A Hyperhemispherical Grid Filter for Orientation Estimation

Florian Pfaff, **Kailai Li**, and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
florian.pfaff@kit.edu, kailai.li@kit.edu, uwe.hanebeck@kit.edu

***Abstract*—Estimating orientations of objects in Euclidean space is an omnipresent challenge in robotics and autonomous systems. A useful representation of orientations involves unit quaternions. While the space of all unit quaternions forms a three-dimensional unit hypersphere, inverting the sign of a quaternion does not change the orientation described by it. Therefore, all possible orientations can be described by considering only a hemisphere of the unit hypersphere. In this paper, we propose a grid filter for arbitrary-dimensional unit hyperhemispheres and apply it to an orientation estimation task and another evaluation scenario. Our approach outperforms previous approaches that consider densities on the entire hypersphere.**

***Index Terms*—Directional estimation, grid filter, quaternion**

I. Introduction

Estimating the orientation of an object in three-dimensional space is of high practical relevance. It is particularly important in the context of robotics, e.g., for controlling the autonomous flight of drones [1]. There are multiple ways to represent orientations. The representation influences the models and has thus a massive impact on the estimation task. Filters tailored to different representations have been developed in the literature. Important representations for which filters have been developed include rotation matrices and unit quaternions.

Nonlinear variants of the Kalman filter should not be directly applied to rotation matrices as they are redundant and can be numerically problematic. However, one can use that the rotation matrices form the Lie group $\mathrm{SO}(3)$ and then apply the invariant extended or unscented Kalman filter [2]. The specific group structure of $\mathrm{SO}(3)$ was considered along with $\mathrm{SE}(3)$ (which considers both the position and orientation of an object in three-dimensional space) in [3]. While such approaches have been applied successfully even for challenging scenarios involving $\mathrm{SE}(3)$ poses, such as visual-inertial odometry [4], they are generally not computationally cheap and it is not possible to increase the number of parameters in the filter to obtain more accurate solutions. Further, for such filters, the description of the uncertainty is generally not suited to precisely model arbitrary uncertainties. For example, problems may arise in cases in which the probability mass is concentrated around multiple vastly different orientations with little probability mass in between.

In this paper, we focus on the representation based on unit quaternions. The domain on which all unit quaternions reside is the three-dimensional unit hypersphere $\mathbb{S}^3$, which can be seen as all points in $\mathbb{R}^4$ of unit norm. The quaternions q and $-q$ describe the same rotation, and thus, $\mathbb{S}^3$ is a double cover of the group of spatial rotations. When describing uncertain rotations using a probability density on $\mathbb{S}^3$, the equivalence of the rotations described by q and $-q$ should be taken into account by ensuring that the probability density at each point is the same as on its antipode.

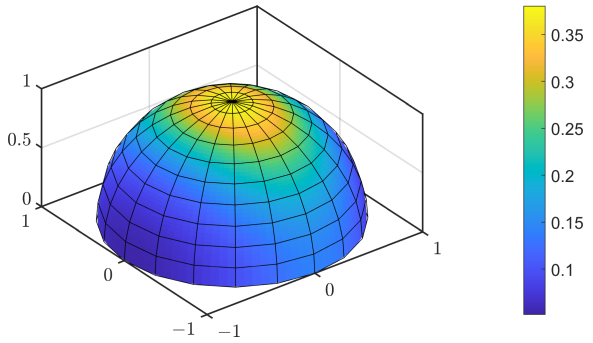


Fig. 1. The density of a Bingham distribution is plotted on the hyperhemisphere and scaled by the factor 2 to ensure normalization.

Different filters that can be used for antipodally symmetric densities on $\mathbb{S}^3$ exist in the literature. First, there is the Bingham filter [5], [6], in which the prior and posterior densities are assumed to be Bingham-distributed. The Bingham distribution is well suited for this estimation task because its density is antipodally symmetric. While the Bingham distribution can represent densities of many shapes on the hypersphere, it has some downsides. Some parameters, such as the normalization constant are expensive to calculate. Further, the Bingham distribution can have two modes or infinitely many along a circle, however, it cannot have, e.g., 4 or 6 modes.

A different approach is to use a grid filter on the hypersphere, which is motivated by grid filters on bounded Euclidean spaces [7]. In [8], [9], the unit sphere and unit hypersphere were considered. Two different approaches for generating the grid were considered. The first is a scheme that provides points that cover the hypersphere homogeneously. More detail on this is provided later. The second scheme is a mode-centric grid, which provides a high resolution around the modes. However, it assumes there are only two modes and additional modes can only be captured with a lower resolution.

In this paper, we provide a grid filter that explicitly considers that q and $-q$ describe the same rotation. We do this by regarding only a hemisphere of $\mathbb{S}^3$, which we denote by $\mathbb{H}^3$. This way, we avoid redundancies and prevent asymmetries, which may arise for grids on the entire sphere due to numerical issues and approximation errors. Our approach is provided in a general way for arbitrary-dimensional hyperhemispheres $\mathbb{H}^d$. Thus, also densities on $\mathbb{H}^2$, such as the one shown in Fig. 1, can be considered. In our novel filter, which we call hyperhemispherical grid filter (HHGF), we use a homogeneous grid on the hyperhemisphere that is based on subdividing the hyperhemisphere into equally sized patches.

While non-homogenous grids can be more efficient per grid point, such grids need to be adapted over time to be useful in scenarios in which the state changes significantly. Further, the underlying assumptions of the grid-generation scheme, such as unimodality, need to be fulfilled to achieve an accurate description of the uncertainty. In contrast, by using a homogeneous grid, we do not make any assumptions about the number of modes. With sufficient grid points, we can properly represent densities whose probability mass is concentrated around multiple vastly different quaternions. Further, using a homogeneous grid, we avoid having large gaps in between the grid points. Gaps can lead to high estimation errors if the function value of the likelihood function for a given measurement is only high in the gaps.

The paper is structured as follows. First, we describe the grid on $\mathbb{H}^d$ used in the filter in the second section. The HHGF itself is described in Sec. III. An evaluation is provided in the fourth section and a conclusion and an outlook in the last section.

II. A Homogeneous Grid for Hyperhemispheres

Before describing grids for hyperhemispheres, we provide a formal definition of the hyperhemisphere. In this paper, we shall always consider the "upper" hemisphere along the last dimension. To ensure the hemisphere does not contain both a point and its antipode, we define the hyperhemisphere $\mathbb{H}^d$ as

$$\begin{aligned}\mathbb{H}^d = \{\underline{x} \in \mathbb{R}^{d+1} : \|\underline{x}\| = 1 \\ \wedge (x_{d+1} > 0 \vee x_{d+1} = 0 \wedge x_d > 0 \\ \vee x_{d+1} = 0 \wedge x_d = 0 \wedge x_{d-1} > 0 \ldots)\} ,\end{aligned}$$

in which the expression in the parentheses continues until $x_1 > 0$ is reached. For example,

$$\begin{aligned}\mathbb{H}^2 = \{\underline{x} \in \mathbb{R}^3 : \|\underline{x}\| = 1 \\ \wedge (x_3 > 0 \vee x_3 = 0 \wedge x_2 > 0 \vee x_3 = 0 \wedge x_2 = 0 \wedge x_1 > 0)\} .\end{aligned}$$

To obtain a grid on $\mathbb{H}^d$, we subdivide the domain into n regions $A_1, \ldots, A_n$. The union of all regions must be equal to the original domain and there should be no overlaps. Such a subdivision is called a partition. For reasons stated in the introduction, we want our grids to cover the domain homogeneously. To achieve the homogeneity, we enforce that all regions in the partition are of equal size. We assign a value to each region, which we refer to as grid value. The grid values are stored in a vector $\underline{\gamma} = [\gamma_1, \ldots, \gamma_n]^\top$. Further, we designate one point in each region to represent the respective region. We choose the centers of the regions and refer to them as grid points $\underline{\beta}_1, \ldots, \underline{\beta}_n$.

In the first subsection, we describe the different interpretations of the grid values (see also [8]). In the second subsection, we outline an equal area partitioning approach for hyperspheres, describe its shortcomings, and explain a modification that makes the approach applicable to hyperhemispheres.

A. Interpretations of the Grid Values

In the first interpretation, we interpret our state space to be discrete. Each discrete state corresponds to one of the n regions and each grid value describes the probability of being in the respective state. The grid values are function values of a probability mass function, which sums up to one. One can derive a filter that is reminiscent of a Wonham filter [10] for this discrete domain. Transforming a system model for a continuous domain into one for a discrete domain is nontrivial and may involve integrals [11].

In the second interpretation, the grid values describe a probability density function defined on the original continuous state space. The grid values can be seen as the function values of the density at the grid points, and an interpolation could be used to obtain the function values for points that are not on the grid (spherical harmonics were used for $\mathbb{S}^2$ in [8]). If an interpolation may yield negative function values, the square roots of the grid values can be interpolated instead. Squaring the function values of this interpolation then provides an interpolation for the actual density [8], [12]. As an alternative, we can define a function on the continuous domain by setting the function values in each region to the respective grid value. We focus on this interpolation to obtain a continuous density because it simplifies our derivations and we believe it is easy to comprehend.

The grid values used by the two interpretations differ but can be transformed into one another. To obtain the function values of a probability density function from the values of the probability mass function used in the first interpretation, one can distribute the probability masses equally in the respective regions. Each grid value for the second interpretation then corresponds to the grid value for the first interpretation divided by the size of the region. To convert a probability density function into a probability mass function, we integrate over the individual regions. Assuming the density function is constant in each region, the grid value for the second interpretation only has to be multiplied by the size of the region to obtain the probability mass for the first interpretation.

B. Equal Area Partitioning for Hemispheres

As we have specified, we aim to partition the hemisphere into regions of equal size. Importantly, our filter will only provide one piece of information on each region. To make statements that are useful and easy to interpret, it makes sense to ensure that each region covers some part of the domain that is easy to comprehend and is small according to some metric other than its total size. Such a metric could be the

largest Euclidean distance between two points in the region. One approach to obtain small regions could be to adapt Lloyd's algorithm [13] to hemispherical domains. However, it is an iterative heuristic to solve a problem that is NP-hard and its convergence rate can be slow [14].

The approach we propose is based on the recursive zonal equal area partitioning algorithm [15] for hyperspheres. In the following paragraphs, we provide a simplified explanation of [15] to give the reader the degree of understanding that is required to comprehend our changes to the algorithm. There is a closed-form formula for subdividing a unit hypersphere into equally sized regions along one axis. We choose the last axis of the Euclidean space in which the hypersphere is embedded. We thus obtain a partitioning comprising n regions (see Fig. 2 for an example) of size $|\mathbb{S}^d|/n$. The uppermost and lowermost regions are called the polar caps. The other regions shall be referred to as collars.

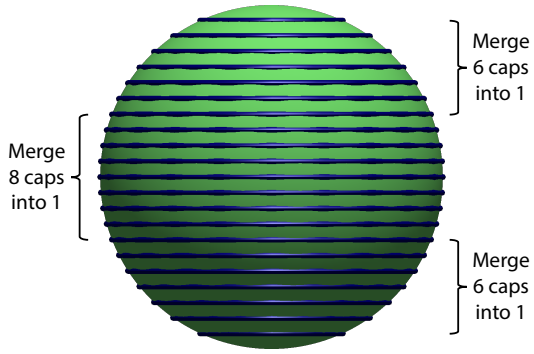
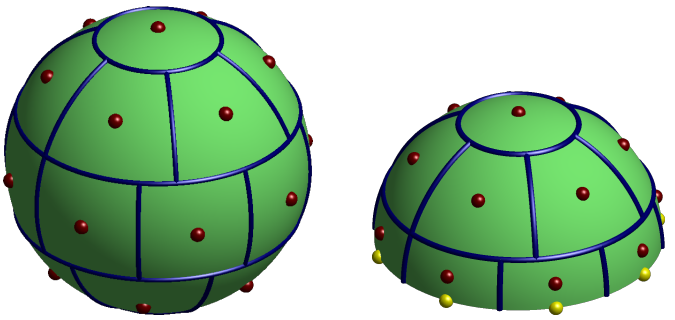


Fig. 2. Naïve subdivision of $\mathbb{S}^2$ into 22 regions of equal size and sketch how they could be joined. We adopted code from [15] to visualize the regions.

If we use the maximum Euclidean distance between two points in a region as an additional measure of size, this naïve subdivision scheme produces rather large regions, as is evident in Fig. 2. Further, even if the probability mass in every region was known, no information could be derived about the azimuth angle. To obtain smaller regions, we consider how collars can be subdivided. This can be achieved by subdividing the upper and lower borders into m equally sized regions while ensuring that the subdivisions of both borders are aligned. The borders of the collars are manifolds with the topology of $\mathbb{S}^{d-1}$. The subdivision can be done recursively until the unit circle $\mathbb{S}^1$ is to be subdivided. The unit circle can be trivially subdivided into m parts by splitting it into regions of size $2\pi/m$.

In the approach in [15], collars of the size of an integer multiple of the region size are generated and subdivided afterward. The result of the subdivision, which is illustrated in Fig. 3a, could also be obtained by joining some collars of the naïve subdivision, which are then split up (see Fig. 2). No changes are done to the polar caps, and thus, the approach in [15] uses the same polar caps as the naïve subdivision. While the algorithm provides all information that is required for a good subdivision (the number of collars and the number of regions in each collar), it is geared toward partitioning the entire hypersphere. In our novel subdivision scheme, we



(a) Partition of $\mathbb{S}^2$ with 22 regions and the respective centers obtained via the original equal area partitioning.

(b) Partition obtained when considering only the 15 regions that at least partially located on the upper hemisphere. The new centers are shown in red and the centers of the original partition in yellow.

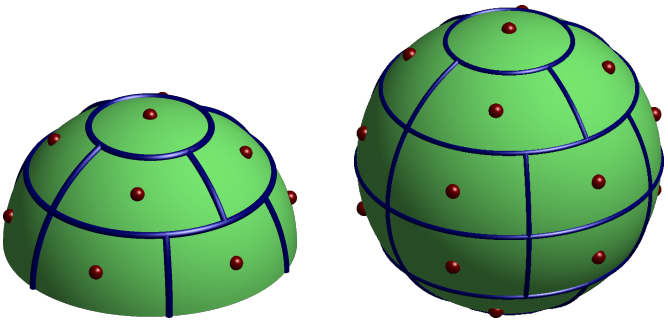
Fig. 3. Illustration of a partition of $\mathbb{H}^2$ based on a partition of $\mathbb{S}^2$ according to [15] with 22 regions.

subdivide only the upper hemisphere. If we cut the sphere in half and maintained the region borders of [15], the regions in the central collar would be cut in half, as illustrated in Fig. 3b. Thus, the regions in that collar would not be of the same size as the other regions. Further, the centers, which we use as grid points, would move (see Fig. 3b).

To prevent this issue, we adjusted the algorithm so that it yields the best even integer number of collars. Then, when subdividing from top to bottom, the boundary of one collar will run along the equator of the hypersphere. The subdivision can then be stopped to obtain a partition of $\mathbb{H}^d$ (see Fig. 4a for an example). Based on this partition, a symmetric partition of the entire hypersphere can be obtained by mirroring all grid points and regions at the center point of the full hypersphere (see Fig. 4). This yields a different partition than the one obtained before, as can be seen when comparing the partitions in Figs. 4b and 3a. If grid values describing a density were assigned to the grid points, the grid values can be copied for the mirrored points. However, all grid values then need to be divided by 2 to ensure the normalization of the density represented by the grid values. In general, determining suitable grid values for the lower hemisphere is not trivial when generating the grid on the lower hemisphere differently, e.g., by mirroring it on the plane through the equator. However, as can be seen in Fig. 4, partitions obtained by mirroring the grid on the center point of the full hypersphere can also be plane symmetric when the number of regions is even for all collars.

III. Hyperhemispherical Grid Filter

In this section, we introduce the HHGF. The approach is conceptually similar to the spherical grid filter (SGF) published in [8]. To be able to also use the filter based on models for the entire hypersphere, we start by regarding the relationship between symmetric densities on the hypersphere and densities on the hyperhemisphere. Then, we introduce how the grid-based representation can be generated for a continuous density, which is important for initializing the filter. Afterward, we explicate the update and prediction steps.



(a) Partition of the hemisphere comprising 11 regions obtained using our novel partitioning scheme.

(b) Symmetric partition of $\mathbb{S}^2$ with 22 regions obtained by mirroring the partition of the hemisphere on the center of the full sphere.

Fig. 4. Illustration showing a partition of $\mathbb{H}^2$ with equally sized regions and a partition of $\mathbb{S}^2$ obtained by mirroring the partition of the hemisphere.

A. Relationship between Symmetric Densities on $\mathbb{S}^d$ and Densities on $\mathbb{H}^d$

Symmetric densities on hyperspheres, such as densities of Bingham distributions, are normalized on the hyperspherical domain $\mathbb{S}^d$. In other words, integrating them over $\mathbb{S}^d$ always yields one. If we merely considered the upper hemisphere of a density on $\mathbb{S}^d$, it would not integrate to one. Due to the symmetry of the density, the upper and lower hemispheres contain the same probability mass, and thus, the integral would be $\frac{1}{2}$. Hence, to derive a normalized density $f^{\text{hemi}}(\underline{x})$ on the hyperhemisphere $\mathbb{H}^d$ based on the symmetric density $f^{\text{symm}}(\underline{x})$ defined on $\mathbb{S}^d$, we can use $f^{\text{hemi}}(\underline{x}) = 2f^{\text{symm}}(\underline{x})$. This will then yield a normalized density on the hyperhemisphere.

One can see that no information is lost by only considering the upper hemisphere by providing a formula for the opposite direction. A symmetric density on the entire hypersphere $f^{\text{sphere}}(\underline{x})$ can be derived from a density on the upper hemisphere $f^{\text{hemi}}(\underline{x})$ by returning $\frac{1}{2}f^{\text{hemi}}(\underline{x})$ if $\underline{x}$ is on the upper hemisphere and returning $\frac{1}{2}f^{\text{hemi}}(-\underline{x})$ otherwise. Stated more formally,

$$f^{\text{sphere}}(\underline{x}) = \begin{cases} \frac{1}{2}f^{\text{hemi}}(\underline{x}) & \text{if } \underline{x} \in \mathbb{H}^d \\ \frac{1}{2}f^{\text{hemi}}(-\underline{x}) & \text{if } \underline{x} \notin \mathbb{H}^d \end{cases} .$$

If $f^{\text{hemi}}(\underline{x})$ was generated from $f^{\text{symm}}(\underline{x})$ according to the formula in the first paragraph, then $f^{\text{sphere}}(\underline{x}) = f^{\text{symm}}(\underline{x})$. Thus, the mapping from symmetric densities on $\mathbb{S}^d$ to densities on $\mathbb{H}^d$ is invertible and a bijection.

B. Density Approximation

We first generate the grid points for the partition of the hyperhemisphere with n regions. The grid values for the first interpretation explained in Sec. II-A can be obtained by determining the probability mass in each region via an integral. By dividing each grid value by the size of the corresponding region, the grid values for the second interpretation can be obtained. These values would ensure that the interpolation that is constant in each region is normalized. However, integrals are often costly, particularly for high-dimensional manifolds. Therefore, we assume that the function we approximate is already constant. Under this assumption, we can simply evaluate the density on the grid points and use the function values as the grid values. With the number of grid points going to infinity, this grid-based approximation converges to a normalized density due to the convergence of the Riemann sum to the Riemann integral of a function. However, for a fixed number of grid points, the density described by the grid values is generally not normalized. We denote such an unnormalized density by $\breve{f}$ and the corresponding grid values by $\breve{\underline{\gamma}}$.

To obtain a normalized density, we calculate the integral of the function $\breve{f}$ described by $\breve{\underline{\gamma}}$. We use that the function is constant in each region and that the integral of a function over a domain is equal to the sum of the integrals over each element in a partition of the domain to obtain

$$\begin{aligned} \int_{\mathbb{H}^d} \breve{f}(\underline{x})\, d\underline{x} &= \sum_{i=1}^{n} \int_{A_i} \breve{f}(\underline{x})\, d\underline{x} = \sum_{i=1}^{n} \breve{\gamma}_i \int_{A_i} 1\, dx \\ &= \frac{|\mathbb{H}^d|}{n} \sum_{i=1}^{n} \breve{\gamma}_i = |\mathbb{H}^d| \text{mean}(\breve{\underline{\gamma}}) \ , \end{aligned} \tag{1}$$

with $|\mathbb{H}^d|$ being the size of the manifold (e.g., 2π for $\mathbb{H}^2$ or $2\pi^2$ for $\mathbb{H}^3$). By scaling all grid points, the entire function can be scaled. Via $\underline{\gamma} = \breve{\underline{\gamma}}/(|\mathbb{H}^d| \text{mean}(\breve{\underline{\gamma}}))$, a new vector $\underline{\gamma}$ is obtained that ensures that the integral over the function it represents is equal to one. If the mean of $\breve{\underline{\gamma}}$ is zero, the normalization step will fail. However, in this case, the grid values do not contain any usable information about the density, and the user should consider using a higher grid resolution to resolve this issue.

C. Update Step

For the update step at time step t, we require a measurement model in the form of a likelihood function $f_t^{\text{L}}(\hat{\underline{z}}_t|\underline{x}_t)$ that provides the probability density that the specific measurement $\hat{\underline{z}}_t$ is obtained when the actual state is $\underline{x}_t$. A conversion to a likelihood function is required when the measurement model is given as an equation $\underline{\boldsymbol{z}}_t = \underline{h}(\underline{\boldsymbol{x}}_t, \underline{\boldsymbol{v}}_t)$, which involves the random variables describing the measurement $\underline{\boldsymbol{z}}_t$, the state $\underline{\boldsymbol{x}}_t$, and the measurement noise $\underline{\boldsymbol{v}}_t$ at time step t. Note that only the sample space of $\underline{\boldsymbol{x}}_t$ needs to be $\mathbb{H}^d$. The sample space of the measurement and the measurement noise can be arbitrary (e.g., $\mathbb{R}^u$ or $\mathbb{S}^u$ $(u \in \mathbb{N}\backslash 0)$). Thus, as long as a formula for the likelihood can be provided, arbitrary measurement noises, even ones that are defined on other manifolds, are supported. Since the likelihood is considered for a fixed measurement $\hat{\underline{z}}_t$ in every time step, the likelihood can be seen as a function of $\underline{x}_t$ that maps from $\mathbb{H}^d$ to $\mathbb{R}_0^+$. Because the variable we condition on changes, the function is not a density in general.

Based on the likelihood, Bayes' rule allows us to derive the posterior density $f_t^{\text{e}}(\underline{x}_t|\hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t)$ that incorporates the information of all measurements until time step t. The posterior density is proportional to $(\propto)$ the unnormalized posterior $\breve{f}_t^{\text{e}}(\underline{x}_t|\hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t)$, which is obtained by multiplying the likelihood $f_t^{\text{L}}(\hat{\underline{z}}_t|\underline{x}_t)$ with the prior density $f_t^{\text{p}}(\underline{x}_t|\hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_{t-1})$ that only considers the measurements until time step $t-1$, i.e.,

$$f_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t) = \frac{f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t) f_t^{\mathrm{p}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})}{\int_{\mathbb{H}^d} f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t) f_t^{\mathrm{p}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})\,d\underline{x}_t} \propto \underbrace{f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t) f_t^{\mathrm{p}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})}_{\breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)} .$$

To provide an update step for our novel filter, we thus need to be able to multiply two densities and normalize the result afterward. With $\underline{\gamma}_t^{\mathrm{p}}$ describing the grid values (i.e., the function values of the prior density at the grid points) at time step t, we obtain the function values of the unnormalized posterior density at the grid points by multiply the grid values with the function values of the likelihood at the grid points. Thus, we obtain the formula

$$\breve{\underline{\gamma}}_t^{\mathrm{e}} = \left[\gamma_{t,1}^{\mathrm{p}} f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{\beta}_1),\quad \cdots,\quad \gamma_{t,n}^{\mathrm{p}} f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{\beta}_n)\right] .$$

The normalization can be implemented as described in Sec. III-B. It should be noted that while the values in $\breve{\underline{\gamma}}_t^{\mathrm{e}}$ correspond to the actual function values of $\breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)$, the normalized values are not identical to the values of the true normalized posterior density. The reason for this is that the probability mass may not be distributed as assumed in our normalization step. Thus, the grid values are only an approximation of the function values of the true posterior density. However, the grid values at least describe a normalized density.

For the multiplication, the likelihood function has to be evaluated n times. Both the multiplication and normalization have a complexity in $O(n)$, and thus, the overall complexity of the update step is in $O(n)$.

D. Prediction Step

For the prediction step, we require the system model in the form of a transition density $f_t^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t)$. A formula for the transition density has to be derived when we have a model $\underline{\boldsymbol{x}}_{t+1} = \underline{a}(\underline{\boldsymbol{x}}_t, \underline{\boldsymbol{w}}_t)$, which involves the random variables describing the state $\underline{\boldsymbol{x}}_t$ and the system noise $\underline{\boldsymbol{w}}_t$ at time step t and the state $\underline{\boldsymbol{x}}_{t+1}$ at time step $t+1$. While the sample space of $\underline{\boldsymbol{x}}_{t+1}$ and $\underline{\boldsymbol{x}}_t$ is $\mathbb{H}^d$, the sample space of the noise can be arbitrary. Thus, as long as we can compute the transition density, the system noise may have an arbitrary distribution on an arbitrary domain.

For our filter, we see the transition density as a function of $\underline{x}_{t+1}$ and $\underline{x}_t$ that maps from $\mathbb{H}^d \times \mathbb{H}^d$ to $\mathbb{R}_0^+$. To obtain a partition of $\mathbb{H}^d \times \mathbb{H}^d$, we generate the Cartesian product of a partition of $\mathbb{H}^d$ with itself. We evaluate the transition density on the Cartesian product of the grid points and store the values in the matrix $\boldsymbol{\Gamma}_t^{\mathrm{T}}$. The entry in the $i$row and jth column of $\boldsymbol{\Gamma}_t^{\mathrm{T}}$ is set to $f_t^{\mathrm{T}}(\underline{\beta}_i|\underline{\beta}_j)$.

Using the Chapman–Kolmogorov equation, the prior density $f_{t+1}^{\mathrm{p}}(\underline{x}_{t+1}|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)$ for the next time step $t+1$ can be provided based on the transition density and the posterior density at time step t. We subdivide the formula on the right-hand side of

$$f_{t+1}^{\mathrm{p}}(\underline{x}_{t+1}|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t) = \int_{\mathbb{H}^d} \underbrace{f_t^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t) f_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)}_{f_t^{\mathrm{j}}(\underline{x}_{t+1},\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)} d\underline{x}_t$$

into two steps. The first is to determine the joint density $f_t^{\mathrm{j}}(\underline{x}_{t+1},\underline{x}_t|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)$ of the states at the current and next time step. The second step is to marginalize out $\underline{x}_t$ to obtain the prior density for the time step $t+1$.

We now consider how to implement these two steps in our grid-based filter. The operations presented in this subsection can only be employed if the grid used for the transition density is the Cartesian product of the grid used for the posterior density. First, we determine the matrix of grid values $\boldsymbol{\Gamma}_t^{\mathrm{j}}$ for the joint density, which contains the function values of $f_t^{\mathrm{j}}(\underline{\beta}_i,\underline{\beta}_j|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)$ for all combinations of grid points $\underline{\beta}_i$ and $\underline{\beta}_j$. We denote the jth column of $\boldsymbol{\Gamma}_t^{\mathrm{j}}$, which describes $f_t^{\mathrm{T}}(\underline{x}_{t+1}|\underline{\beta}_j)$, by $\underline{\gamma}_{t,[:,j]}^{\mathrm{T}}$. Using this notation, $\boldsymbol{\Gamma}_t^{\mathrm{j}}$ can be determined according to

$$\boldsymbol{\Gamma}_t^{\mathrm{j}} = \left[\underline{\gamma}_{t,[:,1]}^{\mathrm{T}}\gamma_{t,1}^{\mathrm{e}},\quad \cdots,\quad \underline{\gamma}_{t,[:,n]}^{\mathrm{T}}\gamma_{t,n}^{\mathrm{e}}\right] .$$

To perform the marginalization, we use the formula for the integral (1) for every grid point on $\underline{x}_{t+1}$ to obtain

$$\forall i \in \{1,\ldots,n\} : \gamma_{t+1,i}^{\mathrm{p}} = |\mathbb{H}^d| \operatorname{mean}\left(\underline{\gamma}_{t,[i,:]}^{\mathrm{j}}\right)$$

as the formula for the grid values of the prior density for the next time step.

While subdividing the formula in the Chapman–Kolmogorov equation into two steps allowed for an easy explanation, better computational performance can be achieved by combining the two steps. To combine the operations, we start by writing out the formula for the mean

$$\gamma_{t+1,i}^{\mathrm{p}} = \frac{|\mathbb{H}^d|}{n} \sum_{j=1}^{n} \gamma_{t,[i,j]}^{\mathrm{j}} ,$$

in which $\gamma_{t,[i,j]}^{\mathrm{j}}$ denotes the element in the ith row and jth column of $\boldsymbol{\Gamma}_t^{\mathrm{j}}$. Then, we write $\gamma_{t,[i,j]}^{\mathrm{j}}$ as the product of the grid values of the transition density $\gamma_{t,[i,j]}^{\mathrm{T}}$ and of the posterior density $\gamma_{t,j}^{\mathrm{e}}$ to obtain

$$\gamma_{t+1,i}^{\mathrm{p}} = \frac{|\mathbb{H}^d|}{n} \sum_{j=1}^{n} \gamma_{t,[i,j]}^{\mathrm{T}} \gamma_{t,j}^{\mathrm{e}} .$$

From this formula, we can see that all grid values of the predicted density can be determined using the scaled matrix–vector product

$$\underline{\gamma}_{t+1}^{\mathrm{p}} = \tfrac{|\mathbb{H}^d|}{n} \boldsymbol{\Gamma}_t^{\mathrm{T}} \underline{\gamma}_t^{\mathrm{e}} .$$

The transition density has to be evaluated n^2 times to generate the matrix $\boldsymbol{\Gamma}_t^{\mathrm{T}}$. However, if the transition density is time invariant, the matrix can be determined once in advance and then used in all time steps. However, the computational complexity of the prediction step is always in $O(n^2)$ due to the matrix–vector multiplication.

IV. Evaluation

In our evaluation, we considered a two-dimensional and a three-dimensional scenario. We compare our implementation of the HHGF, which is available in the GitHub repository of [16], with filters for the entire hypersphere. In the version implemented in [16], the particle filter (PF) supports arbitrary-dimensional hyperspheres. The Bingham filter (BF) does not support two-dimensional scenarios, and thus, we only consider it for the three-dimensional scenario. While the SGF was presented only for $\mathbb{S}^2$ in [8], we have generalized it to arbitrary dimensions for the evaluation in this paper. We refer to the generalized version of the SGF as hyperspherical grid filter (HSGF).

In the first subsection, we describe the scenarios that we consider in our evaluation. The second subsection deals with how we quantify the quality of the filter results. In the last subsection, we provide and discuss the evaluation results.

A. Scenario Descriptions

For both scenarios, we simulated the system behavior from time step 1, the initialization, to time step 10. In each of the 10 time steps, a measurement is obtained. The system model was applied 9 times to get from time step 1 to 10. For both scenarios, we used initial prior densities, transition densities, and likelihood functions that are unimodal on the hyperhemisphere and bimodal on the hypersphere. While it is not guaranteed that the true prior and posterior densities are unimodal (or bimodal) in all time steps, they are in most cases. This facilitates providing point estimates and evaluating the filter results, as explained in the next subsection.

For the PF, SGF, and HSGF, models involving symmetric densities on $\mathbb{S}^d$ were used. The models ensure that all true prior and posterior densities are symmetric. For the HHGF, the corresponding densities for the hyperhemisphere were derived as described in Sec. III-A.

In our description, we provide the models for $\mathbb{S}^d$. The models used for the two scenarios are similar. Hence, we describe both of them at once and highlight the differences. All uncertainties are distributed according to mixtures of von Mises–Fisher (VMF) distributions [17, Sec. 9.3.2] with two components, which are weighted 0.5 each. The mean direction of the second component is at the antipode of the mean direction of the first component. The concentration parameter is identical for both components. Thus, the mixture is antipodally symmetric. The system and measurement models can be seen as simple identity models with a perturbation according to a rotationally symmetric noise term.

The initial prior density is

$$f_1^{\mathrm{p}}(\underline{x}_1) = \frac{1}{2}\big(\mathcal{VMF}(\underline{x}_1; \underline{\mu}_1, 10) + \mathcal{VMF}(\underline{x}_1; -\underline{\mu}_1, 10)\big) \ ,$$

in which the concentration parameter of the VMF distributions is 10 and $\underline{\mu}_1$ is $[0,0,1]^\top$ in the two-dimensional scenario and $[0,0,0,1]^\top$ in the three-dimensional scenario. The measurement noise is also an antipodally symmetric mixture of two VMF distributions with concentration parameter 10. The corresponding likelihood function is

$$f_t^{\mathrm{L}}(\underline{z}_t|\underline{x}_t) = \frac{1}{2}\big(\mathcal{VMF}(\underline{z}_t; \underline{x}_t, 10) + \mathcal{VMF}(\underline{z}_t; -\underline{x}_t, 10)\big) \ .$$

The system noise is distributed according to the same parameters. Thus, the transition density is

$$\begin{aligned} f_t^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t) = \frac{1}{2}\big(&\mathcal{VMF}(\underline{x}_{t+1}; \underline{x}_t, 10) \\ &+ \mathcal{VMF}(\underline{x}_{t+1}; -\underline{x}_t, 10)\big) \ . \end{aligned}$$

To obtain Bingham distributions for all uncertainties for the BF, we drew 100 000 samples from the individual distributions and fitted Bingham distributions to them.

B. Estimates and Error Metric

The mean direction [17, Sec. 9.2.1], which is frequently used to obtain point estimates on $\mathbb{S}^d$, is undefined for antipodally symmetric densities. To obtain point estimates from the results of the filters, we start by converting all filter results into Bingham distributions. For the PF, we fit the samples with a Bingham distribution. Similarly, we interpret the grid points of the SGF and HSGF as samples that are weighted according to the grid values and fit them with a Bingham distribution. For the HHGF, we generate a grid for the entire hypersphere and set the grid values as described at the end of Fig. 4a. A Bingham distribution is then fitted to the symmetric result.

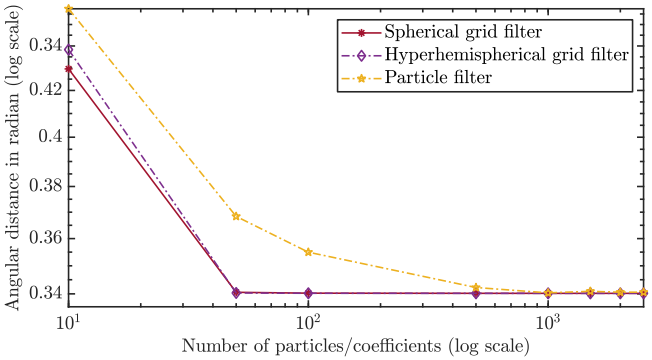
Then, we determine one of the modes of the Bingham distribution (it does not matter which) and use it as the point estimate $\hat{\underline{x}}_{10}$. To quantify the estimation error, we determine the angular distance between the estimate and the true state $\tilde{\underline{x}}_{10}$ and the angular distance between the antipode of the estimate (the second mode of the Bingham distribution) and the true state. The minimum of the two angular distances is then our error metric d. More formally, the error metric is defined as

$$d(\hat{\underline{x}}_{10}, \tilde{\underline{x}}_{10}) = \min(\operatorname{acos}(\hat{\underline{x}}_{10}^\top \tilde{\underline{x}}_{10}), \operatorname{acos}(-\hat{\underline{x}}_{10}^\top \tilde{\underline{x}}_{10})) \ .$$

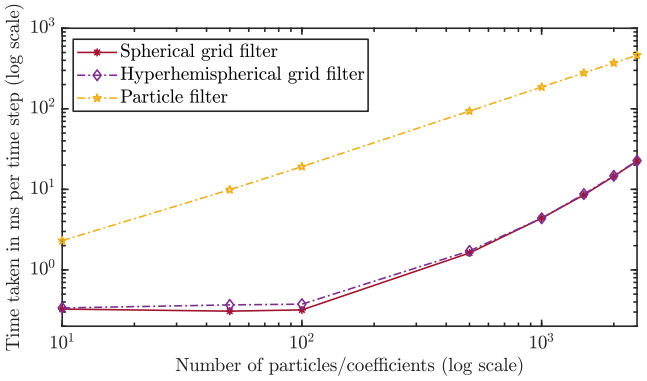
C. Evaluation Results

To obtain reliable results, we determined the errors and run times for 1000 runs and calculated the means. The mean of the errors are given for different numbers of parameters (using the term parameter for both grid points and particles) in Fig. 5b for the first scenario and in Fig. 6b for the second scenario. The corresponding average run times are depicted in Figs. 5a and 6a. The run times were measured on a laptop with an Intel Core i7-7500U CPU and 16 GB of RAM running Matlab 2020a on Windows 10. The run times for operations that only have to be performed once in the beginning, such as determining the grid values for the transition density or sampling from the initial prior density, were not included.

In both scenarios, the HHGF performed best. Compared on a per parameter basis, the HHGF achieved better results than the PF, SGF, and HSGF in almost all configurations. For equal numbers of parameters, the SGF was better than the PF in the two-dimensional scenario. In the three-dimensional scenario, the PF performed better than the HSGF for low numbers of

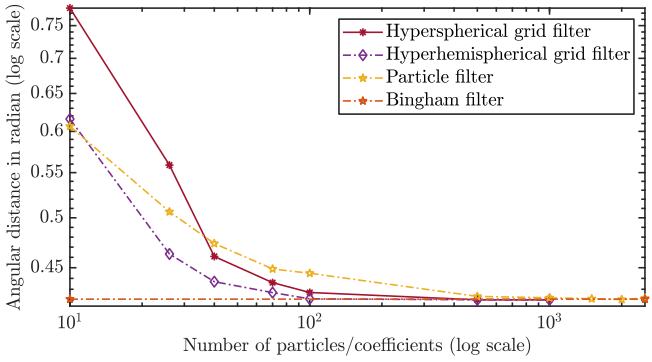


(a) Error over number of grid points or particles.

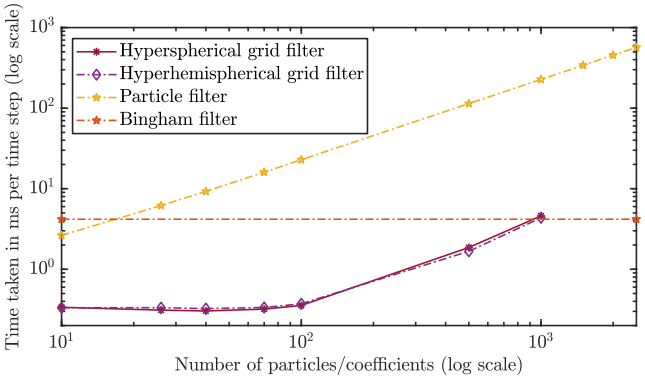


(b) Time over number of grid points or particles.

Fig. 5. Evaluation results for the two-dimensional scenario.



(a) Error over number of grid points or particles.



(b) Time over number of grid points or particles.

Fig. 6. Evaluation results for the three-dimensional scenario. The results of the BF are shown as a straight line because the number of parameters cannot be changed.

parameters. However, the HSGF converged faster to its optimal performance. Since the system and measurement models are identity models and the noise densities can be approximated well using densities of a Bingham distribution, the BF (which uses a fixed number of parameters) performs very well in the three-dimensional scenario. While the grid filters need 500 grid points to surpass the estimation quality of the BF, the PF does not achieve a better estimation quality in the configurations considered.

At some point, the filters stop improving. A probable cause is that at this point, the filters yield almost the same estimate as would be obtained based on the true posterior density. The speed of convergence to the optimal result was similar for the SGF, HSGF, and HHGF. However, the SGF and HSGF required more grid points than the HHGF to achieve a comparable accuracy. We believe this is a result of the worse placement of the grid points. If the grid generated by the equal area partitioning used in the SGF or HSGF is symmetric, half of the points are wasted. One reason for this is that if a grid includes both a certain point and its antipode, the grid values are supposed to be identical for the two points. Further, even if a grid point is not redundant, its placement may be suboptimal. To visualize this, one can mirror all the grid points that were placed on the lower hemisphere onto the upper hemisphere. The grid points on the upper hemisphere will not be as evenly distributed as the grid points used by the HHGF.

The run times of the SGF and HSGF were similar to those of the HHGF. The run times of the grid filters were consistently lower than those of the PF. While a renormalization step is required in the update step of the grid filters, the PF needs to perform a resampling step, which is more expensive. Further, the PF needs to draw one sample of the noise density per particle in the prediction step, which is particularly expensive. Due to this expensive operation, the PF is relatively slow even though its run time increases only linearly in the number of particles. While the matrix–vector multiplication required for the grid filters results in a quadratic increase in the run time with increasing number of grid points, the operation can be performed so efficiently that the grid filters are faster than the PF in the configurations considered. While the grid filters were faster than the BF in most of the considered configurations, the PF was slower when using 50 or more particles.

By combining the errors for different numbers of parameters with the respective run times, one can compare configurations with comparable run times. When taking both the estimation quality and run time into account, the HHGF is superior to all the other filters. In the three-dimensional scenario, the grid filters surpass the estimation quality of the BF at a lower run time than the BF. Further, when comparing the estimation quality at comparable run times, both grid filters are better than the BF. However, the BF performed better than the PF

because the PF takes much longer to reach an estimation quality comparable to that of the BF. When comparing configurations with equal numbers of parameters, the HSGF performed worse than the PF for low numbers of parameters in the three-dimensional scenario. However, the HSGF is considerably faster than the PF. When also considering the run times, the SGF and HSGF perform better than the PF, which is the worst filter in both scenarios.

V. Conclusion and Outlook

In this paper, we presented a novel filter for hyperhemispheres. The filter can be used for almost arbitrary estimation problems involving antipodally symmetric densities on hyperspheres. For example, it can be used for estimating the orientation of an object in three-dimensional space based on unit quaternions. In two evaluation scenarios, the filter has not only proven to be accurate but also fast. While the prediction step is in $O(n^2)$, the most expensive operation is a matrix–vector multiplication, for which highly efficient routines exist. The update step is always in $O(n)$. In the evaluation scenarios, the HHGF was both faster and more accurate than the PF in configurations with equal numbers of parameters. Further, the HHGF surpassed the estimation quality of the BF at lower run times. Because the placement of the grid points takes advantage of the symmetry of the densities, the HHGF required fewer grid points and thus less run time than the SGF and HSGF to achieve an estimation quality that is close to its optimum.

While providing a continuous density based on the grid values is never required in the actual filter, providing interpolations of even higher quality (i.e., ones that are closer to the true density) could be a subject of future work. A possible approach could involve the use of hyperspherical harmonics. After generating a symmetric grid on the entire hypersphere based on the grid on the hyperhemisphere (as explained in Sec. II-B), the grid values could be used to determine the hyperspherical harmonic coefficients. The resulting smooth interpolation has the potential to be closer to the true density than the discontinuous interpolation that yields a function that is constant in each region.

Another subject for future work would be to extend the filter to increase its utility for practical applications. For example, the HHGF could be used as the basis for a filter for $\mathrm{SE}(3)$ that uses Rao–Blackwellization. This filter would be conceptually similar to the filter proposed for $\mathrm{SE}(2)$ in [18]. Finally, the Rao–Blackwellized filter could be generalized to support arbitrary Cartesian products of bounded and linear domains. This would, e.g., allow for integrating angular velocities (which can be measured well using gyroscopes) into the state vector.

Acknowledgment

This work is supported by the German Research Foundation (DFG) under grant HA 3789/16-1.

References

[1] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[2] A. Barrau and S. Bonnabel, "Invariant Kalman Filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018.

[3] T. D. Barfoot and P. T. Furgale, "Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[4] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, 2016.

[5] G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck, "Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 90–105, Dec. 2014.

[6] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Unscented Orientation Estimation Based on the Bingham Distribution," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 172–177, Jan. 2016.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents Series. The MIT Press, 2005.

[8] F. Pfaff, K. Li, and U. D. Hanebeck, "The Spherical Grid Filter for Nonlinear Estimation on the Unit Sphere (to appear)," in *Proceedings of the 1st Virtual IFAC World Congress (IFAC-V 2020)*, Jul. 2020.

[9] K. Li, F. Pfaff, and U. D. Hanebeck, "Grid-Based Quaternion Filter for SO(3) Estimation," in *Proceedings of the 2020 European Control Conference (ECC 2020)*, Virtual, May 2020.

[10] W. M. Wonham, "Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 2, no. 3, pp. 347–369, 1964.

[11] G. Kurz, F. Pfaff, and U. D. Hanebeck, "Discrete Recursive Bayesian Filtering on Intervals and the Unit Circle," in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.

[12] F. Pfaff, K. Li, and U. D. Hanebeck, "Fourier Filters, Grid Filters, and the Fourier-Interpreted Grid Filter," in *Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019)*, Ottawa, Canada, Jul. 2019.

[13] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[14] Y. Lu and H. H. Zhou, "Statistical and Computational Guarantees of Lloyd's Algorithm and Its Variants," *arXiv preprint arXiv:1612.02099*, 2016.

[15] P. Leopardi, "A Partition of the Unit Sphere into Regions of Equal Area and Small Diameter," *Electronic Transactions on Numerical Analysis*, vol. 25, no. 12, pp. 309–327, 2006.

[16] G. Kurz, I. Gilitschenski, F. Pfaff, L. Drude, U. D. Hanebeck, R. Haeb-Umbach, and R. Y. Siegwart, "Directional Statistics and Filtering Using libDirectional," *Journal of Statistical Software*, May 2019.

[17] K. V. Mardia and P. E. Jupp, *Directional Statistics*. John Wiley & Sons, 1999.

[18] G. Kurz, F. Pfaff, and U. D. Hanebeck, "Application of Discrete Recursive Bayesian Estimation on Intervals and the Unit Circle to Filtering on SE(2)," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1197–1206, Mar. 2018.