Discretization of SO(3) Using Recursive Tesseract Subdivision

Gerhard Kurz[1], Florian Pfaff[1], and Uwe D. Hanebeck[1]

***Abstract*— The group of rotations in three dimensions SO(3) plays a crucial role in applications ranging from robotics and aeronautics to computer graphics. Rotations have three degrees of freedom, but representing rotations is a nontrivial matter and different methods, such as Euler angles, quaternions, rotation matrices, and Rodrigues vectors are commonly used. Unfortunately, none of these representations allows easy discretization of orientations on evenly spaced grids. We present a novel discretization method that is based on a quaternion representation in conjunction with a recursive subdivision scheme of the four-dimensional hypercube, also known as the tesseract.**

I. Introduction

Orientations in three dimensions are omnipresent in many problems, for example in robotics [1], [2], computer vision [3], autonomous vehicles, aeronautics [4], [5], crystallography [6], computer graphics [7], etc. In many scenarios, orientations have to be estimated based on noisy sensor measurements or controlled using various kinds of actuators. Due to the uncertainty involved, it often becomes necessary to consider probability distributions on the space of orientations. One possible way to approach this problem is to employ a finite subset of possible rotations and to assign a probability to each discrete rotation.

Aside from approximation of uncertain rotations, discretization on SO(3) has various user applications. For example, in robotic perception, it can be used to generate initial poses for alignment of point clouds. These can be locally refined using an algorithm such as iterative closest point (ICP) to achieve a globally optimal solution [8]. An application in computer vision consists in detecting known 3D objects in 2D images by rendering a 3D model of the object from a set of different perspectives and using pattern recognition to find it in the image [9]. In model predictive control (MPC), it can be used to generate a discrete set of possible control inputs [10] (where the input is an orientation in SO(3)). Finally, in computer graphics, we can use it to pre-render an object or a scene from multiple perspectives, and use the closest one when a particular view is desired to avoid online rendering.

In $\mathbb{R}^3$, orientations can be represented using the special orthogonal group $SO(3)$, which is the group of 3×3 rotation matrices. Rotation matrices are a unique representation of orientations, but they suffer from significant overparameterization – a rotation matrix has nine entries but a rotation in three dimensions only has three degrees of freedom.

For this reason, we will represent orientations using unit quaternions [11], [12]. Quaternions are an extension of complex numbers with three different imaginary units, and hence, they can be parameterized using four-dimensional vectors in $\mathbb{R}^4$. The quaternions with unit norm are commonly used to represent orientations, i.e., each point on the unit sphere S^3 in $\mathbb{R}^4$ can be seen as an orientation. However, the two quaternions q and $-q$ represent the same orientation, so the unit quaternions form a double cover of $SO(3)$. As a result, we can use a hemihypersphere in $\mathbb{R}^4$ as a unique parameterization for $SO(3)$.

By doing so, we reduce the problem of discretization of $SO(3)$ to the problem of discretizing the four-dimensional unit hemisphere $S^3/\{\pm 1\}$. Our novel method is based on recursive subdivision of a four-dimensional cube, also known as the four-dimensional hypercube or tesseract. The surface of the tesseract consists of three-dimensional cubes. However, points on the surface of the tesseract do not, in general, correspond to unit vectors, and thus, cannot be used to represent rotations. For this reason, the resulting points are projected onto the unit hypersphere by normalizing them to have unit length. This discretization induces a Voronoi tessellation [13] on $SO(3)$, i.e., a partition of the space of rotations.

A similar approach has been proposed by Schaefer et al. [14]. The key difference is that their approach is based on subdivisions using tetrahedral and octahedral meshes. In particular, a tetrahedron can be split into four tetrahedra and an octahedron. Moreover, an octahedron can be split into six octahedra and eight tetrahedra. This approach has been used by Glover et al. [15] for evaluation of a recursive rotation estimator based on the Bingham distribution. Other possible discretization approaches consist in uniform discretization of Euler angles (see Sec. V) and in using equal area partitioning of the unit sphere [16]. The latter approach does not, in general, result in a symmetric discretization, however.

There are several advantages to using a tesseract compared with the subdivision method based on octahedra and tetrahedra. Because only one type of polyhedron appears, our algorithm is more straightforward and easier to understand and implement. Also, the resulting discretization has more intuitive symmetries as the tesseract has $90°$ rotational symmetry. Finally, our evaluation shows that the proposed approach also yields the most accurate results when used for approximating probability densities on $SO(3)$.

[1] The authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany (e-mail: gerhard.kurz@kit.edu, florian.pfaff@kit.edu, uwe.hanebeck@ieee.org).

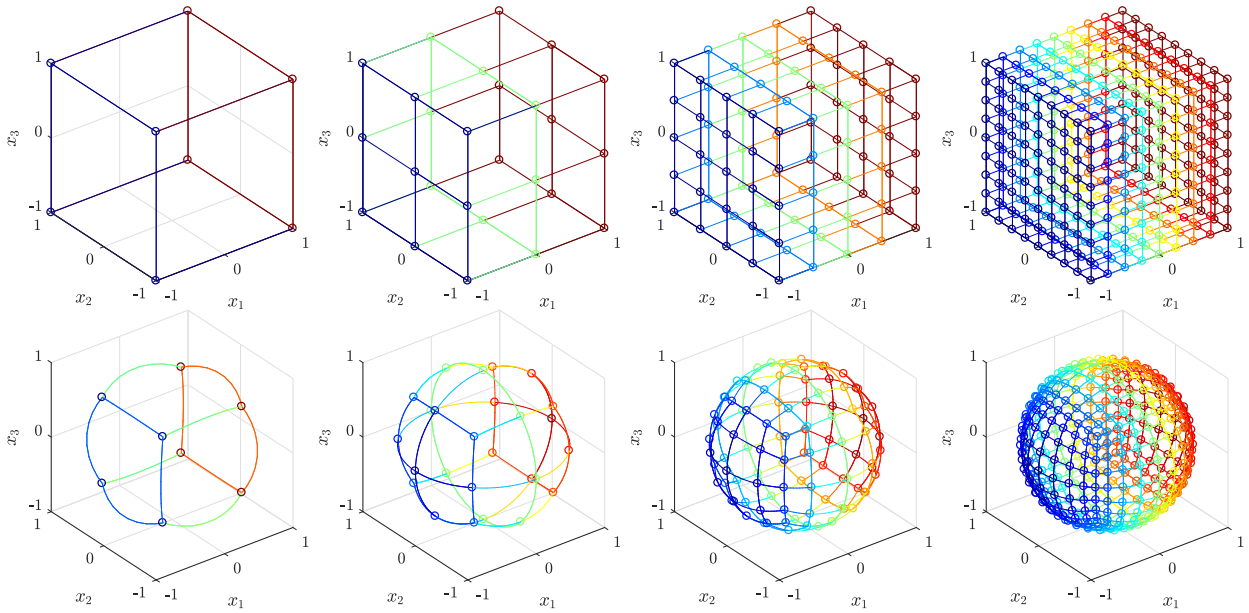

Fig. 1: Examples in 3D before (top) and after (bottom) normalization. Observe that only the surface is subdivided, not the interior of the cube.

II. Cube Surface Subdivision

Before considering the tesseract subdivision algorithm, we introduce the corresponding three-dimensional approach, i.e., a cube subdivision method, to illustrate the basic ideas and concepts.

A three-dimensional cube has eight corners with coordinates

$$[\pm 1, \pm 1, \pm 1] := \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ -1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix} \in \mathbb{R}^{8\times 3} \ .$$

Its surface consists of six two-dimensional squares with coordinates

$$[1, \pm 1, \pm 1] \ , [-1, \pm 1, \pm 1] \ , [\pm 1, 1, \pm 1] \ ,$$
$$[\pm 1, -1, \pm 1] \ , [\pm 1, \pm 1, 1] \ , [\pm 1, \pm 1, -1] \ .$$

We can subdivide a square by adding a new point in the middle of each edge and one point in the center of the rectangle, yielding four squares whose edges have half the length of the original square. This operation can be performed recursively as many times as desired. The same approach is used in quadtrees [17] in the plane. Quadtrees typically do not subdivide all squares equally, but focus on areas of interest that are subdivided more finely than the rest. However, we limit ourselves to uniform subdivisions for now.

Lemma 1. After n subdivision steps, we have $6 \cdot 4^n$ squares with a total of $24 \cdot 4^n$ non-unique corners. As certain corners are shared between multiple squares, the total number of unique points is $6 \cdot 4^n + 2$.

Proof. The first two claims are straightforward. In order to compute the total number of unique points, we proceed as follows.

The surface of a cube consists of 6 squares. Assume that each square has been subdivided into a grid of $m \times m$ points, i.e., there are m^2 points that are unique within each square. In total, we have $6m^2$ points. However, as the squares are touching at the edges and corners, certain points are present in more than one square and we need to subtract those. There are

- 8 corners, which have been counted 3 times, i.e., we need to subtract $8 \cdot 2$,
- 12 edges with $m-2$ points each (excluding the corners) that have been counted twice, i.e., we have to subtract $12 \cdot (m-2)$.

As a result, we obtain

$$6m^2 - 8 \cdot 2 - 12(m-2) \ .$$

For n subdivision steps, i.e., $m = 2^n + 1$, we have

$$6(2^n+1)^2 - 12(2^n - 1) - 16 = 6 \cdot 4^n + 2 \ ,$$

which shows the claim. □

To obtain a tessellation of the surface of the sphere S^2, we normalize all corners to be vectors of unit length. Some examples are depicted in Fig. 1. Even though all squares have an identical size before normalization, the resulting tessellation is not perfectly uniform due to the nonlinear normalization operation, i.e., the distance between neighboring points is not equal everywhere. However, the result is close enough to a uniform tessellation for many

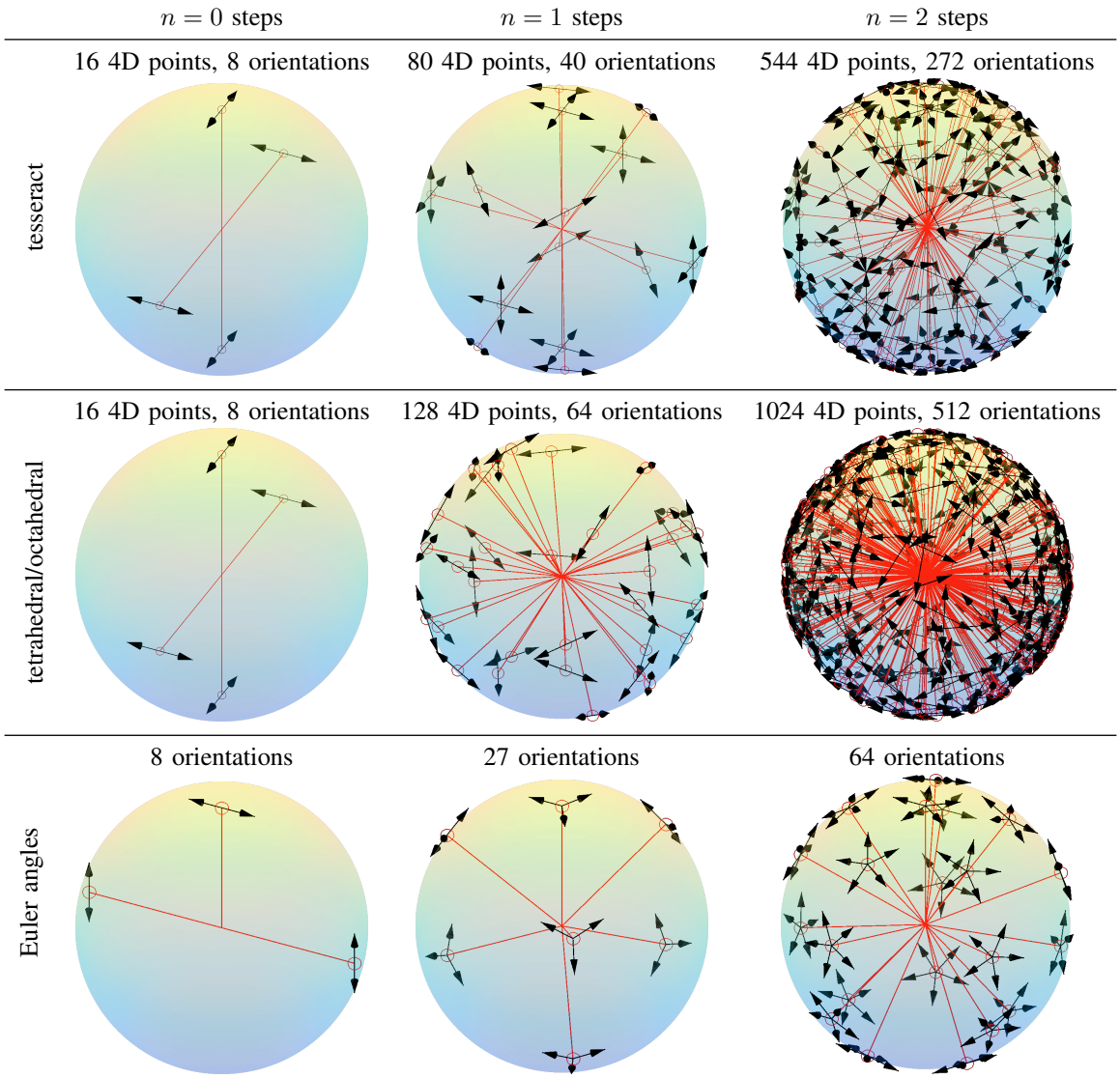

Fig. 2: Examples in 4D for the tesseract subdivision, the tetrahedral/octahedral subdivision, and an Euler angle approach. Each orientation is represented by a point on the unit sphere along with a vector tangential to the sphere that corresponds to the roll angle. The point on the unit sphere can be interpreted as the direction you are facing (two degrees of freedom) along with a vector pointing upwards (one degree of freedom). Note that sometimes, several samples have the same point on the sphere but a different vector pointing upwards.

practical purposes and exhibits a high degree of symmetry. In future work, it may be possible to consider the use of a non-uniform grid on the cube, which is designed to yield a more uniform grid after projection onto the sphere.

III. Tesseract Surface Subdivision

The cube subdivision algorithm introduced above can be generalized to the four-dimensional case as follows. It should be noted, however, that the cube subdivision algorithm is not part of the tesseract subdivision algorithm.

We consider the 4D hypercube (or tesseract). It has 16 corners with coordinates

$$[\pm 1, \pm 1, \pm 1, \pm 1]^T \ .$$

Its three-dimensional surface consists of 8 three-dimensional cubes (embedded in a 4D space) whose corners have the coordinates

$$\mathbf{C}_1 = [1, \pm 1, \pm 1, \pm 1] \ , \tag{1a}$$
$$\mathbf{C}_2 = [-1, \pm 1, \pm 1, \pm 1] \ , \tag{1b}$$
$$\mathbf{C}_3 = [\pm 1, 1, \pm 1, \pm 1] \ , \tag{1c}$$
$$\mathbf{C}_4 = [\pm 1, -1, \pm 1, \pm 1] \ , \tag{1d}$$
$$\mathbf{C}_5 = [\pm 1, \pm 1, 1, \pm 1] \ , \tag{1e}$$
$$\mathbf{C}_6 = [\pm 1, \pm 1, -1, \pm 1] \ , \tag{1f}$$
$$\mathbf{C}_7 = [\pm 1, \pm 1, \pm 1, 1] \ , \tag{1g}$$
$$\mathbf{C}_8 = [\pm 1, \pm 1, \pm 1, -1] \ . \tag{1h}$$

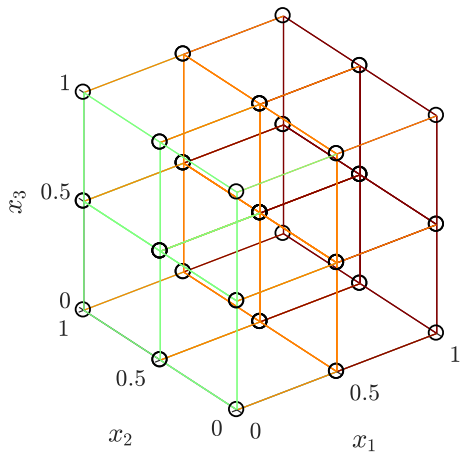

Fig. 3: Subdivision of a 3D cube into eight smaller cubes. This type of subdivision is used for octrees. Observe that unlike in Fig. 1, the interior of the cube is subdivided rather than its surface.

Now, we can subdivide each cube by adding new points in the middle of each edge and a new point in the center. By doing so, we obtain 8 smaller cubes whose edges have half the length of the original edge length. The subdivision is performed in the same way as in octrees [17], [18] in 3D (see Fig. 3).

Lemma 2. After n subdivision steps, we have $8 \cdot 8^n$ cubes with a total of $64 \cdot 8^n$ corners. Certain corners are shared among multiple cubes, so the total number of unique points is $8 \cdot (2^n + 8^n)$.

Proof. Similar to Lemma 1, the first two claims are straightforward. We use a similar approach as before to show the third claim.

The tesseract consists of eight cubes. Consider one cube and assume that there are m points onn each edge, i.e., there is an $m \times m \times m$ grid. Then, one cube has m^3 points that are unique if only the current cube is considered. Since there are 8 cubes, we have $8m^3$ points in total.

Now, we subtract the points occurring multiple times. There are

- 16 corners, counted 4 times $\Rightarrow$ subtract $16 \cdot 3$,
- 32 edges, counted 3 times $\Rightarrow$ subtract $32 \cdot 2 \cdot (m-2)$,
- 24 faces, counted twice $\Rightarrow$ subtract $24 \cdot 1 \cdot (m-2)^2$.

In total, we have

$$8m^3 - 16 \cdot 3 - 32 \cdot 2(m-2) - 24 \cdot (m-2) \ .$$

For n subdivision steps, i.e., $m = 2^n + 1$, we obtain

$$8 \cdot (2^n+1)^3 - 16 \cdot 3 - 32 \cdot 2 \cdot (2^n-1) - 24 \cdot (2^n-1)^2 = 8(2^n+8^n) \ ,$$

which shows the claim. □

The resulting recursive tesseract subdivision method is given in Algorithm 1, which in turn makes use of the cube subdivision method given in Algorithm 2. After applying these algorithms, we normalize the vectors to obtain a tessellation of the four-dimensional hypersphere. The runtime of the proposed algorithms is linear in the number of points in the discretization.

Algorithm 1: Tesseract Subdivision

Input: iterations n
Output: list of points P

```
/* generate tesseract using equations (1a)-(1h)  */
C_0 ← [C_1, ..., C_8];
/* subdivide cubes recursively                   */
for i ← 1 to n do
    C_i ← ∅;
    for j ← 1 to |C_{i-1}| do
        C_i ← [C_i, subdivideCube(C_j)] ;
    end
end
P ← removeDuplicatePoints(C_n);
return P;
```

Algorithm 2: Subdivide Cube

Input: cube $\mathbf{C} \in \mathbb{R}^{8\times 4}$, where each row corresponds to a corner point in $\mathbb{R}^4$
Output: list of 8 cubes C

```
/* compute center of cube                        */
m ← (1/8) Σ_{i=1}^{8} C(i,:);
/* compute subdivision                           */
C ← ∅;
for i ← 1 to 8 do
    /* Generate an axis-aligned cube with given
       opposite corners                          */
    C ← [C, generateCube(C(i,:), m)];
end
return C;
```

As we plan to use the discretization of the hypersphere for quaternions, we have to deal with their property that quaternions q and $-q$ represent the same orientation. Thus, we are actually interested in a tessellation of the hemihypersphere rather than the full hypersphere. It is straightforward to obtain a tessellation of the hemihypersphere based on the previous approach. Due to the symmetry of the tessellation, we can simply perform a regular tesseract subdivision and then remove half of the points in such a way that for any pair $x, -x$ only x is preserved[1].

The resulting tessellation are visualized in Fig. 2. For comparison, we also show the results obtained using the tetrahedral/octahedral subdivision from [14] and using the Euler angle method discussed in Sec. V. It can be seen that the first two approaches generally yield similar results, but the number of points increases more quickly per subdivision step for the tetrahedral/octahedral subdivision.

[1]Note that it is not sufficient to remove all points with $x_1 < 0$ or $x_1 \leq 0$, because there are points with $x_1 = 0$. Half of the points with $x_1 = 0$ are supposed to be removed but the other half has to be kept.

IV. Using the Discrete Approximation for Representation of Uncertain Rotations

The discretization of $SO(3)$ obtained above can be used in many applications, such as estimation, control etc. In the following, we will consider the representation of uncertain orientations as an exemplary application.

Recently, representing uncertain orientations using the Bingham distribution [19] has gained significant interest [20]–[23]. For this reason, we will consider how a Bingham-distributed uncertain orientation can be approximated using a discrete grid obtained from the tesseract subdivision approach.

A Bingham distribution is given by the probability density function (pdf)

$$f(\underline{x}; \mathbf{M}, \mathbf{Z}) = \frac{1}{F(\mathbf{Z})} \cdot \exp\left(\underline{x}^T \mathbf{M}\mathbf{Z}\mathbf{M}^T \underline{x}\right) \ ,$$

where $\underline{x} \in \mathbb{R}^d$ is a unit vector, $\mathbf{M} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix and $\mathbf{Z} = \mathrm{diag}(z_1, \ldots, z_{d-1}, 0) \in \mathbb{R}^{d\times d}$ is a diagonal matrix with $z_1 \leq \cdots \leq z_{d-1} \leq 0$. The term $F(\mathbf{Z})$ refers to the normalization constant (see [24]). It can be seen from the pdf that the Bingham distribution is antipodally symmetric, i.e., $f(\underline{x}) = f(-\underline{x})$, which makes it very appropriate for use with quaternions.

To approximate a given Bingham distribution, we consider the integral of the pdf over the Voronoi region V_i [13] of each grid point

$$w_i = \int_{V_i} f(\underline{x}) \, \mathrm{d}\underline{x} \ ,$$
$$V_i = \{\underline{x} \in S^3 : \|\underline{x} - \underline{x}_i\| \leq \|\underline{x} - \underline{x}_j\| \ \forall j \neq i\}$$

to compute its weight. As solving this integral is only possible numerically and computationally burdensome, we assume that all Voronoi cells are of nearly identical volume and approximate the weights according to

$$w_i = \int_{V_i} f(\underline{x}) \, \mathrm{d}\underline{x} \approx f(\underline{x}_i) \int_{V_i} 1 \, \mathrm{d}\underline{x} \approx c \cdot f(\underline{x}_i) \ ,$$

where c is a normalization constant that ensures $\sum_{i=1}^{n} w_i = 1$. This yields a set of weighted samples that approximate the original continuous distribution. Generally, the finer the grid, the better the approximation.

After normalizing the weights to sum to one, the resulting weighted samples can be interpreted as a discrete set of possible rotations together with their associated probabilities. The key advantage of considering a discrete sample set instead of the original continuous distribution is that it is easy to propagate the samples through nonlinear functions and to reweigh them based on a likelihood. As a result, it is possible to derive a nonlinear discrete filter on $SO(3)$ similar to the discrete circular filter we presented in [25], [26] and the point mass filter proposed in [27].

Of course, the grid-based representation of uncertain orientations is not limited to approximating Bingham distributions. It is also possible to approximate other antipodally symmetric distributions on S^3, for example, the angular central Gaussian [28], the S^3 version of projected Gaussians [29, Sec. III], and Bingham mixtures [2, Sec. IV-D].

V. Evaluation

In the evaluation, we compare our proposed approach with two alternative discretization methods, the tetrahedron/octahedron subdivision method by Schaefer [14] and a naive method based on Euler angles. The Euler angle approach works by considering roll, pitch, and yaw angles $\alpha_r, \alpha_p, \alpha_y$, where $\alpha_r, \alpha_y \in [-\pi, \pi)$ and $\alpha_p \in [-\pi/2, \pi/2]$. For the n-th subdivision step, we consider

$$\begin{aligned}
\alpha_r &= -\pi + 2\pi\frac{i}{n} \ , & i &= 0, \ldots, n-1 \ , \\
\alpha_p &= -\pi/2 + \pi\frac{j}{n} \ , & j &= 0, \ldots, n-1 \ , \\
\alpha_y &= -\pi + 2\pi\frac{k}{n} \ , & k &= 0, \ldots, n-1
\end{aligned}$$

and convert each of the n^3 possible combinations to a unit quaternion. Note that this can result in duplicates because Euler angles are not unique, even when the angles are limited to the appropriate range.

A. Moment-based Evaluation

We consider a Bingham distribution with orientation parameter $\mathbf{M}$ and concentration parameter $\mathbf{Z}$. The orientation parameter is chosen as a random orthogonal matrix $\mathbf{M}$ (see [30, Sec. 2]) to avoid introducing a bias when the symmetry axes of the Bingham distribution align with the symmetry axes of one of the discrete approximations. For the concentration parameters, we investigate several fixed concentration matrices $\mathbf{Z}$ to see how the amount of uncertainty impacts the results.

Due to the antipodally symmetric pdf, a Bingham-distributed random vector $\underline{x}$ always has a first moment of $\mathbb{E}(\underline{x}) = \underline{0}$. However, the second moment $\mathbb{E}(\underline{x} \cdot \underline{x}^T)$ is of great interest because it uniquely characterizes a Bingham distribution, similar to the way the mean and the covariance uniquely describe a Gaussian distribution. As a result, there is a bijection between positive semidefinite 4×4 matrices and Bingham distributions on S^3. Thus, the second moment plays a crucial role in the maximum likelihood parameter estimation algorithm for the Bingham distribution [19, Sec. 5], [24, Sec. III-B].

The second moment of a given Bingham distribution can be computed according to

$$\mathbb{E}(\underline{x} \cdot \underline{x}^T) = \frac{1}{F(\mathbf{Z})} \mathbf{M} \cdot \mathrm{diag}\left(\frac{\partial F(\mathbf{Z})}{\partial z_1}, \ldots, \frac{\partial F(\mathbf{Z})}{\partial z_n}\right) \cdot \mathbf{M}^T \ .$$

Note that this computation involves both the normalization constant and its first derivative. It is possible to compute these values using saddlepoint approximations [31], [24], but for the purpose of the evaluation, we want to avoid the approximation error involved. For this reason, we use the integral representation of the normalization constant by Wood [32, eq. (3.3)], which is slower but much more accurate. The derivative is computed using finite differences.

Then, we compare the true second moment $\mathbf{C}^t$ of the Bingham distribution to the second sample moment $\mathbf{C}^a$ of the

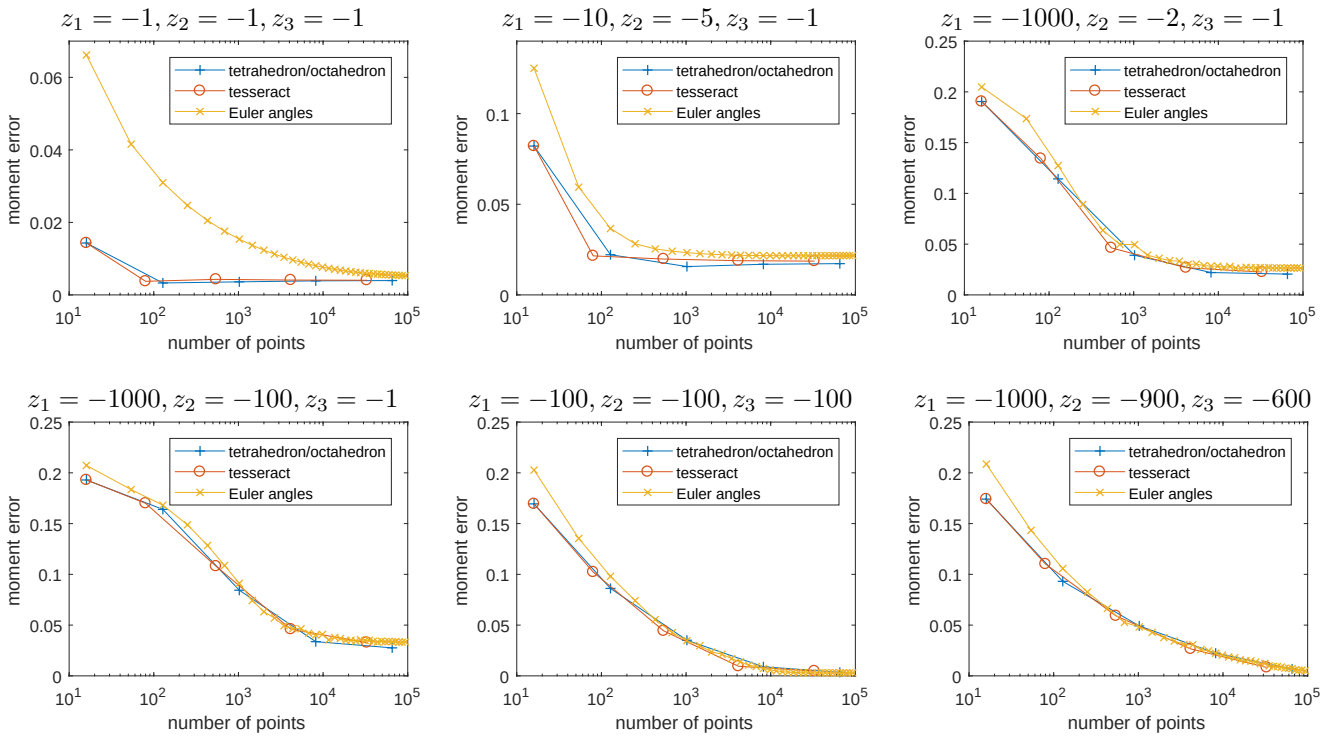

Fig. 4: Moment-based evaluation for different concentration parameters $\mathbf{Z}$ according to the error measure (2).

set of weighted samples obtained based on the discretization of $SO(3)$. For this purpose, we use the error measure

$$\sqrt{\frac{1}{16}\sum_{i=1}^{4}\sum_{j=1}^{4}(\mathbf{C}^t(i,j)-\mathbf{C}^a(i,j))^2}\ . \tag{2}$$

In total, we performed 100 Monte Carlo runs. The results of the moment-based evaluation are shown in Fig. 4. It can be seen that the method based on Euler angles is clearly inferior in some cases. The tesseract subdivision method and the tetrahedron/octahedron subdivision method perform very similarly with a slight advantage for the tesseract method.

B. Expected Distance

In order to evaluate the uniformity of the discretization, we consider the expected distance to the nearest grid point for a random point $\underline{x}$ that is uniformly drawn from the hypersphere. The intuitive motivation is that we seek to avoid large "holes" in the grid, i.e., areas that are not covered by grid points. An upper bound of the distance to the next grid point can be used to derive bounds on the approximation error when applying certain nonlinear functions to the discretized probability density (see [33]).

For this purpose, we consider the expectation value

$$\mathbb{E}(\|\underline{x}-N(\underline{x})\|_2)=\frac{1}{|S^3|}\int_{S^3}\|\underline{x}-N(\underline{x})\|_2\ \mathrm{d}\underline{x}$$

where $|S^3|$ is the surface area of S^3 and

$$N(\underline{x})=\underset{\underline{y}\in\text{grid}}{\arg\min}\,\|\underline{x}-\underline{y}\|_2$$

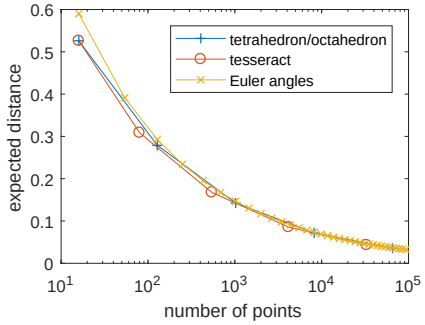

Fig. 5: Evaluation based on the expected distance to the nearest grid point.

is the Euclidean distance to the nearest neighbor of $\underline{x}$. In order to evaluate the integral, we perform a substitution to hyperspherical coordinates $[r,\phi_1,\phi_2,\phi_3]^T$ according to

$$\begin{aligned}x_1&=r\sin(\phi_1)\sin(\phi_2)\sin(\phi_3)\ ,\\ x_2&=r\cos(\phi_1)\sin(\phi_2)\sin(\phi_3)\ ,\\ x_3&=r\cos(\phi_2)\sin(\phi_3)\ ,\\ x_4&=r\cos(\phi_3)\end{aligned}$$

with radius $r=1$ and the corresponding volume correction term $\mathrm{d}\underline{x}=\sin(\phi_2)\sin^2(\phi_3)\,\mathrm{d}\phi_1\,\mathrm{d}\phi_2\,\mathrm{d}\phi_3$. The resulting integral is evaluated numerically. Fig. 5 shows the resulting expected distance. It can be seen that the grid obtained using the Euler angles approach is slightly inferior to the grid produced by the other methods, but the differences are very small.

VI. Conclusion

We have presented a novel method for discretizing the orientation group $SO(3)$. The method is based on using a quaternion representation and discretizing the four-dimensional unit hemisphere. This is achieved by considering a four-dimensional cube, the tesseract, and recursively subdividing its surface, which consists of three-dimensional cubes.

In a simulative evaluation, we have shown the advantages of the proposed approach. It performs significantly better than a naive discretization based on Euler angles while being easier to understand and implement than the tetrahedron/octahedron approach.

In future work, it seems promising to consider an adaptive octree method, i.e., areas of high interest are to be discretized with more subdivision steps than areas of low interest. Moreover, we plan on developing a recursive nonlinear filter based on the proposed discretization, similar to the discrete filters in [25]–[27].

An implementation of the proposed algorithm is available as part of libDirectional [34], a library that focuses on directional statistics and directional estimation.

Acknowledgment

The IGF project 18798 N of the research association Forschungs-Gesellschaft Verfahrens-Technik e.V. (GVT) was supported via the AiF in a program to promote the Industrial Community Research and Development (IGF) by the Federal Ministry for Economic Affairs and Energy on the basis of a resolution of the German Bundestag. This work was also supported by the German Research Foundation (DFG) under grant HA 3789/13-1.

References

[1] J. M. Selig, *Geometric Fundamentals of Robotics*, 2nd ed., ser. Monographs in Computer Science. New York: Springer, 2005.

[2] J. Glover, G. Bradski, and R. Rusu, "Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution," in *Proceedings of Robotics: Science and Systems (RSS 2011)*, Los Angeles, California, USA, 2011.

[3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[4] J. E. Darling and K. J. DeMars, "Analysis of the Gauss–Bingham Distribution for Attitude Uncertainty Propagation," in *Proceedings of the Astrodynamics Specialist Conference*, Vail, Colorado, USA, Aug. 2015.

[5] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, "Novel Quaternion Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174–190, 2006.

[6] B. Stanfill, "Robust Statistical Methods for the Rotation Group," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[7] J. C. Hart, G. K. Francis, and L. H. Kauffman, "Visualizing Quaternion Rotation," *ACM Transactions on Graphics*, vol. 13, no. 3, pp. 256–276, Jul. 1994.

[8] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3d Registration Efficiently and Globally Optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.

[9] M. Ulrich, C. Wiedemann, and C. Steger, "CAD-based Recognition of 3D Objects in Monocular Images." in *ICRA*, vol. 9, 2009, pp. 1191–1198.

[10] A. Hekler, C. Chlebek, and U. D. Hanebeck, "Open-Loop Feedback Control of Nonlinear Stochastic Systems Based on Deterministic Dirac Mixture Densities," in *Proceedings of the 2012 American Control Conference (ACC 2012)*, Montréal, Canada, Jun. 2012.

[11] A. J. Hanson, *Visualizing Quaternions*, 1st ed., ser. The Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann, 2006.

[12] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton University Press, 2002, vol. 66.

[13] H.-S. Na, C.-N. Lee, and O. Cheong, "Voronoi Diagrams on the Sphere," *Computational Geometry*, vol. 23, no. 2, pp. 183–194, 2002.

[14] S. Schaefer, J. Hakenberg, and J. Warren, "Smooth Subdivision of Tetrahedral Meshes," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP '04. New York, USA: ACM, 2004, pp. 147–154.

[15] J. Glover and L. P. Kaelbling, "Tracking 3-D Rotations with the Quaternion Bingham Filter," MIT, Tech. Rep., Mar. 2013.

[16] 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.

[17] H. Samet, "An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures," in *Theoretical Foundations of Computer Graphics and CAD*. Springer Berlin Heidelberg, 1988, pp. 51–68.

[18] H. Eberhardt, V. Klumpp, and U. D. Hanebeck, "Density Trees for Efficient Nonlinear State Estimation," in *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, United Kingdom, Jul. 2010.

[19] C. Bingham, "An Antipodally Symmetric Distribution on the Sphere," *The Annals of Statistics*, vol. 2, no. 6, pp. 1201–1225, Nov. 1974.

[20] M. E. Antone, "Robust Camera Pose Recovery Using Stochastic Geometry," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[21] J. Glover and L. P. Kaelbling, "Tracking the Spin on a Ping Pong Ball with the Quaternion Bingham Filter," in *Proceedings of the 2014 IEEE Conference on Robotics and Automation (ICRA 2014)*, Hong Kong, China, 2014.

[22] 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.

[23] S. Riedel, Z. C. Marton, and S. Kriegel, "Multi-view Orientation Estimation Using Bingham Mixture Models," in *Proc. Robotics (AQTR) 2016 IEEE Int. Conf. Automation, Quality and Testing*, May 2016, pp. 1–6.

[24] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Efficient Bingham Filtering based on Saddlepoint Approximations," in *Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014)*, Beijing, China, Sep. 2014.

[25] 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.

[26] ——, "Application of Discrete Recursive Bayesian Estimation on Intervals and the Unit Circle to Filtering on SE(2) (to appear)," *IEEE Transactions on Industrial Informatics*, 2017.

[27] H. Tidefelt and T. B. Schön, "Robust Point-mass Filters on Manifolds," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 540–545, 2009.

[28] D. E. Tyler, "Statistical Analysis for the Angular Central Gaussian Distribution on the Sphere," *Biometrika*, vol. 74, no. 3, pp. 579–589, 1987.

[29] W. Feiten, M. Lang, and S. Hirche, "Rigid Motion Estimation using Mixtures of Projected Gaussians," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, 2013.

[30] A. Genz, "Numerical Computation of Rectangular Bivariate and Trivariate Normal and t Probabilities," *Statistics and Computing*, vol. 14, no. 3, pp. 251–260, 2004.

[31] A. Kume and A. T. A. Wood, "On the Derivatives of the Normalising Constant of the Bingham Distribution," *Statistics & Probability Letters*, vol. 77, no. 8, pp. 832–837, 2007.

[32] A. T. A. Wood, "Estimation of the Concentration Parameters of the Fisher Matrix Distribution on SO(3) and the Bingham Distribution on S_q, q$\geq$ 2," *Australian Journal of Statistics*, vol. 35, no. 1, pp. 69–79, 1993.

[33] G. Pagès and J. Printems, "Optimal Quadratic Quantization for Numerics: The Gaussian Case," *Monte Carlo Methods and Applications*, vol. 9, no. 2, pp. 135–165, 2003.

[34] G. Kurz, I. Gilitschenski, F. Pfaff, and L. Drude, "libDirectional," 2015. [Online]. Available: https://github.com/libDirectional