Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

The Spherical Grid Filter for Nonlinear Estimation on the Unit Sphere

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: Filters for the unit sphere have to consider its inherent periodic nature. Since the unit sphere is a domain of finite size, suitable grids covering the manifold can be provided. We explain the considerations for the grid generation and provide efficient ways to implement the prediction and update steps of a novel grid filter for this manifold. The filter supports nonlinear system and measurement models in the form of transition densities and likelihoods. In the evaluation, the proposed filter achieves a higher estimation accuracy than competing approaches.

1. INTRODUCTION

Estimation problems on periodic manifolds, such as the circle, the torus, and the sphere, are inherently nonlinear and thus nontrivial. While nonlinear estimation is a large field of research in which no universal approach that is both accurate and fast is known, more and more problems can be solved with success using approaches tailored to the specific problems. In this paper, we consider estimation on the unit sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$, which is the surface of the unit ball in $\mathbb{R}^3$. Modeling uncertainties on spherical domains is of interest, e.g., in geosciences (see, e.g., Watson (1956); Mardia (1981)) and for determining crystal orientations, for example in Chen et al. (2015). Tracking problems on spherical domains arise, e.g., when estimating the orientation of objects for which the roll angle in a roll, pitch, and yaw representation is irrelevant or cannot be determined. Such tracking tasks may involve tracking the direction certain types of antennas are facing or the orientation of rotationally symmetric objects, such as a spear. Other tracking tasks with this topology arise in cases in which only a beam direction is relevant or can be determined, such as in speaker tracking (see Traa and Smaragdis (2014)).

Some existing filters for the unit sphere are based on density assumptions, such as the von Mises–Fisher filter (VMFF) proposed in Chiuso and Picci (1998), of which a nonlinear variant was proposed in Kurz et al. (2016a). An approach that does not depend on a density assumption is the particle filter (PF). While the PF has not been explicitly proposed for the unit sphere, its application to such problems is trivial, and an implementation was released as part of libDirectional (see Kurz et al. (2019)). Another approach is the spherical harmonics filter that was recently proposed in Pfaff et al. (2017). This filter is based on orthogonal basis functions on the unit sphere called spherical harmonics. While this filter often yields better results than the PF, it does not support arbitrary system models in the prediction step.

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

A vastly different approach that is popular for bounded regions in Euclidean spaces is the use of grid filters, as explained in Thrun et al. (2005, Ch. 8). Grid-based filters for the unit circle were regarded in Kurz et al. (2016b); Pfaff et al. (2019). Adapting such filters to the unit sphere is easier and leads to a more efficient filter than adding support for arbitrary transition densities to the spherical harmonics filter. Therefore, we propose a filter for the unit sphere that adopts the ideas of Pfaff et al. (2019). We call it the spherical grid filter (SGF).

The paper is structured as follows. We begin by describing how densities are represented using grid values in the second section. In the third section, we explain how the prediction and filter steps can be implemented for densities in the grid-based representation. In Sec. 4, we compare the SGF with the PF and the VMFF. In the last section, we provide a conclusion and an outlook.

2. DENSITY REPRESENTATIONS

When approximating densities using a grid, one has to decide on the semantic of the n values assigned to the n grid points. In the first subsection of this section, we introduce two closely related interpretations that facilitate the comprehension of the derivations and the validity of the filter. In the second subsection, we introduce our method to represent densities on the unit sphere. In the third subsection, we explain how we represent joint and conditional densities on the unit sphere. This will be essential for the prediction step of our novel filter.

2.1 Interpretations of the Grid Values

For both interpretations, each grid value γ_i can be thought of as carrying information about the probability density function (pdf) in a certain region A_i. By specifying that the union of the regions must cover the entire sphere without any overlapping regions [1], we obtain a partition

[1] Neighboring patches may share points on the boundaries. We do not go into detail on this because the probability mass contained in the boundaries is zero for commonly used densities.

Table 1. Properties of the Two Representations

#	Describes	Domain	Normalization	Convert
1	pmf	Discrete	Sum to 1	Divide by $\lvert A_i \rvert$
2	pdf	Continuous (interpolated)	Interpolation integrates to 1	Multiply with $\lvert A_i \rvert$

$\mathcal{A} = \{A_1, \ldots, A_n\}$ of the domain. One point in each patch is used as the corresponding grid point $\underline{\beta}_i$.

In the first interpretation, the grid value γ_i describes the probability mass in A_i. The value can be obtained by calculating an integral, which is a two-dimensional integral for the unit sphere. The location of $\underline{\beta}_i$ then essentially loses its meaning and the resulting filter resembles a Wonham filter—introduced in Wonham (1964)—in which the individual patches correspond to different states. The vector of grid values $\underline{\gamma} = [\gamma_1, \ldots, \gamma_n]^\top$ can then be interpreted as the list of all probability masses of a probability mass function (pmf). To obtain a pdf based on the grid values, each probability mass γ_i can be distributed evenly in the corresponding patch A_i, leading to a piecewise constant function (see Kurz et al. (2016b)). The function value in each patch corresponds to the grid value divided by the area of the respective patch.

In the second interpretation, which we use for our implementation, the function values of the density at the grid points are used as the grid values. Instead of calculating the integral of the probability density in A_i, we pick one point $\underline{\beta}_i \in A_i$ and assume that the function is constant with a function value of $\gamma_i = f(\underline{\beta}_i)$ in the entire patch. This is reminiscent of the idea behind Riemann sums that are used for numerically approximating (Riemann) integrals. While the approximation based on the grid values is not necessarily normalized, it converges to a normalized density for $n \to \infty$ due to the convergence of Riemann sums to the Riemann integral. By multiplying a grid value γ_i for the second interpretation with the corresponding patch size $|A_i|$, an approximation for the grid value for the first interpretation can be obtained (an approximation is introduced due to the assumption that the function is constant in the patch). The properties of both representations, including ways to convert from one representation to the other, are summarized in Table 1.

For the second interpretation, one way to obtain a continuous density is to use the assumption that the function is constant in the individual patches. However, the grid values can also be used to obtain a different approximation of the original density. Such an approximation preferably converges to the true density for $n \to \infty$. On the unit circle, the smooth interpolation used in Pfaff et al. (2019) based on trigonometric polynomials (i.e., a Fourier series with a finite number of nonzero coefficients) converged much faster than the approximation based on piecewise constant functions used in Kurz et al. (2016b). Since we use the function values at the grid points for this interpretation, their precise locations are evidently relevant. A reasonable choice is to use the center of A_i as $\underline{\beta}_i$.

An important choice is how the domain is partitioned. To obtain representative grid points, it is reasonable to choose a partition that ensures that the boundaries of each patch are reasonably close to a well-chosen corresponding grid point. The filter would also work for patches that are, e.g., long and thin. However, the information derived from the filter may be less useful for the user and point estimates may have higher mean squared errors. As an intuitive example, consider partitioning the unit sphere into patches that are limited only by two azimuth angles and no elevation angles. Then, a high grid value indicates a high probability of being anywhere in the range of the two azimuth angles but it does not provide information about the elevation angle. Besides their shape, the sizes of the patches are of interest. In this paper, we focus on partitions with equally sized patches. Partitions with higher concentrations of grid points in areas of high probability density may be beneficial if suitable prior knowledge is available. However, such grids put an emphasis on existing information and may not work well if, e.g., unexpected measurements occur.

For circles, generating an equidistant grid is trivial. By taking the Cartesian product of the grid points for multiple circles, a grid for hypertoroidal manifolds can be easily generated. As we explain in the next subsection, obtaining a suitable grid is more complicated for the unit sphere.

Remark 1. Partitioning the domain into patches is not essential for the second interpretation when the grid values are used as data points for an interpolation. When interpolating the grid values using orthogonal basis functions, each value can influence all basis functions. Since the basis functions may have a global effect, changing a single grid value might change the entire function, and thus, the grid value does not only affect its neighborhood. However, since generating piecewise constant functions based on the grid values facilitates deriving valid operations for the prediction and update steps, this piecewise constant interpolation will be our focus in the derivation of the filter.

2.2 Representing Densities on the Sphere

Grids on the unit sphere that are generated similar to the naïve grid explained for the hypertorus are equiangular grids (see Colombo (1981)). To generate an equiangular grid, one considers spherical coordinates and takes the Cartesian product of the grid points of two equidistant grids on the azimuth and elevation angles. Due to the differences in the lengths of the circles around the sphere for different elevation angles, the grid points are spaced more densely toward the poles. An advantage of using this grid would be that a fast spherical harmonics transform exists (see Healy et al. (2003)), which is in $O(n^2(\log n)^2)$ for n grid points.

For the reasons mentioned before, we use equally sized patches. Grids based on such partitions also simplify the derivations and the formulae for the prediction and update steps. Partitioning a sphere into patches of equal size is a task with various degrees of freedom and there is no single established approach. One possible approach would be to use a centroidal Voronoi tessellation obtained, e.g., using an adapted version of the algorithm in Lloyd (1982).

In our implementation, we use the fast and efficient recursive zonal equal area (EQ) partitioning algorithm proposed in Leopardi (2006) to obtain equally sized patches. In the partition illustrated in Fig. 1, there are two patches

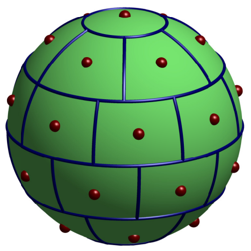


Fig. 1. Illustration of an EQ-based partition with 30 patches. The patches are limited by the blue lines and the grid points are shown in red. The graphic was generated using code by Leopardi (2006).

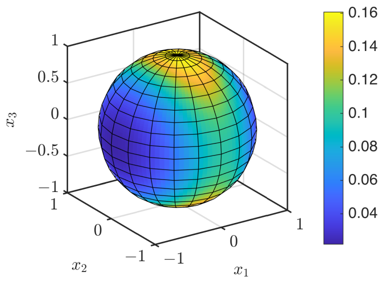
Algorithm 1: Obtaining a Grid-Based Representation

```
Input: Density $f$, desired number of grid points $n$
Output: Grid values $\underline{\gamma}$, grid points $\mathbf{B}$
$\mathcal{A} \leftarrow$ GetEQPartitionS2($n$);
$\mathbf{B} \leftarrow$ GetCenters($\mathcal{A}$);
for $i \leftarrow 1$ to $n$ do
    $\gamma_i \leftarrow f(\underline{\beta}_i)$;            // Store values in $\underline{\gamma}$
end
```

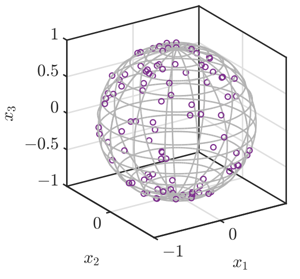
(one on the bottom and one at the top of the sphere) that are limited by only an elevation angle. The other patches are limited by two elevation and two azimuth angles. For each of these patches, there are four outermost points, which are connected by curved lines following the surface of the sphere. Depending on the coordinate system used, the grid points (in our case the centers of the patches) can be stored as either two- or three-dimensional vectors. In our implementation, we stack the 3×1 column vectors in Cartesian coordinates horizontally, yielding a $3 \times n$ matrix $\mathbf{B} = [\underline{\beta}_1, \ldots, \underline{\beta}_n]$ for n grid points. The grid values are stored in an $n \times 1$ vector $\underline{\gamma}$ comprising the function values at the grid points. Pseudocode for obtaining a grid-based representation of a density is provided in Algorithm 1. The partition $\mathcal{A}$ is not stored in our implementation but will be used in the derivation of the filter.

Since continuous pdfs are not required to perform prediction and update steps, we can consider the problem of interpolating the grid values separately from the actual filter. Thus, while we will consider the function to be constant in each patch in our explanation of the prediction and update steps, we interpolate the function values on the grid using spherical harmonics in our implementation. This can be interpreted as an adaption of the idea described (for the unit circle) in Pfaff et al. (2019) to the unit sphere. When directly calculating the spherical harmonic coefficients based on the function values, the interpolation may take on negative function values. To avoid this, we first take the square roots of the grid values. This is possible since the function values of the initial prior are always nonnegative and none of the operations in the prediction and update steps voids their nonnegativity. Then, we interpolate the square roots using spherical harmonics. By squaring the values of the interpolation, the approximation of the actual density is nonnegative everywhere.

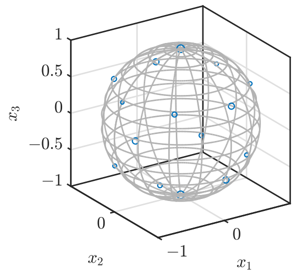
Example 2. In Fig. 2, we provide an example that illustrates different representations and the interpolation of the grid values using spherical harmonics. In Fig. 2a, we



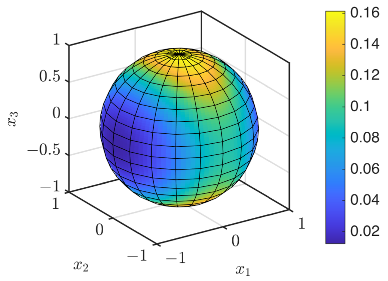
(a) Original density of the Bingham distribution.



(b) Representation of the PF with equally weighted samples.



(c) Representation of the SGF with 15 grid points. The different grid values are represented by the sizes of the circles.



(d) Interpolation of the grid values of the SGF.

Fig. 2. Example of a density and the corresponding representations and interpolations.

show the pdf of a Bingham distribution (see Bingham (1974)) with an identity matrix as its shape matrix $\mathbf{M}$ and a concentration matrix $\mathbf{Z}$ with -2, -0.5, and 0 on the diagonal. In Fig. 2b, we show 100 random samples of the density. This is an example of a representation that a PF with 100 samples may use. The representation used by the SGF with an EQ-based grid with 15 grid points is illustrated in Fig. 2c. An interpolation of the grid values based on spherical harmonics is shown in Fig. 2d. The interpolation clearly resembles the original density, even though only 15 grid points were used.

2.3 Representing Joint and Conditional Densities on the Sphere

The joint density $f^{\mathrm{j}}(\underline{x}_{t+1}, \underline{x}_t)$ is a function of two vectors $\underline{x}_{t+1}$ and $\underline{x}_t$, i.e., a function on $\mathbb{S}^2 \times \mathbb{S}^2$. Similarly, we consider the conditional density $f^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t)$ to be a function of $\underline{x}_{t+1}$ and $\underline{x}_t$. In our implementation, the first sphere of the Cartesian product describes $\underline{x}_{t+1}$ and the second $\underline{x}_t$. A joint density on $\mathbb{S}^2 \times \mathbb{S}^2$ integrates to 1 when integrating over both spheres simultaneously. For a conditional density $f^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t)$, $f^{\mathrm{T}}(\underline{x}_{t+1}|\tilde{\underline{x}}_t)$ describes a valid pdf on the first sphere for every fixed $\tilde{\underline{x}}_t$ on the second sphere.

To obtain a grid for $\mathbb{S}^2 \times \mathbb{S}^2$, we start by generating a grid for $\mathbb{S}^2$ using the EQ partitioning algorithm. Then, the Cartesian product of the set of grid points with itself is used as the grid for $\mathbb{S}^2 \times \mathbb{S}^2$. The function values obtained for all combinations of grid points on the first and second sphere are stored in an $n \times n$ matrix $\mathbf{\Gamma}$. We go into detail on this for conditional densities in Algorithm 2. Each column of $\mathbf{\Gamma}$ describes the grid values for all points on the first sphere for a fixed point on the second sphere. Likewise,

Algorithm 2: Obtaining a Grid-Based Representation for Conditional Densities

```
Input: Conditional density $f$, desired number of grid
       points $n$ for Cartesian product
Output: Grid values $\Gamma$, grid points $\mathbf{B}$
$\mathcal{A} \leftarrow$ GetEQPartitionS2($n$);
$\mathbf{B} \leftarrow$ GetCentroids($\mathcal{A}$);
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $n$ do
        /* Store value in $i$th column and $j$th row of $\Gamma$  */
        $\gamma_{[i,j]} \leftarrow f(\underline{\beta}_i|\underline{\beta}_j)$;
    end
end
```

each row of $\mathbf{\Gamma}$ contains the grid values for all points on the second sphere for a fixed point on the first sphere.

3. THE SPHERICAL GRID FILTER

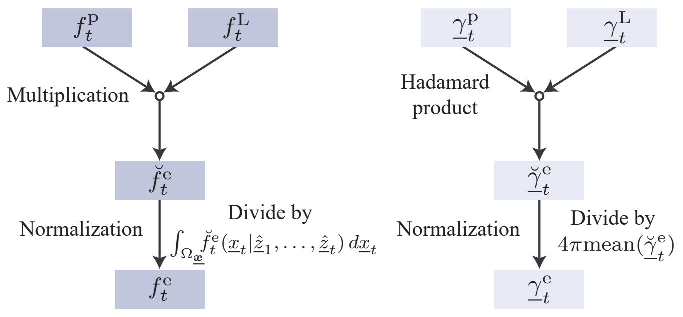
To implement the filter, we need to be able to initialize the vector of grid values (which describes all knowledge we have about the state) and perform update and prediction steps. The length n of the vector is the only parameter of the filter that the user can set freely. n determines the resolution of the grid and higher values generally result in more accurate results. Thus, n should be set as high as feasible while considering available computation power and real time constraints. As the evaluation in Sec. 4 will show, hundreds of grid points can be used at hundreds of Hertz even with an unoptimized implementation on a laptop.

In combination with the partition, the vector of grid values provides us information about the probability density on the entire domain. Unlike in the Kalman filter, this vector does not merely describe a point in the state space. To obtain an estimate in the state space, we treat the grid points as a weighted set of samples and determine their mean direction (see Jammalamadaka and Sengupta (2001, Sec. 2.1)). The vector is initialized by applying Algorithm 1 to the initial prior density f_0^{p} to obtain the vector of grid values $\underline{\gamma}_0^{\mathrm{p}}$. In the first subsection, we regard the update step. The second subsection addresses the prediction step.

3.1 Update Step

An update step can be implemented based on the likelihood function $f_t^{\mathrm{L}}(\underline{z}_t|\underline{x}_t)$ that describes the probability density of the measurement $\underline{z}_t$ when the actual state is $\underline{x}_t$. Since the update step is performed after the measurement is obtained, the measurement is usually fixed. For the fixed measurement $\hat{\underline{z}}_t$, $f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t)$ is only a function of $\underline{x}_t$ and not necessarily a pdf.

Likelihood functions can be provided even when the measurement is not on the unit sphere and instead, e.g., on $\mathbb{R}$ as in Pfaff et al. (2017). If only an equation based on random variables is available, a likelihood function has to be derived from it. For measurements on the sphere, a likelihood function can be provided when the measurement is (after applying a potentially nonlinear vector-valued measurement function $\underline{h}_t$ to the state) perturbed by a measurement noise with a zonal (i.e., rotationally symmetric around a specified axis) pdf. For example, for a von Mises–Fisher-distributed (see Mardia and Jupp (1999, Sec 9.3.2)) noise


f_t^{p}
f_t^{L}
Multiplication
$\breve{f}_t^{\mathrm{e}}$
Normalization
Divide by $\int_{\Omega_{\underline{x}}} \breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)\,d\underline{x}_t$
f_t^{e}
$\underline{\gamma}_t^{\mathrm{p}}$
$\underline{\gamma}_t^{\mathrm{L}}$
Hadamard product
$\breve{\underline{\gamma}}_t^{\mathrm{e}}$
Normalization
Divide by $4\pi\mathrm{mean}(\breve{\underline{\gamma}}_t^{\mathrm{e}})$
$\underline{\gamma}_t^{\mathrm{e}}$

(a) Operations performed on the continuous densities. (b) Operations performed on the grid values.

Fig. 3. Illustration of how the required operations for the update step are implemented in the SGF.

with concentration parameter $\kappa^{\underline{v}}$, the likelihood function is $f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t) = \mathcal{VMF}(\hat{\underline{z}}_t;\underline{h}_t(\underline{x}_t),\kappa^{\underline{v}}) = \mathcal{VMF}(\underline{h}_t(\underline{x}_t);\hat{\underline{z}}_t,\kappa^{\underline{v}})$.

For arbitrary domains, Bayes' rule can be used to describe the posterior density $f_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)$ integrating the new measurement based on the prior density before the update step $f_t^{\mathrm{p}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_{t-1})$ and the likelihood function $f_t^{\mathrm{L}}(\hat{\underline{z}}_t|\underline{x}_t)$. Using Bayes' rule, we obtain

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

in which $\Omega_{\underline{x}}$ denotes the sample space of the state and $\propto$ that the expression following the sign is only equal up to a (nonzero) scaling factor. As illustrated in Fig. 3a, we can thus obtain the posterior density by first determining the unnormalized posterior density $\breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)$, which is the product of the likelihood function and the prior density, and then normalizing the result. We now address how these two operations can be performed based on the grid values.

We start with the vector $\underline{\gamma}_t^{\mathrm{p}}$ comprising function values of the prior density. To obtain the function values of the unnormalized posterior density at the grid points, we multiply each component of the vector $\underline{\gamma}_t^{\mathrm{p}}$ with the likelihood at the corresponding grid point, i.e.,

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

If we store the values of the likelihood function at the grid points in a vector $\underline{\gamma}_t^{\mathrm{L}}$, $\breve{\underline{\gamma}}_t^{\mathrm{e}}$ is equal to the Hadamard (entry-wise) product of $\underline{\gamma}_t^{\mathrm{p}}$ and $\underline{\gamma}_t^{\mathrm{L}}$. If $\underline{\gamma}_t^{\mathrm{p}}$ comprises the actual function values of the prior density, then the values in $\breve{\underline{\gamma}}_t^{\mathrm{e}}$ accurately describe the function values of $\breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)$.

To normalize the result, we determine the integral of the unnormalized posterior density over the entire domain. Instead of integrating over the entire domain, we sum up the integrals over the patches in the partition $\mathcal{A}$, i.e.,

$$\int_{\mathbb{S}^2} \breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)\,dx = \sum_{i=1}^{n}\int_{A_i} \breve{f}_t^{\mathrm{e}}(\underline{x}_t|\hat{\underline{z}}_1,\dots,\hat{\underline{z}}_t)\,dx\ .$$

Now, we use our assumption that the function is constant in each patch. Since $\breve{\underline{\gamma}}_t^{\mathrm{e}}$ describes the function values in the patches, we obtain

Algorithm 3: Update Step of the SGF

```
Input: Current grid values $\underline{\gamma}_t^{\mathrm{p}}$, grid points $\mathbf{B}$,
       likelihood function $f_t^{\mathrm{L}}$, measurement $\underline{\hat{z}}_t$
Output: New grid values $\underline{\gamma}_t^{\mathrm{e}}$
/* Determine unnormalized grid values                    */
for $i \leftarrow 1$ to $n$ do
    $\breve{\gamma}_{t,i}^{\mathrm{e}} \leftarrow \gamma_{t,i}^{\mathrm{p}} f_t^{\mathrm{L}}(\underline{\hat{z}}_t | \underline{\beta}_i)$;
end
$\underline{\gamma}_t^{\mathrm{e}} = \frac{n}{4\pi} \underline{\breve{\gamma}}_t^{\mathrm{e}} / \sum_{i=1}^{n} (\breve{\gamma}_{t,i}^{\mathrm{e}})$;        // Normalize result
```

$$\sum_{i=1}^{n} \int_{A_i} \breve{f}_t^{\mathrm{e}}(\underline{x}_t | \underline{\hat{z}}_1, \ldots, \underline{\hat{z}}_t) \, dx = \sum_{i=1}^{n} \breve{\gamma}_{t,i}^{\mathrm{e}} \int_{A_i} 1 \, dx \; .$$

The surface area of the unit sphere is 4π and all n patches are equally sized, and thus,

$$\sum_{i=1}^{n} \breve{\gamma}_{t,i}^{\mathrm{e}} \int_{A_i} 1 \, dx = \frac{4\pi}{n} \sum_{i=1}^{n} \breve{\gamma}_{t,i}^{\mathrm{e}} = 4\pi \, \mathrm{mean}(\underline{\breve{\gamma}}_t^{\mathrm{e}}) \; . \qquad (2)$$

Thus, we can obtain a vector of grid values describing a normalized posterior density using

$$\underline{\gamma}_t^{\mathrm{e}} = \frac{1}{4\pi \, \mathrm{mean}(\underline{\breve{\gamma}}_t^{\mathrm{e}})} \underline{\breve{\gamma}}_t^{\mathrm{e}} \; .$$

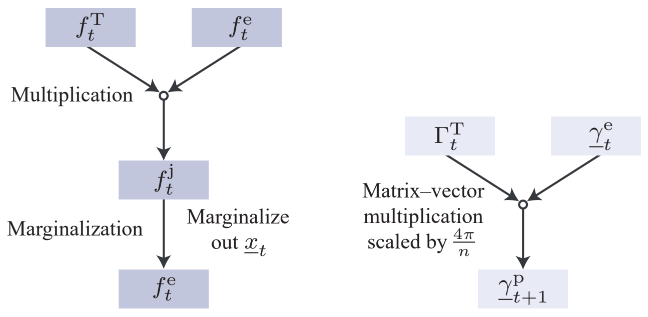
When the values of $\underline{\breve{\gamma}}_t^{\mathrm{e}}$ are all zero, the filter will fail at this point. However, in this case, the vector $\underline{\breve{\gamma}}_t^{\mathrm{e}}$ contains no usable information about the posterior density, and using a higher number of grid points should be attempted. The values in $\underline{\gamma}_t^{\mathrm{e}}$ are, in general, not equal to the function values of the true posterior density. While the utilized normalization constant ensures that the approximation is normalized, the factor may not be identical to the one that would normalize the true unnormalized posterior density. For example, if the true posterior density is highly concentrated and there are no grid points in regions in which the probability density is high, the values obtained when normalizing $\underline{\breve{\gamma}}_t^{\mathrm{e}}$ will be significantly higher than the actual function values of $f_t^{\mathrm{e}}(\underline{x}_t | \underline{\hat{z}}_1, \ldots, \underline{\hat{z}}_t)$ at the grid points.

The operations for the update step based on the grid values are illustrated in Fig. 3b and summarized in pseudocode in Algorithm 3. The computational complexity of the update step is in $O(n)$ and the likelihood function has to be evaluated n times. The presented update step can also be applied similarly for other grids. While the multiplication in (1) does not need to be adapted, changes are generally required for the normalization step as the first equality in (2) may not hold. For non-equally sized patches, the grid values need to be multiplied with the corresponding patch sizes to obtain a suitable normalization constant.

3.2 Prediction Step

For the prediction step, the system model has to be given in form of a transition density $f_t^{\mathrm{T}}(\underline{x}_{t+1} | \underline{x}_t)$ that describes the probability density of being in state $\underline{x}_{t+1}$ when the previous state was $\underline{x}_t$. Based on the transition density, the prior density for the next time step $t+1$ is given by the Chapman–Kolmogorov equation

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


f_t^{T}
f_t^{e}
Multiplication
f_t^{j}
Marginalization
Marginalize out $\underline{x}_t$
f_t^{e}
Γ_t^{T}
$\underline{\gamma}_t^{\mathrm{e}}$
Matrix–vector multiplication scaled by $\frac{4\pi}{n}$
$\underline{\gamma}_{t+1}^{\mathrm{p}}$

(a) Operations performed on the continuous densities.

(b) Operations performed on the grid values.

Fig. 4. Illustration of how the required operations for the prediction step are implemented in the SGF.

It is possible to implement the formula on the right-hand side using two successive operations, as illustrated in Fig. 4a. First, the transition density is multiplied with the posterior density of time step t. This yields a joint density $f_t^{\mathrm{j}}(\underline{x}_{t+1}, \underline{x}_t | \underline{\hat{z}}_1, \ldots, \underline{\hat{z}}_t)$ defined on $\mathbb{S}^2 \times \mathbb{S}^2$ that describes both the knowledge about $\underline{x}_{t+1}$ and $\underline{x}_t$ and considers the information of all measurements until time step t. Then, a marginalization is performed to obtain the prior density that describes only the pdf for the state at time step $t+1$.

We now consider how these operations can be realized based on the grid values. For this, the transition density needs to be converted into a matrix of grid values $\mathbf{\Gamma}_t^{\mathrm{T}}$, as described in Sec. 2.3. As before, the posterior density of time step t is described by a vector of grid values $\underline{\gamma}_t^{\mathrm{e}}$. To easily multiply the transition density with the posterior density, the grids should be compatible. This is the case if the grid used for the posterior density is also used as the basis for the Cartesian product for the grid on $\mathbb{S}^2 \times \mathbb{S}^2$. If the grids are not inherently compatible, it is possible to interpolate the grid values and evaluate one of the two functions on the other grid. This, however, introduces additional approximation errors, may void the normalization of the approximation, and requires special care to prevent negative grid values.

For each grid point $\underline{\beta}_j$ ($j \in \{1, \ldots, n\}$), $\gamma_{t,j}^{\mathrm{e}}$ is the function value of the posterior density at $\underline{\beta}_j$. Further, the density $f_t^{\mathrm{T}}(\underline{x}_{t+1} | \underline{\beta}_j)$ is described by the jth column of $\mathbf{\Gamma}_t^{\mathrm{T}}$, which we denote by $\underline{\gamma}_{t,[:,j]}^{\mathrm{T}}$. To obtain the function values of the joint density at all points in the Cartesian product (i.e., $f_t^{\mathrm{j}}(\underline{\beta}_i, \underline{\beta}_j | \underline{\hat{z}}_1, \ldots, \underline{\hat{z}}_t)$ for all $(i,j) \in \{1, \ldots, n\}^2$), we multiply the function values of $f_t^{\mathrm{T}}(\underline{\beta}_i | \underline{\beta}_j)$ with those of $f_t^{\mathrm{e}}(\underline{\beta}_j | \underline{\hat{z}}_1, \ldots, \underline{\hat{z}}_t)$. Thus, we obtain the formula

$$\Gamma_t^{\mathrm{j}} = \left[\underline{\gamma}_{t,[:,1]}^{\mathrm{T}} \gamma_{t,1}^{\mathrm{e}}, \; \cdots, \; \underline{\gamma}_{t,[:,n]}^{\mathrm{T}} \gamma_{t,n}^{\mathrm{e}} \right] \qquad (3)$$

for the matrix of grid values for the joint density.

We implement the marginalization by calculating the integral for every $\underline{\beta}_i$, which yields the function values of the predicted density on the grid. Using the formula for the integral presented in the update step, we obtain

$$\forall i \in \{1, \ldots, n\} : \gamma_{t+1,i}^{\mathrm{p}} = 4\pi \, \mathrm{mean}\left(\underline{\gamma}_{t,[i,:]}^{\mathrm{j}} \right) \; .$$

The two operations for the prediction step can be combined into one very efficient operation. If $\gamma_{t,[i,j]}^{\mathrm{j}}$ and $\gamma_{t,[i,j]}^{\mathrm{T}}$ denote

Algorithm 4: Prediction Step for Time-Variant Transition Densities

```
Input: Current grid values $\underline{\gamma}_t^{\mathrm{e}}$, transition density $f_t^{\mathrm{T}}$
Output: New grid values $\underline{\gamma}_{t+1}^{\mathrm{p}}$
$n \leftarrow \mathrm{NumberOfElements}(\underline{\gamma}_t^{\mathrm{e}})$;
$\Gamma_t^{\mathrm{T}} \leftarrow \mathrm{GetGridValuesCond}(f_t^{\mathrm{T}}, n)$ // Algorithm 2
$\underline{\gamma}_{t+1}^{\mathrm{p}} \leftarrow \frac{4\pi}{n}\Gamma_t^{\mathrm{T}}\underline{\gamma}_t^{\mathrm{e}}$; // Perform prediction according to (5)
```

the entries at the ith row and jth column of Γ_t^{j} and Γ_t^{T}, respectively, we can calculate $\gamma_{t+1,i}^{\mathrm{p}}$ according to

$$\gamma_{t+1,i}^{\mathrm{p}} = \frac{4\pi}{n}\sum_{j=1}^{n}\gamma_{t,[i,j]}^{\mathrm{j}} = \frac{4\pi}{n}\sum_{j=1}^{n}\gamma_{t,[i,j]}^{\mathrm{T}}\gamma_{t,j}^{\mathrm{e}} \,. \tag{4}$$

The rightmost sum in (4) describes the formula for multiplying the ith row of the matrix Γ_t^{T} with the column vector $\underline{\gamma}_t^{\mathrm{e}}$. As illustrated in Fig. 4b, the column vector $\underline{\gamma}_{t+1}^{\mathrm{p}}$ containing all grid values can be determined using the scaled matrix–vector product

$$\underline{\gamma}_{t+1}^{\mathrm{p}} = \tfrac{4\pi}{n}\Gamma_t^{\mathrm{T}}\underline{\gamma}_t^{\mathrm{e}} \,. \tag{5}$$

Adjustments to this formula may be required for other grids than the EQ-based grid. While the formula for the joint density (3) remains valid, the marginalization has to be adjusted to respect the sizes of the individual patches.

Pseudocode for tracking tasks with time-variant system models is given in Algorithm 4. If the transition density is time invariant, the matrix of grid values Γ^{T} can be precalculated and used as an input instead of f_t^{T}. One should verify that $\mathrm{NumberOfElements}(\Gamma^{\mathrm{T}}) = \mathrm{NumberOfElements}(\underline{\gamma}_t^{\mathrm{e}})^2$ to ensure that (assuming the same grid generation scheme is used) the grid values are based on compatible grids. The computational complexity of the prediction step is in $O(n^2)$ due to the matrix–vector multiplication. Further, for time-variant transition densities, n^2 function evaluations of f_t^{T} are required in each prediction step. This effort is only required once for time-invariant transition densities.

4. EVALUATION

For our evaluation, we simulated the time steps 0 to 9 of a scenario with a simple measurement model and a complicated system model. We compare the newly proposed SGF with the PF and the nonlinear variant of the VMFF. The SGF is evaluated for different numbers of grid points and the PF for different numbers of particles. The estimation error is assessed by determining the angular distance (see Kennedy and Sadeghi (2013, Sec. 7.2.1))

$$d_{\mathrm{ang}}(\tilde{\underline{x}}_9, \hat{\underline{x}}_9^{\mathrm{e}}) = \mathrm{acos}(\tilde{\underline{x}}_9^\top \hat{\underline{x}}_9^{\mathrm{e}})$$

between the true state $\tilde{\underline{x}}_9$ and the posterior estimate $\hat{\underline{x}}_9^{\mathrm{e}}$ at the last time step. To obtain reliable results, we take the mean of the errors of 10000 runs.

The code for the PF and VMFF is part libDirectional (see Kurz et al. (2019)), and the latest version in the GitHub repository also contains the SGF. All evaluations were performed on a laptop with an Intel Core i7-7500U CPU and 16 GB of RAM, running Matlab 2020a on Windows 10. In the remainder of this section, we first describe the scenario in detail and then present the evaluation results.

4.1 Evaluation Scenario

The random variable $\underline{\boldsymbol{x}}_0$ describing the initial state is distributed according to the von Mises–Fisher distribution $\mathcal{VMF}(\underline{x}_0; [0,0,1]^\top, 100)$, which is rather concentrated and thus challenging to approximate. The system function

$$\underline{a}(\underline{x}_t, \underline{u}) = \frac{\alpha\underline{x}_t + (1-\alpha)\underline{u}}{\|\alpha\underline{x}_t + (1-\alpha)\underline{u}\|}$$

is based on the normalized linear interpolation function, which was also considered in Kurz et al. (2016a). After applying the system function, the result is perturbed by von Mises–Fisher-distributed noise with concentration parameter 100. The transition density for this system model can be written as

$$f_t^{\mathrm{T}}(\underline{x}_{t+1}|\underline{x}_t) = \mathcal{VMF}(\underline{x}_{t+1}; \underline{a}(\underline{x}_t, \underline{u}), 100) \,.$$

We set $\underline{u} = [0,1,0]^\top$ and keep the vector constant for all time steps. Explained verbally, the state is attracted to $\underline{u}$ but the applied system noise causes it to jump around and never quite reach the target point. Due to the time invariance of the model, the grid values of the transition density only have to be determined once.

The measurement model is simpler. The measurement function is the identity function and the measurements are merely perturbed by von Mises–Fisher-distributed noise with concentration parameter 100. The likelihood function is thus

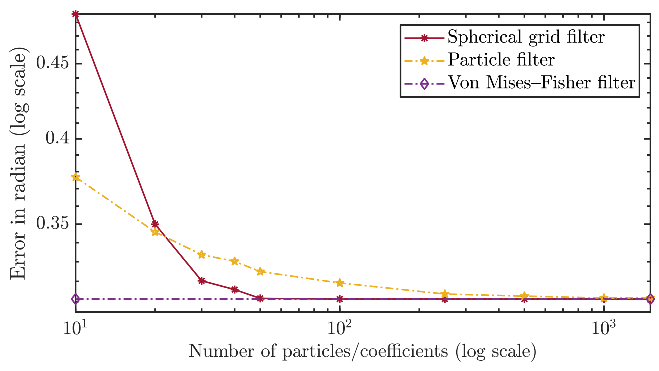
$$f_t^{\mathrm{L}}(\underline{z}_t|\underline{x}_t) = \mathcal{VMF}(\underline{z}_t; \underline{x}_t, 100) \,.$$

4.2 Evaluation Results

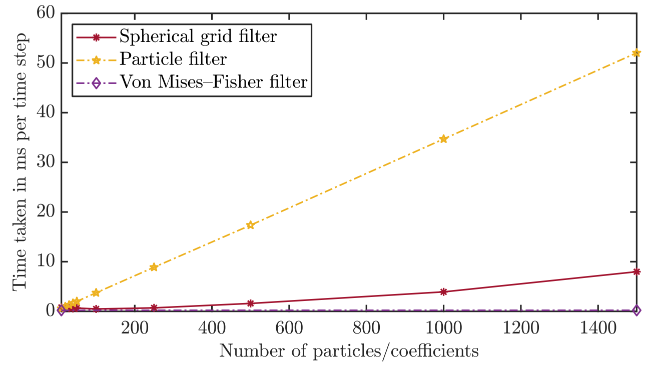
The mean of the errors is shown for all filters and all numbers of parameters in Fig. 5a. Using 100 grid points, the SGF achieves an estimation accuracy that is close to its optimal accuracy. The PF converges slowly and even with 1500 particles, it does not reach the estimation accuracy of the VMFF and SGF.

The run times per time step are shown for all filters in Fig. 5b. For the PF, the main computational cost is caused by sampling the von Mises–Fisher distribution for each particle in the prediction step. This explains the linear increase in computation time with the number of particles. The VMFF is faster than all configurations of the SGF and PF. Using 100 grid points, the SGF achieves better results than the VMFF at approximately twice the run time. With a run time of less than 5 ms, the tracking can run at 200 Hz for this configuration of the SGF (using other programming languages may result in even lower run times). The SGF is only slightly better than the VMFF because the scenario lends itself well to the use of the VMFF. Noises with other distributions (e.g., multimodal ones) could significantly reduce the estimation quality of the VMFF. The increase in the run time of the SGF is slow compared with the PF. While the complexity of the SGF is in $O(n^2)$, it is very fast in practice because the matrix–vector multiplication is implemented efficiently in Matlab.

Since the SGF was faster and (except for very few grid points) more accurate than the PF, the SGF performed better overall. The VMFF was very fast and yielded high-quality results. The SGF provided even better estimates at run times that are suitable for many real time applications.



(a) Errors for different numbers of parameters.



(b) Run times for different numbers of parameters.

Fig. 5. Evaluation results. The results for the VMFF are shown as horizontal lines because the number of parameters cannot be varied.

5. CONCLUSION AND OUTLOOK

In this paper, we proposed the SGF as a novel approach to filtering on the unit sphere. A fast and efficient algorithm was used to generate the required grids. By taking the Cartesian product of such a grid with itself, a suitable grid for the transition density was obtained. To perform prediction and update steps, it must merely be possible to evaluate the transition density and likelihood function on suitable grids. The SGF allows providing a pdf via interpolation, is fully deterministic, and outperformed the PF in a scenario involving highly concentrated densities.

In future work, partitions with varying patch sizes could be considered. Further, the SGF could be generalized to arbitrary-dimensional spheres. Moreover, other manifolds, such as a hemisphere of the hypersphere $\mathbb{S}^3$, which can be used to represent rotations in the form of unit quaternions, could be considered. Based on generalizations of the SGF, filters for relevant Cartesian products of periodic and Euclidean domains could be derived via Rao–Blackwellization.

ACKNOWLEDGMENT

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

REFERENCES

Bingham, C. (1974). An Antipodally Symmetric Distribution on the Sphere. *The Annals of Statistics*, 2(6).

Chen, Y.H., Wei, D., Newstadt, G., DeGraef, M., Simmons, J., and Hero, A. (2015). Parameter Estimation in Spherical Symmetry Groups. *IEEE Signal Processing Letters*, 22(8).

Chiuso, A. and Picci, G. (1998). Visual Tracking of Points as Estimation on the Unit Sphere. *The Confluence of Vision and Control, Lecture Notes in Control and Information Science*, 237.

Colombo, O.L. (1981). Numerical Methods for Harmonic Analysis on the Sphere. Technical report, Ohio State University.

Healy, Jr., D.M., Rockmore, D.N., Kostelec, P.J., and Moore, S.S.B. (2003). FFTs for the 2-Sphere—Improvements and Variations. *Journal of Fourier Analysis and Applications*, 9(4).

Jammalamadaka, S.R. and Sengupta, A. (2001). *Topics in Circular Statistics*. World Scientific.

Kennedy, R.A. and Sadeghi, P. (2013). *Hilbert Space Methods in Signal Processing*. Cambridge University Press.

Kurz, G., Gilitschenski, I., and Hanebeck, U.D. (2016a). Unscented von Mises–Fisher Filtering. *IEEE Signal Processing Letters*, 23(4).

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

Kurz, G., Pfaff, F., and Hanebeck, U.D. (2016b). 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.

Leopardi, P. (2006). A Partition of the Unit Sphere into Regions of Equal Area and Small Diameter. *Electronic Transactions on Numerical Analysis*, 25(12).

Lloyd, S. (1982). Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2).

Mardia, K.V. (1981). Directional Statistics in Geosciences. *Communications in Statistics—Theory and Methods*, 10(15).

Mardia, K.V. and Jupp, P.E. (1999). *Directional Statistics*. John Wiley & Sons.

Pfaff, F., Kurz, G., and Hanebeck, U.D. (2017). Filtering on the Unit Sphere Using Spherical Harmonics. In *Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2017)*. Daegu, Republic of Korea.

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

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series. The MIT Press.

Traa, J. and Smaragdis, P. (2014). Multiple Speaker Tracking With the Factorial von Mises–Fisher Filter. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2014)*.

Watson, G.S. (1956). Analysis of Dispersion on a Sphere. *Geophysical Journal International*, 7.

Wonham, W.M. (1964). Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 2(3).