16th International Conference on Information Fusion
Istanbul, Turkey, July 9-12, 2013

Rigid Motion Estimation using Mixtures of Projected Gaussians

Wendelin Feiten[1], Muriel Lang[2] and Sandra Hirche[2]

***Abstract*—Modeling the position and orientation in three-dimensional space is important in many applications. In robotics, the position and orientation of objects as well as the rigid motions of robots are derived from sensor data that are uncertain. The uncertainties of these sensor data result in position and orientation uncertainties that can be very widely spread or have several peaks.**

In this paper we describe a class of probability density functions (*pdf*) on the group of rigid motions that allows for modeling wide-spread and multi-modal *pdf* and offers most of the operations that are available for the mixtures of Gaussians on Euclidean space. The use of this class of *pdf* is illustrated with an example from robotic perception.

I. Introduction

In many applications, the position and orientation of objects in the three-dimensional space or the motion of objects through the three-dimensional space need to be modeled. Various approaches to this task have been explored in the past, in different areas.

In the estimation of the motion of airplanes, generally the rotation is described by the three angles of roll, pitch and yaw. In computer graphics, the rotation in three dimensions is generally described by unit quaternions. In robotics, the Rodrigues vector is often used as a parameterization of the three-dimensional rotation. These rotation representations are then combined with the translation to model the rigid motion, which is equivalent to the position and orientation in space. The standard representation in mathematics and also in robotics is the rotation matrix, in robotics generally enhanced with an additional column representing the translation.

In most cases, where position and orientation are derived from measurements, this is not a certain information, but it depends on random observations. This means that in addition to the parameterization of the rigid motions a probability density function (*pdf*) over these parameters needs to be given.

In order to cover a wide range of applications, the following requirements should be satisfied by such a class of *pdf*:

Expressive Power: The class should be able to approximate any *pdf.*

Information Fusion: For two *pdf* from the class pertaining to the same pose, there should be an algorithm to calculate the resulting pose estimate as a *pdf* from this class that best explains both prior estimates.

Information Propagation: For two *pdf* pertaining to two subsequent motions, there should be an algorithm to calculate the *pdf* of the composition of the two motions.

Efficiency: The algorithms should be fast enough and the space requirements should be small enough to allow for the use of the *pdf* in real time applications.

In section II we will investigate how these requirements are addressed in different approaches and draw our conclusions as to the concepts we want to combine and extend in our approach. The resulting class of *pdf* together with the algorithms according to the requirements is then developed in the subsequent sections. In section III we explain how the Gaussian distribution in $\mathbb{R}^3$ is used to represent a peaked *pdf* on the rotations and how this approach generalizes to represent also widely spread or multimodal *pdf* using mixture distributions (we call this the *Mixtures of projected Gaussians* or *MPG*). We then extend this from rotations to the rigid motions in section IV. Section V describes the information fusion and the uncertainty propagation with the mixture distributions on the rigid motions. To keep the number of mixture elements small, similar elements are merged and negligible ones are disregarded. A background noise is introduced in section VI and also an approach to keep the computation time at a reasonable size. In section VII we then illustrate the use of this class of *pdf* in the field of robotic perception and conclude in section VIII with a short summary of the properties of the new class of *pdf* and the potential further development.

This presentation of our new class of *pdf* on rigid motions builds on our previous publications [1], [2], [3] and gives an overview of the approach.

II. Related Work

Representing the position and orientation of objects in three-dimensional space, or equivalently, representing the rigid motion, has been under investigation for such a long time that giving a complete outline of the history seems inadequate. Admitting that much more work has been done, and giving credit to it, we report here the work that we used to develop our approach. Along with a short assessment of the crucial ingredients of the respective work, we explain which of them we decided to integrate in our approach and why.

The first design decision is, which parameterization of the rotations and of the rigid motions to use.

Rotation matrices are very well understood and easy to visualize (their columns form the basis of the new coordinate frame). Also, extended with the translation vector as an additional column and an additional row $(0,0,0,1)$ to a homogeneous transformation matrix, they represent the rigid motions in a way that composition of motions is done by matrix multiplication. However, the rigid motions have inherently only six degrees of freedom, and this representation has twelve parameters. This means that six equality constraints are needed on the parameter space to determine the valid

[1] W. Feiten is with Siemens Corporate Technology, Siemens AG, D-81739 Munich wendelin.feiten@siemens.com

[2] M. Lang and S. Hirche are with the Institute for Information-Oriented Control, Faculty of Electrical Engineering and Information Technology, Technische Universität München, D-80290 Munich muriel.lang@tum.de

978-605-86311-1-3 ©2013 ISIF

subset of parameters. For this reason we decided against this representation.

Euler angles, for example the parameter set *roll, pitch, yaw*, need no equality constraints to reduce the dimension, because the rotation has three degrees of freedom. The Euler angles work well enough as long as their values remain small. If they become large, the parameterization can become singular. There is an orientation for which the parameters can not be uniquely determined (gimbal lock). Since we want to be able to deal with a wide range of orientations, we decided against Euler angles.

Rodrigues vectors, like Euler angles, use only three parameters for the representation of the orientation. This means that no equality constraints are needed, and that a *pdf* can be induced on the rotations from a *pdf* on $\mathbb{R}^3$ (see [4]). But for Rodrigues vectors there is no efficient way, to the best of our knowledge, to determine the parameters of a composition of rigid motions from the parameters for each individual motion. This also makes it technically more difficult to determine the *pdf* of the composition from the *pdf* of the two components.

The **unit quaternions** represent the rotations using four parameters, so one equality constraint is needed to reduce the dimension. The composition of rotations corresponds to multiplication of unit quaternions, and there is no singularity (except for one symmetry that is easy to handle). They have been used for a long time (see [5]) and are now in wide use in the computer graphics community (see [6]). Not only can the rotation be described with quaternions, but also the rigid motions can be described by **dual quaternions** (they will be explained below). The dual quaternions only need one equality constraint, are unique for a rigid motion (except the easy to handle symmetry in one quaternion) and have a closed form for the composition of rigid motions. Their advantage is that the composition of rigid motions corresponds to the multiplication of dual quaternions, so that there is a closed form for the parameters of the composition (see [6]). Goddard and Abidi [7], [8] use this formalism to capture the correlation between rotation and translation, and also use the derivatives to estimate the dynamical state of a system. The dual quaternions seem to us to be the best match with our requirements.

This takes us to the decision which distribution types to use.

A very common distribution type to describe arbitrary *pdf* is the description with a **sample set**. The expressive power is high, composition is easy, but this comes at a price. In high dimensions (and for wide spread distributions, six dimensions is high) the sample sets need to be very large in order to be representative of the *pdf*. Also, the fusion of two particle sets is not straight forward and computationally expensive. So we decide to keep the sample set description as a fallback option, but to develop a representation that is still general, but more efficient in terms of memory and computation time.

Since we want to use the same operations on the *pdf* of our class that are available for the Gaussians and the mixtures of Gaussians, we specifically look at those distributions that are derived from the Gaussian distributions in Euclidean space. Basically we distinguish two different approaches.

In the first approach, the *pdf* is defined by a Gaussian in $\mathbb{R}^4$ with zero mean, taking the values on the unit sphere $S_3 \subset \mathbb{R}^4$ and renormalizing with the value of the integral over the sphere. This is also known as the Bingham distribution. This type of distribution has been used by [9] and by [10]. One convenient property of this distribution is that the values for antipodal points on the sphere are identical. This makes sense because antipodal points (more precisely the unit quaternions corresponding to antipodal points) represent the same rotation. However, two objections need to be made. The first one is that calculating the integral is very expensive (see [11]). Glover [10] therefore computes a lookup table of normalizing constant approximations using standard floating point arithmetic. The second objection is that the distribution resulting from a composition of motions is not a Bingham distribution, so an additional approximation step is needed to stay in the Bingham distributions.

In the second approach, the unit sphere is treated as a manifold, and is in turn (locally) parameterized by suitable tangent spaces. The value of the *pdf* on the unit sphere is defined by the value of a Gaussian on the corresponding tangent space. This approach has for example been taken by Choe [12] and Goddard [7]. Both approaches do not meet all of our requirements: Choe does not take the translations into account, and Goddard uses one Gaussian on one tangent space only. This is a suitable approach in case that the initial estimate and the information to be fused is highly certain, but we need to be more general.

This requirement can be met by using mixture distributions. Mardia et al. [13] use a mixture of bivariate von Mises distributions to describe a general distribution to specify the secondary structure of proteins. They use a *pdf* on the torus, not on the sphere, but the technique carries over.

Finally, we carry to our approach the common technique of approximating the *pdf* of a non-linear function of random variables with Gaussian distributions. We use the linearization based on the Jacobian of the non-linear function. Alternatively, the approach from unscented Kalman filters used by Kraft [14] could also be carried over to our approach.

Combining the selected techniques from the related work and adding a fair bit (as described below) we arrive at the following approach:

The rigid motions are parameterized with dual quaternions. The rotation part of the dual quaternion is locally parameterized by a tangent plane at a tangent point, the bijection between tangent plane and unit sphere is given by the central projection. The *pdf* on the dual quaternions is a mixture distribution. Each element of the mixture is derived from a Gaussian on the six-dimensional Euclidean space that is the Cartesian product of the space of translation vectors and the tangent plane.

This approach allows for carrying over many operations from Gaussian mixtures to our new class of *pdf*. Because they are derived from the central projection of Gaussians we call them *Mixtures of Projected Gaussians*.

III. Modeling Rotation Uncertainties

For didactic reasons, and because modeling the uncertainty of orientation in three dimensions is an interesting topic in itself, we first restrict ourselves to investigate the rotation part of a rigid motion. We start with a short review of the parameterization of the rotations by unit quaternions.

A. Rotation Represented by Unit Quaternions

A quaternion is a number $\mathbf{q} = a + bi + cj + dk$, where i, j, k are three imaginary parts and $a, b, c, d \in \mathbb{R}$. Hence we can identify the quaternions $\mathbb{H}$ with $\mathbb{R}^4$, and the unit quaternions with the 3D unit sphere $S_3 \subset \mathbb{R}^4$.

The unit quaternion

$$\mathbf{q} = \cos(\theta/2) + \sin(\theta/2)(ui + vj + wk) \tag{1}$$

represents the rotation by θ around the unit length vector $(u, v, w)^\top$ in the following way.
A point $\mathbf{p} = (x, y, z)^\top \in \mathbb{R}^3$ is represented as the imaginary quaternion

$$\mathbf{q}_\mathrm{p} = xi + yj + zk. \tag{2}$$

The rotated point $\mathbf{p\prime}$ is represented by the imaginary quaternion

$$\mathbf{q}_{\mathbf{p\prime}} = \mathbf{q} * \mathbf{q}_\mathrm{p} * \overline{\mathbf{q}}, \tag{3}$$

where $\overline{\mathbf{q}} = a - bi - cj - dk$ is the conjugate of $\mathbf{q}$ and $*$ is the quaternion multiplication. From (3) it follows that $\mathbf{q}$ and $-\mathbf{q}$ both represent the same orientation. This is the reason, why we prefer a probability density function on S_3 that is antipodal symmetric. To avoid confusion about which rotation quaternion to use, we always choose the one, which ensures shortest rotation, i.e. the one in the same hemisphere, except for 180° rotations, where both rotation quaternions are on the equator.

B. Probability Density Function on S_3

We derive the *pdf* on S_3 from the multivariate Gaussian distributions $\mathscr{N}(\mu, \Sigma)$ on a tangent space to S_3. The tangent space is in our case 3D, and it is determined by the tangent point $\mathbf{p} \in S_3$ and a base for the tangent space. For the point $\mathbf{p}_0 = (1, 0, 0, 0)^\top \in S_3$ we define the base of the tangent space to be

$$\mathbf{B}_0 = \{\mathbf{b}_1^0, \mathbf{b}_2^0, \mathbf{b}_3^0\} = \{(0,1,0,0)^\top, (0,0,1,0)^\top, (0,0,0,1)^\top\}. \tag{4}$$

Points on S_3 can be interpreted as quaternions. The canonical base [2] at any other point $\mathbf{p} \in S_3$ is given by

$$\mathbf{B}_\mathbf{p} = \{\mathbf{p} * \mathbf{b}_1^0 * \overline{\mathbf{p}},\ \mathbf{p} * \mathbf{b}_2^0 * \overline{\mathbf{p}},\ \mathbf{p} * \mathbf{b}_3^0 * \overline{\mathbf{p}}\}. \tag{5}$$

This defines a tangent space $TS_\mathbf{p} = TS(\mathbf{p}, B_\mathbf{p})$ that is isomorphic to the Euclidean space $\mathbb{R}^3$. A Gaussian distribution $\mathscr{N}(\mu, \Sigma)$ on $\mathbb{R}^3$ trivially induces a Gaussian distribution on $TS(\mathbf{p}, B_\mathbf{p})$, and that distribution induces a distribution on S_3 via the central projection

$$\Pi_\mathbf{p} : TS_\mathbf{p} \rightarrow S_3$$
$$\Pi_\mathbf{p}(\mathbf{p}_t) = \{-\frac{\mathbf{p}_t}{\|\mathbf{p}_t\|}, \frac{\mathbf{p}_t}{\|\mathbf{p}_t\|}\}. \tag{6}$$

Each point on the tangent space $TS_\mathbf{p}$ is projected to two opposite points on the sphere, as central projection means projection along a straight line through the center of the sphere. Hence, this line crosses the tangent space in one point and the sphere in two points. This fits perfectly the topology of unit quaternions. Points on the sphere, that are orthogonal to the tangent point are defined to be projected to the compact closure of the tangent space.

The central projection $\Pi_\mathbf{p}$ is used to define a probability distribution on S_3 according to a probability distribution on $TS_\mathbf{p}$. A projected Gaussian [1]

$$\mathscr{PG} := \mathscr{N}(\mathbf{p}, \mu, \Sigma) \text{ on } S_3 \tag{7}$$

can be derived from a Gaussian distribution $\mathscr{N}(\mu, \Sigma)$ on $\mathbb{R}^3$ with the projection $\Pi_\mathbf{p}$. To obtain a continuous completion of the *pdf* on S_3, we define the density to be zero for all points on the equator of $\mathbf{p}$. Up to a normalizing factor C, the value of the projected Gaussian at the projected point $\Pi_\mathbf{p}(\mathbf{q})$ is the same as the Gaussian kernel at point $\mathbf{q}$,

$$\mathscr{N}(\mathbf{p}, \mu, \Sigma)(\Pi_\mathbf{p}(\mathbf{q})) := C\mathscr{N}(\mu, \Sigma)(\mathbf{q}). \tag{8}$$

The normalizing factor C is found by taking the integral of this projected *pdf* over S_3 [3]. In the subsection VI-C we give an efficient algorithm for an approximation to this integral of arbitrary precision. The projected Gaussian density *pg* is a probability density function over the rotations in 3D, or equivalently over the orientations on S_3. Figure 1 shows a projected Gaussian density on the sphere S_2 projected from the tangent plane $TS_{\text{North Pole}} \simeq \mathbb{R}^2$ with the north pole as point of contact.

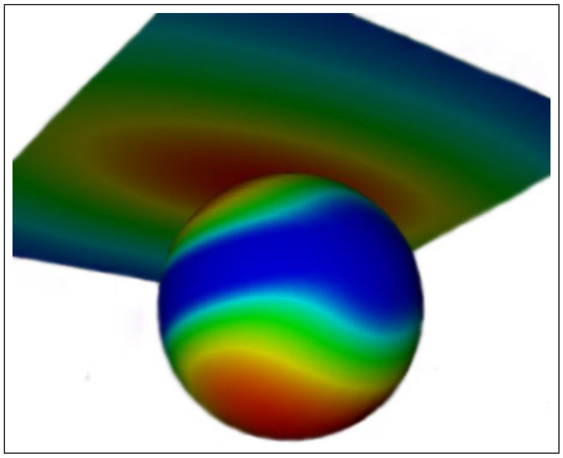

Fig. 1. Projected Gaussian density *pg* on the sphere S_2, obtained by central projection $\Pi_{\text{North Pole}}$ of a 2D Gaussian on the tangent space $TS_{\text{North Pole}}$.

C. Mixture Distribution

The distribution described above is peaked, i.e. it has one orientation with maximal probability density, and away from this maximum the probability decreases rapidly.

Since we need to be a lot more general, we use the analogon of the mixture of Gaussians in Euclidean space, i.e. we extend our class of distributions to combinations of kernels as introduced above.

A mixture of projected Gaussians (MPG) is defined as

$$\mathscr{M} = \sum_{i=1}^{n} \lambda_i \mathscr{PG}_i, \tag{9}$$

where $\mathscr{PG}_i$ are the projected Gaussian kernels and λ_i is the weighting factor with $\sum_i \lambda_i = 1$, $\lambda_i \geq 0\ \forall i$ and $n \in \mathbb{N}$ [1]. With the back projection $\Pi_{\mathbf{p}}^{-1} : S_3 \rightarrow TS_{\mathbf{p}}$, from sphere to tangent space, this mixture $\Pi_{\mathbf{p}}^{-1}(\mathscr{M})$ is identical to a mixture of Gaussians in the Euclidean space.

A MPG is a *pdf* on S_3 that we use to estimate orientations parameterized by unit quaternions.

IV. Modeling Rigid Motion Uncertainties

The results of the previous section III are extended from rotations to rigid motions in this section. The special Euclidean group $SE(3)$ is the group of rigid motions in $\mathbb{R}^3$. A rigid motion is equivalent to a pose consisting of a 3D orientation and a 3D position. First, a parameterization for the rigid motions is needed. Then, the distribution on the special orthogonal group $SO(3)$ of rotations has to be extended to $SE(3)$. We use the dual quaternions to represent the rigid motion on $S_3 \times \mathbb{R}^3$.

A. Rigid Motion Represented by Dual Quaternions

The ring of the dual quaternions is defined as

$$\mathbb{H}_D = \{\mathbf{dq} \mid \mathbf{dq} = \mathbf{q}_{\text{real}} + \varepsilon \cdot \mathbf{q}_{\text{dual}}\ \&\ \mathbf{q}_{\text{real}}, \mathbf{q}_{\text{dual}} \in \mathbb{H}\}, \tag{10}$$

where ε is a dual unit, having the property $\varepsilon^2 = 0$ [15].

As introduces above, a unit quaternion $\mathbf{q}_r$ represents a 3D rotation. A 3D translation is represented by the imaginary quaternion $\mathbf{q}_t = xi + yj + zk$. Together they define a rigid motion $(\mathbf{q}_r, \mathbf{q}_t)$ in 6D. Hence, isomorphisms exist between the representations

$$SE(3) \simeq (\mathbf{q}_r, \mathbf{q}_t) \simeq S_3 \times \mathbb{R}^3 \tag{11}$$

for rigid motions. The dual quaternion representing the rigid motion $(\mathbf{q}_r, \mathbf{q}_t)$, can be calculated explicitly by

$$\mathbf{dq} := \mathbf{q}_r + \varepsilon \cdot \frac{1}{2} \mathbf{q}_t * \mathbf{q}_r, \tag{12}$$

where $*$ denotes the quaternion multiplication. To transform a point $\mathbf{p} \in \mathbb{R}^3$ by a rigid motion parameterized by a dual quaternion $\mathbf{dq}$, the point needs to be represented by an imaginary quaternion $\mathbf{q}_{\text{p}}$ as in (2). Then the transformed point $\mathbf{p\prime}$ is represented by the imaginary quaternion $\mathbf{dq} ** \mathbf{q}_{\mathbf{p\prime}} ** \overline{\mathbf{dq}}$, where $**$ is the dual quaternion multiplication and $\overline{\mathbf{dq}} = \overline{\mathbf{q}}_{\text{real}} + \varepsilon \cdot \overline{\mathbf{q}}_{\text{dual}}$ is the conjugate of $\mathbf{dq}$.

B. Probability Density on $S_3 \times \mathbb{R}^3$

The projected Gaussians, introduced in section III-B to define a *pdf* on S_3 can easily be extended to a *pdf* on $S_3 \times \mathbb{R}^3$ and by (11) to a *pdf* on the rigid motions. The central projection $\Pi_{\mathbf{p}}$ needs to be extended to not only handle the rotation parameter, but also the translation part. We investigate the Cartesian set product of the tangent space $TS_{\mathbf{p}}$ and the space of translations $\mathbb{R}^3$. It is isomorphic to the Euclidean space $\mathbb{R}^6$,

$$TS_{\mathbf{p}} \times \mathbb{R}^3 \simeq \mathbb{R}^6. \tag{13}$$

We define a new central projection from the Cartesian set product $TS_{\mathbf{p}} \times \mathbb{R}^3$ to the Cartesian set product $S_3 \times \mathbb{R}^3$ of the sphere with the space of translations. In the new projection, the mapping of the first three components from the tangent space $TS_{\mathbf{p}}$ to the hypersphere S_3 are equal to the central projection mapping defined in section III-B. The mapping of the last three components in the translation space $\mathbb{R}^3$ is defined to be the identity. We denote the extended central projection with $\Pi_{\mathbf{p}}$ as before,

$$\Pi_{\mathbf{p}} : \underbrace{TS_{\mathbf{p}} \times \mathbb{R}^3}_{\simeq\ \mathbb{R}^6} \rightarrow \underbrace{S_3 \times \mathbb{R}^3}_{\simeq\ SE(3)}. \tag{14}$$

The projected Gaussian can similarly be extended from a *pdf* on S_3 to $S_3 \times \mathbb{R}^3$. Again, everything remains the same, but a Gaussian distribution $\mathscr{N}(\mu, \Sigma)$ on $\mathbb{R}^6$ is projected by the extended central projection. Accordingly, the projected Gaussian

$$\mathscr{PG} := \mathscr{N}(\mathbf{p}, \mu, \Sigma) \text{ on } S_3 \times \mathbb{R}^3 \tag{15}$$

is now defined on the special Euclidean group. Still it holds, that up to a normalizing factor C, the value of the projected Gaussian at the projected point $\Pi_{\mathbf{p}}(\mathbf{q})$ is the same as the Gaussian kernel at point $\mathbf{q}$,

$$\mathscr{N}(\mathbf{p}, \mu, \Sigma)(\Pi_{\mathbf{p}}(\mathbf{q})) := C \cdot \mathscr{N}(\mu, \Sigma)(\mathbf{q}). \tag{16}$$

On the extended projected Gaussians, the definition of a mixture distribution stays exactly the same as in section III-C. The MPG is defined as

$$\mathscr{M} = \sum_{i=1}^{n} \lambda_i \mathscr{PG}_i, \tag{17}$$

where $\mathscr{PG}_i$ are the projected Gaussian kernels and λ_i the weighting factors [1]. Now, a MPG is a *pdf* on $S_3 \times \mathbb{R}^3$ that can be used to estimate poses in the special Euclidean group of rigid motions. Its advantage compared to a single projected Gaussian is its wide applicability.

We do not distinguish between the notations on the special orthogonal group and the special Euclidean group, because in the general case, the restriction to purely orientation is directly obtained by setting the translation 0.

V. Operations On Mixtures Of Projected Gaussians

The fusion of information is an important requirement for many applications using probability densities. We explain two basic operations on MPG, the fusion and the propagation of uncertainty.

A. Information Fusion

Pose information encoded in two MPGs, can be fused in two steps. First, the projected Gaussians have to be transfered to a common tangent space and second, each $\mathscr{PG}_i$ from one mixture $\mathscr{M}_1$ needs to be fused with every $\mathscr{PG}_j$ from the other mixture $\mathscr{M}_2$. The first step is performed by double projection of the Gaussian kernels $\mathscr{PG}_i$ and $\mathscr{PG}_j$ from the tangent spaces $TS_{\mathbf{p}_i}$ and $TS_{\mathbf{p}_j}$ to the common one $TS_{\mathbf{p}_{ij}}$, i.e. $\Pi^{-1}_{\mathbf{p}_{ij}}(\Pi_{\mathbf{p}_i}(\mathscr{PG}_i))$ and $\Pi^{-1}_{\mathbf{p}_{ij}}(\Pi_{\mathbf{p}_j}(\mathscr{PG}_j))$. In case the tangent points anyway represent the same orientation, $\mathbf{p}_i \in \{\mathbf{p}_j, -\mathbf{p}_j\}$, set $\mathbf{p}_{ij} = \mathbf{p}_i$. For all $\mathbf{p}_i \notin \{\mathbf{p}_j, -\mathbf{p}_j\}$, the new tangent point $\mathbf{p}_{ij}$ is calculated as the weighted midpoint on the sphere S_3 between the two former tangent points $\mathbf{p}_i$ and $\mathbf{p}_j$,

$$\mathbf{p}_{ij} = \frac{\lambda_i \mathbf{p}_i + \lambda_j \mathbf{p}_j}{\|(\lambda_i \mathbf{p}_i + \lambda_j \mathbf{p}_j)\|}. \tag{18}$$

Recall, that the weights of the projected Gaussians $\mathscr{PG}_i$ and $\mathscr{PG}_j$ are given by λ_i and λ_j, which express the reliability of the information content of the projected Gaussians.

The fusion of two projected Gaussian densities itself is similar to the fusion of two Gaussian densities $\varphi_1 \sim \mathscr{N}(\mu_1, \Sigma_1)$ and $\varphi_2 \sim \mathscr{N}(\mu_2, \Sigma_2)$ on $\mathbb{R}^n$ pertaining to the same phenomenon [7],

$$\begin{aligned} &\varphi_3 \sim \mathscr{N}(\mu_3, \Sigma_3) \\ &\mu_3 = (\Sigma_1 + \Sigma_2)^{-1} (\Sigma_1 \mu_2 + \Sigma_2 \mu_1) \\ &\Sigma_3 = \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1}. \end{aligned} \tag{19}$$

We can generalize this to projected Gaussian densities $pg_1 \sim \mathscr{N}(\mathbf{p}_1, \mu_1, \Sigma_1)$ and $pg_2 \sim \mathscr{N}(\mathbf{p}_2, \mu_2, \Sigma_2)$ only if the tangent spaces are reasonably close, i.e. the angle between the tangent points should be small enough such that the transfer of projected Gaussians between tangent spaces does not cause too big distortions. The new projected Gaussian kernels $\mathscr{PG}_{ij}$ that come about by successive fusion get new weighting coefficients to evaluate the plausibility, whether the Gaussian kernels to be fused pertain the same phenomenon. The coefficients are calculated as $\lambda_{ij} := \lambda_i \lambda_j \alpha_{ij} \gamma_{ij}$, where

$$\alpha_{ij} = \mathrm{e}^{\left(-a \arccos((\mathbf{p}_i^\top \mathbf{p}_j)^2)\right)} \tag{20}$$

penalizes projected Gaussians with detached tangent points, because that makes it difficult to share a tangent space. The factor $a \in \mathbb{R}$ determines the penalty increase for a growing angle between the tangent points. The factor γ_{ij} is calculated as

$$\gamma_{ij} = 1 - (\mu_i - \mu_j)(\Sigma_i + \Sigma_j)^{-1}(\mu_i - \mu_j)^\top \tag{21}$$

and expresses whether the transferred kernels are compatible according to a coincidence measure, based on the Mahalanobis distance. That means, even when the projected Gaussians can share a tangent space, they still can be incompatible, because differences of the μs and the Σs can make it unlikely that both distributions describe the same phenomenon.

Generally the rotation part of μ in $\mathscr{PG}$ is not zero. Since it is advantageous to work with projected Gaussians that have zero mean in the corresponding Gaussian on the tangent space, we approximate the projected Gaussian kernel by one that is transfered to the according tangent point $\mathbf{p}$ by central projecting to the sphere and then reprojecting to the new tangent space. The centered projected Gaussian kernel is called $\mathscr{PG}_0 := \mathscr{N}(\mathbf{p}, 0, \Sigma)$.

B. Propagation of Uncertainty

Deriving the *pdf* of a composition of rigid motions from the *pdf* of the components is important in robotics. For instance the resulting uncertainty in a kinematic chain needs to be estimated based on the joint uncertainties. In perception, the cumulated effect of sensor uncertainty and feature uncertainty on the pose of an object needs to be estimated.

To calculate this propagation in our framework, we define the composition of two projected Gaussians $\mathscr{PG}_1 := \mathscr{N}(\mathbf{p}_1, 0, \Sigma_1)$ and $\mathscr{PG}_2 := \mathscr{N}(\mathbf{p}_2, 0, \Sigma_2)$ as

$$\mathscr{PG}_3 := \mathscr{N}\left(\mathbf{p}_1 * \mathbf{p}_2, 0, J_\gamma \Sigma_\gamma J_\gamma^\top\right), \tag{22}$$

where J_γ is a Jacobian matrix, defined below, and Σ_γ is the block diagonal matrix of Σ_2 and Σ_1. In formula (22), the Jacobian is $J_\gamma = \frac{\partial f(\varphi_2, \varphi_1)}{\partial(\varphi_2, \varphi_1)}\Big|_{(0,0)}$, where $f(\varphi_2, \varphi_1)$ denotes the reprojection of the product $**$ of the dual quaternions, representing the transformations, that are encoded in $\Pi_{\mathbf{p}_1}(\varphi_1)$ and $\Pi_{\mathbf{p}_2}(\varphi_2)$, i.e.

$$f(\varphi_2, \varphi_1) = \Pi^{-1}_{\mathbf{p}_3}(\Pi_{\mathbf{p}_2}(\varphi_2) ** \Pi_{\mathbf{p}_1}(\varphi_1)) \tag{23}$$

This composition formula allows us to propagate uncertainty, encoded in a projected Gaussian, to another 6D pose by an uncertain rigid motion.

VI. Efficiency Issues

A disadvantage of information fusion is that the number of mixture elements is rapidly growing. For a mixture $\mathscr{M}_1$ having n_1 elements, and another mixture $\mathscr{M}_2$ having n_2 elements, the fused mixture $\mathscr{M}_{12}$ has $n_1 n_2$ elements. To keep the calculation time reasonably small, we replace similar projected Gaussians by a new projected Gaussian, that contains the information of the old ones. We call this procedure merge.

In some applications, there is background noise that can hardly be modeled efficiently by a small number of Gaussian kernels. An example is the estimation of an object pose from a feature on the object, where the feature might be mismatched, i.e. not pertain to the object. Then we want to model the degree of reliability in the weights that we give to the projected Gaussians. But the weights of the mixture elements must sum to 1. Hence, we add a component to the mixture that expresses the uncertainty, that the feature might be an outlier. This uncertainty is best modeled by a component with 'uniform' distribution, as we have no information about the true pose of the outlier.

A third issue of saving memory space and processing time, which will be explained in this section, is an efficient approximation of the renormalization factor.

A. Merge

Reducing the number of mixture elements by merging can be realized iteratively. All pairs of projected Gaussians in the mixture $\{\mathscr{PG}_i, \mathscr{PG}_j\}, i \neq j$, are tested for similarity according to the Kullback-Leibler discrimination (KLD). According to [16], this is the ideal cost function for Gaussian mixture reduction. For a reliable comparison, each pair needs to be transferred to a common tangent space $TS_{\mathbf{p}_{ij}}$ as in V-A. Instead of calculating the KLD directly, which is computationally expensive, we define a dissimilarity measure $B((\lambda_i, \mathscr{PG}_i), (\lambda_j, \mathscr{PG}_j))$ analog to the one [17] proposes, but for MPG instead of regular Gaussian mixtures. The dissimilarity measure B is an upper bound for the (non-symmetric) Kullback-Leibler discrimination of $(\mathscr{PG}_i, \mathscr{PG}_j)$ and $(\mathscr{PG}_j, \mathscr{PG}_i)$. The two mixture elements, which are the closest, according to the dissimilarity measure B, are replaced through their moment-preserving merge $(\lambda_{ij}, \mathscr{PG}_{ij})$, with

$$\begin{aligned}
&\lambda_{ij} = \lambda_i + \lambda_j \text{ and } \mathbf{p}_{ij} \text{ defined as in (18)} \\
&\mu_{ij} = \frac{1}{\lambda_{ij}}(\lambda_i\mu_i + \lambda_j\mu_j) \\
&\Sigma_{ij} = \frac{1}{\lambda_{ij}}\left(\lambda_i\Sigma_i + \lambda_j\Sigma_j + \frac{\lambda_i\lambda_j}{\lambda_{ij}}(\mu_i - \mu_j)(\mu_i - \mu_j)^\top\right).
\end{aligned} \tag{24}$$

B. Background Noise

The unit distribution $\mathscr{U}$ is introduced as first component to the mixture

$$\mathscr{M} = \lambda_0\mathscr{U} + \sum_{i=1}^{n} \lambda_i \mathscr{PG}_i, \tag{25}$$

where $\sum_i \lambda_i = 1$. On compact sets the distribution $\mathscr{U}$ would be the uniform distribution, but in our case we have to define a distribution over $SE(3) = S_3 \times \mathbb{R}^3$. To emphasize the definition of the distribution on $SE(3) = S_3 \times \mathbb{R}^3$, we call $\mathscr{U}$ unit distribution. It is characterized by the fusion with other kernels according to [3]

$$\begin{aligned}
\mathscr{PG} &= \mathscr{U} \times \mathscr{PG} \\
\mathscr{PG} &= \mathscr{PG} \times \mathscr{U} \\
\mathscr{U} &= \mathscr{U} \times \mathscr{U}.
\end{aligned} \tag{26}$$

The fusion of pose information is not affected by the introduction of the unit distribution, as it has the properties of a unit element. If the unit distributions of both mixtures are fused, $\mathscr{U}$ remains unchanged and gets the sum of the two former weights as new weight. In case that a unit distribution $\mathscr{U}_0$ with weight λ_0 and an arbitrary projected Gaussian kernel PG_k with weight λ_k is fused, we obtain PG_k with the new weight $\lambda_{0k} = \lambda_0\lambda_k$. On iterative fusion, the unit distribution weight converges to 0 exponentially fast with the number of steps.

C. Renormalization

In section III-B we mention a renormalization constant C, that makes the difference between a Gaussian and a projected Gaussian. To calculate this factor C, we need to integrate the projection over $S_3 \times \mathbb{R}^3$, what is equivalent to calculating

$$I = \int_{\mathbb{R}^6} f(u,v,w) g(u,v,w,x,y,z)\ \mathrm{d}(u,v,w,x,y,z), \tag{27}$$

where g is the density of a Gaussian in $\mathbb{R}^6$ and

$$f(u,v,w) := \frac{1}{(1+u^2+v^2+w^2)^2} \tag{28}$$

is the volume correction term in the integration by substitution. We denote $\mathbf{p} := (u,v,w,x,y,z)$ and use the parameters Σ and μ of the Gaussian density g.

$$I = C_g \int_{\mathbb{R}^6} f(u,v,w)\, \mathrm{e}^{-0.5\left(\mathbf{p}^\top - \mu\right)^\top \Sigma^{-1}\left(\mathbf{p}^\top - \mu\right)}\, \mathrm{d}\mathbf{p}, \tag{29}$$

where

$$C_g = \frac{1}{(2\pi)^3\sqrt{\det\Sigma}}. \tag{30}$$

This integral can either be calculated via Monte Carlo integration or using sparse grids methods [18]. Monte Carlo methods are either slow or provide no exact results. Sparse grids methods overcome the curse of dimensionality to some extent, but have problems with Gaussian densities and are also computationally expensive [19]. Therefore, we take advantage of the special structure of the integrand and adopt a different approach. The factor f is a bell shaped curve depending only on the square of the radius

$$f(u,v,w) = \frac{1}{(1+r^2)^2} \text{ with } r^2 := u^2+v^2+w^2. \tag{31}$$

It can be shown that f can be approximated to arbitrary precision with exponential functions

$$\max_{\mathbb{R}} \left| \frac{1}{(1+r^2)^2} - \sum_{i=1}^{n} a_i \mathrm{e}^{-b_i r^2} \right| < \varepsilon \tag{32}$$

Letting R be the 6×6 matrix with the first three diagonal elements 1 and all other entries 0, instead of the original integral (29), we now calculate two integrals of the type $I(b)$, to approximate I.

$$\begin{aligned}
I(b) &= \int_{\mathbb{R}^6} \mathrm{e}^{-br^2}\, \mathrm{e}^{-0.5\left(\mathbf{p}^\top - \mu\right)^\top \Sigma^{-1}\left(\mathbf{p}^\top - \mu\right)}\, \mathrm{d}\mathbf{p} \\
&= \int_{\mathbb{R}^6} \mathrm{e}^{-b\mathbf{p}R\mathbf{p}^\top - 0.5\left(\mathbf{p}^\top - \mu\right)^\top \Sigma^{-1}\left(\mathbf{p}^\top - \mu\right)}\, \mathrm{d}\mathbf{p} \\
&= \int_{\mathbb{R}^6} \mathrm{e}^{-0.5\left[\mathbf{p}(2bR)\mathbf{p}^\top + \left(\mathbf{p}^\top - \mu\right)^\top \Sigma^{-1}\left(\mathbf{p}^\top - \mu\right)\right]}\, \mathrm{d}\mathbf{p}
\end{aligned}$$

The sum of quadratic terms is again a quadratic term. So in order to complete the square for the sum

$$S := \mathbf{p}\,(2bR)\,\mathbf{p}^\top + \left(\mathbf{p}^\top - \mu\right)^\top \Sigma^{-1} \left(\mathbf{p}^\top - \mu\right), \tag{33}$$

we find the minimum of the sum, i.e. the 'mean value' $\mu_0 = (2b\Sigma R + Id)^{-1}\mu$. Using this new μ_0 in the quadratic form (33) above, we find the constant term

$$C_Q = \mu_0^\top\,(2bR)\,\mu_0 + (\mu_0 - \mu)^\top \Sigma^{-1} (\mu_0 - \mu), \tag{34}$$

such that

$$S = \left(\mathbf{p}^\top - \mu_0\right)^\top \left(2bR + \Sigma^{-1}\right) \left(\mathbf{p}^\top - \mu_0\right) + C_Q. \tag{35}$$

With the shorthand notation $\Sigma_b^{-1} := 2bR + \Sigma^{-1}$, the approximate integral becomes

$$I(b) = \mathrm{e}^{-0.5C_Q} \int_{\mathbb{R}^6} e^{-0.5\left(\mathbf{p}^\top - \mu_0\right)^\top \Sigma_b^{-1} \left(\mathbf{p}^\top - \mu_0\right)} \, \mathrm{d}\mathbf{p}.$$

The integrand is a multivariate normal distribution up to the normalizing factor of a Gaussian in $\mathbb{R}^6$, so the value of the integral is the inverse of that factor, and

$$I(b) = \mathrm{e}^{-0.5C_Q} \left(2\pi\right)^3 \sqrt{det\left(\Sigma_b\right)}. \tag{36}$$

From (30) and (36) we get a closed form approximation for C at little computational cost, with $n = 2$ in (32)

$$C \cong C_g \left(a_1 I(b_1) + a_2 I(b_2)\right). \tag{37}$$

The approximation coefficients a_i and b_i are the same for each projection Π_p and can be calculated offline. Using this method, the calculation time for fusion is 50 times faster than using Monte Carlo with 2000 samples, what corresponds to a similar accuracy.

VII. Illustration of the MPG information fusion

To demonstrate that the framework works properly, we simulate an example of a robotic application. The pose of a box shape object should be determined under the usage of the class of probability distributions mixtures of projected Gaussians. The probability density function of a pose is visualized by drawing a number of samples from the distribution. Hence, we first have to explain how a 6D pose is visualized.

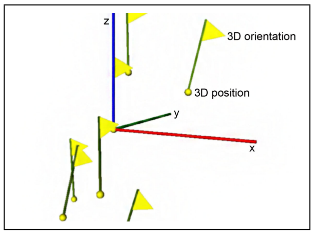

Fig. 2. Flags determine the full 6D Pose and replace drawing coordinate systems at each sample.

Each flag in Figure 2 represents a sample from the estimated probability distribution. It stands at the 3D point of the sample and its orientation determines the orientation of the sample. That is to say, the flag pole equals the z-axis and its pennant looks in direction of the x-axis. Working with right-handed coordinate systems, the y-axis is also determined.

We now show how the wide spread *pdf* is used in the context of pose estimation. If for example an edge of an object is detected in a camera image, there are only certain object poses under which this edge is visible at all. In Figure 3 to the left this is shown, only that for clarity here the possible camera positions are outlined with a dashed line. Similarly, a point feature like the letter F to the right is only visible under certain poses. These visibilities are interpreted as widely spread *pdf* describing the pose of the object. The pose estimate is narrowed down by fusing these *pdf* using the operators described above.

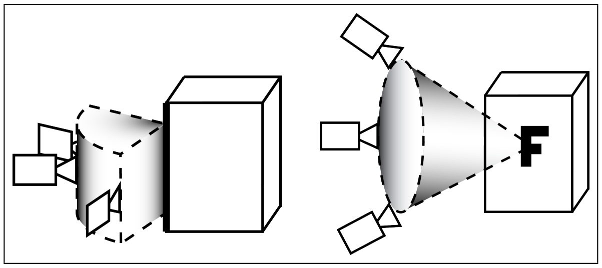

Fig. 3. Viewing cylinder segment of camera positions one the left and viewing cone on the right to detect front left object edge as outer edge, respectively surface feature 'F'.

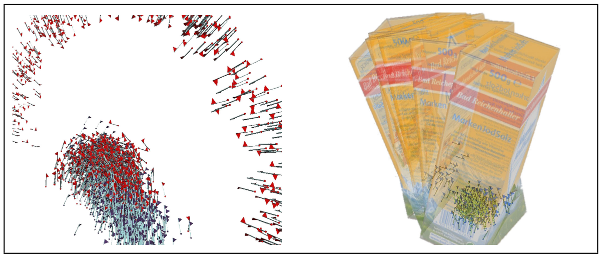

Fig. 4. **left:** Independent object pose estimations using an edge feature (red flags) and a surface feature (blue flags). **right:** Resulting object pose information, after fusion of the red and blue mixtures in left image.

In Figure 4, this principle is illustrated using a simulated salt box. To the left, the red flags depict the possible object poses resulting from an edge of the box, and the blue flags depict the possible object poses resulting from a point feature on the surface of the box.

Note that the edge in the image is interpreted as one of eight short edges on the object, four on the top, for on the bottom. This results in many kernels, that together represent the possible object poses (upside down or upright, facing into four directions, and spread out according to the visibility range). The resulting *pdf* is thus widely spread as shown by the red flags.

We assume that the point feature is unique for the object. This means the corresponding pose *pdf* is more focused and has fewer kernels, as shown by the blue flags.

Fusing the object pose distributions from both features, the impossible poses are eliminated. For example, since the point feature says the object is upright, all poses from the line feature where the object could be upside down are eliminated. Similarly, the poses where the salt box is not pointing towards the camera are ruled out. The mixture resulting from the fusion of both estimates is shown in green. It is already is clearly visibly more peaked. In Figure 4 right the object is shown at the corresponding poses - this is not sufficiently concentrated for most manipulation tasks.

The estimate is sharpened by including more features in the fusion process. In the example shown in Figure 5 we used seven features. Only six of them actually are features of the object, four edge features and two surface features on the front side of the box.

One feature is a mismatch which is not on the box but part of the background noise.

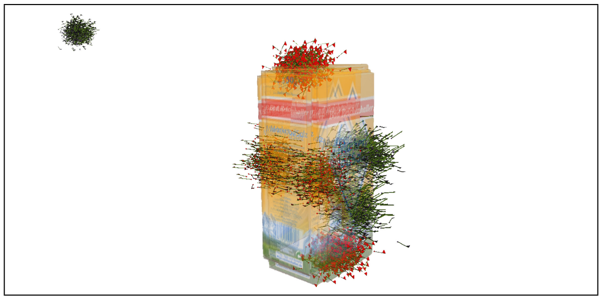

Fig. 5. Seven features used for estimating the pose of the box object. Three Features are surface features, whereof one is an outlier in the background and four features are edge features.

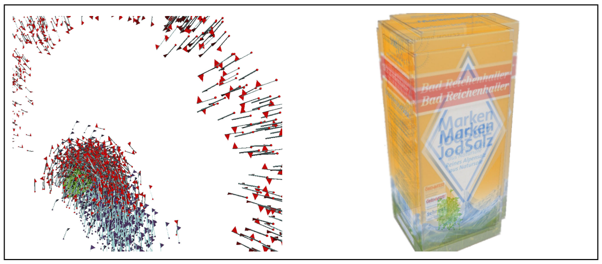

Fig. 6. Estimation of the object pose after fusion of all seven features, visualized by green samples. **left:** For easy comparison, additionally the object pose information from an edge and a surface feature are visualized. **right:** Visualization of target as several samples, drawn from final distribution.

Fusing the *pdf* of these features results in a peaked probability distribution describing the object. Figure 6 shows in both images (in green) the resulting probability distribution.

Compared to the estimate from just two features, shown in Figure 6 on the left, the estimate from seven features, shown on the right, is sufficiently accurate to grasp the object. This is true even with the outlier included in the fusion process, because the possibility of each feature being an outlier, captured in the unit introduced above, makes the fusion process robust with respect to outliers.

VIII. Conclusion

In this paper we describe a class of *pdf* on the rigid motions which is well suited for problems in robotics and robotic perception, but can also be used in other applications. This class of *pdf* has many useful properties. It has a large expressive power, being able to describe wide spread *pdf* and to model the correlation between position and orientation. Further, it has closed form expressions for the fusion of information and for the propagation of uncertainty. The computations are made efficient by special approximation techniques for renormalization and for modeling of background noise. Since the class is based on mixtures of Gaussian distributions, more extensions are possible. We have implemented sampling from the distribution as well as fitting a distribution from this class to a sample set using a variant of the EM algorithm. Further techniques have been adapted from the mixtures of Gaussians to fuse base elements or to drop small base elements. A full implementation of the class of *pdf* and of the operations has been made in Python, and the approach has been validated in simulation. A re-implementation in C++ is under way. The code will be available open source as soon as we finish an easy to handle toolbox.

Acknowledgment

This work was funded in parts under the ARTEMIS Joint Undertaking as part of the project R3-COP, from the German Federal Ministry of Education and Research (BMBF) under grant no. 01IS10004E, and was supported in part within the DFG excellence initiative research cluster Cognition for Technical Systems - CoTeSys (www.cotesys.org).

References

[1] Wendelin Feiten, Pradeep Atwal, Robert Eidenberger, and Thilo Grundmann. 6D Pose Uncertainty in Robotic Perception. In Torsten Kröger and Friedrich M. Wahl, editors, *Advances in Robotics Research*, pages 89–98. Springer Berlin Heidelberg, 2009.

[2] Wendelin Feiten and Muriel Lang. Mpg - a framework for reasoning on 6 dof pose uncertainty. Corporate Technology, Intelligent Systems & Control, Siemens AG, D-80200 Munich, 2011. ICRA Workshop on Manipulation Under Uncertainty.

[3] Muriel Lang and Wendelin Feiten. MPG - Fast Forward Reasoning on 6 DOF Pose Uncertainty. In *7th German Conference on Robotics (ROBOTIK 2012)*, Munich, Germany, May 2012.

[4] Christoph Hertzberg, Ren Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *CoRR*, pages –1–1, 2011.

[5] J. Stuelpnagel. On the parameterization of the three-dimensional rotation group. In *NASA Technical Documents*. NASA, January 1948.

[6] L Kavan, S Collins, Carol O'Sullivan, and J Zara. Dual quaternions for rigid transformation blending. *Technical report TCDCS200646 Trinity College Dublin*, 2009.

[7] James Samuel Goddard. *Pose And Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering*. PhD thesis, University of Tennessee, Knoxville, 1997.

[8] J. S. Goddard and M. A. Abidi. Pose and motion estimation using dual quaternion-based extended kalman filtering. *SPIE Conf. on Three-Dimensional Image Capture and Applications*, 3313:189–200, January 1998.

[9] Seth Teller and Matthew E. Antone. Robust camera pose recovery using stochastic geometry. Technical report, Proceedings of the AOIS-2001, 2001.

[10] Jared Glover, Gary Bradski, and Radu Bogdan Rusu. Monte carlo pose estimation with quaternion kernels and the bingham distribution. Los Angeles, USA, 2011. Submitted to Robotics: Science and Systems (RSS 2011).

[11] J. J. Love. Bingham statistics. In *Encyclopedia of Geomagnetism & Paleomagnetism*, pages 45–47. Springer, Dordrecht, The Netherlands, 2007.

[12] Su Bang Choe. *Statistical analysis of orientation trajectories via quaternions with applications to human motion*. PhD thesis, UNIVERSITY OF MICHIGAN, 2006.

[13] K. V. Mardia, C. C. Taylor, and G.K. Subramaniam. Protein bioinformatics and mixtures of bivariate von mises distributions for angular data. *Biometrics*, 63(2):505–512, June 2007.

[14] E. Kraft. A quaternion-based unscented kalman filter for orientation tracking. *Information Fusion*, 1:47–54, 2003.

[15] Erhan Ata and Yusuf Yayli. Dual unitary matrices and unit dual quaternions. *Differential Geometry - Dynamical Systems*, 10:1–12, 2008.

[16] Air Force Inst of Tech Wright-Patterson AFB OH School of Engineering, Management, and J.L. Williams. *Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter*. Storming Media, 2003.

[17] Andrew R. Runnalls. A kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, 2007.

[18] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, May 2004.

[19] Dirk Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, München, August 2010.