{>0}$, $b \in \mathbb{R}_{>0}$, $\theta \in [-\pi/2; \pi/2]$; sphere radius $r \in \mathbb{R}_{>0}$
Ensure: sphere center $\mathbf{s} = [x_0 \ y_0 \ z_0]^T \in \mathbb{R}^3$

- 1: $\mathbf{a}_1 := \mathbf{e}(0)$, $\mathbf{a}_2 := \mathbf{e}(\pi)$ ▷ Ellipse major axis end-points from Eq. (12)
- 2: $\cos 2\alpha$, $\mathbf{w} := \text{ConeFromEllipse}(\mathbf{a}_1, \mathbf{a}_2)$ ▷ Cone opening angle & direction from Eqs. (24–25)
- 3: $\mathbf{s} := \text{SphereCenterFromEllipse}(r, \cos 2\alpha, \mathbf{w})$ ▷ Sphere center from Eq. (27)
- 4: **return** \mathbf{s}

Algorithm 3 Direct3pFit

Require: points $\mathcal{P} = \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^2, i = \{1, \dots, n\}, n \geq 3\}$; camera matrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$; sphere radius $r \in \mathbb{R}_{>0}$
Ensure: sphere center $\mathbf{s} = [x_0 \ y_0 \ z_0]^T \in \mathbb{R}^3$

- 1: $\mathbf{Q} := \text{Normalize}(\mathcal{P}, \mathbf{K})$ ▷ Normalization by Eq. (2) & calc. unit vectors
- 2: **if** $n = 3$ **then**
- 3: $\cos \alpha$, $\mathbf{w} := \text{ConeFromPoints}(\mathbf{Q}^{-T})$ ▷ Cone opening angle & direction from Eq. (17)
- 4: **else**
- 5: $\cos \alpha$, $\mathbf{w} := \text{ConeFromPoints}(\mathbf{Q}^{+T})$ ▷ Cone opening angle & direction from Eq. (23)
- 6: **end if**
- 7: $\mathbf{s} := \text{SphereCenterFromPoints}(r, \cos \alpha, \mathbf{w})$ ▷ Sphere center from Eq. (28)
- 8: **return** \mathbf{s}

detectors and several properties of these arcs are exploited, e.g. overall gradient distribution, arc-support directions, and polarity. This method is sensitive to the setting of its three parameters. Thus, it was tuned on each test sequence in *HQ* to achieve accurate results.

AAMB (Meng et al. 2020): After segmenting the edges into elliptic arcs, the so-called digraph-based arc adjacency matrix (AAM) is constructed. All arc combinations are then generated through bidirectionally searching the AAM for ellipse candidates. The next step is the estimation of the cumulative-factor-based cumulative matrices (CM). At last, the ellipses are fitted to each candidate through the double eigen-decomposition of the CMs using the Jacobi method.

Some novel papers deal with sphere positioning and contain ellipse fitting too. The following paragraphs summarize the approaches and the innovation of our method in contrast to these in addition to the complexity analysis in Sect. 4.

LLSB (Sun et al. 2016): A linear least-squares-based sphere estimation method is presented in this paper, which requires high-quality ellipse contours. Its limitation is the white spherical target in front of a blackboard, as mentioned in the introduction. This method reconstructs sphere centers effectively using two constraints as well. However, the paper did not reveal how to interpret them independently. The approach detects the sphere center directly, while ellipse estimation is not exposed in detail, but the parameters can be expressed.

IMM (Shi et al. 2019): An implicit minimal method is proposed for sphere fitting similarly to our initial attempt, described in Appendix B. The sphere center estimation is composed of a particular and a free solution in the null space, which problems are solved by SVD and Gauss-Jordan's elimination. Both projected ellipse and sphere parameters can be computed directly, minimizing an algebraic error function.

CB (Hajder et al 2020, Tóth et al 2020): Our former approach deals with circle-based sphere estimation, which is suitable especially for narrow-angle lenses when the a/b ratio of the projected ellipse is close to one. In a RANSAC process, the minimal solver is a three-point circle estimator with a well-tuned error threshold. *LSQDir* is applied to find the ellipse parameters to the largest consensus set. The sphere center is obtained from the implicit ellipse parameters with Levenberg-Marquardt optimization.

In order to evaluate the performance of the algorithms, both minimal and non-minimal cases were tested contaminated by zero-mean Gaussian noise. We also varied the distance of the sphere centers from the focal point both along the vertical and optical axes. We modeled several situations both in MATLAB and in Blender. We assumed that the camera intrinsics were calibrated, and the sphere radius was known.

Moreover, we tested real scenarios with two pre-calibrated cameras. We compared the algorithms by estimating the parameters of the observed sphere in both cameras. Since the cameras and spheres were static, the 3D translation between the sphere centers should coincide with the translation of the two cameras and, thus, can be used for measuring the error. Finally, a camera-LiDAR calibration application revealed the relevance of these algorithms.

5.1 Experimental Settings

The applied synthetic, photorealistic, and real camera intrinsics are listed in Table 3. These parameters were assumed to be known during the tests. The perspective camera model was applied in all cases. The digital camera was a Hikvision MV-CA020-20GC sensor with high-quality Fujinon SV-0614H lenses. The radial distortion of the lenses was negligible.

Table 3 The intrinsic camera parameters in pixels of the three test setups discussed in Sects. 5.3, 5.4 and 5.5

Setup	Environment	Image resolution	Focal length		Principal point	
			f_u	f_v	u_0	v_0
Matlab-1	Synthetic	qHD: 960 × 540	1050	1050	480	270
Matlab-2	Synthetic	HD: 1920 × 1080	800	675	960	540
Blender	Photorealistic	HD: 1920 × 1080	800	675	960	540
Hikvision	Real-world	WUXGA: 1920 × 1200	1262.6	1267.4	934.6	659.5

The ground truth ellipse parameters were also required to compare the different estimators in the synthetic tests. They were delivered from the sphere parameters, as we discussed in Sect. 2.2. In the real-world tests (Sects. 5.5 and 5.6), *3pFit* was executed in a RANSAC framework to find the ellipse in the image and *Direct3pFit* to find the estimated sphere center based on the best consensus set

5.2 Optimization of *Direct3pFit*

Levenberg-Marquardt minimization (LMM) can help in numerical optimization (Hartey and Zisserman 2004) when the data set is noisy. The formula of the LMM problem was

$$\arg \min_s \|d(Sphere2Ell(s, r), \mathcal{P})\|^2, \tag{29}$$

where s is initially estimated with the proposed *Direct3pFit*, the function d gives the Euclidean difference vector between a parametric ellipse and an inlier pointset, and \mathcal{P} is the set of the 2D inlier points. The function d is a quadric; the sought root can be approximated numerically.

LMM found a local optimum in a few steps in many experiments. The most likely reason is that although the noise occurs in the image plane, the cone axis is close to perpendicular to that plane even if the ellipse lies far from the principal point but still visible, *i.e.*, near a corner of the image (Hajder et al. 2020). Equations (17) and (23) minimize a geometric error. After that, the algorithm always takes exact steps, and no approximation is needed. As the geometric error is linear w.r.t. the inverse of the cosine of the angle, *Direct3pFit* obtains the (optimal) solution in one step without any iterations. The inverse of the cosine is the ratio of the sphere center—camera distance and the radius. The fitting error in the image space, *i.e.*, the ellipse-points distance, can be minimized via numerical optimization. As the proposed method gives the initial value, which is also geometric error-based minimization, the numerical optimization usually cannot improve the result. However, the optimization of (29) via LMM is preferred in degenerated cases.

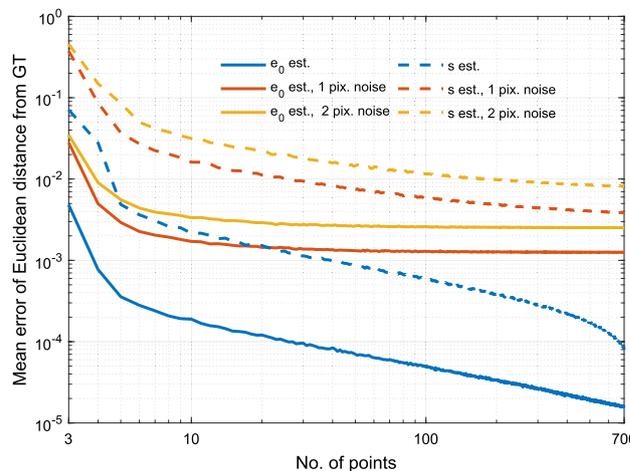


Fig. 6 The mean error in the ellipse (e_0 ; continuous curves; meters) and sphere centers (s ; dashed curves; meters) plotted as a function of the sample size used for the estimation. For estimating e_0 , we ran the method in Sect. 3.1 (*3pFit*). The sphere centers were estimated from e_0 by the method in Sect. 3.2 (*Ell2Sph*). The results in the noise-free and noisy cases are shown by color

5.3 Synthesized Accuracy and Ill-Conditioning Tests

Synthetic tests implemented in MATLAB for competitive-quantitative evaluation demonstrate how the three-point algorithms behave in noisy and ill-conditioned cases. Camera parameters are listed in the first two rows of Table 3.

The effect of the inlier number. In the first experiment presented in Fig. 6, the ellipse and sphere center estimation mean error was evaluated *w.r.t.* the number of points using *3pFit* and *Direct3pFit* in case of no noise, 1 pixel and 2 pixel white Gaussian noise in the image. The sphere parameters were $s = [-0.95 \ 0.35 \ 3.00]^T$ and $r = 0.35$, and the tests were repeated 1000 times with different point sets containing 3–700 inliers. We observed that the sphere center error is about 10^{1.5} times more than the ellipse center error in this configuration. For overdetermined cases, 50 was a sufficient number of fitting points to get a millimeter-precise result with this setup during synthetic tests. Noise-free scenarios and at least 50 points gave a sub-millimeter result. In general, more contour points gradually reduced the error rate. For real-world scenarios, accurate inlier localization was a bottleneck of the sphere detection, as the algorithm in this setup

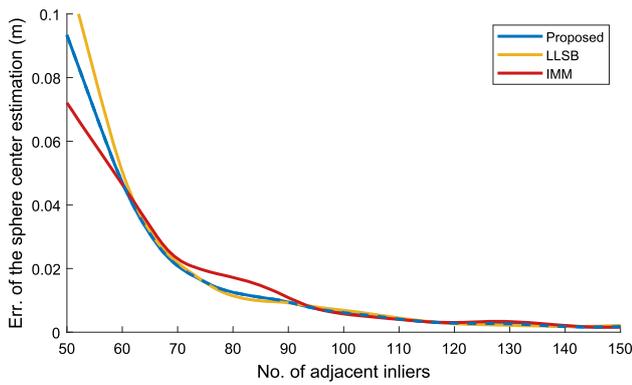


Fig. 7 The smoothed error in sphere centers plotted as a function of the sample size used for the estimation. The inliers were ill-conditioned, and only the results between 50 and 150 points were analyzed since the error function converges to zero, *i.e.*, the 7.9% of the ellipse contour was enough for millimeter precise sphere fitting. Some measurements are highlighted in Fig. 8 for visual comparison; further evaluation is presented in Table 4

cannot approach centimeter precision with 2 pixel noise even with about 500 points. Nonetheless, in general, more contour points gradually reduced the error rate.

Ill-conditioned tests. To support this observation and test the robustness, we generated ill-conditioned scenarios containing only adjacent inliers, *i.e.*, an arc segment. This problem occurs when the ellipse is blurred, and only a short contour segment can be detected. We compared the results to the two other sphere detectors *IMM* (Shi et al. 2019) and *LLSB* (Sun et al. 2016) as well presented in Fig. 7 and Fig. 8. The methods using 40 adjacent inliers from the total ~ 2000 points gave unsatisfactory results: ellipses with large semi-axes as the inliers are nearly fitting to a line. 47, 48, and 51 inliers

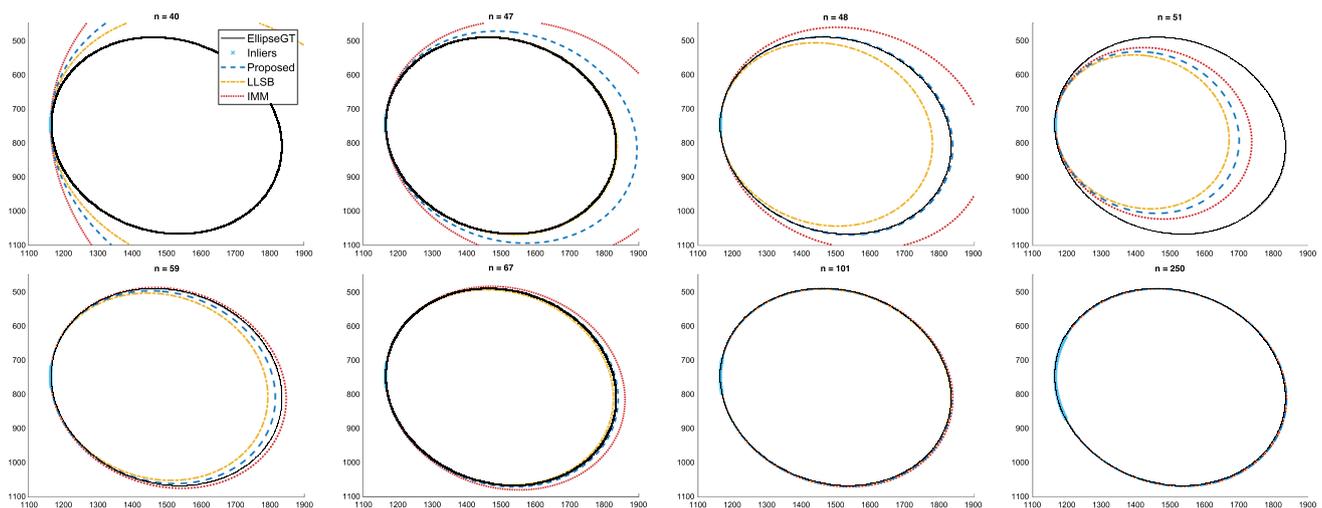


Fig. 8 Ill-conditioned test cases present the robustness of the sphere fitting methods. The fewer the inlier number (n), the more degraded the fitted ellipses displayed in pixels. Nonetheless, the proposed algorithm is reliable from $n \geq 250$ points. The ellipse contour is constructed by about 2000 pixels

have a varying, unstable impact. From 59 to 101 inliers, the parameters converged to the optimal values and reached centimeter-precise results. Finally, the ellipses are ideal in case of 250 or more inliers. *LLSB* and the proposed algorithm overperform *IMM* in case of 70–100 inliers. The compared quantitative results and runtime are listed in Table 4. The measurements executed on an Intel(R) i7-8665U 1.90 GHz with 16 GB RAM verified the accuracy of the methods, while the runtime of the proposed approach was significantly better: within a second for 1894 run.

Stability analysis. More than 25000 noiseless test cases with 1000 inliers were generated with varying sphere parameters. The top image of Fig. 9 showcases the reliability of the proposed method and *LLSB*: the highest errors were around 10^{-10} and 10^{-8} , respectively. However, the two peaks around 10^{-7} and 10^{-2} reveals the stability problems of *IMM* in the zoomed, bottom image. Additionally, we evaluated the effect of the proposed implementation improvements presented in Sect. 4. The optimization steps did not influence the results of the overdetermined cases. In contrast, the same test data set but with a minimal three-point fitting revealed that the algorithm is faster by 3.71% with the same accuracy. This property is especially beneficial in using robust detectors with a high outlier ratio.

5.4 Photorealistic Tests

Further tests based on photorealistic images were generated by Blender. The camera parameters, listed in the third row of Table 3, and the sphere radius $r = 0.25$ m were fixed while the sphere location was changed in the scene. We described the sphere locations using the following terms.

Table 4 Mean error (SE ; meters), standard deviation (SD ; meters), root-mean-square deviation ($RMSD$; meters) of the sphere center estimation methods in 1894 degenerate scenarios with varying inlier ratio

Method	SE	SD	$RMSD$	Δt
Proposed	0.0122	0.0755	0.0764	0.8450
<i>LLSB</i>	0.0140	0.0871	0.0882	6.7538
<i>IMM</i>	0.0214	0.1377	0.1393	157.2325

Results present the stability of the algorithms. The runtime (Δt ; section) of the proposed method is clearly the best. These properties make it the most suitable to apply in robust detectors (e.g. RANSAC). The time measurements were repeated 1000 times and averaged on Intel(R) i7-8665U 1.90 GHz with 16 GB RAM

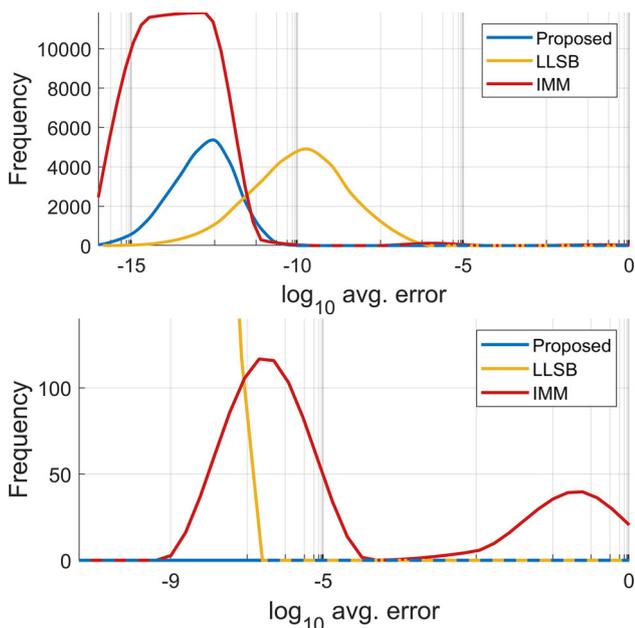


Fig. 9 Stability analysis of the three-point methods. The top image presents the distribution of the average error of 25000 different scenarios in a logarithmic scale. In the bottom image, the zoomed diagram shows the stability error of *IMM* around zero

The depth coordinates and a varying step set were defined as $z_k = 0.75 + 0.05 \cdot k$ and $step_k = 0.05 + 0.065 \cdot k$ where $k \in \{0, \dots, 25\}$. We defined new set of x and y coordinates for every depth value z_k as $x_i = step_k \cdot i$ and $y_j = step_k \cdot j$ where $i \in \{0, \dots, 9\}$ and $j \in \{0, \dots, 3\}$. This process evaluated $10 \times 4 \times 26 = 1040$ sphere positions in total. All sphere contours were continuous and fully visible in the input images. The image sequence of Fig. 10 on the left shows some generated input image indexed with $[i, j, k]$. As no noise was added to the images, the inlier set \mathcal{P} was obtained directly by Canny edge detection.

The results for the test evaluation are listed in Table 5. The high-quality input images caused a sub-pixel error rate for every tested method. Our *3pFit* method approximated the length of the major semi-axis the best and gave the most

precise sphere localization. Although the estimation of b and \mathbf{e}_0 were not outperform some other methods, the difference is minimal, and the constraint-based approach generated precise sphere positions. The time demand of the proposed method is also notable compared to the others. Note, that implementation of *AAMB* and *HQ* was provided in advance; hence, the edge detection was different which affected the runtime. The methods *LSQDir*, *LLSB*, and *IMM* got the same input pointset after executing Canny edge detection. *Ell2Sphere* algorithm helps to estimate the sphere positions for the comparison based on the general five-point methods *HQ*, *AAMB*, and *LSQDir*.

The middle image in Fig. 10 presents the median sphere center estimation error *w.r.t.* the distance from the optical axis projected into the image and measured in pixels, which means the $\mathbf{p} - \mathbf{e}_s$ distance. This diagram concludes that the further distance—*i.e.*, higher a/b ratio—favors a more precise estimation in case of high picture quality, especially in case of the *HQ* method. One key component of the proposed method is to estimate the major ellipse axis end-points, and circle-like sphere projections do not favor this process. The optimal distance is around 300 – 600 pixel distance for *LSQDir*, *LLSB*, *IMM*, and the proposed method followed by an increasing error. If the depth of the sphere is normal but the distance from the optical axis is relatively large, the ellipse approximates to another conic section: the parabola (see Fig. 3). Hence the inflected problem is similar to the circle-like cases.

The depth test on the right image of Fig. 10 reports the same sphere center estimation error *w.r.t.* the z coordinate of the sphere. This setup had an optimum at $z = 1.4$ m, but the difference from the higher error rate is around 1 mm. On the first hand, the further distance from the camera increases the rounding error as we convert the point between pixels and the metric system. On the other hand, the horizontal and vertical views of the camera influence the visibility of the sphere in the image. The closer depth means a smaller range for sphere coordinates x and y , which causes unfavorable, circle-like ellipse parameters as in the previous test. Some examples are plotted on the left in Fig. 10, labeled as *image*[0, 0, 0], *image*[9, 0, 0], *image*[0, 3, 0], and *image*[9, 3, 0] among the input images. We think the optimum was the point when the spheres were relatively further from the camera and the optical axis, and the raster images caused minimal measurement error in the contour points. The sphere fitting was generally stable in the case of varying depth parameters. Note that the trend of *AAMB* is worse around 1–2 mm in depth and optical axis distance tests.

5.5 Real-World Stereo Camera Tests

During real-world scenarios, we used Hikvision MV-CA020-20GC cameras to measure the precision of the sphere center estimation (see detailed camera parameters in Table 3). A

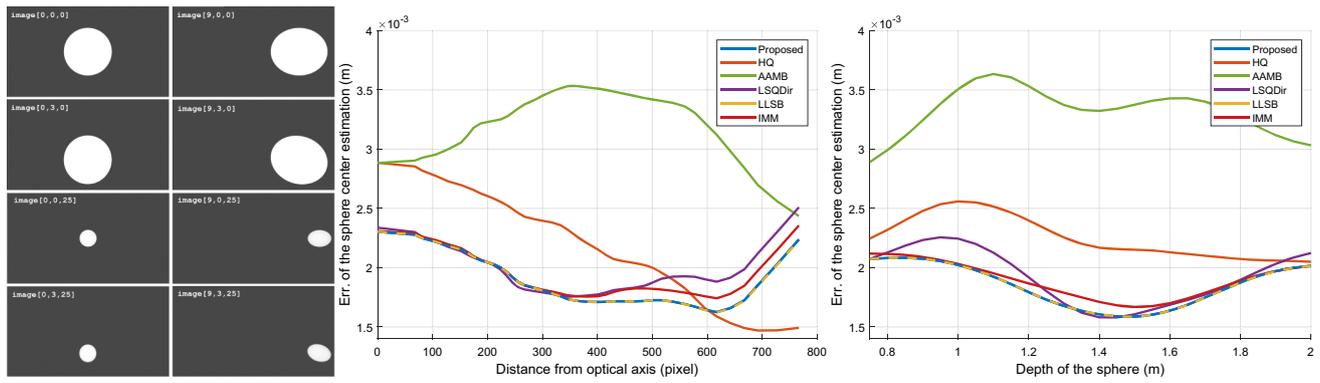


Fig. 10 Input images of the photorealistic setup (left). The indices of the pictures, denoted by $image[i, j, k]$ in the upper left corners, determine the position of the sphere in the scene. The details of the setup are discussed in Sect. 5.4. Sphere estimation error *w.r.t.* the distance from the optical axis (middle) and *w.r.t.* depth of the sphere (right)

Table 5 Photo-realistic results of 1040 images. n_o , Δt , \mathcal{SE} , and \mathcal{SD} denote the outlier number, the runtime, the mean error, and the standard deviation, respectively

Method	n_o	Δt	a		b		e_0		s	
			\mathcal{SE}	\mathcal{SD}	\mathcal{SE}	\mathcal{SD}	\mathcal{SE}	\mathcal{SD}	\mathcal{SE}	\mathcal{SD}
Proposed	1	0.6895	0.3123	0.2969	0.2379	0.2510	0.8047	0.1583	0.0019	0.0005
HQ (Lu et al. 2020)	28 [†]	435.2723	0.4122	0.3003	0.3133	0.2261	0.8095	0.1689	0.0023	0.0006
AAMB (Meng et al. 2020)	5	41.6001	0.3752	0.3125	0.2832	0.2313	0.6275	0.1337	0.0032	0.0006
LSQDir (Halir and Flusser 1998)	29	137.3629	0.3203	0.3182	0.2306	0.2406	0.8060	0.1611	0.0020	0.0006
LLSB (Sun et al. 2016)	39	1.1967	0.3126	0.2973	0.2380	0.2511	0.8055	0.1603	0.0019	0.0005
IMM (Shi et al. 2019)	116	21.0683	0.2994*	0.2892*	0.2280*	0.2456*	1.4053	2.8123	0.1225	0.4172

The ellipse parameters a , b , e_0 are in pixels, and the sphere position s is in meters. *89 out of the 116 outlier cases were filtered to generate IMM results for parameters a and b . Hence, in these columns, the second most competitive results were also highlighted

[†]Note that HQ failed to find any ellipse in 25 cases from the 28 outlier scenario

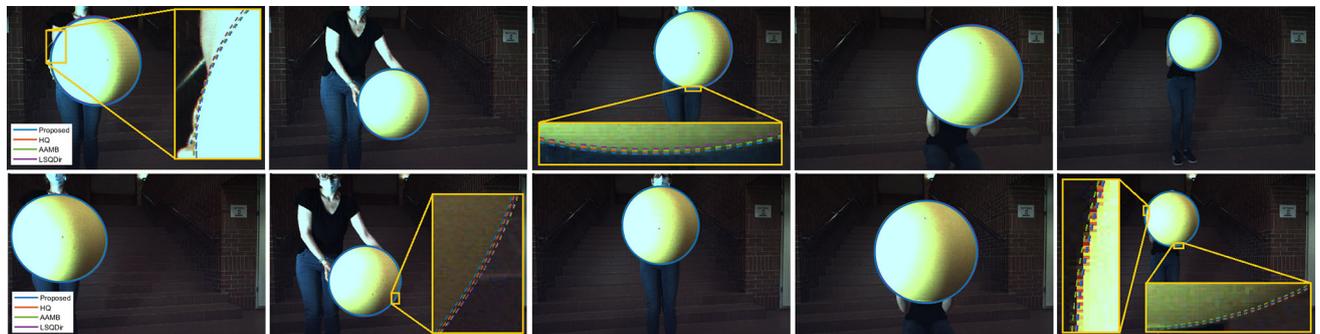


Fig. 11 Stereo image pairs with fitted ellipses. Every column demonstrates a different setup; the top and bottom row shows the right and left camera images taken simultaneously, respectively. The results of

the different methods are similar as the ellipses cover each other in the original images. Yellow rectangles show magnified image parts that illustrate the results more detailedly

sphere with 0.25 m radius was placed 68 times in the scene. Some image pairs with the fitted ellipses are pictured in Fig. 11. The sphere-camera positions were not measured directly, but after a GT transformation: the cameras were translated along axis x by 21 cm. We refined this prior knowledge of the extrinsic parameters using the

checkerboard-based (Zhang 2000; Heikkila and Silven 1997) Stereo Camera Calibrator in Matlab Computer Vision Toolbox. Therefore, the sphere center was estimated in the two camera coordinate systems based on the paired images, and the error was given as the translation error of the sphere centers.

Table 6 The tested methods and the applied sub-steps of the RANSAC method in the stereo camera experiment

Tested methods	The applied methods in RANSAC algorithm		
	Minimal solver	Refitting method	Sphere center from ellipse parameters
<i>Proposed</i>	<i>3pFit</i>	Single step: <i>Direct3pFit</i> (see Table 7)	
<i>HQ</i> (Lu et al. 2020)	<i>HQ</i>	<i>HQ</i>	<i>Ell2Sphere</i>
<i>AAMB</i> (Meng et al. 2020)	<i>AAMB</i>	<i>AAMB</i>	<i>Ell2Sphere</i>
<i>LSQDir</i> (Halir and Flusser 1998)	<i>LSQDir</i>	<i>LSQDir</i>	<i>Ell2Sphere</i>
<i>CB</i> (Hajder et al. 2020; Tóth et al. 2020)	Circle fitting	<i>LSQDir</i>	From implicit ellipse with LMM

Table 7 Comparison of the proposed 2-step and simplified algorithms during the real tests

	<i>3pFit+Ell2Sphere</i> vs. <i>Direct3pFit</i>	
	<i>SE</i>	<i>SD</i>
Camera#1	1.5173×10^{-4}	2.3683×10^{-4}
Camera#2	4.5716×10^{-5}	6.3819×10^{-5}
Trans. err	1.7182×10^{-4}	2.3535×10^{-4}

SE and *SD* denote the mean error and standard deviation. Neither the sphere center estimations of the cameras nor the translation error between the sphere centers showed any observable difference except for some subcentimeter-submillimeter deviation. However, the *Direct3pFit* is less computational demanding as it can be seen in the algorithms of Sect. 4

For real images, edge detection and outlier filtering are required; accordingly, the methods were tested using RANSAC. The sub-steps are compared in Table 6. The proposed process finds the best ellipse contour point set with *3pFit* and *Direct3pFit* computes the sphere center in a single step. The general five-point methods *HQ*, *AAMB*, and *LSQDir* were supplemented with *Ell2Sphere* to facilitate the comparison after ellipse refitting. *CB* applies a circle-based minimal method and Levenberg-Marquardt optimization on implicit ellipse parameters from *LSQDir*.

At first, we examined the difference between our proposed 2-step sphere center estimation via the ellipse fitting and the direct method that directly generates the sphere from the points. The results listed in Table 7 justify that the two approaches had similar results, and the results differed only due to rounding errors. Hence, applying *Direct3pFit* in a single operation during refitting is an appropriate choice.

The main results of the experiments over the concurrent methods were plotted in Fig. 12. The horizontal axis exhibits all of the test cases. The five dots are the measured errors in every column based on the same input using varying methods. The results are ordered based on the error values of the proposed method for easier visual comparison. The fewer points below the dark blue points, the better the proposed fitting algorithm. The color of the lowest dots gives the ratio of the best estimation listed in the upper left corner of the diagram. Missing dots in a column mean that a result is an outlier

or the error rate is higher than 6 cm. The outlier ratio are also listed above, and the most conspicuous issue was the high rate of *LSQDir* and the deviation of the error of *CB*. The other approaches had only negligible wrong results. Using the proposed algorithm, matching the sphere center estimation with the two cameras gave 1 – 6 cm error while the average distance of the sphere and the cameras was around 1–2 m. In general, the error was not linear *w.r.t.* the distance of the sphere and the cameras, but close sphere images (as the stereo image example in fourth column of Fig 16) definitely results in a smaller error. The positions of purple points in some of the last test cases indicate that *LSQDir* produced the best, only 2 cm error. In these instances, sphere-camera distance was long (like in fifth column of Fig 16) meaning that *LSQDir* was robust despite the outlier rate. Our method outperformed the rival ones giving the best results for 69.70% of the test cases.

Figure 13 represents the distribution of the same data series categorized into 0.5 cm long intervals in a histogram-like manner. The proposed method performed better within the 1.5–2 cm error rate, while most cases were between 2 and 3 cm. Remarkable that 9 test results of *CB* were below 1.5 cm error. In precision, the *AAMB* method lagged at the beginning. The drawback of *CB* is that 41.86% of the tests were above the 6 cm error and cannot be listed in the histogram.

For a more profound analysis, Fig. 14 shows the sphere location error along the three main coordinate axes separately. During the estimation for the coordinates *x* and *y*, the spread was below 2 cm. The mean error was not accurate around zero. This may be caused by the light direction affecting the edge positions. The plot suggests that the detectable error comes from the wrong estimation, mainly along axis *z*, *i.e.*, the methods are sensitive in terms of depth estimation. However, 50% of the estimation was still within the ± 2 cm error level. The advantage of the proposed method also appeared along axis *z*: the standard deviation was reduced from 6 to 4 cm compared to other methods. It can be seen along every axis that these camera instincts negatively affect the circle-based results of *CB* as they have the most significant spread in all three directions.

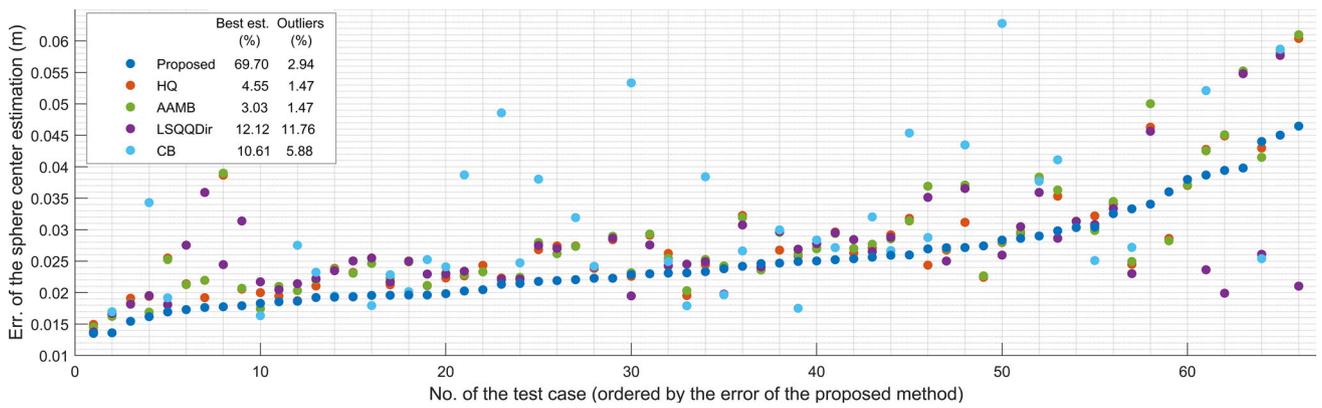


Fig. 12 The error between the sphere center estimations with a stereo camera using 68 test cases. The measurements are sorted by the error of the proposed algorithm. Every data quartet along a vertical line is the estimated error of the same setup by the different methods. The distance of the sphere and the camera centers were between 1 and 2 m on

average. Further distances do not show significantly higher error; the image quality and illumination principally influence the results. A typical error is between 1–4 cm. The proposed algorithm performs better in 84% of the testing scenarios

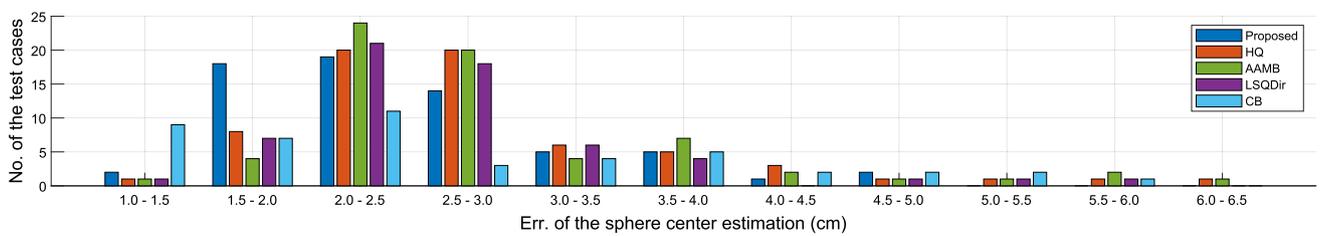


Fig. 13 The distribution of the error between the sphere center estimations with stereo camera generated from the data in Fig. 12. Our method was very dominant in the regions below 2 cm error, and no case generated more than 5 cm error opposite to the other methods

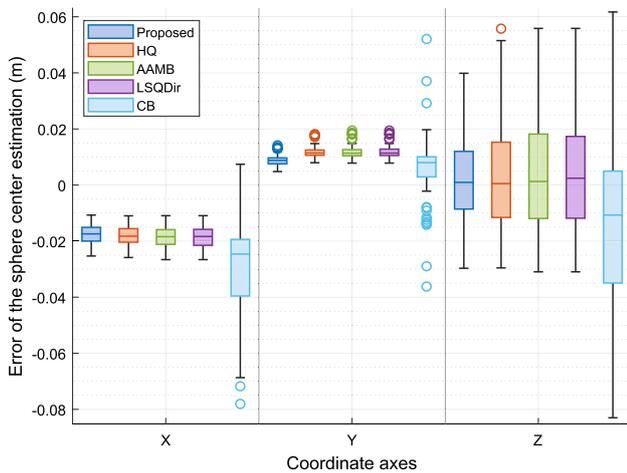


Fig. 14 Boxplots of sphere center estimation errors. Testing data and rival methods are the same as in Fig. 12

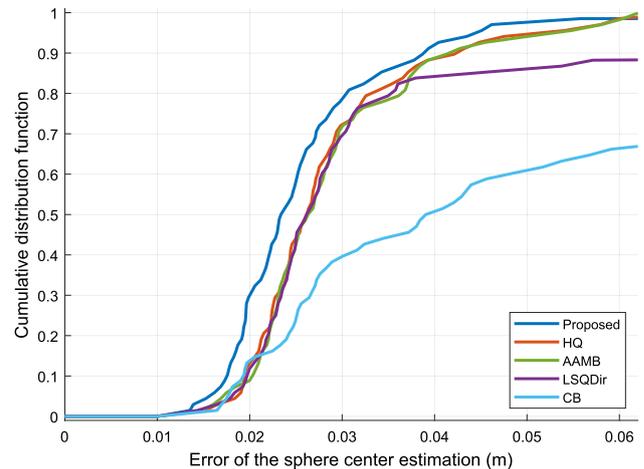


Fig. 15 The cumulative distribution (CDF) of the real tests. The data sets were smoothed by Gaussian-weighted moving average with three neighboring data points. The CDF of the proposed method has a minor error with 2–4 mm while reaching the exact value of the CDF

The cumulative distribution function (CDF) shown in Fig. 15 has very similar results as a continuous function. The steepest parts of the functions are similarly around 1.5–3 cm. The mean error values for the methods are listed in the bottom right corner. In total, the proposed method precedes

the others with 2–4 mm. Due to the high amount of outliers and less accurate estimations, the *LSQDir* and *CB* does not reach the top value within 6 cm.



Fig. 16 An input image sequence from Camera2 of Fig. 17. The estimated sphere centers regarding these images are the blue points around $y = 1.8$ m in Fig. 18

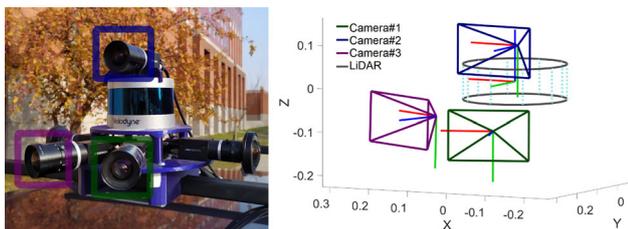


Fig. 17 The schematic figure of the sensor calibration using the estimated extrinsic parameters (right) represents a realistic adjustment considering the photo of the mounted setup (left). In this experiment, the three perspective cameras (green, blue, purple) were calibrated from the equipped five, the two sideways fisheye cameras were not installed in our system

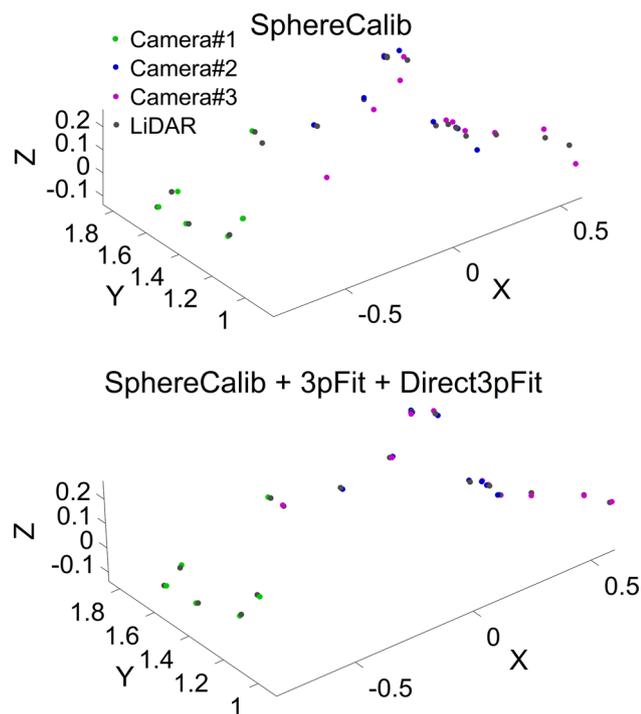


Fig. 18 The estimated sphere centers registered into the LiDAR coordinate system show the measurement error and spread before (top) and after (bottom) the improvements in the calibration pipeline. The sphere was placed typically to 1–2.5 m far from the sensors. The calibration was accomplished using 6, 9, 11, and 19 sphere centers from Camera#1 (green), Camera#2 (blue), Camera#3 (purple), and the LiDAR (grey). The result of the improved pointset registration gave the setup plotted on the left side in Fig. 17

5.6 Real-World Application: Camera-LiDAR Calibration

In this part of the paper, we illustrate the utility of the proposed algorithms via a multi-camera and LiDAR calibration experiment for an autonomous driving system. In Tóth et al 2020, we presented how spheres are suitable for LiDAR-camera calibration, which has three main steps: (i) pointcloud-based sphere center estimation, (ii) image-based sphere center estimation, and (iii) relative pose estimation by pointset registration. The theoretical contribution of this paper was applicable in the image-based sphere recognition, as *3pFit* was executed in a RANSAC framework to find the ellipse in the image and *Direct3pFit* to find the estimated sphere center based on the best consensus set. This section aims to present the results of the improved calibration process.

A Velodyne VLP-16 LiDAR sensor and three Hikvision cameras (see the left picture in Fig. 17) were mounted on our test vehicle. Initially, the cameras under the LiDAR were oriented about $\pm 20^\circ$ to the forward direction, and the one on the top was looking ahead, but the displacement errors during the mounting process and the car movement motivated us to apply a precise calibration. The cameras and the LiDAR were hardware-triggered to take the images, and the pointcloud was captured during a single turnaround of the LiDAR simultaneously. The calibration input was one continuous capture with 4 frames per second with half of the ordinary image resolution (see Hikvision camera intrinsics in Table 3) while one walked with a sphere back and forth in front of the sensor setup. The method calibrated every camera to the LiDAR-based on 6, 9, and 11 image-pointcloud pair. The number of the test images and LiDAR scans depended on the visibility of the sphere from different points of view. Some recordings contained the sphere in two camera images; hence, only 19 LiDAR clouds were processed in total due to the overlap in the field of view.

One image sequence of Camera#2 can be seen in Fig. 16. The mean error and standard deviation compared to the original *SphereCalib* (Tóth et al 2020) method of this example were listed in Table 8. The novel methods improved the calibration significantly: the mean error was reduced from 5.48 to 1.25 cm with sub-centimeter standard deviation results. The 3D point registration gives an error within a few centimeters.

Table 8 Mean error (SE ; meters) and stranded deviation (SD ; meters) of the sphere center registration between the LiDAR and the cameras before and after improving our calibration pipeline with the new ellipse and sphere estimation algorithms

		Camera#1	Camera#2	Camera#3	Accumulated
SphereCalib (Tóth et al 2020)	SE	0.0635	0.0362	0.0694	0.0548
	SD	0.0266	0.0431	0.0488	0.0448
SphereCalib + $3pFit$ + $Direct3pFit$	SE	0.0121	0.0137	0.0115	0.0125
	SD	0.0025	0.0096	0.0054	0.0070

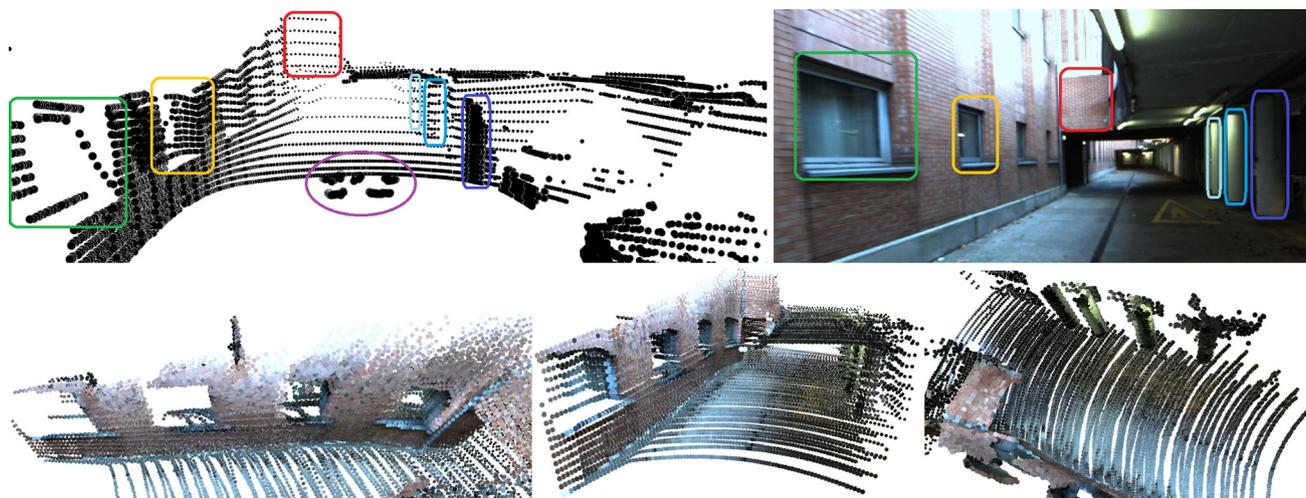


Fig. 19 To validate the sphere-based calibration results, we registered colored pointcloud sequence by the iterative closest point (ICP, Besl and McKay (1992)) algorithm (bottom; same pointcloud from different points of view) and displayed one original snapshot of the scene (top left: pointcloud; top right: camera image). On the top, the framed regions with the same color cover similar areas and are easy to identify in the colored 3D points. The points circled with purple belong to the

ego car. On the bottom, the coloring around the window sills and the eaves (bottom left) exemplify the precision of the extrinsic calibration parameters, including the image-based sphere estimation. The dividing black line (bottom middle) and yellow road markings (bottom right) are also recognizable, and the road is in shadow after the third pillar (bottom right), like in the camera image. The main pillars with some intense greenish light are also observable (Color figure online)

The calibration result is compared to the actual setup in Fig. 17. The origins and the coordinate axis directions represent a realistic sensor setup. Especially the rotation parameters are accurate, as the coloring shows in the further qualitative examples. This estimation is essential, as the displacement can be more easily guaranteed during mounting, but the rotation of the cameras is more often error-prone than planned.

Figure 18 visualizes the estimated 6, 12, 10, and 19 sphere centers of Camera#1, Camera#2, Camera#3, and the LiDAR, respectively, after transforming them in the LiDAR coordinate system. This qualitative comparison confirms the results in Table 8 as the error can be observed in the point pair registration before applying $3pFit$ and $Direct3pFit$ in the top image. After the improvements, the sphere centers nearly overlap in the bottom image, as the error rate is minimal.

In Fig. 19, the colored and registered pointcloud sequence is a visual illustration of the reliability. As we had neither GPS nor IMU data, the iterative closest point (ICP, Besl and McKay (1992)) algorithm concatenated the LiDAR snapshots to map the environment and the movement of the

vehicle. Several landmarks and objects are well-colored: the window sills, the basement pillars, the gutter, the road sign, and some shadows.

In conclusion, the calibration experiments revealed that spheres are practical for 2D–3D data registration. A further application of spheres is panoramic image stitching, as the projected ellipse centers can be matched between two cameras, which gives the homography between the images (Hartley and Zisserman 2004). We plan to expand the autonomous test vehicle with additional sensors and use sphere-based calibration for adequate reliability.

6 Conclusion

This paper proposed a simple analytical three-point algorithm for parameter estimation of a sphere and its projected ellipse, which needs no numerical optimization. We demonstrated that the contour points of the sphere in the image form a special ellipse. Based on this particular setup, two geometric constraints have been applied here: the *projectional*

constraint and the *axial constraint*. Experimental results show that the proposed ellipse fitting method outperforms the rival techniques in various environments. Our analysis also reveals that the introduced constraints and some simplification steps improve the precision of the proposed algorithm. Therefore, this method is faster even than the rival three-point approaches. Given these results, we believe that the proposed fitting algorithms provide accurate parameters to assist various algorithms in utilizing spheres.

Acknowledgements EFOP–3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies—The Project is supported by the Hungarian Government and co-financed by the European Social Fund. Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund. L. Hajder thanks the support of the “Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

Funding Open access funding provided by Eötvös Loránd University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Perspective sphere projection

The implicit form of a sphere can be written as

$$(\mathbf{x} - \mathbf{s})^T (\mathbf{x} - \mathbf{s}) - r^2 = 0, \tag{A.1}$$

where $\mathbf{s} = [x_0 \ y_0 \ z_0]^T$ and r are the sphere center and radius, respectively. A spatial point $\mathbf{x} \in \mathbb{R}^3$ is on the surface of the sphere if (A.1) holds.

The sphere is a special kind of quadric. The implicit equation of a general quadric is

$$[\mathbf{x}^T \ 1] \mathbf{M} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0, \tag{A.2}$$

where $\mathbf{M} = \mathbf{M}^T$ is a 4×4 symmetric matrix. It can be rewritten as

$$[\mathbf{x}^T \ 1] \begin{bmatrix} \tilde{\mathbf{M}} \mathbf{b} \\ \mathbf{b}^T \ v \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0, \tag{A.3}$$

where the 3×3 matrix $\tilde{\mathbf{M}}$ is also symmetric.

For a sphere, the matrix \mathbf{M} , representing the quadric, can be written as

$$\mathbf{M} = \begin{bmatrix} \mathbf{I} & -\mathbf{s} \\ -\mathbf{s}^T & \mathbf{s}^T \mathbf{s} - r^2 \end{bmatrix}. \tag{A.4}$$

Thus, for a sphere, $\tilde{\mathbf{M}} = \mathbf{I}$, $\mathbf{b} = -\mathbf{s}$, and $v = \mathbf{s}^T \mathbf{s} - r^2$.

Contour of a projected quadric. For a pin-hole camera, the camera intrinsic parameters are frequently represented by the so-called camera matrix \mathbf{K} . The projection of a point is given as follows: $\lambda [u \ v \ 1]^T = \mathbf{K}\mathbf{x}$, where λ is the so-called projective depth and $\mathbf{u} = [u \ v \ 1]^T$ is the homogeneous representation of the image points. Therefore, a spatial point \mathbf{x} on the reprojected ray of image point \mathbf{u} can be obtained as $\mathbf{x} = \lambda \mathbf{K}^{-1} \mathbf{u}$.

If the reprojected ray is substituted, we get the following implicit equation:

$$[\lambda \mathbf{u}^T \mathbf{K}^{-T} \ 1] \begin{bmatrix} \tilde{\mathbf{M}} \mathbf{b} \\ \mathbf{b}^T \ v \end{bmatrix} \begin{bmatrix} \lambda \mathbf{K}^{-1} \mathbf{u} \\ 1 \end{bmatrix} = 0.$$

Then $\lambda^2 \mathbf{u}^T \mathbf{K}^{-T} \tilde{\mathbf{M}} \mathbf{K}^{-1} \mathbf{u} + 2\lambda v \mathbf{b}^T \mathbf{K}^{-1} \mathbf{u} + v = 0$. This is a quadratic equation, therefore, a ray and a non-degenerate quadric have at most two intersection points. Contour points produce a single intersection in which case the discriminant is zero: $(2v \mathbf{b}^T \mathbf{K}^{-1} \mathbf{u})^2 - 4 (\mathbf{u}^T \mathbf{K}^{-T} \tilde{\mathbf{M}} \mathbf{K}^{-1} \mathbf{u}) v = 0$. Therefore,

$$(\mathbf{b}^T \mathbf{K}^{-1} \mathbf{u})^2 = v \mathbf{u}^T \mathbf{K}^{-T} \tilde{\mathbf{M}} \mathbf{K}^{-1} \mathbf{u}. \tag{A.5}$$

This equation represents a conic in the image.

Special case: projection of spheres. For spheres, $\mathbf{b} = -\mathbf{s}$, $v = \mathbf{s}^T \mathbf{s} - r^2$, and $\tilde{\mathbf{M}} = \mathbf{I}$. Moreover, if the cameras are calibrated as it is written in (2) then $\mathbf{K} = \mathbf{I}$. Therefore, (A.5) can be modified as

$$(\mathbf{s}^T \mathbf{u})^2 = (\mathbf{s}^T \mathbf{s} - r^2) \mathbf{u}^T \mathbf{u}. \tag{A.6}$$

If the coordinates within the vectors are considered, the formula becomes $(ux_0 + vy_0 + z_0)^2 = (x_0^2 + y_0^2 + z_0^2 - R^2)(u^2 + v^2 + 1)$. One can rewrite it as

$$\begin{aligned} & (R^2 - y_0^2 - z_0^2)u^2 + (R^2 - x_0^2 - z_0^2)v^2 + \\ & 2x_0y_0uv + 2x_0z_0u + 2y_0z_0v + R^2 - x_0^2 - y_0^2 = 0. \end{aligned} \tag{A.7}$$

Appendix B Implicit approach with SVD

The equation system using three independent ellipse contour points $\mathbf{p}_i = [\hat{u}_i \hat{v}_i 1]^T$, $i \in \{1, 2, 3\}$ can be written in matrix form, based on (4), as follows:

$$\begin{bmatrix} \hat{u}_1^2 & \hat{u}_1 \hat{v}_1 & \hat{v}_1^2 & \hat{u}_1 & \hat{v}_1 & 1 \\ \hat{u}_2^2 & \hat{u}_2 \hat{v}_2 & \hat{v}_2^2 & \hat{u}_2 & \hat{v}_2 & 1 \\ \hat{u}_3^2 & \hat{u}_3 \hat{v}_3 & \hat{v}_3^2 & \hat{u}_3 & \hat{v}_3 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (\text{A.8})$$

A more compact form is $\mathbf{M}\Phi = \mathbf{0}$, where $\mathbf{M} \in \mathbb{R}^{3 \times 6}$ is the matrix containing the polynomial functions of the coordinates and Φ is the column vector containing the six ellipse coefficients. Applying Singular Value Decomposition (SVD) $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ yields the basis vectors for the null space of \mathbf{M} . The last three column vectors of \mathbf{V} give the three right null vectors \mathbf{v}_4 , \mathbf{v}_5 , and \mathbf{v}_6 of \mathbf{M} . The linear combination of those gives the solution for the ellipse parameters:

$$\Phi(\gamma, \delta) = \mathbf{v}_4 + \gamma \mathbf{v}_5 + \delta \mathbf{v}_6. \quad (\text{A.9})$$

The weights $\gamma, \delta \in \mathbb{R}$ for the linear combination should be computed considering the constraints, discussed in (10). Substituting the expressions into C_1 and C_2 leads us to an equation system of third degree polynomials containing two variables.

Unfortunately, a closed formula to γ and δ expressed by the null space vector elements is not attainable, because there is no general solution of the following equation system:

$$\begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix} [\gamma^3 \ \gamma^2 \delta \ \gamma^2 \ \gamma \delta^2 \ \gamma \delta \ \gamma \ \delta^3 \ \delta^2 \ \delta \ 1]^T = \mathbf{0}, \quad (\text{A.10})$$

where every coefficient g_i of \mathbf{g} and h_i of \mathbf{h} are third degree polynomials of some of the v_{jk} values. v_{jk} denotes the k th element of \mathbf{v}_j , $i \in \{1, \dots, 10\}$, $j \in \{4, 5, 6\}$, $k \in \{1, \dots, 6\}$.

To the best of our knowledge, these coefficients cannot be factorized and do not contain any usable special property to find an explicit solution to γ and δ . Hence, we applied the symbolic solver built-in function in MATLAB on the equations containing the concrete values of vectors \mathbf{v}_4 , \mathbf{v}_5 , and \mathbf{v}_6 , but it is time-consuming which makes this solution not recommended. Shi et al. (2019) tried to solve it similarly in an implicit way, but by different techniques, and also found a solution only with multiple roots.

References

Barath, D., & Matas, J. (2021). Graph-cut ransac: Local optimization on spatially coherent structures. *IEEE Transactions on Pattern Anal-*

ysis and Machine Intelligence. <https://doi.org/10.1109/TPAMI.2021.3071812>

- Besl, P., & McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256. <https://doi.org/10.1109/34.121791>
- Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: delving into high quality object detection. In: *2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018*, pp. 6154–6162, <https://doi.org/10.1109/CVPR.2018.00644>
- Chum, O., & Matas, J. (2005). Matching with PROSAC—progressive sample consensus. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR 2005), 20–26 June 2005, San Diego, CA, USA*, pp. 220–226, <https://doi.org/10.1109/CVPR.2005.221>
- Dong, W., Roy, P., Peng, C., et al. (2021). Ellipse R-CNN: Learning to infer elliptical object from clustering and occlusion. *IEEE Transactions on Image Processing*, 30, 2193–2206. <https://doi.org/10.1109/TIP.2021.3050673>
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. <https://doi.org/10.1145/358669.358692>
- Fitzgibbon, A., Pilu, M., & Fisher, R. (1996). Direct least squares fitting of ellipses. In: *Proceedings of 13th international conference on pattern recognition* (Vol. 1, pp. 253–257). <https://doi.org/10.1109/ICPR.1996.546029>
- Fitzgibbon, A. W., Pilu, M., & Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 476–480. <https://doi.org/10.1109/34.765658>
- Glaeser, G., Stachel, H., & Odehnal, B. (2016). *The Universe of Conics*. Heidelberg: Springer Spektrum Berlin.
- Guan, J., Deboeverie, F., Slembrouck, M., et al. (2015). Extrinsic calibration of camera networks using a sphere. *Sensors*, 15(8), 18985–19005. <https://doi.org/10.3390/s150818985>
- Hajder, L., Tóth, T., & Pusztai, Z. (2020). Automatic estimation of sphere centers from images of calibrated cameras. In: *Proceedings of the 15th international joint conference on computer vision, imaging and computer graphics theory and applications, VISIGRAPP 2020, Volume 4: VISAPP, Valletta, Malta, February 27–29, 2020*. SCITEPRESS, pp. 490–497. <https://doi.org/10.5220/0009164304900497>
- Halir, R., & Flusser, J. (1998). Numerically stable direct least squares fitting of ellipses. In: *Proceedings 6th international conference in central Europe on computer graphics and visualization*. WSCG
- Hartley, A., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2 edn). Cambridge: Cambridge University Press, <https://doi.org/10.1017/CBO9780511811685>
- Heikkila, J., & Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pp. 1106–1112. <https://doi.org/10.1109/CVPR.1997.609468>
- Hoffmann, C. M. (1989). *Geometric and Solid Modeling: An Introduction*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Hough, P.V. (1962). Method and means for recognizing complex patterns US Patent
- Jabbar, A., Mendes, A., Chalup, S.K. (2019). Comparing ellipse detection and deep neural networks for the identification of drinking glasses in images. In: *Proceedings of 12th international conference computer vision systems, ICVS 2019, Thessaloniki, Greece, September 23–25, 2019*, pp. 319–329, https://doi.org/10.1007/978-3-030-34995-0_29

- Klüng, A., & Meli, F. (2005). Self calibration method for 3d roundness of spheres using an ultraprecision coordinate measuring machine. In: *Proceedings of the 5th EUSPEN international conference*, pp 193–196.
- Liu, Z., & Qiao, H. (2009). Multiple ellipses detection in noisy environments: A hierarchical approach. *Pattern Recognition*, 42(11), 2421–2433. <https://doi.org/10.1016/j.patcog.2009.01.028>
- Liu, Z., Qiao, H., & Xu, L. (2007). Investigation on multisets mixture learning based object detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(8), 1339–1351. <https://doi.org/10.1142/S0218001407005971>
- Liu, Z., Wu, Q., Wu, S., et al. (2017). Flexible and accurate camera calibration using grid spherical images. *Optics Express* 25, 13, 15,269–15,285. <https://doi.org/10.1364/OE.25.015269>
- Lu, C., Xia, S., Shao, M., et al. (2020). Arc-support line segments revisited: An efficient high-quality ellipse detection. *IEEE Transactions on Image Processing*, 29, 768–781. <https://doi.org/10.1109/TIP.2019.2934352>
- Lu, Y., & Payandeh, S. (2010). On the sensitivity analysis of camera calibration from images of spheres. *Computer Vision and Image Understanding*, 114, 8–20. <https://doi.org/10.1016/j.cviu.2009.09.001>
- Mai, F., Hung, Y.S., & Zhong, H., et al. (2007). A hierarchical approach for fast and robust ellipse extraction. In: *2007 IEEE international conference on image processing*, pp V-345–V-348, <https://doi.org/10.1109/ICIP.2007.4379836>
- Matsuoka, R., & Maruyama, S. (2016). Eccentricity on an image caused by projection of a circle and a sphere. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* III-5:19–26. <https://doi.org/10.5194/isprs-annals-III-5-19-2016>
- Meng, C., Li, Z., Bai, X., et al. (2020). Arc adjacency matrix-based fast ellipse detection. *IEEE Transactions on Image Processing*, 29, 4406–4420. <https://doi.org/10.1109/TIP.2020.2967601>
- Nair, P. S., & Saunders, A. T., Jr. (1996). Hough transform based ellipse detection algorithm. *Pattern Recognition Letters*, 17(7), 777–784. [https://doi.org/10.1016/0167-8655\(96\)00014-1](https://doi.org/10.1016/0167-8655(96)00014-1)
- Prasad, D. K., Leung, M. K. H., & Quek, C. (2013). Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46(5), 1449–1465. <https://doi.org/10.1016/j.patcog.2012.11.007>
- Rosin, P. L. (1993). A note on the least squares fitting of ellipses. *Pattern Recognition Letters*, 14(10), 799–808. [https://doi.org/10.1016/0167-8655\(93\)90062-1](https://doi.org/10.1016/0167-8655(93)90062-1)
- Ruijl, T., Franse, J., & van Eijk, J. (2001). Ultra precision cmm aiming for the ultimate concept. In: *Proceedings of the 2nd EUSPEN international conference*, pp 234–237, <https://doi.org/10.1051/ijmqe/2014009>
- Shi, K., Li, X., & Xu, H. et al. (2019). 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, pp 65–70, <https://doi.org/10.5220/0008096300650070>
- Sun, J., He, H., & Zeng, D. (2016). Global calibration of multiple cameras based on sphere targets. *Sensors*, 16(1), 77. <https://doi.org/10.3390/s16010077>
- Tóth, T., Pusztai, Z., & Hajder, L. (2020). Automatic lidar-camera calibration of extrinsic parameters using a spherical target. In: *2020 IEEE international conference on robotics and automation, ICRA 2020, Paris, France, May 31–August 31, 2020*. IEEE, pp 8580–8586, <https://doi.org/10.1109/ICRA40945.2020.9197316>
- Xie, Y., & Ji, Q. (2002). A new efficient ellipse detection method. In: *16th international conference on pattern recognition*, pp 957–960 vol.2, <https://doi.org/10.1109/ICPR.2002.1048464>
- Xiong, F., et al. (2021). Lens-Free optical detection of thermal motion of a submillimeter sphere diamagnetically levitated in high vacuum. *Physical Review Applied*. <https://doi.org/10.1103/physrevapplied.16.1011003>
- Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1), 59–76. [https://doi.org/10.1016/S0262-8856\(96\)01112-2](https://doi.org/10.1016/S0262-8856(96)01112-2)
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334. <https://doi.org/10.1109/34.888718>
- Zuliani, M., Kenney, C.S., & Manjunath, B.S. (2005). The multi-RANSAC algorithm and its application to detect planar homographies. In: *Proceedings of the 2005 international conference on image processing, ICIP 2005, Genoa, Italy, September 11–14, 2005*, pp 153–156, <https://doi.org/10.1109/ICIP.2005.1530351>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.