Spherocylinder Pose from Silhouette

Erol Ozgur[1]
erol.ozgur@sigma-clermont.fr
Mohammad Alkhatib[1]
mohammad.alkhatib@sigma-clermont.fr
Youcef Mezouar[1]
youcef.mezouar@sigma-clermont.fr
Emmanuel Buc[2]
ebuc@chu-clermontferrand.fr
Bertrand Le Roy[3]
bertrand.leroy@chu-st-etienne.fr
Adrien Bartoli[2]
adrien.bartoli@gmail.com

[1] Clermont Auvergne INP
Institut Pascal
Clermont-Ferrand, France
[2] University Hospital
Clermont-Ferrand, France
[3] University Hospital
Saint-Etienne, France

Figure 1: Examples of spherocylindrical objects.

Abstract

A spherocylinder is a cylinder capped at one end by a hemisphere of equal radius. We present `ScyPoSe` (Spherocylinder Pose from Silhouette), a theory and method for estimating the spherocylinder 5-DoF pose from its silhouette in a single 2D image. `ScyPoSe` exploits the geometric property that the spherically-normalised silhouette of a spherocylinder is formed of circular arcs on the camera-centred unit sphere: two great-circle arcs from the cylinder and a single small-circle arc from the hemisphere. This structure reduces pose estimation to a sequence of constrained plane fitting problems, admitting simple, fast, and convex solutions for both the minimal 5-point and redundant $n > 5$ point cases. We further introduce `Robust-ScyPoSe`, a RANSAC-based extension that jointly estimates pose, classifies silhouette points, and rejects erroneous points. Experiments on synthetic and real data demonstrate that `Robust-ScyPoSe` pushes the state of the art forward in accuracy and robustness.

1 Introduction

The spherocylinder is a simple geometric primitive governed by only one shape parameter -the radius- and five pose parameters. Its study is appealing; first, it fits many real-world objects: pills, missiles, submarines, fingertips and ultrasound probes (figure 1), to name but a few; second, its pose is fully encoded by its 2D silhouette (*aka* occluding contour [1, 4]). In other words, one could use a spherocylinder model to recover the spatial location of a real object, whether well-textured or untextured, with minimal modelling burden (just a radius parameter, without texture mapping), and without requiring point correspondences. Interestingly, an unknown radius leads to a one-parameter pose family expressible in closed-form.

Yet, estimating spherocylinder pose from a single calibrated perspective image remains surprisingly underexplored. The silhouette consists of two line segments from the cylinder flanking one elliptical curve from the hemisphere. These primitives belong to different geometric families; this heterogeneous composition makes a unified treatment hard.

Only one prior work explicitly addresses single-view spherocylinder pose estimation [3]. It follows a two-stage strategy: first estimating the cylindrical component from line geometry, then recovering the hemispherical component through ellipse- and cone-based fitting. While intuitive, this loose coupling introduces architectural heterogeneity into the estimation pipeline — the cylinder is handled through projective line geometry [6], while the hemispherical cap is treated through conic or quadric formulations [10] — mixing different geometric representations and alternating between linear and nonlinear solvers. Robustified variants further sacrifice geometric rigour for speed, replacing true geometric distances with algebraic proxies [2, 9]. Despite the simplicity of the shape, a coherent, closed-form, geometrically meaningful formulation for single-view spherocylinder pose estimation from silhouette observations has yet to emerge.

We change the perspective entirely. Projecting silhouette points onto the camera-centred unit sphere — spherical normalisation — dissolves the heterogeneity: the two cylinder lines become great-circle arcs and the hemisphere curve becomes a small-circle arc. Each arc independently obeys cocircularity and coplanarity, while the spherocylinder geometry further ties them together through joint constraints. The problem thus reduces to a sequence of constrained plane-fitting problems — linear, convex, and geometrically meaningful — under a single unified framework.

This insight yields two algorithms. Silhouette points of a spherocylinder naturally fall into three parts: those lying on each of the two cylinder silhouette lines and those lying on the hemisphere contour. Knowing which point belongs to which part is referred to as classification. `ScyPoSe` assumes classification is known and estimates the full 5-DoF pose from classified silhouette points, delivering the first minimal 5-point closed-form solution alongside a redundant variant that exploits all available points for maximum accuracy. `Robust-ScyPoSe` drops this requirement entirely, jointly estimating pose, classifying points, and rejecting erroneous points — deployable directly on raw, unprocessed contours. We summarise our contributions as follows:

- We identify and exploit consistent *independent constraints* and *joint constraints* coupling the cylinder and hemisphere geometry within a unified plane-fitting framework.
- We derive `ScyPoSe`, the first minimal 5-point closed-form solver for spherocylinder pose estimation and an $n > 5$ point solver based on linear least squares rounds.
- We propose `Robust-ScyPoSe`, which jointly estimates pose, classifies silhouette points, and rejects erroneous ones — requiring no prior point classification.

We validate `ScyPoSe` and `Robust-ScyPoSe` on synthetic and real data. Real-data experiments cover two applications: ultrasound probe pose estimation, which could enable visual servoing for tumour localisation during laparoscopic surgery, and finger pose estimation, which could enable collaborative grasping and assembly in cobotics.

2 Methodology

We introduce the notation and constraints, describe the plane-fitting approach, and present `ScyPoSe` together with its robust variant `Robust-ScyPoSe`.

2.1 Notation and Modelling

General points. We use italics for scalars and bold for vectors and matrices. Underlines denote unit vectors such as $\underline{\mathbf{v}} \in \mathbb{S}^2$, where $\mathbb{S}^2$ is the unit sphere centred at $\mathbf{0}_{3\times 1}$. Function η spherically normalises a vector, hence projects its tip point onto $\mathbb{S}^2$, as:

$$\underline{\mathbf{x}} = \eta(\mathbf{X}) = \mathbf{X}/\|\mathbf{X}\|, \quad \mathbf{X} \in \mathbb{R}^3. \tag{1}$$

Overbars denote homogeneous coordinate vectors such as $\bar{\mathbf{v}} = [\mathbf{v}^\top, 1]^\top$. We use the standard camera coordinate system, with the optical centre $\mathbf{O} = \mathbf{0}_{3\times 1}$ as origin, the principal axis as $\underline{\mathbf{z}} = [0,0,1]^\top$ and the image axes parallel to $\underline{\mathbf{x}} = [1,0,0]^\top$ and $\underline{\mathbf{y}} = [0,1,0]^\top$. Matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$ represents the camera intrinsics, containing the principal point $\mathbf{p}_o \in \mathbb{R}^2$ and focal lengths $f_x \in \mathbb{R}^+$ and $f_y \in \mathbb{R}^+$ in pixels. For distortion-corrected images, point $\mathbf{X} \in \mathbb{R}^3$ projects to image point $\mathbf{x} \in \mathbb{R}^2$ via $\bar{\mathbf{x}} \propto \mathbf{K}\mathbf{X}$. An image point $\mathbf{x}$ backprojects to a unit ray $\underline{\mathbf{q}} = \eta(\mathbf{K}^{-1}\bar{\mathbf{x}})$.

Spherocylinder pose parameterisation. A spherocylinder caps a cylinder with a hemisphere, both of radius $R \in \mathbb{R}^+$. The hemisphere centre $\mathbf{H} \in \mathbb{R}^3$ and cylinder axis direction $\underline{\mathbf{u}} \in \mathbb{S}^2$ define the spherocylinder pose, represented jointly as $\xi = [\mathbf{H}^\top, \underline{\mathbf{u}}^\top]^\top \in \mathbb{R}^3 \times \mathbb{S}^2$. The pose has 5 DoF as rotational symmetry about $\underline{\mathbf{u}}$ leaves one DoF unobservable. Unlike Plücker coordinates, which represent the 4 DoF of infinite lines via direction and moment, the spherocylinder pose anchors the hemisphere centre $\mathbf{H}$. We assume the spherocylinder axis is not collinear with the optical axis, so the cylinder's silhouette consists of two image lines $\ell_+, \ell_- \in \mathbb{R}^3$. The image lines $\ell_\pm$ backproject to tangent planes $\Pi_\pm$, with outward-pointing unit normals $\underline{\mathbf{m}}_\pm = \eta(\mathbf{K}^\top \ell_\pm)$. The subscript $\pm$ refers to either or both cylinder silhouette lines.

Image observations. We use a point set $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and a corresponding label set $\mathcal{L} = \{l_1, \ldots, l_n\}$, where $l_i \in \{\mathtt{h}, \mathtt{c}_+, \mathtt{c}_-, \mathtt{o}\}$ denotes whether point $\mathbf{x}_i$ belongs to the hemisphere ($\mathtt{h}$), a cylinder edge ($\mathtt{c}_\pm$), or is an outlier ($\mathtt{o}$). Each point $\mathbf{x}_i \in \mathcal{P}$ with $l_i \neq \mathtt{o}$ backprojects to a ray $\underline{\mathbf{q}}_i \in \mathbb{S}^2$ passing through the 3D contour generator. We use $\mathcal{Q} = \{\underline{\mathbf{q}}_1, \ldots, \underline{\mathbf{q}}_n\}$. For given labels, we extract point subsets with the shorthand $\mathcal{P}_* = \{\mathbf{x}_i \,|\, l_i = *\}$; *e.g.*, $\mathcal{P}_\mathtt{h}$ are the hemisphere points.

2.2 Spherocylinder Pose Constraints

2.2.1 Constraining the Spherocylinder Pose

A sphere inscribed within a cylinder retains a single translational DoF along the cylinder's central axis. A ray from the optical centre that is tangent to the sphere's contour generator resolves this DoF, yet yields two solutions symmetric about the ray within the cylinder. Therefore, a single spherically-normalised point ray from the hemisphere silhouette pins both candidate hemisphere centres to two discrete positions along the cylinder's axis. Consequently, 5 points — 2 per cylinder silhouette line and 1 from the hemisphere — reduce the spherocylinder to exactly two candidate poses.

2.2.2 Geometric Properties

We characterise the backprojections of the spherocylinder's silhouette on $\mathbb{S}^2$. We use the prefix *sphero* for figures on $\mathbb{S}^2$ and for planes cutting $\mathbb{S}^2$. A *sphero-circle* $\mathcal{C} \subset \mathbb{S}^2$ is supported by a *sphero-plane* Π with unit normal $\underline{\mathbf{n}} \in \mathbb{S}^2$ and offset $d \in (-1,1)$, such that every $\underline{\mathbf{q}} \in \mathcal{C}$ satisfies $\underline{\mathbf{n}}^\top \underline{\mathbf{q}} = d$. For $d \neq 0$ and $|d| < 1$, Π is offset from $\mathbf{O}$ and $\mathcal{C}$ is a *sphero small-circle*, arising from the spherically-normalised silhouette of a sphere or hemisphere [5]. For $d = 0$, Π passes through $\mathbf{O}$ and $\mathcal{C}$ is a *sphero great-circle*, arising from a spherically-normalised cylinder silhouette line. The spherocylinder, being a cylinder capped with a hemisphere, therefore yields a concatenation of circular arcs on $\mathbb{S}^2$: two great-circle arcs supported by sphero-planes Π_+ and Π_- (one per cylinder silhouette line), and a single small-circle arc supported by sphero-plane Π (from the hemisphere).

2.2.3 Geometric Constraints

We use geometric properties to derive two sets of constraints on the points on $\mathbb{S}^2$ forming each of the spherocylinder arcs.

Independent constraints. Each arc points $\mathcal{Q}_{\mathtt{h}}$, $\mathcal{Q}_{\mathtt{c}_+}$ and $\mathcal{Q}_{\mathtt{c}_-}$, independently satisfies:

1. *Cocircularity.* The points are trivially circular, as they lie on a sphero-circle.

2. *Coplanarity.* Cocircularity implies coplanarity, which may be considered a relaxation of cocircularity in $\mathbb{R}^3$. Consequently, the points are coplanar and supported by a sphero-plane.

Joint constraints. This is our primary theoretical contribution, entirely overlooked by existing loosely-coupled methods. While an isolated hemisphere arc has 3 DoF on $\mathbb{S}^2$, its attachment to the cylinder imposes two further constraints on the hemisphere's sphero-plane $\Pi = [\underline{\mathbf{n}}^\top, -d]^\top$. They arise from two planes naturally associated with the contour generators. The cylinder generator plane Π_{cg} spans the two silhouette edges, while the hemisphere generator plane Π_{hg} contains the circular contour generator of the hemisphere. In figure 2, on the axial view, Π_{hg} materialises as a light purple disk, cut by Π_{cg} as a solid purple line. These two planes meet along a line of direction $\underline{\mathbf{m}}$, shown as the same solid purple line. This yields 2 joint constraints:

1. *Parallelism.* Since Π_{hg} is parallel to the sphero-plane Π (light purple rectangle), the direction $\underline{\mathbf{m}}$ is also parallel to Π, forcing $\underline{\mathbf{m}} \perp \underline{\mathbf{n}}$. This holds for any hemisphere centre lying along the cylinder's axis, and the direction is recoverable from the cylinder's outward-pointing normals as $\underline{\mathbf{m}} = \eta(\underline{\mathbf{m}}_+ - \underline{\mathbf{m}}_-)$.

2. *Iso-depth.* The depth of sphero-plane Π is fixed at $d_\perp = \cos(\alpha)$ by the angular spacing of the cylinder's arcs for an hemisphere centre translated along the cylinder axis to closest point to the optical centre — that is, when the hemisphere depth equals the cylinder depth. It is thus recoverable as $d_\perp = \underline{\mathbf{m}}^\top \underline{\mathbf{m}}_+$.

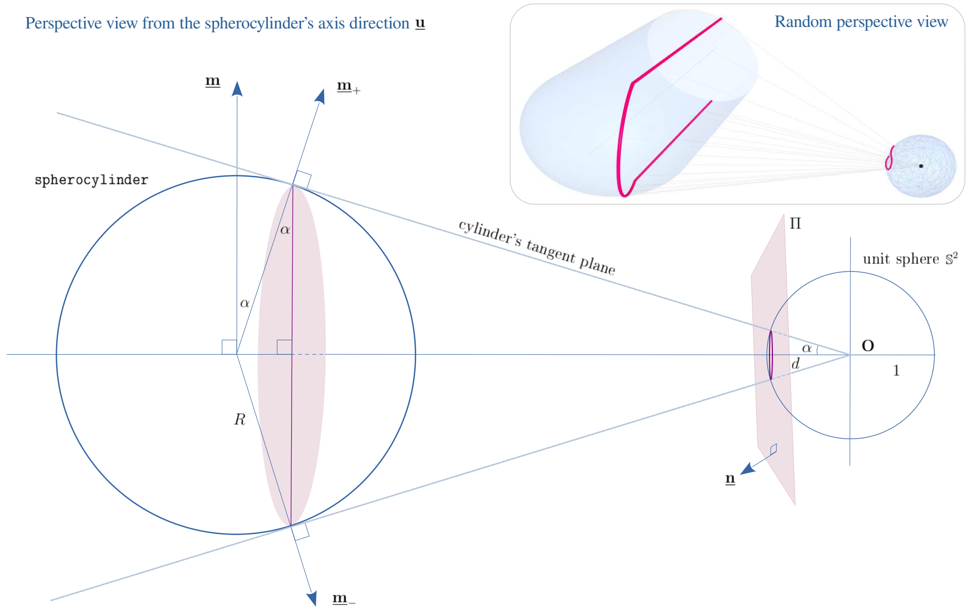


Figure 2: Spherocylinder geometry with a view along its axis $\underline{\mathbf{u}}$. The inset shows a second viewpoint, including the contour generator and its spherically-normalised contour. The cylinder's tangent planes define the axis $\underline{\mathbf{u}} = \eta(\underline{\mathbf{m}}_+ \times \underline{\mathbf{m}}_-)$, while sphero-plane Π supports the hemisphere's spherically-normalised points as a small-circle arc. Spherocylinder geometry imposes two joint geometric constraints on Π such that $\Pi \parallel \underline{\mathbf{m}}$ (*parallelism*) and depth $d_\perp = \underline{\mathbf{m}}^\top \underline{\mathbf{m}}_+$ (*iso-depth*).

2.3 Recovering Sphero-circles by Fitting Sphero-planes

2.3.1 Principle and Overview

We exploit the geometric constraints to build `ScyPoSe` and `Robust-ScyPoSe`. Both follow the same two-step pipeline: *(i)* fitting sphero-planes to the spherically-normalised silhouette arcs, then *(ii)* recovering the pose from the fitted planes. In step *(i)*, Π_+ and Π_- are first fitted to the two great-circle arcs under *coplanarity* alone (section 2.3.2), establishing the *joint constraints* on Π; Π is then fitted to the small-circle arc under both *coplanarity and joint constraints* (section 2.3.3). Step *(ii)* (section 2.4) recovers the hemisphere centre and cylinder axis from the three fitted planes, yielding the 5-DoF pose.

The two algorithms differ in their inputs and robustness. `ScyPoSe` takes silhouette pixels $\mathcal{P}$ and their labels $\mathcal{L}$, while `Robust-ScyPoSe` takes only the silhouette pixels $\mathcal{P}$ and performs point classification internally, producing the labels $\mathcal{L}$. `Robust-ScyPoSe` additionally verifies the conformity of the spherically-normalised points via the *cocircularity constraint*, increasing robustness. We first present the general and minimal sphero-plane fitting solutions and the recovery of their corresponding sphero-circles, then detail each algorithm in turn.

2.3.2 Sphero Great-Circle Recovery

We derive general m-point and minimal 2-point solvers for sphero-plane fitting, from which the great-circles can be recovered.

General $m \geqslant 2$ point sphero great-circle plane fitting. Since every great-circle plane passes through the optical centre, the plane's depth is zero. We therefore seek the linear plane $\Pi_\pm$ minimising the sum of squared orthogonal distances to the spherically-normalised cylinder points $\mathcal{Q}_{\mathtt{c}_+}$ or $\mathcal{Q}_{\mathtt{c}_-}$. Let $\mathbf{A}_{m\times 3}$ be the matrix containing these points. Its singular value decomposition ($\mathtt{svd}$) yields the least singular vector as the optimal plane normal $\underline{\mathbf{m}}_\pm \in \mathbb{S}^2$, giving $\Pi_\pm = [\underline{\mathbf{m}}_\pm^\top, 0]^\top$.

Minimal 2-point sphero great-circle plane recovery. Since every great-circle plane passes through the optical centre, only two spherically-normalised points $\underline{\mathbf{q}}_i$ and $\underline{\mathbf{q}}_{i'}$ with $l_i = l_{i'} = \mathtt{c}_+$ or $l_i = l_{i'} = \mathtt{c}_-$ suffice. The plane normal is $\underline{\mathbf{m}}_\pm = \eta(\underline{\mathbf{q}}_i \times \underline{\mathbf{q}}_{i'})$, and the great-circle plane is $\Pi_\pm = [\underline{\mathbf{m}}_\pm^\top, 0]^\top$.

Sphero great-circle recovery. The intersection $\Pi_\pm \cap \mathbb{S}^2$ defines a sphero great-circle $\mathcal{C}_\pm$ uniquely. It has centre at the origin $\mathbf{O}$, normal $\underline{\mathbf{m}}_\pm$ and unit radius $r = 1$.

2.3.3 Constrained Sphero Small-Circle Recovery

We derive general m-point, 2-point and minimal 1-point solvers for sphero-plane fitting, from which the small-circles can be recovered. The general m-point and 2-point solvers enforce only the *parallelism* constraint, whereas the minimal 1-point solver additionally enforces the *iso-depth* constraint.

General $m > 2$ point constrained sphero small-circle plane fitting. We seek the plane minimising the sum of squared orthogonal distances to the spherically-normalised hemisphere points $\mathcal{Q}_{\mathtt{h}}$, subject to the joint constraint *parallelism* $\Pi \parallel \underline{\mathbf{m}}$.

Let $\underline{\mathbf{q}}_0$ be the centroid of the spherically-normalised points and $\mathbf{A}_{m\times 3}$ the matrix of centred points $\underline{\mathbf{q}}_i - \underline{\mathbf{q}}_0$. Let $\mathbf{B}_{3\times 2} = \mathtt{null}(\underline{\mathbf{m}})$ span the 2D subspace perpendicular to $\underline{\mathbf{m}}$. We project the centred points onto this subspace $\mathbf{Q}_{m\times 2} = \mathbf{A}_{m\times 3}\mathbf{B}_{3\times 2}$. This confines the plane's tilt to a rotation about $\underline{\mathbf{m}}$. We decompose $\mathbf{Q}_{m\times 2}$ into its singular vectors using an $\mathtt{svd}$. The least singular vector gives the optimal constrained normal $\mathbf{v} \in \mathbb{S}^1$ within the 2D subspace. We map the normal $\mathbf{v}$ back to 3D $\underline{\mathbf{n}} = \eta(\mathbf{B}_{3\times 2}\mathbf{v})$ yielding the constrained 3D normal. We correct the normal's sign as $\underline{\mathbf{n}} = \mathtt{sign}(\underline{\mathbf{q}}_0^\top \underline{\mathbf{n}})\underline{\mathbf{n}}$ so that it points toward the spherically-normalised points. Finally, we compute the depth as $d = |\underline{\mathbf{q}}_0^\top \underline{\mathbf{n}}|$ and form the sphero-plane $\Pi = [\underline{\mathbf{n}}^\top, -d]^\top$.

Although d is computed, we ultimately rely only on the sphero-plane normal $\underline{\mathbf{n}}$ to recover the hemisphere centre on the cylinder axis via triangular similarity, as detailed in section 2.4.

2-point constrained sphero small-circle plane recovery. Given spherically-normalised points $\underline{\mathbf{q}}_i$ and $\underline{\mathbf{q}}_{i'}$ with $l_i = l_{i'} = \mathtt{h}$, we enforce parallelism $\Pi \parallel \underline{\mathbf{m}}$ and recover the sphero-plane normal as $\underline{\mathbf{n}} = \eta(\underline{\mathbf{m}} \times (\underline{\mathbf{q}}_i - \underline{\mathbf{q}}_{i'}))$. The plane's signed distance to origin is $d' = \underline{\mathbf{q}}_j^\top \underline{\mathbf{n}}$ where $j \in \{i, i'\}$ is any of the 2 points. Following this, we form the plane $\Pi = [\mathtt{sign}(d')\underline{\mathbf{n}}^\top, -d]^\top$ where $d = |d'|$.

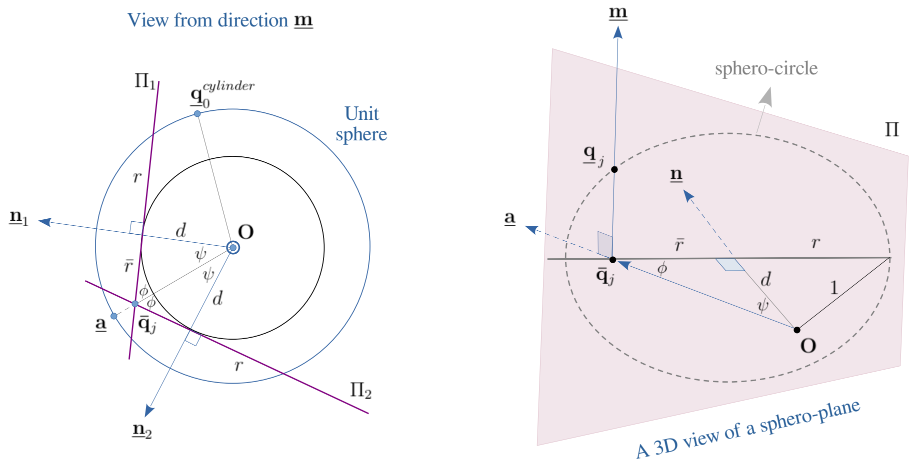


Figure 3: Minimal 1-point constrained sphero small-circle plane solutions for the hemisphere in an iso-depth configuration (*i.e.*, hemisphere and cylinder depths coincide).

Minimal 1-point constrained sphero small-circle plane recovery. The minimal 1-point small-circle solver estimates the hemisphere's sphero-plane from a single normalised silhouette point via a two-step pipeline. The first step, *iso-depth alignment*, solves an idealised coincident-depth configuration: by assuming equal hemisphere and cylinder depths, the solver enforces parallelism and iso-depth joint constraints to produce a fictive spherocylinder pose whose hemisphere sphero-plane already encodes the cylinder's true depth. The second step, *pose resolution*, lifts this idealisation to the general case where depths differ: triangular similarity yields the spatial displacement between the fictive and true hemisphere centres, correcting the sphero-plane to recover the true pose.

Iso-depth alignment: There exist two sphero-plane solutions on the camera-centred unit sphere (figure 3). We recover both solutions then select the correct one as follows. We project the hemisphere point $\underline{\mathbf{q}}_j$ onto the bisection plane $[\underline{\mathbf{m}}^\top, 0]^\top$ of the cylinder's tangent planes using $\bar{\mathbf{q}}_j = \underline{\mathbf{q}}_j - (\underline{\mathbf{q}}_j^\top \underline{\mathbf{m}})\,\underline{\mathbf{m}}$. We normalise the projected point to obtain a unit direction vector $\underline{\mathbf{a}} = \eta(\bar{\mathbf{q}}_j)$ with $\underline{\mathbf{a}} \perp \underline{\mathbf{m}}$. Since $\underline{\mathbf{n}} \perp \underline{\mathbf{m}}$, we can align $\underline{\mathbf{a}}$ onto a solution normal $\underline{\mathbf{n}}$ by a $\mp\psi$ rotation about $\underline{\mathbf{m}}$:

$$\underline{\mathbf{n}}_{1,2} = \mathtt{exp}(\mp\psi\,[\underline{\mathbf{m}}]_\times)\,\underline{\mathbf{a}} \tag{2}$$

where $[\underline{\mathbf{m}}]_\times$ is the skew-symmetric matrix of $\underline{\mathbf{m}}$ and $\psi = \mathtt{acos}(d/\|\bar{\mathbf{q}}_j\|)$. Rotation about $\underline{\mathbf{m}}$ enforces parallelism $\Pi \parallel \underline{\mathbf{m}}$, while the angle ψ enforces iso-depth with $d = d_\perp = \underline{\mathbf{m}}^\top \underline{\mathbf{m}}_+$. The correct solution is selected as the normal most aligned with the centroid $\underline{\mathbf{q}}_0^{cylinder}$ of the cylinder's spherically-normalised silhouette points:

$$\underline{\mathbf{n}} = \underset{\underline{\mathbf{n}} \in \{\underline{\mathbf{n}}_1, \underline{\mathbf{n}}_2\}}{\mathtt{arg\,max}}(\underline{\mathbf{n}}^\top \underline{\mathbf{q}}_0^{cylinder}). \tag{3}$$

Here $\underline{\mathbf{q}}_0^{cylinder}$ indicates the surface of the finite physical body enclosed within the infinite virtual cylinder. Once the correct solution is selected, we form the plane $\Pi = [\underline{\mathbf{n}}^\top, -d]^\top$.

Pose resolution: The iso-deph alignment produces a translated hemisphere from the true one, as though the hemisphere were in a fictive *iso-depth* configuration — that is, on a ro-

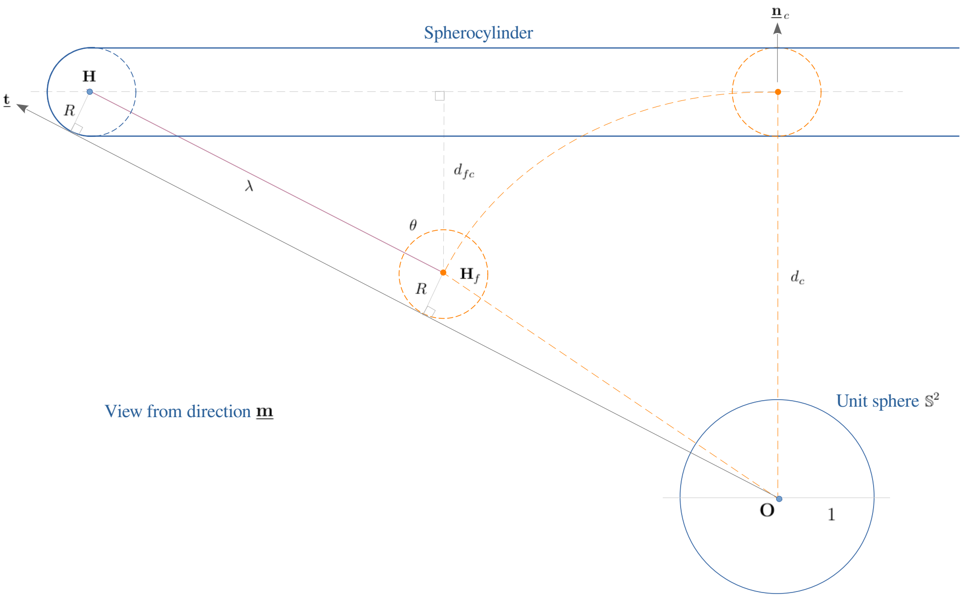


Figure 4: Minimal 1-point solution for the hemisphere in a general case (*i.e.*, hemisphere and cylinder depths differ) when silhouette point ray $\underline{\mathbf{q}}_j$ lies on the bisection plane ($\underline{\mathbf{q}}_j = \underline{\mathbf{t}}$).

tated version of the true cylinder about $\underline{\mathbf{m}}$ — while remaining tangent to the point ray $\underline{\mathbf{q}}_j$ and having its centre on the bisection plane $[\underline{\mathbf{m}}^\top, 0]^\top$. Let this translation be $\lambda\underline{\mathbf{t}}$. We generate two tangent directions to the fictive sphere on the bisection plane $\underline{\mathbf{t}}_{1,2} = \mathtt{exp}(\mp\alpha[\underline{\mathbf{m}}]_\times)\underline{\mathbf{n}}$ where $\alpha = \mathtt{acos}(d_\perp)$ is the half opening angle of the cylinder's tangent planes, and select the direction most aligned with ray $\underline{\mathbf{q}}_j$ using $\underline{\mathbf{t}} = \mathtt{argmax}(\underline{\mathbf{t}}^\top\underline{\mathbf{q}}_j)$ where $\underline{\mathbf{t}} \in \{\underline{\mathbf{t}}_1, \underline{\mathbf{t}}_2\}$. The direction $\underline{\mathbf{t}}$ corresponds to the displacement direction when ray $\underline{\mathbf{q}}_j$ lies on the bisection plane (figure 4). When the hemisphere silhouette is fully visible, exactly one such ray exists — giving the solution. The offset λ is then recovered by triangular similarity. The fictive hemisphere centre is $\mathbf{H}_f = (R/r)\underline{\mathbf{n}}$ where $r = \sqrt{1-d^2}$. Since this fictive *iso-depth* configuration corresponds to a rotated version of the true cylinder, the true cylinder shares the same depth $d_c = R/r$ but along direction $\underline{\mathbf{n}}_c = -\eta(\underline{\mathbf{m}}_+ + \underline{\mathbf{m}}_-)$. The distance from the fictive hemisphere centre to the cylinder axis is then $d_{fc} = d_c - \mathbf{H}_f^\top\underline{\mathbf{n}}_c$. Considering the right triangle formed by $\mathbf{H}_f$, the true hemisphere centre $\mathbf{H}$, and the closest point on the cylinder axis to $\mathbf{H}_f$, the angle θ at $\mathbf{H}_f$ satisfies both $\mathtt{cos}(\theta) = \underline{\mathbf{t}}^\top\underline{\mathbf{n}}_c$ and $\mathtt{cos}(\theta) = d_{fc}/\lambda$, yielding $\lambda = d_{fc}/\mathtt{cos}(\theta)$. The true hemisphere centre is thus $\mathbf{H} = \mathbf{H}_f + \lambda\underline{\mathbf{t}}$. Finally, the sphero-plane's normal is corrected as $\underline{\mathbf{n}} = \eta(\mathbf{H})$ and its depth updated as $d = \underline{\mathbf{q}}_j^\top\underline{\mathbf{n}}$, yielding the corrected sphero-plane $\Pi = [\underline{\mathbf{n}}^\top, -d]^\top$.

Small-circle recovery. The intersection $\Pi \cap \mathbb{S}^2$ uniquely defines the sphero small-circle $\mathcal{C}$. It follows from the geometry in figure 3 that $\mathcal{C}$ has normal $\underline{\mathbf{n}}$, centre $d\underline{\mathbf{n}}$ and radius $r = \sqrt{1-d^2}$.

2.4 Pose Estimation from Classified Silhouette Points

`ScyPoSe` estimates the 5-DoF spherocylinder pose from a minimum of 5 points.

Algorithm. Algorithm 1 gives the full procedure. The inputs are the spherocylinder's sil-

Algorithm 1 `ScyPoSe`

Inputs: Spherocylinder's silhouette pixels $\mathcal{P}$, labels $\mathcal{L}$, radius R, and camera intrinsics $\mathbf{K}$

Outputs: Spherocylinder pose $\xi = [\mathbf{H}^\top, \underline{\mathbf{u}}^\top]^\top \in \mathbb{R}^3 \times \mathbb{S}^2$

1: $\Pi_+ = \texttt{Sphero-gc-plane}(\mathcal{P}, \mathcal{L}, \mathbf{K})$ // first great-circle plane
2: $\Pi_- = \texttt{Sphero-gc-plane}(\mathcal{P}, \mathcal{L}, \mathbf{K})$ // second great-circle plane
3: $\Pi = \texttt{Sphero-sc-plane}(\mathcal{P}, \mathcal{L}, \mathbf{K}, \Pi_+, \Pi_-)$ // hemisphere small-circle plane
4: $\mathbf{H} = \texttt{Hemisphere Centre}(\Pi, \Pi_+, \Pi_-, R)$
5: $\underline{\mathbf{u}} = \texttt{Cylinder Axis}(\Pi_+, \Pi_-)$

houette points $\mathcal{P}$ and their labels $\mathcal{L}$, the spherocylinder's prescribed radius $R \in \mathbb{R}^+$ in metric units, and the camera intrinsic matrix $\mathbf{K}$. The output is the spherocylinder's 5-DoF pose $\xi \in \mathbb{R}^3 \times \mathbb{S}^2$. Lines 1–2 fit the sphero great-circle planes Π_+ and Π_- to the spherically-normalised cylinder silhouette points, respectively. Line 3 fits the constrained sphero small-circle plane Π from the hemisphere's spherically-normalised silhouette points under the joint constraints. Line 4 recovers the hemisphere centre $\mathbf{H}$ on the cylinder axis via triangular similarity. First, the cylinder depth is computed as $d_c = R/\texttt{sin}(\alpha)$, where $\alpha = \texttt{acos}(\underline{\mathbf{m}}^\top \underline{\mathbf{m}}_+)$. The hemisphere depth then follows as $d_h = d_c/(\underline{\mathbf{n}}^\top \underline{\mathbf{n}}_c)$, where $\underline{\mathbf{n}}_c = -\eta(\underline{\mathbf{m}}_+ + \underline{\mathbf{m}}_-)$ is the unit direction toward the closest point on the cylinder axis. Finally, the hemisphere centre is obtained as $\mathbf{H} = d_h \underline{\mathbf{n}}$. Line 5 recovers the cylinder axis $\underline{\mathbf{u}} = \eta(\underline{\mathbf{m}}_+ \times \underline{\mathbf{m}}_-)$.

2.5 Robust Pose Estimation from Unclassified Silhouette Points

`Robust-ScyPoSe` is a sampling-based method that jointly estimates the pose, classifies unlabelled silhouette points, and rejects erroneous points, exploiting all independent (coplanarity, cocircularity) and joint constraints. It follows the same steps as `ScyPoSe`, but after recovering each candidate sphero-plane via a minimal solution, it additionally scores the candidate sphero-plane by counting inliers on the corresponding sphero-circle via the cocircularity constraint.

Point conformity to a sphero-circle. Let $\underline{\mathbf{a}}$ represent a spherically-normalised point as in figure 3. We project it along its normalisation ray onto the candidate sphero-circle plane Π, yielding $\bar{\mathbf{q}}_j$. We then compute its distance to the sphero-circle centre on the plane via triangle similarity:

$$\bar{r} = \frac{d}{\underline{\mathbf{a}}^\top \underline{\mathbf{n}}} \sqrt{1 - (\underline{\mathbf{a}}^\top \underline{\mathbf{n}})^2} \tag{4}$$

where $\underline{\mathbf{n}}$ is the sphero-plane normal and d its offset. The point conforms to the sphero-circle if $|r - \bar{r}| < \tau$, where $r = \sqrt{1-d^2}$ is the sphero-circle radius and τ is a small threshold.

Algorithm. Algorithm 2 gives the full procedure. The inputs are the spherocylinder's unclassified silhouette pixels $\mathcal{P}$, the spherocylinder's prescribed radius $R \in \mathbb{R}^+$ in metric units,

Algorithm 2 `Robust-ScyPoSe`

Inputs: Spherocylinder's unclassified silhouette pixels $\mathcal{P}$ and radius R, camera intrinsics $\mathbf{K}$ and RANSAC threshold $\tau_{\texttt{pix}}$ (*e.g.*, 1 pixel).
Outputs: Spherocylinder pose $\xi = [\mathbf{H}^\top, \underline{\mathbf{u}}^\top]^\top \in \mathbb{R}^3 \times \mathbb{S}^2$ and pixel labels $\mathcal{L}$.
1: $\mathcal{Q} = \texttt{Spherical Normalisation}(\mathcal{P}, \mathbf{K})$ // equation (1)
2: $\{\Pi_1, \mathcal{X}_1\} = \texttt{RANSAC Sphero-gc-plane}(\mathcal{Q}, \tau_{\texttt{gc}})$ // 2-point great-circle solver
3: $\mathcal{Q} = \texttt{Remove Points}(\mathcal{Q}, \mathcal{X}_1)$
4: $\{\Pi_2, \mathcal{X}_2\} = \texttt{RANSAC Sphero-gc-plane}(\mathcal{Q}, \tau_{\texttt{gc}})$ // 2-point great-circle solver
5: $\mathcal{Q} = \texttt{Remove Points}(\mathcal{Q}, \mathcal{X}_2)$
6: $\{\Pi_+, \Pi_-, \mathcal{Q}_{\texttt{c}_+}, \mathcal{Q}_{\texttt{c}_-}\} = \texttt{Orient Normals Outward from Cylinder}(\Pi_1, \Pi_2, \mathcal{X}_1, \mathcal{X}_2)$
7: $\mathcal{Q} = \texttt{Remove Points Out Of Cylinder}(\mathcal{Q}, \Pi_+, \Pi_-)$
8: $\{\Pi, \mathcal{Q}_{\texttt{h}}\} = \texttt{RANSAC Sphero-sc-plane}(\mathcal{Q}, \tau_{\texttt{sc}}, \Pi_+, \Pi_-)$ // k-point small-circle solver
9: $\mathbf{H} = \texttt{Hemisphere Centre}(\Pi, \Pi_+, \Pi_-, R)$
10: $\underline{\mathbf{u}} = \texttt{Cylinder Axis}(\Pi_+, \Pi_-)$
11: $\mathcal{L} = \texttt{Reconstruct Labels}(\mathcal{P}, \mathbf{K}, \mathcal{Q}_{\texttt{c}_+}, \mathcal{Q}_{\texttt{c}_-}, \mathcal{Q}_{\texttt{h}})$

the camera intrinsic matrix $\mathbf{K}$ and the RANSAC threshold $\tau_{\texttt{pix}}$. The outputs are the spherocylinder's 5-DoF pose $\xi \in \mathbb{R}^3 \times \mathbb{S}^2$ and point labels $\mathcal{L}$. Line 1 spherically normalises the full silhouette $\mathcal{P}$ to $\mathcal{Q} \subset \mathbb{S}^2$. Line 2 fits a great-circle plane to the normalised points via RANSAC, iterating over random 2-point samples and applying the 2-point minimal solver at each iteration. Consensus sets are formed by testing each point's conformity to the candidate sphero great-circle against a scaled threshold $\tau_{\texttt{gc}} = d_{\texttt{gc}}(\tau_{\texttt{pix}}/f_{\texttt{max}})$, where $f_{\texttt{max}} = \texttt{max}(f_x, f_y)$ is the camera's maximum focal length in pixels and $d_{\texttt{gc}} = 0.001$ is a small value that replaces the ideal great-circle plane distance of zero to avoid zero threshold. The algorithm terminates with an m-point great-circle plane fit on the largest consensus set $\mathcal{X}_1$. This yields Π_1, and $\mathcal{X}_1$ is classified as the inliers of the first cylinder edge. Line 3 removes the first edge's inliers $\mathcal{X}_1$ from the points $\mathcal{Q}$. Line 4 repeats Line 2 on the remaining points to fit the second great-circle plane Π_2 and get the inliers $\mathcal{X}_2$ of the second cylinder edge. Line 5 removes the second edge's inliers $\mathcal{X}_2$ from the remaining points $\mathcal{Q}$. Line 6 orients the two great-circle normals outward from the cylinder silhouette, giving $\Pi_\pm$ and $\mathcal{Q}_{\texttt{c}_\pm}$. Line 7 discards points lying outside the cylinder silhouette. Line 8 fits a small-circle plane to the remaining points via RANSAC, iterating over random k-point samples and applying the k-point minimal solver, where $k \in \{1, 2\}$. Consensus sets are formed by testing each point's conformity to the candidate sphero small-circle against a scaled threshold $\tau_{\texttt{sc}} = d_{\texttt{sc}}(\tau_{\texttt{pix}}/f_{\texttt{max}})$ where $d_{\texttt{sc}} = \underline{\mathbf{m}}^\top \underline{\mathbf{m}}_+$. The algorithm terminates with an m-point constrained small-circle plane fit on the largest consensus set. This yields Π, and its inliers give the hemisphere contour $\mathcal{Q}_{\texttt{h}}$. Lines 9–10 recover the pose as in algorithm 1. Finally, line 11 reconstructs the pixel labels $\mathcal{L}$ from the classified inlier sets.

Remark. Noise may cause $\|\bar{\mathbf{q}}_j\| < d$, making $\psi = \texttt{acos}(d/\|\bar{\mathbf{q}}_j\|)$ imaginary. This happens when $\|\bar{\mathbf{q}}_j\| \approx d$ near the hemisphere–cylinder junction. Within `RANSAC Sphero-sc-plane` loop, we handle this by defining $\psi = \texttt{acos}(\texttt{min}(d/\|\bar{\mathbf{q}}_j\|, 1))$.

3 Experiments and Results

We evaluated variants of `Robust-ScyPoSe` against `Robust-SpherO` [5] on synthetic data and against `LUP` [3] on real data. All variants use two 2-point great-circle solvers and one k-point small-circle solver ($k \in \{1,2\}$). The great-circle planes are recovered via two successive 2-point RANSACs; the variants differ only in how the small-circle plane is recovered: `ScyPoSe5pi` exhaustively tests each remaining point with the 1-point solver; `ScyPoSe5p` runs a single 1-point RANSAC; `ScyPoSe6p` runs a single 2-point RANSAC.

3.1 Evaluation Metrics

We evaluated using absolute and relative error metrics. An absolute error measures the deviation of the estimated pose from ground truth for a single image. The position error e_H is the Euclidean distance (*mm*) between estimated $\mathbf{H}$ and ground-truth $\mathbf{H}^*$ hemisphere centres, while the orientation error e_u is the angle (*degrees*) between the estimated $\underline{\mathbf{u}}$ and ground-truth $\underline{\mathbf{u}}^*$ cylinder axis directions. A relative error measures how consistently a method estimates pose changes between two images, independently of any absolute offset and thus invariant to global pose bias. The position consistency error e_H^{rel} is the Euclidean distance between the estimated and ground-truth hemisphere centre displacements (*mm*), while the orientation consistency error e_u^{rel} is the absolute difference between the estimated and ground-truth angular displacements of the cylinder axis direction (*degrees*).

3.2 Experiments on Synthetic Data

We evaluated `ScyPoSe5p` through controlled stress tests on synthetic data against noise, outliers, and occlusion (see figure 5, the first 3 graphs), and compare against `Robust-SpherO`, which estimates the hemisphere centre independently of the cylinder, isolating the benefit of our joint hemisphere-cylinder constraint. Each trial samples $1{,}000$ random spherocylinder poses with radius $R = 5$ mm at depths between $10R$ and $30R$, typical of a laparoscopic ultrasound probe. The silhouette consists of 100 points on each line and 50 points on the hemisphere, reduced accordingly in the occlusion tests (e.g., 10% occlusion on the hemisphere and 10% on each line). `Robust-SpherO`, the state-of-the-art, falls outside the graphs' limits: it performs poorly on the hemisphere and degrades progressively with increasing depth, consistent with the results reported in [5].

We further evaluate the robustness of `ScyPoSe5p` near the axial-alignment degeneracy, varying the cylinder axis deviation from $10°$ down to $6°$ from the optical axis while keeping the hemisphere centre on the optical axis (see figure 5, the last graph). For a spherocylinder of radius R and length L, the optical axis lies entirely inside the spherocylinder below a deviation of $\arctan(R/L)$ (about $5°$ for $R = 5$ mm, $L = 55$ mm), marking the onset of the degeneracy.

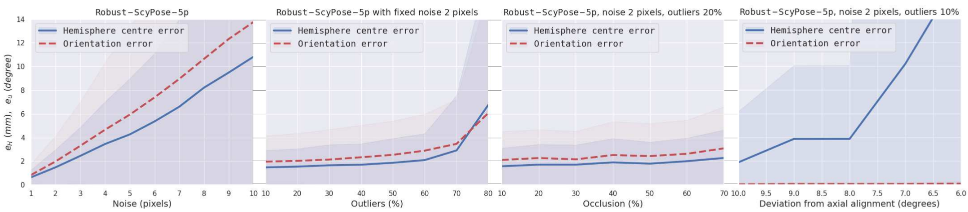


Figure 5: Robustness of `ScyPoSe5p` to increasing noise (pixels), outlier ratio (%), occlusion (%), and to proximity to the axial-alignment degeneracy (°).

3.3 Experiments on Real Data

We evaluated `ScyPoSe5pi`, `ScyPoSe5p`, and `ScyPoSe6p` against `LUP` [3] on laparoscopic ultrasound probe and human finger pose estimation. Each experiment is repeated 100 times per image and method, and we report the mean and standard deviation of the results.

3.3.1 Laparoscopic Ultrasound Probe Pose Estimation

Dataset. We used the public dataset [7], which provides ground-truth probe poses obtained via marker-based tracking. It comprises images from four surgical sequences (S1–S4), from which we selected 30 frames in which the probe hemisphere is fully visible. The probe has a radius of $5\,mm$ and, in a typical laparoscopic procedure, is positioned 10–$15\,cm$ from the camera. Since the marker largely occludes the probe head, we segmented the probe silhouette contours manually. Manual segmentation was used to cope with the presence of the markers; however, the `ScyPoSe` framework is fully compatible with existing neural networks for markerless, fully automatic end-to-end deployment, as demonstrated in qualitative results (figure 7). Figure 6 shows example frames from each sequence with the estimated pose overlaid. Figure 7 illustrates a markerless case: the probe silhouette is first segmented using [11] (top-right image), then passed to `Robust-ScyPoSe` for pose recovery. The estimated pose is overlaid on the image (bottom-left image), with classified pixels highlighted in colour (bottom-right image).

Results. In terms of absolute errors (Table 1), `ScyPoSe6p` achieves the lowest overall centre position error (27.16 ± 4 mm) and axis orientation error ($8.66 \pm 2.3°$), outperforming `LUP`, which exhibits high variance (e.g., 55.48 ± 226 mm on S3). Across sequences, S1 is the easiest for all methods (centre position ≈ 6 mm, axis orientation $\approx 5°$), while S2 and S3 are more challenging for position estimation, likely due to the probe tip pointing away from the camera. `ScyPoSe5p` and `ScyPoSe5pi` perform comparably throughout.

Table 1: Averaged absolute errors per surgical sequence and overall. Bold: best.

	S1 (8 images)		S2 (5 images)		S3 (9 images)		S4 (8 images)		All (30 images)	
Method	e_H *(mm)*	e_u *(deg)*	e_H *(mm)*	e_u *(deg)*	e_H *(mm)*	e_u *(deg)*	e_H *(mm)*	e_u *(deg)*	e_H *(mm)*	e_u *(deg)*
`ScyPoSe5pi`	6.19 ± 0.6	5.09 ± 2.3	51.84 ± 8.5	6.46 ± 1.8	47.76 ± 8.45	16.71 ± 2.8	15.87 ± 11.4	4.54 ± 2.0	28.85 ± 7.1	8.66 ± 2.3
`ScyPoSe5p`	6.21 ± 0.6	$\mathbf{5.00 \pm 2.3}$	51.59 ± 8.3	6.47 ± 1.8	47.39 ± 8.55	$\mathbf{16.68 \pm 2.7}$	16.73 ± 12.1	4.58 ± 2.0	28.93 ± 7.3	$\mathbf{8.64 \pm 2.3}$
`ScyPoSe6p`	$\mathbf{5.87 \pm 0.9}$	5.11 ± 2.3	51.71 ± 8.1	$\mathbf{6.44 \pm 1.8}$	$\mathbf{47.05 \pm 5.91}$	16.71 ± 2.7	$\mathbf{10.71 \pm 2.3}$	$\mathbf{4.53 \pm 2.0}$	$\mathbf{27.16 \pm 4.0}$	8.66 ± 2.3
`LUP`	6.08 ± 0.9	5.43 ± 2.3	$\mathbf{50.27 \pm 8.5}$	6.49 ± 2.1	55.48 ± 226	17.00 ± 4.8	13.98 ± 8.4	4.91 ± 2.5	30.37 ± 71	8.94 ± 3.1

Regarding relative errors (Table 2), `ScyPoSe6p` again achieves the lowest overall position consistency error (4.73 ± 3.8 mm) and orientation consistency error ($3.0 \pm 1.8°$), significantly lower than `LUP` (13.85 ± 106 mm and $4.0 \pm 2.3°$). All `Robust-ScyPoSe` variants outperform `LUP`.

Table 2: Averaged relative errors per surgical sequence and overall. Bold: best.

	S1 (8 images)		S2 (5 images)		S3 (9 images)		S4 (8 images)		All (30 images)	
Method	e_H^{rel} *(mm)*	e_u^{rel} *(deg)*	e_H^{rel} *(mm)*	e_u^{rel} *(deg)*	e_H^{rel} *(mm)*	e_u^{rel} *(deg)*	e_H^{rel} *(mm)*	e_u^{rel} *(deg)*	e_H^{rel} *(mm)*	e_u^{rel} *(deg)*
`ScyPoSe5pi`	$\mathbf{0.90 \pm 0.5}$	$\mathbf{3.8 \pm 1.6}$	11.76 ± 8.1	3.1 ± 1.2	9.33 ± 8.6	3.2 ± 2.4	1.96 ± 3.4	2.1 ± 1.6	5.52 ± 5.0	3.1 ± 1.8
`ScyPoSe5p`	0.95 ± 0.5	3.8 ± 1.7	11.43 ± 7.6	3.1 ± 1.2	9.09 ± 8.8	$\mathbf{3.0 \pm 2.3}$	2.35 ± 4.0	$\mathbf{2.1 \pm 1.6}$	5.51 ± 5.1	3.0 ± 1.8
`ScyPoSe6p`	1.40 ± 0.9	3.8 ± 1.8	$\mathbf{10.59 \pm 7.3}$	$\mathbf{3.1 \pm 1.2}$	$\mathbf{7.23 \pm 5.1}$	3.1 ± 2.3	$\mathbf{1.61 \pm 3.0}$	2.2 ± 1.6	$\mathbf{4.73 \pm 3.8}$	$\mathbf{3.0 \pm 1.8}$
`LUP`	1.43 ± 0.8	3.9 ± 2.0	12.35 ± 9.2	3.3 ± 1.6	31.02 ± 337	6.1 ± 3.1	7.88 ± 10.7	2.4 ± 2.2	13.85 ± 106	4.0 ± 2.3

Runtimes. All methods are implemented in GNU Octave and evaluated on the same machine, with an average of $1,500$ silhouette points per image across the four surgical sequences. `ScyPoSe5p` is the fastest (63 ± 41 *ms*), followed by `ScyPoSe6p` (145 ± 97 *ms*). `LUP` is the slowest by a large margin (7232 ± 1726 *ms*), as its runtime scales poorly with the number of silhouette points.

Conclusion. Overall, all `Robust-ScyPoSe` variants consistently outperform `LUP` across both absolute and relative error metrics as well as inference speed, with `ScyPoSe6p` emerging as the most robust.

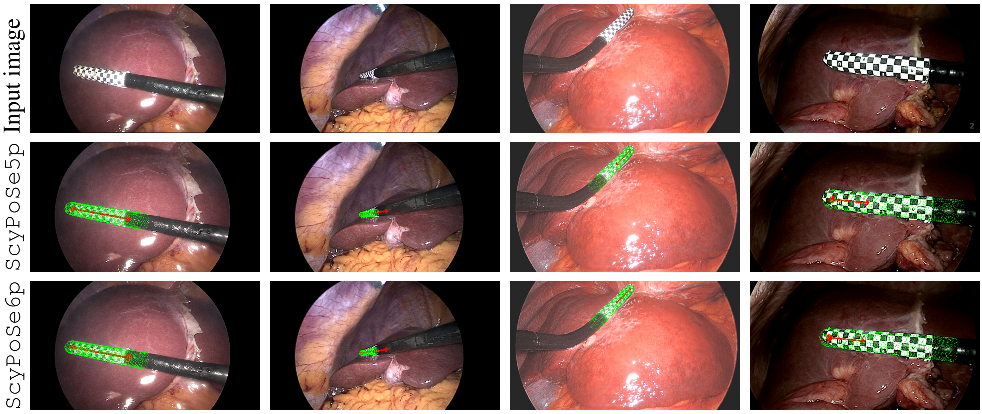


Figure 6: Forward projections of the ultrasound probe's head model onto example images from surgical sequences S1–S4. The first row shows the input images; the second and third rows show the model overlaid using poses estimated by `ScyPoSe5p` and `ScyPoSe6p`, respectively.

3.3.2 Human Finger Pose Estimation

Dataset. We captured a single RGB-D image of a hand using an Intel RealSense camera at approximately $30\,cm$ — the distance at which the sensor delivers peak depth accuracy. We derived ground-truth finger poses and radii from the depth channel. From the RGB image, we segmented the hand using [8], then isolated each finger's silhouette contour manually and passed it as input to the methods.

Results. Table 3 reports absolute errors for all five fingers. `ScyPoSe6p` achieves the lowest overall estimation errors. It yields $4.46 \pm 1.09\,mm$ position and $3.9 \pm 2.1°$ orientation error. `ScyPoSe5p` and `ScyPoSe5pi` rank second in performance. Figure 8 shows spherocylinder models overlaid on the hand fingers using the estimated poses.

Table 3: Averaged absolute errors per finger and overall. Bold: best.

	Thumb		Index		Middle		Ring		Pinky		All (5 fingers)	
Method	e_H (mm)	e_u (deg)	e_H (mm)	e_u (deg)	e_H (mm)	e_u (deg)	e_H (mm)	e_u (deg)	e_H (mm)	e_u (deg)	e_H (mm)	e_u (deg)
`ScyPoSe5pi`	5.40 ± 0.29	5.8 ± 1.9	9.09 ± 0.25	8.0 ± 2.7	2.77 ± 1.45	1.8 ± 0.5	4.55 ± 2.20	$\mathbf{4.0 \pm 3.2}$	6.82 ± 2.55	4.6 ± 2.8	5.72 ± 1.35	4.8 ± 2.2
`ScyPoSe5p`	4.22 ± 0.32	4.9 ± 2.7	8.95 ± 0.33	7.5 ± 3.9	4.18 ± 3.51	7.3 ± 4.0	4.40 ± 2.27	4.5 ± 3.7	4.28 ± 1.26	3.4 ± 1.1	5.21 ± 1.54	5.5 ± 3.1
`ScyPoSe6p`	$\mathbf{3.57 \pm 0.35}$	$\mathbf{4.3 \pm 2.4}$	$\mathbf{7.02 \pm 0.25}$	$\mathbf{6.0 \pm 3.0}$	$\mathbf{2.54 \pm 0.12}$	$\mathbf{1.3 \pm 0.3}$	$\mathbf{3.61 \pm 1.18}$	5.1 ± 3.0	5.57 ± 3.55	$\mathbf{3.0 \pm 2.1}$	$\mathbf{4.46 \pm 1.09}$	$\mathbf{3.9 \pm 2.1}$
`LUP`	5.05 ± 0.33	5.7 ± 2.0	8.83 ± 0.21	7.0 ± 2.1	5.24 ± 0.63	1.7 ± 0.3	3.96 ± 2.16	5.6 ± 3.6	$\mathbf{3.86 \pm 2.72}$	4.4 ± 2.5	5.39 ± 1.21	4.9 ± 2.1

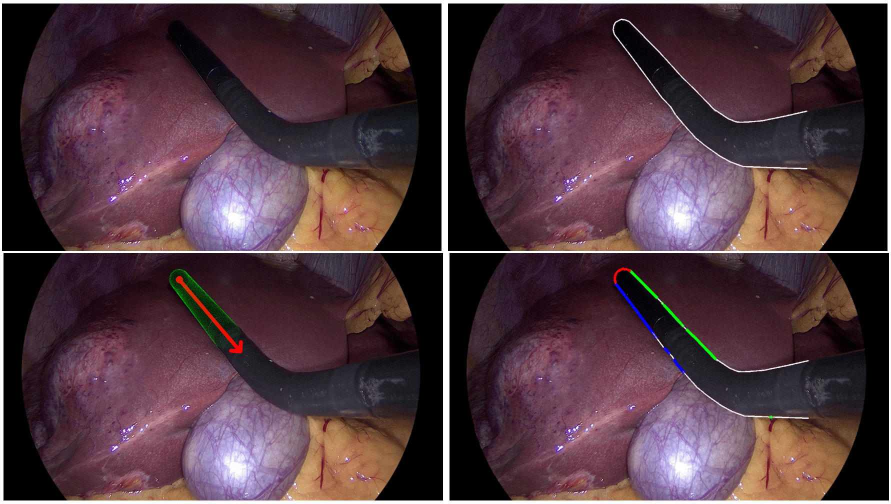
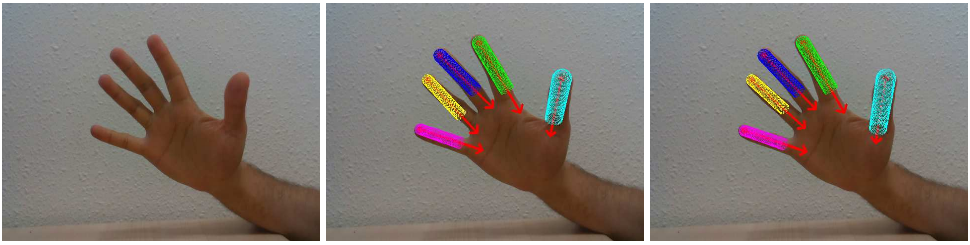


Figure 7: Qualitative result on a markerless ultrasound probe. First row: input image and the extracted contour. Second row: overlaid spherocylinder from the estimated pose and the classified pixels (red: hemisphere; green and blue: cylinder edges; white: outliers).



Input image | `ScyPose5p` | `ScyPose6p`

Figure 8: Projected spherocylinder models onto the fingers using the estimated poses.

Conclusion. `Robust-ScyPoSe` variants consistently outperform `LUP` on hand fingers pose estimation.

4 Conclusion

We have proposed `ScyPoSe`, a method to estimate spherocylinder pose from a single calibrated view of its silhouette. The method is accurate, computationally efficient, and straightforward to implement. We have also proposed its robustified extension `Robust-ScyPoSe` that works with unclassified silhouette points and rejects outliers.

Future work will address two degenerate cases: *(i)* affine views, where perspective cues vanish and depth become ambiguous, and *(ii)* a spherocylinder axis collinear with the optical axis, where the cylinder's silhouette lines become nearly unobservable.

Acknowledgement

This work is funded by the projects ANR JCJC - IMMORTALLS and Interreg Sudoe Programme (European Regional Development Fund) - REMAIN (S1/1.1/EO111).

References

[1] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision (IJCV)*, 9(2):83–112, 1992.

[2] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least squares fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):476–480, 1999.

[3] M. M. Kalantari, E. Ozgur, M. Alkhatib, E. Buc, B. Le Roy, R. Modrzejewski, Y. Mezouar, and A. Bartoli. Markerless ultrasound probe pose estimation in mini-invasive surgery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[4] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13(3):321–330, 1984.

[5] E. Ozgur, M. Alkhatib, Y. Mezouar, and A. Bartoli. Reconstructing a sphere and the camera focal length from a single view by fitting planes. *International Journal of Computer Vision (IJCV), special issue: selected papers from BMVC 2024*, 134 (151), 2026.

[6] L. Quan. Conic reconstruction and correspondence from two views. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(2):151–160, 1996.

[7] N. Rabbani, L. Calvet, Y. Espinel, B. Le Roy, M. Ribeiro, E. Buc, and A. Bartoli. A methodology and clinical dataset with ground-truth to evaluate registration accuracy quantitatively in computer-assisted laparoscopic liver resection. *Computer Methods in Biomechanics and Biomedical Engineering (CMBBE): Imaging & Visualization*, 10(4): 441–450, 2022.

[8] L. Ruzicka, B. Kohn, and C. Heitzinger. Tipsegnet: Fingertip segmentation in contactless fingerprint imaging. *Sensors*, 25(6):1824, 2025.

[9] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(11): 1115–1138, 1991.

[10] T. Toth and L. Hajder. A minimal solution for image-based sphere estimation. *International Journal of Computer Vision (IJCV)*, 131 (6):1428–1447, 2023.

[11] Z. Zhang and K. Zhang. Farsee-net: Real-time semantic segmentation by efficient multi-scale context aggregation and feature space super-resolution. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8411–8417, 2020.