

Calibration by correlation using metric embedding from non-metric similarities

Andrea Censi, *Student Member, IEEE*, Davide Scaramuzza, *Member, IEEE*

Abstract—This paper presents a new intrinsic calibration method that allows us to calibrate a generic single-view point camera just by waving it around. From the video sequence obtained while the camera undergoes random motion, we compute the pairwise time correlation of the luminance signal for a subset of the pixels. We show that, if the camera undergoes a random uniform motion, then the pairwise correlation of any pixels pair is a function of the distance between the pixel directions on the visual sphere. This leads to formalizing calibration as a problem of *metric* embedding from *non-metric* measurements: we want to find the disposition of pixels on the visual sphere, from similarities that are an *unknown* function of the distances. This problem is a generalization of multidimensional scaling (MDS) that has so far resisted a comprehensive observability analysis (can we reconstruct a metrically accurate embedding?) and a solid generic solution (how to do so?). We show that the observability depends both on the *local geometric properties* (curvature) as well as on the *global topological properties* (connectedness) of the target manifold. It follows that, in contrast to the Euclidean case, on the sphere we can recover the scale of the points distribution, therefore obtaining a metrically accurate solution from non-metric measurements. We describe an algorithm that is robust across manifolds and can recover a metrically accurate solution when the metric information is observable. We demonstrate the performance of the algorithm for several cameras (pin-hole, fish-eye, omnidirectional), and we obtain results comparable to calibration using classical methods. Additional synthetic benchmarks show that the algorithm performs as theoretically predicted for all corner cases of the observability analysis.

Index Terms—intrinsic camera calibration, metric embedding, catadioptric cameras, pin-hole cameras, fish-eye cameras



1 INTRODUCTION

In many applications, from classic photogrammetry tasks to autonomous robotics, camera calibration is a necessary preliminary step before using the camera data [1]. Calibration is necessary even for off-the-shelf cameras, as the properties of an optical system typically differ substantially from the stated manufacturer’s specifications. *Extrinsic* camera calibration is concerned with recovering the pose (position and orientation) of the camera with respect to another camera, or another reference frame of interest. *Intrinsic* camera calibration is concerned with estimating the origin and direction of the line of sight of each pixel; this information allows us to put into correspondence the image of an object with the position of the object in the world. Some scientific applications require estimating other characteristics of the optical system, such as the point-spread function. In this paper, we focus on intrinsic camera calibration for central cameras.

In a central camera, the lines of sight of every pixel intersect in a single point. Therefore, the intrinsic calibration information consists of the direction of each pixel on the visual sphere (\mathbb{S}^2). If a camera is non central, then one needs to know, for each pixel, also its spatial position (in \mathbb{R}^3) in addition to its direction (in \mathbb{S}^2). A non-central camera can be approximated as a central camera only if the displacement of each pixel’s origin is negligible with respect to the distance to the objects in the scene. This assumption is generally satisfied in applications such as robotics, but might not be satisfied for more uncommon applications and optical systems.

A general description of how the properties of lenses, mirrors, and sensors contribute to the geometry of the optical system is outside of the scope of this paper; a recent tutorial is given by Sturm *et. al.* [2].

Established techniques for intrinsic calibration: The most widely used techniques for intrinsic camera calibration (from now on, simply “calibration”) require the use of a known calibration pattern, and that the cameras optics can be well represented by a restricted family of models. Several calibration software tools are available online as open source. The Matlab Calibration Toolbox [3] works for pin-hole cameras and implements a mix of techniques appeared in the literature [4, 5]. The model used for pin-hole cameras is parametrized by the center of projection, the focal length, and radial and tangential distortion, which accounts for the possibility of the image sensor being non perpendicular to the optical axis. Other calibration toolboxes [6–13] can be used for calibrating omnidirectional catadioptric cameras, obtained by placing a mirror on top of a conventional camera, such that the optical axis coincides with the mirror’s axis, or with fish-eye cameras (dioptric). The parameters of the model are the center of projection in image coordinates and the profile of the radial distortion. These methods are relatively simple to use. In most of them, the user prints out a calibration pattern consisting of a black and white checkerboard, and collects several pictures of the pattern from different points of view. A semi-interactive procedure is used to identify the corners of the calibration pattern. Given this information, the software automatically solves for the calibration parameters. The algorithms rely on the fact that the pattern is known to lie on a plane, which allows recovering the parameters of the homography describing the world-to-image transformation, and that the nonlinear parts of the model (e.g., distortion) are simple enough that they can

- A. Censi is with the Control & Dynamical Systems department, California Institute of Technology. E-mail: andrea@cds.caltech.edu.
- D. Scaramuzza is with the AI Lab, Department of Informatics, University of Zurich. E-mail: davide.scaramuzza@ieee.org

be recovered using generic nonlinear optimization.

Innovative calibration techniques: Recently, there have been several works to improve on these techniques, to make them more flexible by enlarging the family of optics considered, or making the calibration procedure more convenient. Grossberg and Nayar [14] describe a method for calibrating an arbitrary imaging system, in which the pixels are allowed to have an arbitrary configuration on the visual sphere, that is based on an active display. Espuny and Gil [15] describe a technique that does not require a known image pattern, but is based on known sensor motion.

Calibration in robotics: The last decade has seen the introduction of autonomous robotic systems where navigation and object recognition are performed with vision sensors. In these systems, precise calibration is needed to ensure the safety of operation. Moreover, for robots, calibration is considered a “lifelong” activity that should be carried as autonomously as possible, as eventually any equipment degrades and must be re-calibrated over the course of its operating life; therefore, techniques that allow autonomous unsupervised calibration are especially valuable. Several calibration techniques have been designed to run autonomously embedded in a robotic architecture (e.g., [16, 17]); in these systems, a reduced set of camera calibration parameters (e.g., focal length) can be treated as another state variable, and then jointly estimated by a Bayesian filter together with the other states, such as the robot pose and the features in a map.

Calibration by correlation: In this paper, we will describe an approach to intrinsic camera calibration based exclusively on low-level statistics of the raw pixel streams, such as the inter-pixel correlation. To the best of our knowledge, Grossmann *et. al.* [18] were the first to propose this idea for the problem of camera calibration, albeit they were inspired by work done in developmental robotics and related fields [19–21].

The basic premise is that the statistics of the raw pixel stream contain information about the sensor geometry. Let $y_i(t)$ be the luminance perceived at the i -th pixel at time t . If we compare the sequences $\{y_i(t)\}_t$ and $\{y_j(t)\}_t$ for the i -th and j -th pixel, we expect to find that they are more similar the closer the two pixels are on the visual sphere. The geometry of the sensor can be recovered if one can find a statistics of the two sequences that is precisely a function of the pixels distances. More formally, let $s_i \in \mathbb{S}^2$ be the direction of the i -th pixel on the visual sphere, and let $d(s_i, s_j)$ be the geodesic distance on the sphere between the directions s_i and s_j . Let $\varphi : \mathbb{R}^T \times \mathbb{R}^T \rightarrow \mathbb{R}$ indicate a real-valued statistics of two sequences of length T . For example, the statistics φ can be the sample correlation, the mutual information, or any other information-theoretical divergence between two sequences, such as the “information distance” [18]. Define the *similarity* Y_{ij} between two pixels using φ :

$$Y_{ij} = \varphi(\{y_i(t)\}_t, \{y_j(t)\}_t).$$

The assumption that must be verified for the method to work, which we will call the *monotonicity condition*, is that the similarity is a function f of the pixel distance:

$$Y_{ij} = f(d(s_i, s_j)), \quad (1)$$

and that this f is monotonic, therefore, invertible.

Grossmann *et. al.* assume to know the function f , obtained with a separate calibration phase, by using a sensor with known intrinsic calibration experiencing the same scene as the camera being calibrated. Therefore, using the knowledge of f , one can recover the distances from the similarities: $d(s_i, s_j) = f^{-1}(Y_{ij})$. They describe two algorithms for recovering the pixel positions given the inter-pixel distances. The first algorithm is based on *multidimensional scaling* (which we will recall in the following sections) and solves for all pixel directions at the same time. The authors observe that this method is not robust enough for their data, and propose a robust nonlinear embedding method, inspired by Sammon [22] and Lee *et. al.* [23]. This second algorithm is iterative and places one pixel per iteration on the sphere, trying to respect all constraints with previously placed points.

Compared with traditional calibration methods, the calibration-by-correlation approach is attractive because it does not require a parametric model of the camera geometry, control of the sensor motion, or particular properties of the scene. However, the results reported by Grossmann *et. al.* do not compare favorably with traditional methods. The authors focus their quantitative analysis mainly on the accuracy of the estimation and inversion of the function f . They find that, for the information distance, f is reliably invertible only for $d(s_i, s_j) \leq 30^\circ$.¹ For large field of view, the estimated distributions appear significantly “shrunk” on the visual sphere.² Moreover, they find that, in practice, the function f is sensitive to the scene content; in their data, they find that applying the function f^{-1} estimated with a calibration rig to the data taken from a different camera leads to over-estimation of the angular distances.³

Contribution: In this paper, we start from the same premise of Grossmann *et. al.*, namely that it is possible to find a statistics of the pixel stream that depends on the pixel distance. However, rather than assuming the function f known, we formulate a joint optimization problem, in which we solve for both the directions $\{s_i\}$ and the function f . In this way, there is no need for a preliminary calibration phase with a sensor of known geometry. However, the problem becomes more challenging, requiring different analytic and computational tools.

Paper outline: Section 2 gives a formal description of the joint optimization problem. Section 3 discusses the conditions under which one can expect a monotonic relation between pixel distance and pixel statistics. We show that, if the camera undergoes uniform random motion, then necessarily all pairwise statistics between pixel values must depend on the pixel distance only. This suggests that a good way to collect data for camera calibration is to wave it around as randomly as possible, a theory we verify in practice.

Section 4 gives an observability analysis of the prob-

1. Compare Fig. 6 in [18], which shows the graph of f as a function of distance; and Fig 8ab, which shows the error for estimating f^{-1} .
2. See Section 4.4.1 and Fig. 13 in [18]. Note the shrinkage of the distribution (no quantitative measure is given in the paper).
3. See Section 4.4.2 in [18].

lem. The observability depends both on the manifold’s *local geometric properties* (curvature) as well as on *global topological properties* (connectedness). In \mathbb{R}^m , the scale is not observable, but, surprisingly, it is observable in \mathbb{S}^2 and other spaces of nonzero curvature, which makes the problem more constrained than in Euclidean space.

Section 5 discusses the performance measures that are adequate for the problem. The Procrustes error (i.e., alignment up to rotations) is an intuitive choice, but it is not admissible because it is not invariant to all symmetries of the problem. We use the *Spearman score* as an admissible and observable performance measure.

Section 6 describes our algorithm, which is an extension of the classical Shepard-Kruskal (SK) algorithm [24–27]. The major extension is an extra step necessary to recover the correct scale when it is observable; this step is critical for accurate calibration.

Section 7 discusses the experimental results for the case of camera calibration. The algorithm is evaluated for three different cameras: a pin-hole (45° FOV), a fish-eye (150° FOV), and an omnidirectional catadioptric camera ($360^\circ \times 100^\circ$ FOV). The results obtained are comparable with those obtained using conventional methods. Section 8 presents the results on a mix of real and synthetic datasets, which are meant to include all corner cases of the observability analysis. This shows that our algorithm is generic and works for other relevant cases in addition to the case of camera calibration.

Finally, Section 9 concludes the paper and discusses some possible directions for future work.

Supplemental materials: Appendix A contains the proofs that have been omitted for reasons of space. Appendix B contains the complete statistics and visualization for the benchmarks discussed. References to materials in Appendix A/B are written with the “A–” or “B–” prefix; e.g., Definition A-5. The hyperlinks link to an online copy of the documents. The attached multimedia materials include the source code to the algorithms and the intermediate processed results for the test cases discussed. Datasets and code are available at http://purl.org/censi/2012/camera_calibration.

2 PROBLEM FORMULATION

Let \mathbb{M} be a Riemannian manifold, and let d be its geodesic distance (introductory references for the differential geometry concepts used in this paper are [28–30]). We formalize the problem of metric embedding from non-metric measurements as follows.

Problem 1. Given a symmetric matrix $\mathbf{Y} \in \mathbb{R}^{n \times n}$, estimate the set of points $\mathcal{S} = \{s_i\}_{i=1}^n$ in a given manifold \mathbb{M} , such that $Y_{ij} = f(d(s_i, s_j))$ for some (unknown) monotonic function $f: [0, \infty) \rightarrow \mathbb{R}$.

Without loss of generality, we assume the similarities to be normalized so that $-1 \leq Y_{ij} \leq 1$ and $Y_{ii} = 1$. This implies $f(0) = 1$, and that f is nonincreasing. For camera calibration, the manifold \mathbb{M} will generally be the unit sphere \mathbb{S}^2 ; however, we formulate a slightly more generic problem. We will be especially interested in showing how the observability of the problem changes if \mathbb{M} is chosen to be \mathbb{S}^1 (the unit circle) or \mathbb{R}^m instead of \mathbb{S}^2 .

If the function f was known, it would be equivalent to know directly the matrix of distances. The problem of finding the positions of a set of points given their distance matrix is often called “metric embedding”. In the Euclidean case ($\mathbb{M} = \mathbb{R}^m$), the problem is classically called *Multidimensional Scaling* (MDS), and was first studied in psychometry in the 1950s. Cox and Cox [31] describe the statistical origins of the problem and give an elementary treatment, while France and Carroll [27] give an overview of the algorithmic solutions.

The scenario described in Problem 1 is sometimes called *non-metric* multidimensional scaling. The word “non-metric” is used because the metric information, contained in the distances $d(s_i, s_j)$, is lost by the application of the unknown function f . In certain applications, it is not important for the reconstructed points to be recovered accurately. For example, in psychometry, one might use these techniques essentially for visualization of high-dimensionality datasets; in that case, one only wants a *topologically correct* solution. If that is the case, one can just choose an arbitrary \tilde{f} different from the true f ; as long as $f(0) = \tilde{f}(0)$, the results will be topologically correct. However, in the camera calibration setting, we are explicitly interested in obtaining a *metrically accurate* solution. Note that Problem 1 is a chicken-and-egg problem in the two unknowns f and $\{s_i\}_{i=1}^n$: knowing the function f , one can estimate the distances as $f^{-1}(Y_{ij})$, and use standard MDS to solve for $\{s_i\}_{i=1}^n$; conversely, knowing the distances, it is trivial to estimate f . But is it possible to estimate both at the same time? To the best of our knowledge, there has not been any claim about whether *accurate metric embedding from non-metric measurements* is possible. In this paper, we will show that the answer depends on the properties of the manifold \mathbb{M} . Specifically, while for \mathbb{R}^m the scale is not observable, we show that accurate metric embedding is possible for \mathbb{S}^2 . Consequently, it is possible to calibrate a camera from any statistics that respects the monotonicity condition (1), even if the function f is a priori unknown.

Other problems with an equivalent formalization: We briefly mention several other problems that can be formalized in the same way, and that will motivate us to solve the problem in a slightly more generic way than what strictly needed for camera calibration. In developmental robotics and similar fields [19–21], a common scenario is that an agent starts from zero knowledge of its sensors, and its first concern is to recover the geometry of the sensor (possibly a camera, but also a range-finder or other robotic sensor) by considering simple statistics of the sensor streams. In *sensor networks* (see, e.g., [32]), one basic problem is localizing the nodes in space based on relative measurements of wi-fi strength. Assuming the signal is a function of the distance, we arrive to the same formalization, using \mathbb{R}^n as the target manifold. More generally, this formalization covers many embedding problems in machine learning, where the data is assumed to be in a metric space, but the available similarities, perhaps obtained by comparing vectors of *features* of the data, cannot be interpreted directly as distances in the original metric space.

3 WHEN IS SIMILARITY A FUNCTION OF THE PIXELS DISTANCE?

The basic assumption of this paper is that it is possible to find a statistics of the pixel luminance that satisfies the monotonicity condition (1). We state a result that guarantees that *any* pairwise statistics is asymptotically a function of the distance between the pixels, if the camera undergoes uniformly random motion, in the sense that the camera’s orientation (a rotation matrix \mathbf{R}) is uniformly distributed in $\text{SO}(3)$ (the set of rotation matrices).

Proposition 2. *If the probability distribution of the camera orientation \mathbf{R} is uniform in $\text{SO}(3)$, the expectation of a function of the luminance of two pixels depends only on the pixel distance: for all functions $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, there exists a function $f : \mathbb{R}^+ \rightarrow \mathbb{R}$, such that*

$$\mathbb{E}\{g(y(s_i), y(s_j))\} = f(d(s_i, s_j)),$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation with respect to \mathbf{R} .

Proof: See Section A-2 of Appendix A. \square

In particular, this is valid for the correlation between pixel values, as the correlation can be written as $\text{corr}(y_i, y_j) = \mathbb{E}\{g(y(s_i), y(s_j))\}$ with $g(y(s_i), y(s_i)) = (y(s_i) - \bar{y})(y(s_i) - \bar{y})$. Most other similarity statistics can be written in the same fashion.

When is similarity monotonic?

Proposition 2 ensures that (1) holds for some function f , but it does not ensure that such function f is monotone. To find conditions that guarantee that f is monotone it is necessary to introduce some model of the environment. Essentially, f might not be monotone if there is some long-range “structure” in the environment. We describe an artificial counter example in which f is not monotone.

Example 3. Imagine a room, shaped like a parallelepiped with the base of size $L \times L$ and height $H \gg L$ (Fig. 1). Suppose an omnidirectional camera is suspended in the middle of the room, equidistant from the walls, ceiling, and floor. From that position, the images of ceiling and floor gets projected on the visual sphere in an area contained in a spherical cap of radius $\delta = 2 \arccos(H/\sqrt{H^2 + L^2})$. For example, for $L = 5\text{m}$ and $H = 10\text{m}$, we obtain $\delta \simeq 28^\circ$. This implies that, if two pixels observe the ceiling at the same time, they cannot be more than 28° apart. Assume that the floor and the ceiling are painted of a uniform white, and the walls have very intricate black-white patterns, well approximated by white noise. We let the camera undergo random rotational motion, and we compute the correlation of the pixel luminance. Consider now two pixels at distance $d(s_i, s_j) = 60^\circ$. Note that any two pixels at this distance will never look both at the ceiling at the same time, because the apparent size of the ceiling is $\delta = 28^\circ$. Hence, there are three possibilities: 1) they are both looking at the walls; 2) one is looking at the walls, another at the ceiling; 3) one is looking at the walls, another at the floor. In all cases, one is looking at the white noise on the walls. Therefore, the

correlation of two pixels streams at distance 60° is 0: $f(60^\circ) = 0$. Consider now two pixels at distance 180° (one exactly opposite to the other on the visual sphere). For these pixels, there are two possibilities: 1) they are both looking at the walls; 2) one looks at the ceiling, the other at the floor. Because floor and ceiling are the same color, the luminance of these two pixels has a slight positive correlation: $f(180^\circ) > 0$. Therefore, the function f is not monotonic, because $f(0^\circ) = 1$, $f(60^\circ) = 0$, and $f(180^\circ) > 0$.

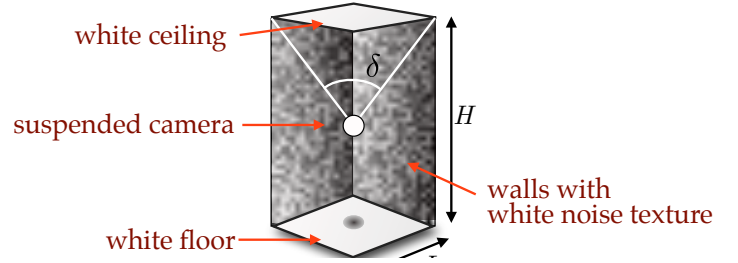


Figure 1: Environment used in Example 3.

4 OBSERVABILITY ANALYSIS

A *symmetry* of an estimation problem is any joint transformation of the unknowns (in this case, the directions $S = \{s_i\}$ and the function f) that does not change the observations (in this case, the similarities Y_{ij}). Studying the observability of the problem means describing what symmetries are present. In this section, we first give a tour of the symmetries of this problem, before presenting the main result in Proposition 8.

Isometries: It is easy to see that the similarities Y_{ij} are preserved by the isometries of the domain \mathbb{M} .

Definition 4. An *isometry* of \mathbb{M} is a map $\varphi : \mathbb{M} \rightarrow \mathbb{M}$ that preserves distances: $d(\varphi(s_i), \varphi(s_j)) = d(s_i, s_j)$. We denote the set of all isometries of \mathbb{M} by $\text{Isom}(\mathbb{M})$.

Sliding: This symmetry exists if f is not informative enough. Define the “informative radius” of f as follows.

Definition 5. For a function $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}$, let $\text{infr}(f)$ be the maximum r such that f is invertible in $[0, r]$.

If the set S has two components distant more than $\text{infr}(f)$ from each other, one component can be isometrically moved independently of the other, without changing the observations (Fig. 2b); we call this “sliding”.

Linear warping: We define a *linear warping* as a map that scales the inter-point distances uniformly by a constant.

Definition 6. A linear warping of \mathbb{M} is a map $\varphi_\alpha : \mathbb{M} \rightarrow \mathbb{M}$ such that, for some $\alpha > 0$, for all $s_1, s_2 \in \mathbb{M}$,

$$d(\varphi_\alpha(s_1), \varphi_\alpha(s_2)) = \alpha d(s_1, s_2),$$

If a linear warping exists, then it is a symmetry of the problem. In fact, suppose that $(f, \{s_i\})$ is a solution

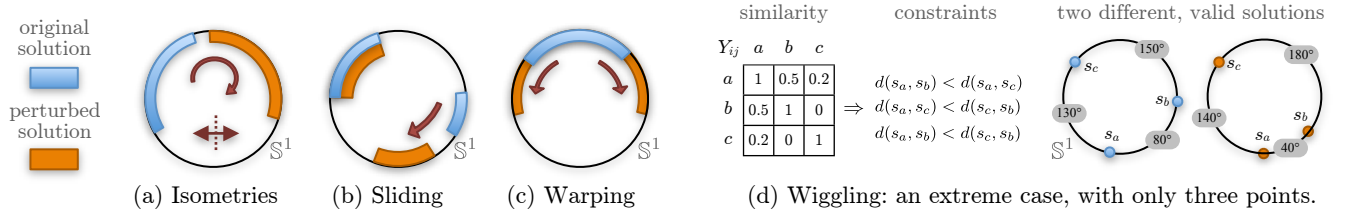


Figure 2: Symmetries of the estimation problem, illustrated in the case $\mathbb{M} = \mathbb{S}^1$. (a) Isometries (for \mathbb{S}^1 , rotations and reflections) are unobservable because they preserve distances between points. (b) If f is non invertible on the whole domain, disconnected components of \mathcal{S} can move isometrically independently from each other; we call this “sliding”. (c) In manifolds with zero curvature (e.g., \mathbb{R}^n and \mathbb{S}^1 , but not \mathbb{S}^2) the scale is not observable; formally, a linear warping (Definition 6) does not violate the problem constraints. (d) If the set of points is finite, the constraints are not violated by small perturbations of the points, called “wiggling”.

of the problem. Construct another solution $(f', \{s'_i\})$, with $f' = \frac{1}{\alpha}f$ and $s'_i = \varphi_\alpha(s_i)$. We would not be able to distinguish between these two solutions, as they would give the same observations.

Wiggling: A peculiar aspect of Problem 1 is that the unknowns $\mathcal{S} = \{s_i\}_{i=1}^n$ live in a continuous space \mathbb{M} , but the observations $Y_{ij} = f(d(s_i, s_j))$ are actually equivalent to a set of *discrete inequalities*, a fact which is very well explained by Agarwal *et. al.* [33]. In fact, because the function f is completely unknown (the only constraint being its monotonicity), all that we can infer about the inter-point distances from the matrix \mathbf{Y} is their ordering: for all $(i, j), (k, l)$, if $Y_{ij} \geq Y_{kl}$, then we can infer $d(s_i, s_j) \leq d(s_k, s_l)$, but nothing more. Therefore, the sufficient statistics in the matrix Y_{ij} is the *ordering* of the entries, not their specific *values*. This raises the question of whether precise metric reconstruction is possible, if the available observations are a set of discrete inequalities. In fact, given a point distribution $\{s_i\}$ of n points, the position of the generic point s_j is constrained by n^2 inequalities (many of which redundant). Inequalities cannot constrain a specific position for s_j in the manifold \mathbb{M} ; rather, they specify a small finite area in which all constraints are satisfied. Therefore, for each solution, an individual point has a small neighborhood in which it is free to “wobble” without violating any constraint. In general, we call these perturbations “wiggling”⁴.

Definition 7. A *wiggling* of a set $\{s_i\} \subset \mathbb{M}$ is a map $\varphi : \mathbb{M} \rightarrow \mathbb{M}$ that preserves the ordering of the distances: for all i, j, k, l : $d(s_i, s_j) < d(s_k, s_l) \Leftrightarrow d(\varphi(s_i), \varphi(s_j)) < d(\varphi(s_k), \varphi(s_l))$.

The size of the allowed wiggling decreases with the density of points; for n points uniformly distributed in \mathbb{M} , one can show that the average wiggling space is in the order of $o(1/n)$ per single point (i.e., keeping the others fixed). In the limit as the points become dense, it is possible to show that wiggling degenerates to rigid linear warpings. For very sparse distributions, the effect of wiggling can be quite dramatic (Fig. 2d).

Main result: The following proposition establishes which of the previously described symmetries are

4. Note that, by these definition, isometries \subseteq linear warping \subseteq wiggling; studying when the inclusions of these transformations classes is the basis of the observability analysis reported in Appendix A.

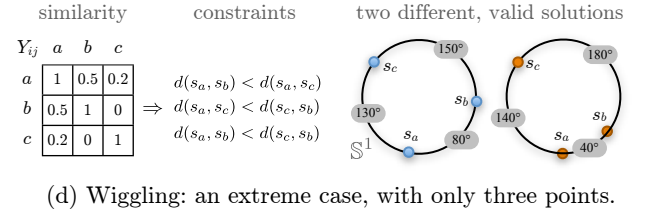


Table 1: Observability classes

class	space	curvature	extra assumptions	symmetries
A	$\mathbb{S}^{\geq 2}$	> 0	-	wiggling, isometries
B	\mathbb{S}^1	0	$\text{rad}(\mathcal{S}) + \text{infr}(f) \geq 2\pi$	wiggling, isometries
C	\mathbb{S}^1	0	$\text{rad}(\mathcal{S}) + \text{infr}(f) < 2\pi$	wiggling, isometries, linear warpings
D	\mathbb{R}^n	0	-	wiggling, isometries, linear warpings
A	\mathbb{H}^n	< 0	-	wiggling, isometries

present in the problem, as a function of the geometric properties of the space \mathbb{M} , the point distribution, and the function f .

Proposition 8. Assume the set $\mathcal{S} = \{s_i\}$ is an open subset of \mathbb{M} whose closure has only one connected component. Let the available measurements be $Y_{ij} = f(d(s_i, s_j))$, where $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}$ is a monotone function with $\text{infr}(f) > 0$. Then:

- If \mathbb{M} has nonzero curvature (e.g., \mathbb{S}^2), then it is possible to recover f exactly, and \mathcal{S} up to isometries.
- If \mathbb{M} has zero curvature:
 - If \mathbb{M} is simply connected (e.g., \mathbb{R}^m), then it is possible to recover f only up to scale, and \mathcal{S} up to isometries plus a “linear warping” (Definition 6).
 - If \mathbb{M} is not simply connected (e.g., \mathbb{S}^1), the scale can be recovered if $\text{infr}(f)$ is large enough. For \mathbb{S}^1 , this happens if

$$\text{rad}(\mathcal{S}) + \text{infr}(f) \geq \pi, \quad (2)$$

where $\text{rad}(\mathcal{S})$ is the radius of \mathcal{S} (Definition A-1).

The observability breaks down as follows:

- “Sliding” occurs if \mathcal{S} has multiple components with Hausdorff distance greater than $\text{infr}(f)$.
- If \mathcal{S} has a finite number of points, there is a “wiggling” uncertainty, in the order of $o(1/n)$ for uniform distributions.

Proof: See Section A-3 in Appendix A. \square

The results are summarized in Table 1 and the various observability classes are labeled A–D for later reference. We note that the observability results depend both on the *local geometrical properties* of the space (curvature) as well as the *global topological properties* (connectedness).

5 MEASURING PERFORMANCE

The performance of an algorithm must be measured in a way compatible with the observability of the problem. We expect an error score to be *invariant*, meaning that it is conserved by the symmetries of the problem. If a score is not invariant, it is measuring something that is not possible to estimate. We expect an error score to be *complete*, meaning that it is minimized only by the solutions of the problem. If an error score is not complete, it cannot be used to distinguish solutions from non solutions. Finally, we wish the error score to be *observable*, in that it can be computed from the data, without the ground truth.

Distances-based performance measures: In our case, several classical error measures, widely used in other contexts, do not satisfy all these properties. The Procrustes error is defined as the mean distance between the solution $\{s_i\}_{i=1}^n$ and the ground truth $\{\bar{s}_i\}_{i=1}^n$, after choosing the best isometry that makes the two sets overlap [34].

Definition 9. The *Procrustes error* e_{pr} is defined as:

$$e_{\text{pr}}(\{\bar{s}_i\}, \{s_i\}) \triangleq \min_{\varphi \in \text{Isom}(\mathbb{M})} \frac{1}{n} \sum_{i=1}^n d(\bar{s}_i, \varphi(s_i)). \quad (3)$$

This error score is unsuitable in our case, because, while it is invariant to isometries, it is not invariant to the other symmetries, namely linear warpings and wiggings. This means that, if we are considering an instance of the problem where the scale is not observable, using the Procrustes error can produce misleading results (we will show this explicitly in Section 8). Moreover, there is the problem that not all points contribute equally to this performance measure. When aligning the two points sets, the points near the center of the distribution will be always more aligned, and the errors will accumulate for the points at the borders of the distribution. To eliminate this problem, we can consider the error on the inter-point distances rather than the absolute position of the points.

Definition 10. The *mean relative error* e_r is the mean error between the inter-point distances:

$$e_r(\{\bar{s}_i\}, \{s_i\}) \triangleq \frac{1}{n^2} \sum_{i,j=1}^n |d(\bar{s}_i, \bar{s}_j) - d(s_i, s_j)|. \quad (4)$$

This error function is still invariant to isometries, does not need an optimization problem to be solved, and all pairs of points contribute equally. Moreover, it can be easily modified to be invariant to linear warpings: because linear warpings scale the distances uniformly, we achieve invariance by optimizing over an unknown scale.

Definition 11. The *mean scaled relative error* e_{sr} is the relative error after the optimal warping:

$$e_{\text{sr}}(\{\bar{s}_i\}, \{s_i\}) \triangleq \min_{\alpha > 0} \frac{1}{n^2} \sum_{i,j=1}^n |d(\bar{s}_i, \bar{s}_j) - \alpha d(s_i, s_j)|. \quad (5)$$

This is invariant to warpings. However, it is still not invariant to wiggings. To achieve invariance to wiggling we have to change approach.

Spearman-correlation-based performance measures: We introduce the *Spearman score*: an invariant, complete, and observable score for all observability classes. It is based on the idea of *Spearman correlation*, which measures a possibly *nonlinear* dependence between two variables, in contrast with the usual correlation, which can only assess linear dependence. The Spearman correlation is a common tool in applied statistics, but it is not widely used in engineering. The idea is that, to assess nonlinear relations, we should consider not the value of each datum, but rather their *order* (or *rank*) in the sequence.

Definition 12. Let $\text{order} : \mathbb{R}^n \rightarrow \text{Perm}(n)$ be the function that computes the order (or rank) of the elements of a vector. For example, $\text{order}([2012, 1, 15]) = [2, 0, 1]$.

Definition 13. The *Spearman correlation* between two sequences x, y is the Pearson correlation of their order vectors: $\text{spear}(x, y) \triangleq \text{corr}(\text{order}(x), \text{order}(y))$.

Lemma 14. The *Spearman correlation detects any nonlinear monotonic relation*: $\text{spear}(x, y) = \pm 1$ if and only if $y = g(x)$ for some monotonic function g .

We use this fact to check whether there exists a monotonic function f such that $Y_{ij} = f(d(s_i, s_j))$. Given a solution $\{s_i\}_{i=1}^n$, we compute the corresponding distance matrix, and then compute the Spearman correlation of the distance matrix to the similarity matrix. To that end, we need to first unroll the matrices into a vector using the operator $\text{vec} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n^2}$.

Definition 15. The *Spearman score* of a solution $\{s_i\}_{i=1}^n$ is the Spearman correlation between the (flattened) similarity matrix and the (flattened) distance matrix $\mathbf{D} = [\mathbf{D}_{ij}] = [d(s_i, s_j)]$:

$$\rho_{\text{sp}}(\{s_i\}) \triangleq |\text{spear}(\text{vec}(\mathbf{Y}), \text{vec}(\mathbf{D}))|. \quad (6)$$

The Spearman score is invariant to all symmetries of the problem, including wiggings, which by definition preserve the ordering of the distances. It is also complete because if $\rho_{\text{sp}}(\{s_i\}, \mathbf{Y}_{ij}) = 1$, then there exists an f such that $Y_{ij} = f(d(s_i, s_j))$.

If the data is corrupted by noise, $\rho_{\text{sp}} = 1$ might not be attainable. In that case, it makes sense to normalize the score by the score of the ground truth.

Definition 16. The *Normalized Spearman score* is

$$\rho_{\text{sp}}^*(\{\bar{s}_i\}, \{s_i\}) \triangleq \frac{\rho_{\text{sp}}(\{s_i\})}{\rho_{\text{sp}}(\{\bar{s}_i\})}. \quad (7)$$

Table 2 summarizes the properties of the performance measures discussed.

6 ALGORITHM

We describe an extension of the classic Shepard-Kruskal algorithm (SK) [24–27] that we call SKv+w (SK variant + warping). The basic idea of SK is to use standard MDS⁵ on Y_{ij} to obtain a first guess for $\{s_i\}$. Given this guess,

5. Given an $n \times n$ distance matrix \mathbf{D} , the best embedding in \mathbb{R}^m can be found by solving for the top m eigenvectors of an $n \times n$ semidefinite positive matrix corresponding to a “normalized” version of \mathbf{D} [27, 34].

Algorithm 1 The SKv+w embedding algorithm for a generic manifold \mathbb{M} .

Input: similarities $\mathbf{Y} \in \mathbb{R}^{n \times n}$, manifold-specific functions: $\text{MDS}_{\mathbb{M}}$, $\text{distances}_{\mathbb{M}}$, $\text{init}_{\mathbb{M}}$. **Output:** $\mathbf{S} \in \mathbb{M}^n$.

```

1  for  $\mathbf{D}^0$  in  $\text{init}_{\mathbb{M}}(\text{order}(\mathbf{Y}))$ : # Some manifolds need multiple starting points
2   $\mathbf{S}^0 = \text{MDS}_{\mathbb{M}}(\mathbf{D}^0)$  # Compute first guess by MDS
3  for  $k = 1, 2, \dots$  until  $s^k$  converged:
4   $\mathbf{D}^k = \text{distances}_{\mathbb{M}}(\mathbf{S}^{k-1})$  # Compute current distances
5   $\mathbf{D}_*^k = \text{vec}^{-1}(\text{sorted}(\text{vec}(\mathbf{D}^k))[\text{order}(\text{vec}(\mathbf{Y}))])$  # Nonparametric fitting and inversion of  $f$ .
6   $\mathbf{S}^k = \text{MDS}_{\mathbb{M}}(\mathbf{D}_*^k)$  # Embed according to the modified distances.
7   $s^k = \text{spearman\_score}(\mathbf{S}^k, \mathbf{Y})$  # Use the Spearman score for checking convergence
8   $\mathbf{S}^* = \mathbf{S}^{k^*}$ , where  $k^* = \arg \max_k s^k$  # Find best iteration according to the score.
9  if  $\mathbb{M}$  is  $\mathbb{S}^m$ ,  $m \geq 2$ : # Find optimal warping factor to embed in the sphere.
10  $\mathbf{D}^* = \text{distances}_{\mathbb{M}}(\mathbf{S}^*)$ 
11  $\alpha^* = \arg \min_{\alpha} \sigma_{m+1}^{\alpha} / \sigma_{m+2}^{\alpha}$ , where  $\{\sigma_i^{\alpha}\} = \text{singular\_values}(\cos(\alpha \mathbf{D}^*))$ 
12 return  $\text{MDS}_{\mathbb{M}}(\alpha^* \mathbf{D}^*)$  # Embed the warped distances
13 return  $\mathbf{S}^*$ 

```

\mathbb{M} -specific initializations: $\text{init}_{\mathbb{R}^m}(\mathbf{oY}) \triangleq \mathbf{oY}$; $\text{init}_{\mathbb{S}^m}(\mathbf{oY}) \triangleq \{\pi \mathbf{oY} / n^2, 2\pi \mathbf{oY} / n^2\}$.

Table 2: Properties of performance measures

obs. class \rightarrow	invariant?				complete?				observable?
	A	B	C	D	A	B	C	D	
Procrustes error (3)	✗†	✗†	✗§	✗§	✓	✓	✗	✗	✗
Relative error (4)	✗†	✗†	✗§	✗§	✓	✓	✗	✗	✗
Scaled relative error (5)	✗†	✗†	✗§	✗§	✓	✓	✓	✓	✗
Spearman score (6)	✓	✓	✓	✓	✓	✓	✓	✓	✓

†: Not invariant to wiggings.

§: Not invariant to linear warping and wiggings.

one can obtain a rough estimate \tilde{f} of f ; given \tilde{f} , one can apply \tilde{f}^{-1} to Y_{ij} to obtain an estimate of the distances D_{ij} ; then one solves again for $\{s_i\}$ using MDS. The SK algorithm does not give accurate metric reconstruction. Our goal was to obtain a general algorithm that could work in all corner cases of the observability analysis. The algorithm described here will be shown to be robust across a diverse set of benchmarks on different manifolds, with a vast variation of shapes of f and noise levels. To this end, we extended the SK algorithm in several ways. In the following, some parts are specific to the manifold: we indicate by $\text{MDS}_{\mathbb{M}}$ a generic implementation of MDS on the manifold \mathbb{M} , so that $\text{MDS}_{\mathbb{R}^n}$ is the classical Euclidean MDS, and $\text{MDS}_{\mathbb{S}^n}$ is the spherical MDS employed by Grossmann *et. al.*.

EM-like iterations (lines 3–7 of Algorithm 1): A straightforward extension is to iterate the alternate estimation of $\{s_i\}$ and f in an EM-like fashion. This modification has also been introduced in other SK variants [27]. This iteration improves the solution, but still does not give metrically accurate solutions.

Choice of first guess for the distance matrix (line 1): Assuming that the similarities have already been normalized ($-1 \leq Y_{ij} \leq 1$), the standard way to obtain an initial guess D_{ij}^0 for the distance matrix is to linearly scale the similarities, setting $D_{ij}^0 \propto 1 - Y_{ij}$. This implies that, given the perturbed similarities $Y_{ij}^* = g(Y_{ij})$ for some monotone function g , the algorithm starts from a different guess and has a different trajectory. However, because the sufficient statistics $\text{order}(Y_{ij}^*) = \text{order}(Y_{ij})$

is conserved, we expect the same solution. The fix is to set $D_{ij}^0 \propto \text{order}(Y_{ij})$ (making sure the diagonal is zero), so that the algorithm is automatically invariant to the shape of f .

Multiple initializations (line 1): We observed empirically that multiple initializations are necessary for the case of \mathbb{S}^m . In particular, if one scales D_{ij}^0 such that $0 \leq D_{ij}^0 \leq \pi$, all solutions generated have diameter $\leq \pi$; if one scales D_{ij}^0 such that $0 \leq D_{ij}^0 \leq 2\pi$, all solutions have diameter $\geq \pi$. Extensive tests show that one of the two starting points always allows convergence to the true solution (the other being stuck in a local minimum). In Algorithm 1 this is represented by a manifold-specific function $\text{init}_{\mathbb{M}}$ returning the list of initial guesses for \mathbf{D} .

Non-parametric inversion of f (line 5): We have to find some representation for f , of which we do not know the shape, and use this representation to compute f^{-1} . In this kind of scenarios, a common solution is to use a flexible parametric representation for f , such as splines or polynomials. However, parametric fitting is typically not robust to very noisy data. A good solution is to use completely *non-parametric* fitting of f . Suppose we have two sequences $\{x_i\}, \{y_i\}$ which implicitly model a noisy relation $y_i = f(x_i) + \text{noise}$ for some monotone f . Our goal is to estimate the sequence $\{f^{-1}(y_i)\}$. Let $\text{sorted}(\{x_i\})$ be the sorted sequence $\{x_i\}$. Then non-parametric inversion can be obtained by using the order of $\{y_i\}$ to index into the sorted $\{x_i\}$ array⁶:

$$\{f^{-1}(y_i)\} \simeq \text{sorted}(\{x_i\})[\text{order}(\{y_i\})].$$

This is seen in line 5 applied to the (unrolled) distance and similarity matrices.

Spearman Score as convergence criterion (line 3): The iterations are stopped when the Spearman score converges. In practice, we observed that after 4–7 iterations the score has negligible improvement for all benchmarks. This score is also used to choose the best solution among multiple initializations (line 8).

6. The square brackets here indicate indexing into the array, as in most programming languages (e.g., Python).

Warping recovery phase (lines 9–12): The most important change we introduce is a “warping recovery” phase that changes the qualitative behavior of the algorithm in the case of \mathbb{S}^m , $m \geq 2$. As explained in the observability analysis, in curved spaces the scale of the points distribution is observable. However, the SK algorithm (i.e., lines 3–7 of Algorithm 1) cannot compensate what we call a linear warping (Definition 6); in fact, it is easy to see that if \mathbf{D}_0 is a fixed point of the loop, also $\alpha\mathbf{D}_0$, for $\alpha > 0$, is a fixed point. In other words, the “null space” of the Shepard-Kruskal algorithm appears to be the group of linear warpings. Therefore, we implemented a simple algorithm to find the scale that best embeds the data onto the sphere, based on the fact that if \mathbf{D} is a distance matrix for a set of points on \mathbb{S}^m , then the cosine matrix $\cos(\mathbf{D})$ must have rank $m + 1$. Therefore, to find the optimal scale, we look for the optimal $\alpha > 0$ such that $\cos(\alpha\mathbf{D})$ is closest to a matrix of rank 3. This is implemented in lines 9–12, where the ratio of the $(m + 1)$ -th and the $(m + 2)$ -th singular value is chosen as a robust measure of the rank.

While it would be interesting to observe the improvements obtained by each variation to the original algorithm, for reasons of space we focus only on the impact of the warping recovery phase. We call SKv the SKv+w algorithm *without* the warping recovery phase (i.e., without the lines 9–12).

Algorithm complexity: The dominant cost of SKv+w lies in the truncated SVD decomposition needed for $\text{MDS}_{\mathbb{M}}$ in the inner loop; the exact decomposition takes $O(n^3)$, which is, in practice, in the order of 5 ms for $n = 100$ and 500 ms for $n = 1000$ on current hardware⁷. There exist faster approximations to speed up the MDS step; see, e.g., the various Nystrom approximations [35].

7 CAMERA CALIBRATION RESULTS

We divide the experimental evaluation in two sections. In this first section, we describe the results of the method for camera calibration. In Section 8, we describe a series of experiments with artificial datasets to demonstrate the various corner cases of the observability analysis.

Hardware: We use three different cameras, covering all practical cases for imaging systems: a perspective camera (“FLIP” in the following), a fish-eye camera (“GOPRO”), and an omnidirectional catadioptric camera (“OMNI”). FLIP: The Flip Mino HD [36] is a \$100 consumer-level video recorder (Fig. 3a). It has a 45° FOV; it has a 3X optical zoom, not used for these logs. GOPRO: The GOPRO camera [37] is a \$300 rugged professional-level fish-eye camera for outdoor use (Fig. 4a). The field of view varies slightly between 127° and 170° according to the resolution chosen; for our tests, we chose a resolution corresponding to a 150° field of view. OMNI: We used a custom-made omnidirectional catadioptric camera (Fig. 5a). This is a small, compact system very popular for micro aerial platforms, such as quadrotors [38, 39]. The camera is created by connecting a

perspective camera to a hyperbolic mirror. The resulting field of view is 360° (horizontally) by 100° (vertically). The images have much lower quality than the FLIP and GOPRO (Fig. 4b). Table 3 summarizes the statistics of the three datasets.

Table 3: Dataset statistics

camera	FOV	fps	resolution	subsampling	n	length
FLIP	45°	30	1280×720	24 × 24 grid	1620	57416
GOPRO	150°	30	1280×720	24 × 24 grid	1620	29646
OMNI	360°	20	640×480	8 × 8 grid	1470	13131

Manual calibration: We calibrated the cameras using conventional techniques, to have a reference to which to compare our method. We calibrated the FLIP using the Matlab Calibration Toolbox [3], which uses a pin-hole model plus second-order distortion models. We calibrated the GOPRO and the OMNI using the OCamCalib calibration toolbox [13], using a fourth-order polynomial for describing the radial distortion profile [6, 7, 40]. Both methods involve printing out a calibration pattern (a checkerboard), taking several pictures of the board, then identifying the corners of the board using a semi-interactive procedure. Some examples of the calibration images used are shown in Fig. 3c and 4c.

Data collection: The environment in which the log is taken influences the spatial statistics of the images. The data logs were taken in a diverse set of environments. For the FLIP, the data was taken outdoors in the Caltech campus, which has a large abundance of natural elements. For the GOPRO, the data was taken in the streets of Philadelphia, a typical urban environment. For the OMNI, the data was taken indoors in a private apartment and an office location. Examples of the images collected are shown in Fig. 3b, 4b, 5b; the full videos are available on the website. In all cases, the cameras were held in one hand and waved around “randomly”, trying to exercise at least three degrees of freedom (shoulder, elbow, wrist), so that the attitude of the camera was approximately uniformly distributed in $\text{SO}(3)$. We did not establish a more rigorous protocol, as these informal instructions produced good data. Data taken by exercising only one degree of freedom of the arm (e.g., forearm, with the wrist being fixed) did not satisfy the monotonicity assumption. Another example of data that we tried that did not satisfy the assumption was data from an omnidirectional camera mounted on a car⁸.

Data processing: For all cameras, the original RGB stream of each pixel was converted to a one-dimensional signal by computing the luminance. We also subsampled the original images with a regular grid so that we could work with a reduced number of points. For the OMNI data, we used masking to only consider the annulus around the center (Fig. 5c), therefore excluding the reflection of the camera in the mirror and the interior of the box which lodged the camera. We used the correlation between the pixel luminance values as the similarity statistics: $Y_{ij} = \text{corr}(y_i(t), y_j(t))$,

⁷. Tests executed using Numpy 1.5, BLAS compiled with Intel MKL, on a 2.67Ghz Intel Xeon core.

⁸. Because the car motion is mostly planar, a portion of the pixels always observes the sky (a featureless scene), while others observe the road (a scene richer in features).



Figure 3: The FLIP camera is a consumer-level portable video recorder. The data for calibration is taken while walking in the Caltech campus, with the camera in hand, and "randomly" waving the arm, elbow, and wrist.



Figure 4: The GOPRO camera is a rugged consumer camera for outdoors use. It uses a fish-eye lens with 170° field of view.

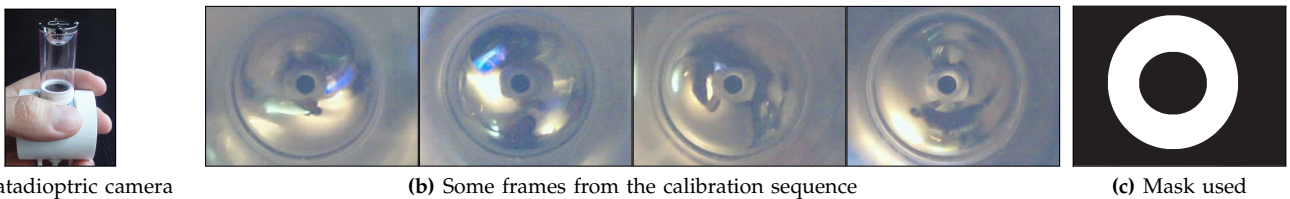
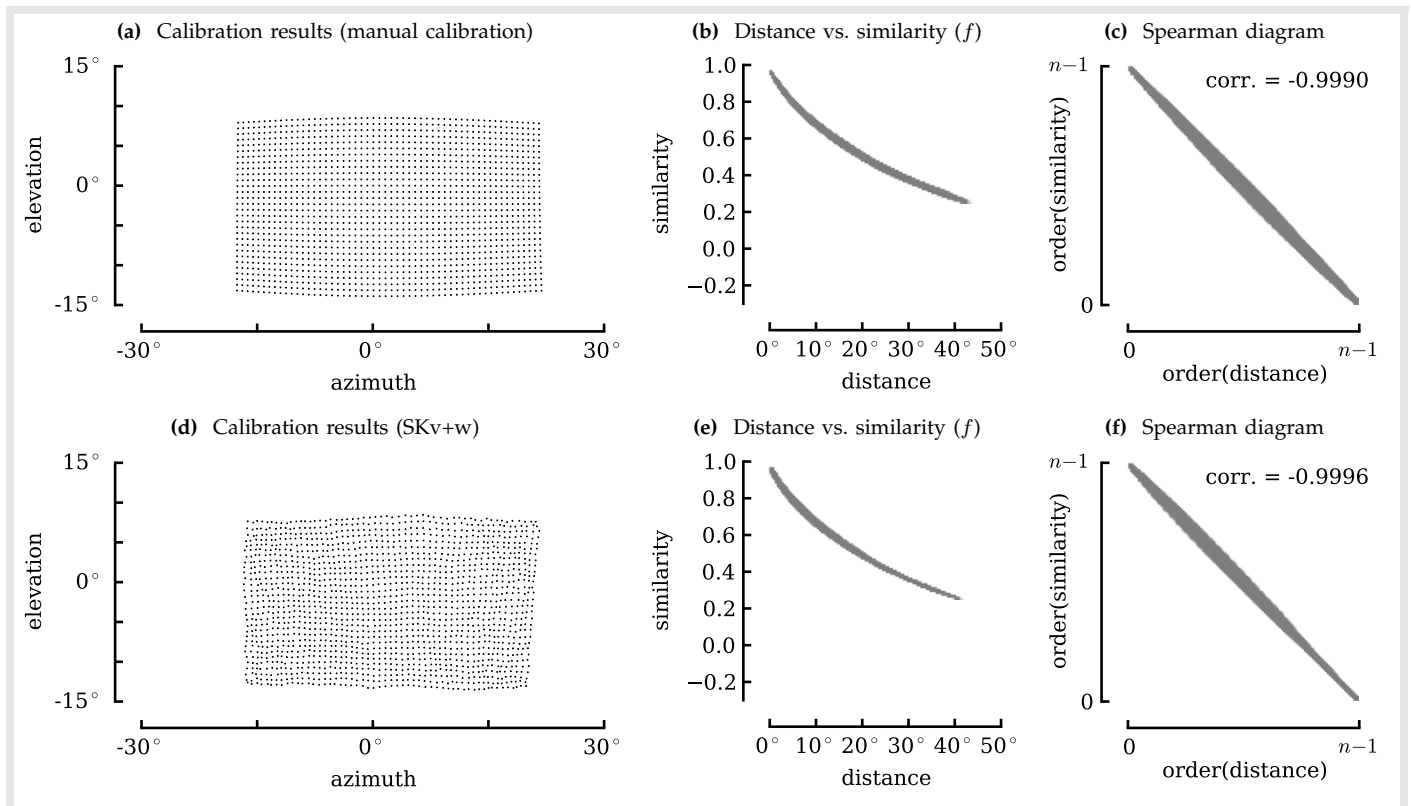


Figure 5: Note the small dimensions of this omnidirectional catadioptric camera, very well suited for aerial robotics applications. The data quality is much lower than for the FLIP and GOPRO data.

Figure 6: Calibration results for the FLIP data using $\text{corr}(y)$ as the similarity statistics. See top of page 10 for explanation.



Legend for Figure 6,7,8: The first row (fig. a,b,c) shows the results of calibration using conventional methods, while the second row (d,e,f) shows the results of our algorithm. The first column (a, d) shows the points distribution on the sphere, displayed using azimuth/elevation coordinates. The second column (b, e) shows the joint distribution of pixel distance ($d(s_i, s_j)$) and pixels similarities (Y_{ij}), which, in this case, is the correlation. This is the function f that we should fit. Finally, the third column (c, f) shows $\text{order}(d(s_i, s_j))$ vs. $\text{order}(Y_{ij})$ and their correlation, from which we derive the Spearman score.

Figure 7: Calibration results for the GOPRO data using $\text{corr}(y)$ as the similarity statistics. See above for explanation.

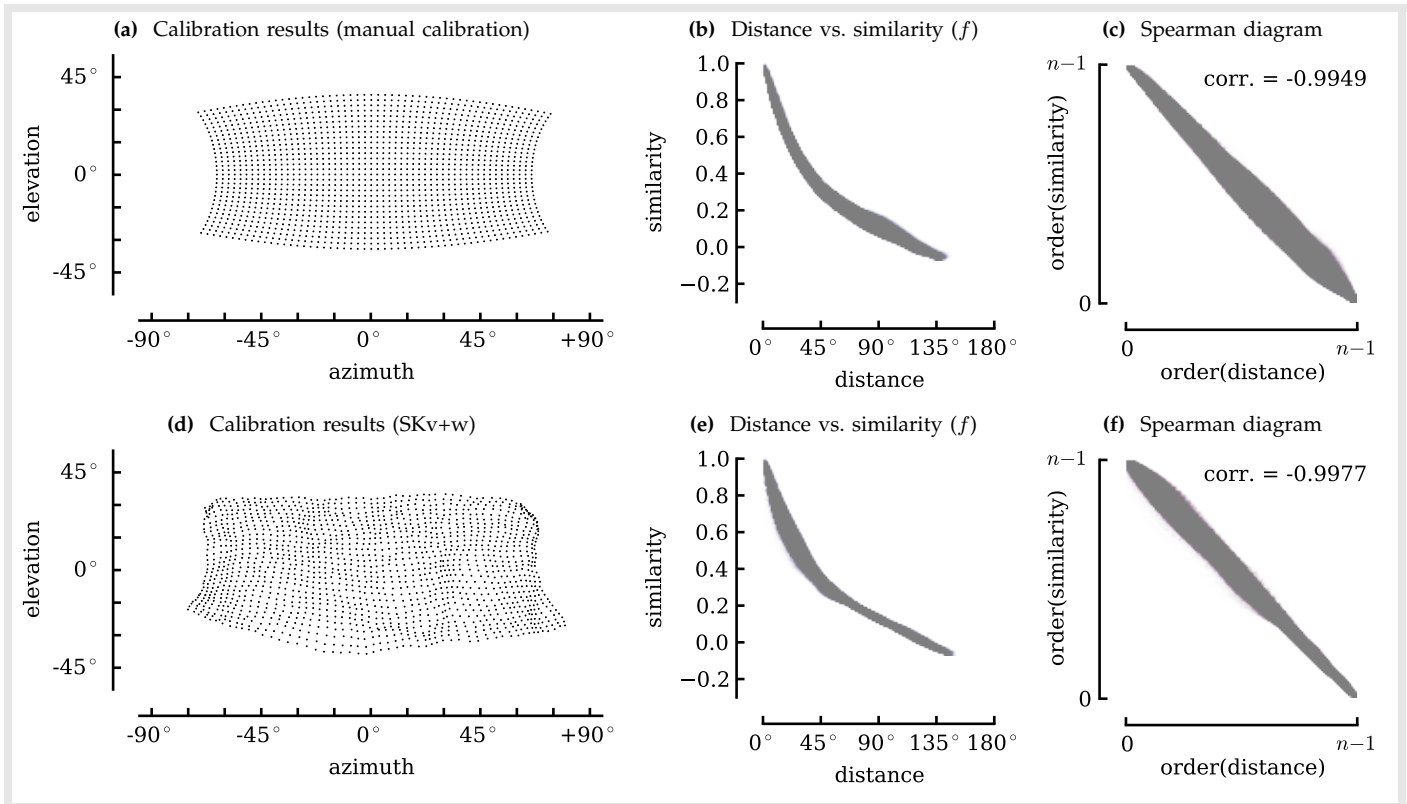
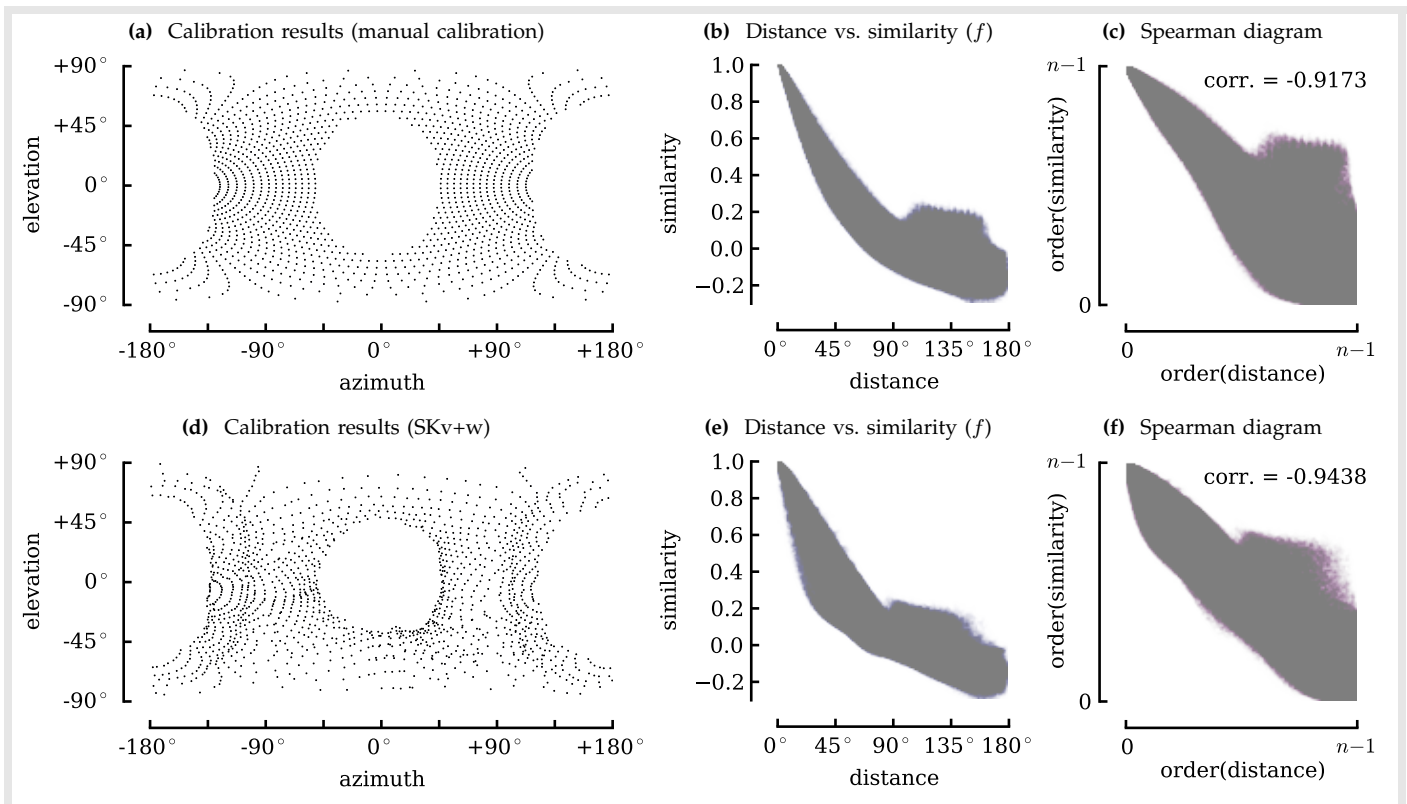


Figure 8: Calibration results for the OMNI data using $\text{corr}(y)$ as the similarity statistics. See the top of this page for explanation.



where $y_i(t)$ indicates the luminance of the i -th pixel at time t . This simple statistics was the most useful across cameras (Section 7.2 discusses other possible choices of the similarity statistics). We found that the monotonicity condition is well verified for all three cameras. To plot these statistics, we assume the calibration results obtained with conventional techniques as the ground truth. The joint distribution of the similarity Y_{ij} and the distance $d(s_i, s_j)$ is shown in Fig. 6b, 7b, 8b. For these logs, the spatial statistics were quite uniform: at a distance of 45° , the inter-pixel correlation was in the range 0.2–0.3 for all three cameras. For the GOPRO and OMNI data, the correlation is 0 at around 90° . The correlation is negative for larger distances. The different average luminance between sky and ground (or floor and ceiling) is a possible explanation for this negative correlation. The OMNI data is very noisy for distances in the range 90° – 180° , as the sample correlation converges more slowly for larger distances. To check that the monotonicity condition is satisfied, regardless of the shape of f , it is useful to look at the Spearman diagrams in Fig. 6c, 7c, 8c, for the FLIP, GOPRO, and OMNI, respectively. These diagrams show, instead of similarity (Y_{ij}) versus distance ($d(s_i, s_j)$), the order of the similarities ($\text{order}(Y_{ij})$) versus the order of the distances ($\text{order}(d(s_i, s_j))$). The correlation of those gives the Spearman score (Definition 15). If there was a perfectly monotonic relation between similarity and distance, the Spearman diagram would be a straight line, regardless of the shape of f , and the Spearman score would be 1 (Lemma 14).

7.1 Calibration results

The complete statistics are presented in Appendix B in Table B-1. The results of manual calibration and calibration using our method are graphically shown in Fig. 6d, 7d, 8d. The plots show the data using spherical coordinates (azimuth/elevation).⁹ There is a number of intuitive remarks that can be made on the results by direct observations of the resulting point distributions (or, better, its 3D equivalent). For the FLIP data (Fig. 6d) the reconstructed directions lie approximately on a grid, as expected. For this data, and the GOPRO as well, the estimated points are more regular at the center of the field of view than on the borders. This is probably due to the fact that the pixels at the border have less constraints. The estimated FOV is very similar to the result given by the manual calibration (43° instead of 45°). For the GOPRO data (Fig. 7d) the shape of the sensor is well reconstructed, except for the two upper corners of the camera. The estimated FOV matches the manual calibration (153° instead of 150°). For the OMNI data (Fig. 8d) the shape of the sensor is overall well reconstructed, but it is more noisy than the FLIP or GOPRO. This is to be expected as the monotonicity relation is not as well respected (Fig. 8e).

9. It is challenging to visualize 3D data with 2D projections, because any projection will distort some part of the data. For this reason, we also provide 3D visualization using MATLAB .fig figures, which allows the user to rotate in 3D the data. Click the following links to access the .fig files: [📌 mino](#), [📌 GOPRO](#), [📌 OMNI](#).

Table 4: Calibration results (normalized Spearman score)

dataset			norm. Spearman score ρ_{sp}^*			
S	FOV	f	$g.\text{truth}$	SKv	SKv+w	MDS
FLIP	45°	$\text{corr}(y)$	1	0.9998	1.0006	0.9709
GOPRO	150°	$\text{corr}(y)$	1	1.0027	1.0029	0.9702
OMNI	360°	$\text{corr}(y)$	1	1.0288	1.0288	0.9831

(See complete results in Table B-1b)

Table 5: Calibration results (Procrustes error)

dataset			Procrustes error		
S	FOV	f	SKv	SKv+w	MDS
FLIP	45°	$\text{corr}(y)$	24.05°	0.74°	15.16°
GOPRO	150°	$\text{corr}(y)$	4.72°	3.53°	6.20°
OMNI	360°	$\text{corr}(y)$	9.48°	9.48°	32.43°

(See complete results in Table B-1c)

It can be concluded that our method gives results reasonably close to manual calibration, even for cases like the OMNI where the monotonicity condition holds only approximately. As predicted by the observability analysis, the scale can be reconstructed even without knowing anything about the function f .

We now look at quantitative performance measures. As explained before, the only admissible performance measure is the Spearman score, shown in Table 4. When judged by this performance measure, the SKv+w algorithm is slightly better than the manual calibration (the normalized Spearman score is larger than 1). In other words, the estimated distribution is actually a better fit of the similarity data than the manual calibration results. This implies that the imprecision in the estimate is a limitation of the input data rather than of the optimization algorithm; to obtain better results, we should improve on the data rather than improving the algorithm.

The Procrustes error (Equation 3) is the most intuitive performance measure (but not invariant to wiggling). The results are shown in Table 5. The error with respect to manual calibration is an average of 0.7° for the FLIP data, 3.5° for the GOPRO data, and 9.5° for the OMNI data. The table shows both the results with and without the warping phase (SKv+w and SKv, respectively). This makes it clear that the warping phase is necessary to obtain a good estimate of the directions, especially for the FLIP data. The difference is lower for the GOPRO data and negligible for the OMNI data. Intuitively, the warping phase takes advantages of what can be called “second-order” constraints, in the sense that they allow us to establish the scale at small FOV, but they disappear as the FOV tends to zero, because a small enough section of \mathbb{S}^2 looks flat (like \mathbb{R}^2). Finally, it is clear that the accuracy of MDS is much lower than SKv or SKv+w. This fact is best seen visually in Fig. B-1j in the supplemental materials. In general, MDS obtains topologically correct solutions, but the scale is never correctly recovered, or the data appears otherwise deformed.

These results seem to outperform the results shown in Grossmann *et al.*: compare, for example, Figure 13 in [18]. Note that their method assume that the function f is known, obtained through a separate calibration

phase. In principle, with much more information, their results should be better. Without having access to their data, we can only speculate on the reason. Perhaps the simplest explanation is that they do not “wave around” the camera for collecting the data; and therefore the monotonicity condition might not be as well satisfied. Moreover, they use a similarity statistics which has very low informative radius (30°), which might cause problems, even though the robust nonlinear embedding algorithm they use should be robust to this fact.

7.2 Results for different similarity statistics

Proposition 2 ensures that *any* statistics is a function of the pixel distance, but this result is limited in three ways: 1) it is only an asymptotic result, valid as time tends to infinity; 2) it assumes a perfectly uniform attitude distribution; and 3) it does not ensure that the function f is invertible (monotonic). Therefore, it is still an engineering matter to find a statistics which is 1) robust to finite data size; 2) robust to a non-perfectly uniform trajectory; and 3) has a large invertible radius. An exhaustive treatment of this problem is outside the scope of this paper and delegated to future work. Here, we briefly show the results for three other statistics in addition to the luminance correlation. All statistics are defined as the correlation of an instantaneous function of the luminance and can be efficiently computed using streaming methods. The first variant consists in applying an instantaneous contrast transformation $c : y \mapsto y^2$ to the luminance before computing the correlation:

$$Y_{ij} = \text{corr}(c(y_i(t)), c(y_j(t))) \quad (8)$$

The second statistic is the correlation of the temporal derivative $\dot{y} = \frac{d}{dt}y$ of the luminance:

$$Y_{ij} = \text{corr}(\dot{y}_i(t), \dot{y}_j(t)) \quad (9)$$

This was inspired by recent developments in neuromorphic hardware [41]. Finally, we consider the correlation of the *sign* of the luminance change, as it is invariant to contrast transformations:

$$Y_{ij} = \text{corr}(\text{sgn}(\dot{y}_i(t)), \text{sgn}(\dot{y}_j(t))). \quad (10)$$

Table 6 shows the Spearman score obtained by using these on the OMNI data (the most challenging dataset). We find, in this case, that the contrast-scaled luminance (8) is slightly better than the simple correlation; the solution found is qualitatively similar (Fig. B-7d). The two other similarity statistics (9) and (10) have much lower scores; for them, the monotonicity assumption is not well verified: their distributions are not informative for large distances (Fig. B-8, B-9). It is clear that there is a huge design space for similarity statistics. In the end, we did not find any statistic which was better than the simple correlation *uniformly* for all our three data sets. Therefore, we consider this an open research question.

8 ADDITIONAL EVALUATION

The main purpose of this section is to provide experimental results that cover all corner cases of the observability analysis in Proposition 8.

Table 6: Results with different similarity statistics

dataset		Spearman score	
S	f	g.truth	SKv+w
OMNI	$\text{corr}(y)$	0.9173	0.9438
OMNI	$\text{corr}(c(y))$	0.9212	0.9465
OMNI	$\text{corr}(\dot{y})$	0.8550	0.9211
OMNI	$\text{corr}(\text{sgn}(\dot{y}))$	0.8739	0.9077

(See complete results in Table B-3a)

Observability class A : The observability class A corresponds to distributions in \mathbb{S}^2 where the scale is observable, and it is possible to reconstruct the directions up to wiggling. To illustrate the effect of wiggling, we generated some synthetic datasets, so that the effect of wiggling can be seen independently of the measurement noise. We use as the ground truth the distribution given by manual calibration, and we use as our function f the kernel $f_{\text{exp}}(d) = \exp(-0.52d)$, which is the exponential kernel that best fits the FLIP data (Fig. 6b). The results are shown in Table 7. We can see that the Spearman score for SKv+w is 1, meaning that the solution found is a perfect solution to the problem. However, the Procrustes error is 1.25° . This is the practical demonstration that the Procrustes error is not admissible. This synthetic experiment gives us a sense of what is the accuracy in the directions domain that we can obtain in practice, even if we had perfect measurements of the correlation or other statistics. The Procrustes error for the real data is in the order of 1° ; this means that the contribution of the noise is commensurable to wiggling, and the results would benefit more from using a denser grid rather than longer logs. The Procrustes error is negligible for the omnidirectional data. This shows that an omnidirectional directions distributions makes the problem overall more constrained.

Table 7: Benchmarks for class A (\mathbb{S}^2) with synthetic data

dataset			Spearman score		Procrustes error	
S	FOV	f	g.truth	SKv+w	g.truth	SKv+w
FLIP	45°	f_{exp}	1	1.000	0°	1.25°
GOPRO	150°	f_{exp}	1	1.000	0°	0.90°
OMNI	360°	f_{exp}	1	1.000	0°	0.00°

(See complete results in Table B-2a and B-2c)

Observability class B : The observability class B corresponds to distributions on \mathbb{S}^1 where the scale is observable due to the non-simply connected topology and a function f with large informative radius. For this set of benchmarks we used synthetic data. We used three different functions f , shown in Table 8. The function

Table 8: Kernels used in synthetic benchmarks

f	infr(f)
$f_{\text{lin}}(d) = 0.5 - 0.5d$	180°
$f_{\text{smooth}}(d) = \cos^3(d)$	180°
$f_{\text{steep}}(d) = \max\{\cos^3(d), 0\}$	90°

$f_{\text{lin}}(d)$ is linear in the distance d . The function $f_{\text{smooth}}(d)$ is nonlinear in the distance, but still invertible on the

whole domain. The function f_{steep} is equal to f_{smooth} for $d \in [0, 90^\circ]$ and 0 for $d \geq 90^\circ$. This implies that $\text{infr}(f_{\text{steep}}) = 90^\circ$. We simulated random distributions of points in \mathbb{S}^1 with 315° FOV. This satisfies condition (2) because $315^\circ/2 + 90^\circ \geq 180^\circ$. Table 9 shows the diameter of the estimated distribution. In all cases, SKv recovers the scale of the distribution. Instead MDS recovers the scale correctly only for linear f .

Table 9: Benchmarks for class B (\mathbb{S}^1 , observable scale)

dataset			diameter	
S	FOV	f	MDS	SKv
random dist.	315°	f_{lin}	318°	318°
random dist.	315°	f_{steep}	127°	312°
random dist.	315°	f_{smooth}	114°	316°

(See complete results in Table B-4c and B-4a)

Observability class C: The observability class C corresponds to the case of \mathbb{S}^1 in which the scale is not observable, because f is not informative enough. For this class, we use a mix of synthetic data and real data. For the synthetic data, we simulated a random distribution of directions on \mathbb{S}^1 with FOV 45° and 90° . For the real data, we extracted the center scan line of the FLIP and GOPRO data. We know that these pixels lie approximately on a great circle of \mathbb{S}^2 , therefore this data can be embedded in \mathbb{S}^1 . Table 10 shows the Spearman score and the Procrustes error obtained by $\text{MDS}_{\mathbb{S}^n}$ and SKv. The most interesting fact about these benchmarks is that they clearly show that, without a proper observability analysis, one might reach erroneous conclusions by considering a non-admissible performance measure. If one were to compare MDS and SKv only on the Procrustes error, one would conclude erroneously that MDS performs better than SKv. However, in reality, SKv gives better results, as can be seen from the Spearman score. What is happening is that, for the non-admissible measure, the intrinsic bias of MDS is better adjusted to the dataset bias.

Table 10: Benchmarks for class C (\mathbb{S}^1 , unobservable scale)

dataset			Spearman score		Procrustes error	
S	FOV	f	MDS	SKv	MDS	SKv
FLIP (center)	23°	$\text{corr}(y)$	0.9706	0.9999	13.52°	25.62°
random dist.	45°	f_{smooth}	0.9987	0.9999	7.47°	20.08°
GOPRO (center)	75°	$\text{corr}(y)$	0.9592	0.9988	8.58°	13.01°
random dist.	90°	f_{smooth}	0.9853	0.9997	9.22°	8.41°

(See complete results in Table B-5c and B-5a)

Observability class D: The observability class D corresponds to the Euclidean case, where the scale is not observable. For this class we used exclusively synthetic data. We simulated a random distribution of points on the $[0, 1] \times [0, 1]$ square, and for the similarities we used the three functions f_{lin} , f_{steep} , f_{smooth} . The data in Table 11 is a simple verification of the fact that MDS provides a correct reconstruction only if similarity is a linear function of the distance, while SKv provides the correct solution regardless of the shape of f .

Table 11: Benchmarks for class D (\mathbb{R}^2)

dataset		norm. Spearman score ρ_{sp}^*		
S	f	g. truth	MDS	SKv
random square	f_{lin}	1	1.000	1.000
random square	f_{steep}	1	0.7706	1.000
random square	f_{smooth}	1	0.9182	1.000

(See complete results in Table B-6a)

9 CONCLUSIONS

This paper presented a new calibration method that does not need any known calibration pattern (e.g., checkerboards), known camera motion, or another calibrated apparatus. It does not have any assumption on the camera model, and is therefore able to calibrate any single-view point cameras. We have shown that calibration-by-correlation is an instance of the problem of metric embedding from non-metric measurements, which appears naturally in many other fields. So far, it has implicitly been assumed that it was not possible to recover the metric information (scale of the distribution) from non-metric measurements. We have given a comprehensive discussion of the observability of the problem, showing that it depends both on the local geometrical characteristics (curvature) as well as the global topological properties (connectedness) of the particular manifold considered. While in Euclidean space it is *never* possible to recover the scale, it *is* possible in spaces of nonzero curvature, or in non-simply connected manifolds such as \mathbb{S}^1 . We have presented an optimization algorithm based on the classical Shepard-Kruskal algorithm, with several additions that make it robust across manifolds and a wide variety of benchmarks. The main addition is a “warping recovery phase” that we have shown necessary to obtain the correct scale in the spherical case. In addition to the camera calibration problem, we evaluated the algorithm on a series of synthetic benchmarks, making sure it works as expected in all corner cases of the observability analysis. Therefore, it will likely be useful for problems other than camera calibration that have a similar formalization.

Future work: On the theoretical side, there is still much to do in the analysis of Problem 1. We have provided an observability analysis, but what is missing is some result, in the spirit of the Cramér–Rao bound, which would answer the question of what noise level can be tolerated on f to obtain a given accuracy on metric reconstruction. Because the observations can be interpreted as a set of inequalities, while the unknowns live in a continuous space, standard techniques cannot be used. On the algorithmic side, there is still much to do about proving the convergence properties of the algorithm. The algorithm appears to be robust across diverse benchmarks, but we do not have proofs of global convergence. The difficulty in the analysis is that the core of the algorithm is a mix of “continuous” operations (e.g. computing the SVD of a matrix) and “discrete” operations (e.g., reordering the elements of the matrix); the interaction of these continuous and discrete operations is hard to analyze with conventional techniques. Several other extensions

are motivated by the problem of camera calibration. As noted before, it is an open question whether one can find a better similarity statistics than the correlation. It would also be useful to study extensions of the problem to non-monotonic functions f , or that allow for the statistics to vary in different parts of the field of view. This would allow to use as source data logs in which the camera attitude is not uniform in $SO(3)$, such as the logs from autonomous vehicles. It would also be useful to extend this method to non-central cameras, where each pixel has a direction $s_i \in \mathbb{S}^2$ and a spatial position in \mathbb{R}^3 . Finally, our most immediate current work consists in integrating this method with parametric methods that use a prior knowledge of the camera model [7], to obtain the best of both worlds.

REFERENCES

- [1] T. A. Clarke and J. G. Fryer. "The Development of Camera Calibration Methods and Models". In: *The Photogrammetric Record* 16.91 (1998), pp. 51–66 DOI:10.1111/0031-868X.00113.
- [2] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. "Camera Models and Fundamental Concepts Used in Geometric Computer Vision". In: *Foundations and Trends in Computer Graphics and Vision* 6.1–2 (2011), pp. 1–183 DOI:10.1561/06000000023.
- [3] J.-Y. Bouguet. *The Matlab Calibration Toolbox*. (url).
- [4] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22.11 (2002) DOI:10.1109/34.888718.
- [5] D. Gennery. "Generalized camera calibration including fish-eye lenses". In: *Int. J. of Computer Vision* 68.3 (2006) DOI:10.1007/s11263-006-5168-1.
- [6] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion". In: *Proceedings of IEEE International Conference of Vision Systems (ICVS)*. 2006 DOI:10.1109/ICVS.2006.3.
- [7] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A Toolbox for Easy Calibrating Omnidirectional Cameras". In: *Int. Conf. on Intelligent Robots and Systems*. 2006 DOI:10.1109/IROS.2006.282372.
- [8] D. Scaramuzza and R. Siegwart. "Vision Systems: Applications". In: ed. by G. Obinata and A. Dutta. inTech, 2007. Chap. A Practical Toolbox for Calibrating Omnidirectional Cameras. ISBN: 978-3-902613-01-1 (url).
- [9] C. Mei and P. Rives. "Single View Point Omnidirectional Camera Calibration from Planar Grids". In: *Int. Conf. on Robotics and Automation*. Rome, Italy, Apr. 2007, pp. 3945–3950 DOI:10.1109/ROBOT.2007.364084.
- [10] C. Mei. *Omnidirectional camera calibration toolbox for MATLAB*. (url).
- [11] J. Kannala and S. S. Brandt. "A generic camera model and calibration method for conventional, wide-angle and fish-eye lenses". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28.8 (2006), pp. 1335–1340 DOI:10.1109/TPAMI.2006.153.
- [12] J. Kannala. *Camera calibration Toolbox for Generic Lenses for MATLAB*. (url).
- [13] D. Scaramuzza. *OCamCalib: Omnidirectional Camera Calibration Toolbox for Matlab*. (url).
- [14] M. Grossberg and S. Nayar. "The Raxel Imaging Model and Ray-Based Calibration". In: *Int. J. of Computer Vision* 61.2 (Feb. 2005), pp. 119–137 DOI:10.1023/B:VISI.0000043754.56350.10.
- [15] F. Espuny and J. B. Gil. "Generic self-calibration of central cameras from two "real" rotational flows". In: *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*. 2008 (url).
- [16] A. Martinelli, D. Scaramuzza, and R. Siegwart. "Automatic Self-calibration of a Vision System during Robot Motion". In: *Int. Conf. on Robotics and Automation*. Orlando, FL, May 2006, pp. 43–48 DOI:10.1109/ROBOT.2006.1641159.
- [17] G. Antonelli, F. Caccavale, F. Grossi, and A. Marino. "Simultaneous calibration of odometry and camera for a differential drive mobile robot". In: *Int. Conf. on Robotics and Automation*. 2010 DOI:10.1109/ROBOT.2010.5509954.
- [18] E. Grossmann, J. A. Gaspar, and F. Orabona. "Discrete camera calibration from pixel streams". In: *Computer Vision and Image Understanding* 114.2 (2010), pp. 198–209. ISSN: 1077-3142 DOI:10.1016/j.cviu.2009.03.009.
- [19] M. Boerlin, T. Delbruck, and K. Eng. "Getting to know your neighbors: unsupervised learning of topography from real-world, event-based input". In: *Neural computation* 21.1 (2009) DOI:10.1162/neco.2009.06-07-554.
- [20] J. Stober, L. Fishgold, and B. Kuipers. "Sensor Map Discovery for Developing Robots". In: *AAAI Fall Symposium on Manifolds Learning and Its Applications*. 2009 (url).
- [21] J. Modayil. "Discovering sensor space: Constructing spatial embeddings that explain sensor correlations". In: *Int. Conf. on Development and Learning*. 2010 DOI:10.1109/DEVLRN.2010.557885.
- [22] J. W. Sammon. "A Nonlinear Mapping for Data Structure Analysis". In: *IEEE Transactions on Computers* 18 (5 1969), pp. 401–409 DOI:10.1109/T-C.1969.222678.
- [23] R. C. T. Lee, J. R. Slagle, and H. Blum. "A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space". In: *IEEE Transactions on Computers* 26 (3 1977), pp. 288–292. ISSN: 0018-9340 DOI:10.1109/TC.1977.1674822.
- [24] R. Shepard. "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function (Part I)". In: *Psychometrika* 27.3 (1962), pp. 125–140 DOI:10.1007/BF02289630.
- [25] R. Shepard. "The analysis of proximities: Multidimensional scaling with an unknown distance function (Part II)". In: *Psychometrika* 27.3 (1962), pp. 219–246. ISSN: 0033-3123 DOI:10.1007/BF02289621.
- [26] J. B. Kruskal. "Multidimensional scaling by optimizing goodness of fit to a nonparametric hypothesis". In: *Psychometrika* 29.1 (1964), pp. 1–27 DOI:10.1007/BF02289565.
- [27] S. L. France and J. J. Carroll. "Two-Way Multidimensional Scaling: A Review". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 99 (2010), pp. 1–18 DOI:10.1109/TSMCC.2010.2078502.
- [28] M. do Carmo. *Riemannian Geometry*. Birkhauser, 1994. ISBN: 3-540-20493-8.
- [29] J. Rotman. *An introduction to the theory of groups*. Springer-Verlag, 1995. ISBN: 0387942858.
- [30] J. Ratcliffe. *Foundations of hyperbolic manifolds*. Vol. 149. Graduate Texts in Mathematics. Springer, 2006 DOI:10.1007/978-0-387-47322-2.
- [31] T. Cox and M. Cox. *Multidimensional Scaling*. Boca Raton, FL: Chapman & Hall / CRC, 2001. ISBN: 1-58488-094-5.
- [32] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz. "Localization from connectivity in sensor networks". In: *IEEE Transactions on Parallel and Distributed Systems* 15.11 (2004), pp. 961–974 DOI:10.1109/TPDS.2004.67.
- [33] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie. "Generalized Non-metric Multidimensional Scaling". In: *Eleventh International Conference on Artificial Intelligence and Statistics*. 2007 (url).
- [34] J. C. Gower and G. B. Dijksterhuis. *Procrustes problems*. Vol. 30. Oxford Statistical Science Series. Oxford, UK: Oxford University Press, 2004. ISBN: 978-0-19-851058-1.
- [35] J. C. Platt. "FastMap, MetricMap, and Landmark MDS are all Nystrom algorithms". In: *In Proceedings of 10th International Workshop on Artificial Intelligence and Statistics*. 2005, pp. 261–268 (url).
- [36] Pure Digital Technologies. *The Flip MINO HD website*. (url).
- [37] Woodman Labs. *The GOPRO Camera website*. (url).
- [38] C. Gimkiewicz, C. Urban, E. Innerhofer, P. Ferrat, S. Neukom, G. Vanstraelen, and P. Seitz. "Ultra-miniature catadioptrical system for an omnidirectional camera". In: ed. by H. Thienpont, P. V. Daele, J. Mohr, and M. R. Taghizadeh. Vol. 6992. 1. Strasbourg, France: SPIE, 2008, 69920J DOI:10.1117/12.779988.
- [39] S. Weiss, D. Scaramuzza, and R. Siegwart. "Monocular-SLAMDbased navigation for autonomous micro helicopters in GPS-denied environments". In: *Journal of Field Robotics* 28.6 (2011), pp. 854–874. ISSN: 1556-4967 DOI:10.1002/rob.20412 (url).
- [40] M. Ruffi, D. Scaramuzza, and R. Siegwart. "Automatic Detection of Checkerboards on Blurred and Distorted Images," in: *Int. Conf. on Intelligent Robots and Systems*. Nice, France, Sept. 2008, pp. 3121–3126 DOI:10.1109/IROS.2008.4650703.
- [41] P. Lichtsteiner, C. Posch, and T. Delbruck. "A 128 × 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 43.2 (Feb. 2008), pp. 566–576. ISSN: 0018-9200 DOI:10.1109/JSSC.2007.914337.