MPG - Fast Forward Reasoning on 6 DOF Pose Uncertainty

Muriel Lang, Wendelin Feiten
Siemens Corporate Technology, Intelligent Systems and Control, Germany

Topic: **Research and Development /** modeling, planning and control
Keywords: pose estimation, probabilistic inference, forward perception

Abstract

Estimation of the pose of an object, or equivalently *estimation of a rigid motion*, is a prerequisite to many tasks in robotics like perception and manipulation. We want to estimate object poses, consisting of orientation and position of a target object, with 6 degrees of freedom (DOF). We use dual quaternions for the representation of the pose and Mixtures of Projected Gaussians as probability density function over all possible object poses.
The framework can deal with widely spread density functions and provides closed form calculations of their fusion. Further the framework allows for the compositions of motion. In this paper we present two improvements of our framework: an explicit treatment of uncertainties due to approximations and a fast integration over the 6-dimensional space of poses. Finally the improved framework is demonstrated using different types of features in combination.

1 Introduction

The first prerequisite for object pose estimation is an appropriate *representation for the pose*. The probably most common representations of rotation in three dimensions are rotation matrix, Euler angles, Rodrigues vector and unit quaternions. For a detailed explanation of our decision to use dual quaternions as pose representation, see [1].

Further we need a *probability density function* in the Euclidean space that fulfills our requirements. These are independence of the coordinate system, fusion and composition of weak information and a small number of parameters to assure quick calculation time. A particle set might be the most frequently used approach, but requires a large number of samples and there is no analytic formula to fuse probability density functions. Since a single sensor does not provide sufficient certain information, a technique is needed to fuse a number of weak estimates of the pose that have a widely spread probability density function. Antone [2] suggests to use the Bingham distribution in order to represent weak information. By now it is known that propagated uncertain information only can be approximated by Bingham distributions. Love [3] states that the renormalization of the Bingham distribution is computationally expensive. Glover [4], who also works with a mixture of Bingham distributions, recommends to create a precomputed lookup table of approximations of the normalizing constant using standard floating point arithmetic. Mardia et al. [5] use a mixture of bivariate von Mises distributions. They fit the mixture model to a data set using the expectation maximization algorithm as this allows for modeling widely spread distributions. We chose Mixtures of projected Gaussians (MPG), which is a convex combination of density functions, called projected Gaussians (PG), see Figure 1.

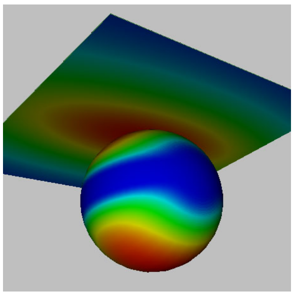


Figure 1: PG on S_2

To construct such a PG, we select a tangent point on the sphere S_3, which is represented by a unit quaternion, and then take the Cartesian set product of the according 3D tangent space with the 3 dimensional space of translations $\mathbb{R}^3$. In this 6D Euclidean space a 6D Gaussian distribution is defined. When we project the Gaussian back to the unit sphere S_3 by central projection and renormalize the result, this induces a distribution over 6D poses. The MPG was first presented by Feiten et al. [6] and then expanded in [7]. With these requirements, we were able to develop a framework for pose estimation with weak information.

2 Forward Perception in the Sensor Model

We constructed a *sensor model* to be able to estimate the pose of a target object in $S_3 \times \mathbb{R}^3$ given the data we receive from the robot's stereo camera system and a 3D sensor. The robot already has a 3D model of the target object and was taught several feature points on its surface. As our framework is open to use different feature types, we worked with SIFT and edge features. From the data taken by the robot, we obtain information about the *position and orientation of features*, that are detected by the cameras and recognized to lie on the object's surface. The sensors of the robot provide 3D points of features, which are assigned to lie on the target object. Further we need the angle of the rotation in the image for each feature. We know from heuristics the positions from which the features can be recognized. Thus with this angle we can model the probability distribution describing all possible orientations. In addition with the appropriate 3D point describing the position of the feature we obtain the mixture distribution which determines each feature pose. In our framework a feature always contains 6D pose information. Our approach includes 6D information but also allows for the restriction to 3D information about the position. In this case the unit distribution has to be set as probability density function to describe the orientation.

As the feature was recognized on the surface of the target object, we can also deduce the rigid motion that transforms from feature to object pose from the 3D model of the object. This shows the need of composition of information. As the probability distribution for the object pose, which we get from one feature is widely spread, we want to fuse the information we get from several features in order to obtain a sufficiently peaked probability distribution. The advantage of this approach is that iteratively features can be added until a given threshold for the exactness of the estimated pose is reached. We call this *forward reasoning*, since we do not need to go back to original measurements. The framework does not require a certain feature type. Any pose information we get can be added.

2.1 The Role of the Identity in a MPG

As one can never be sure about the validity of the recognized features we decided to introduce a kind of white noise to *model the weakness of the information* we get from the robot's sensors. This was realized by including the unit distribution as a component U to the mixture M estimating the feature: $M = \lambda_0 \cdot U + \lambda_1 \cdot PG_1 + \ldots + \lambda_n \cdot PG_n$, $\sum_i \lambda_i = 1$, $\lambda_i \geq 0 \forall i$. We add the identity to the convex combination of projected Gaussian kernels as first entry and renormalize again over the sum, because the identity among probability density functions corresponds to the unit distribution in a mixture. Calculation with the Mixture elements is as follows: $U \times PG = PG$, $PG \times U = PG$, $U \times U = U$.

When we want to *fuse pose information*, which we get from different detected features, that are expected to be part of the same object, we fuse each summand of the first with each summand of the second convex combination. For an explanation of the algorithm of fusion see [1] or [6]. The new weight π_{ij} for each resulting mixture element is the product $\pi_{ij} = \gamma_{ij} \cdot \lambda_i \cdot \lambda_j$, where λ_i and λ_j are the weights of the mixtures' entries when fusing PG_i and PG_j and γ_{ij} is the remaining weight of the mixture element after fusion. Of course, afterwards the sum $\sum_{i,j} \pi_{ij}$ has to be renormalized again to provide a probability density function. In the case that we fuse a unit distribution with weight λ_0 and an arbitrary PG kernel which has weight λ_k, we obtain the same PG with the new weight $\pi_{0k} = \lambda_0 \cdot \lambda_k$. As well as in the case that two unit distributions shall be fused, the weights α_{ij} and δ_{ij} equal 1. In practice, for the mixtures M_1 and M_2 describing the object pose obtained from feature 1 and 2, we start with the weights $\lambda_1 = \lambda_2 = 0.5$ of the unit distributions U_1 and U_2. For the fused distribution $U_{1,2}$ we then get the weight $\pi_{1,2} = \lambda_1 \cdot \lambda_2 = 0.25$. Hence the weight of the unit distribution converges to 0 exponentially fast with the number of steps.

A disadvantage of the fusion of information is, that the resulting Mixture M has $\prod_i \# M_i$ elements. To handle this quickly growing number of the elements of the convex combination, we approximate the mixture. That is to say we drop the mixture elements with the lowest weights π_i until a given threshold is reached, that says how much impreciseness we allow. To retain a probability density function, we arise the weight of the unit distribution about the missing weight.

2.2 Fast Integration of the PG Base Element

Another problem we have to handle, pops up on the determination of a projected Gaussian: The PG base element pg is a density function on $S_3 \times \mathbb{R}^3$. Up to a normalization constant, it is the projection of a 6D Gaussian g on $\mathbb{R}^6$ onto $S_3 \times \mathbb{R}^3$, where the first three components u, v, w are a central projection from a 3D tangent space to S_3 and the last three ones x, y, z are mapped as identity. From the substitution rule for integration we have with $q = (u, v, w, x, y, z)$ and $f(u, v, w) = \frac{1}{(1+u^2+v^2+w^2)^2}$:

$$\int_{S_3 \times \mathbb{R}^3} pg\,(q) \quad \mathrm{d}q = \int_{\mathbb{R}^6} f(u,v,w) \cdot \; g\,(u,v,w,x,y,z)\; \mathrm{d}\,(u,v,w,x,y,z)$$

As mentioned before, g is a Gaussian, so with some suitable constant C, covariance matrix Σ, mean μ, $f(u, v, w)$ as above and taking q to be a column now, we have:

$$C \cdot \int_{\mathbb{R}^6} f(u,v,w) \cdot \mathrm{e}^{-0.5 \cdot (q-\mu)^T \cdot \Sigma^{-1} \cdot (q-\mu)}\, \mathrm{d}\,(u,v,w,x,y,z)$$

One way to calculate the integral is via Monte Carlo integration, where we used the Gaussian itself as a model. But unfortunately, in 6D we needed up to 2000 samples to obtain reasonably small errors. Also numerical integration is no alternative, as it suffers the so-called curse of dimensionality. Integrals in six dimensional space are notoriously hard to calculate, even if sparse grids methods [8] are used. These overcome the curse of dimensionality to some extent, but have problems with Gaussian densities and require more than 2000 function evaluations in 6D (see [9]). We therefore take advantage of the special structure of the integrand and adopt a different approach. The factor of the exponential is a bell shaped curve depending only on the square of the radius $r^2 = u^2 + v^2 + w^2$ and can be approximated with exponential functions. With suitable parameters a_i and b_i we found the maximal distance $\max_{\mathbb{R}} \left| f(r) - \sum_{i=1}^{2} a_i \cdot \mathrm{e}^{-b_i \cdot r^2} \right|$ to be small enough for our purposes. So instead of the original integral, we now need to integrate one or two terms of the type

$$\begin{aligned} I_a &= \int_{\mathbb{R}^6} a \cdot \mathrm{e}^{-b \cdot r^2} \cdot \mathrm{e}^{-0.5 \cdot (q-\mu)^T \cdot \Sigma^{-1} \cdot (q-\mu)} \, \mathrm{d}q \\ &= \int_{\mathbb{R}^6} a \cdot \mathrm{e}^{-0.5 \cdot \left[q^T (2b \cdot R) \cdot q + (q-\mu)^T \cdot \Sigma^{-1} \cdot (q-\mu) \right]} \, \mathrm{d}q \end{aligned}$$

where R is a 6×6 matrix with the first three diagonal elements 1 and all other entries 0.

The sum of quadratic terms is again a quadratic term. So in order to complete the square for $q^T (2b \cdot R) \cdot q + (q-\mu)^T \cdot \Sigma^{-1} \cdot (q-\mu)$ we find the minimum of the sum, i.e. the 'mean value' $\mu_0 = (2b \cdot \Sigma \cdot R + Id)^{-1} \cdot \mu$. Using this new μ_0 in the quadratic form above, we find the constant term $C = \mu_0^T (2b \cdot R) \cdot \mu_0 + (\mu_0 - \mu)^T \cdot \Sigma^{-1} \cdot (\mu_0 - \mu)$. Thus we get for the approximate integral I_a:

$$I_a = a \cdot \mathrm{e}^{-0.5 \cdot C} \cdot \frac{(2\pi)^3}{\det (2b \cdot R + \Sigma^{-1})}$$

For our specific case, we have approximated the integrand in a way that the error vanishes at the boundaries. This is due to the remaining factor which is still a Gaussian and thus very well behaved, and due to the approximation of the other factor being uniform over the range. As we use always the same type of projection, the first factor always is the same and we need to solve the approximation task, i.e. finding the approximating sum of exponential functions, only once. The approximating integral I_a allows for a closed form solution at little computational cost.

3 Results

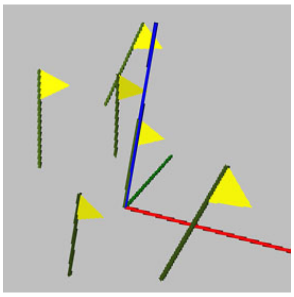
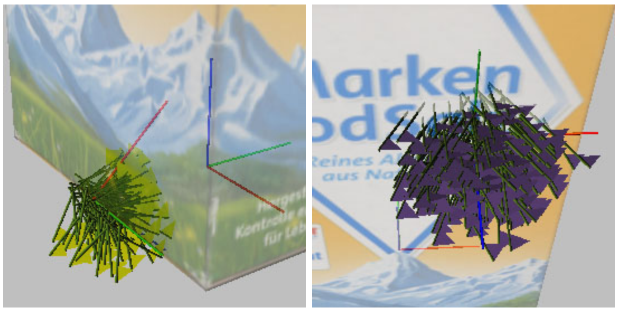


Figure 2: 6D Pose

The probability density function of a pose is visualized by drawing a number of samples from the distribution. So we first explain how a 6d pose is visualized. Each flag represents a sample from the estimated probability distribution. If the sample is drawn from a density function of a feature pose, it has a green pole, if it is the density function of an object pose, it has a silver blue pole. The flag 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. As we only work with right-handed coordinate systems, this determines the direction of the y-axis.

We know from heuristics that a high percentage of SIFT features can be recognized from poses which are rotated up to 20 degrees from head-on view, edge features can be rotated 45 degrees around the axis equal to the edge (see Figure 3).



(a) Edge Feature (b) SIFT Feature

Figure 3: Visualization of Features

From the edge feature 3(a) one can not deduce whether the object stands upside down or not. Thus we have to estimate both possible directions and all resulting object poses.

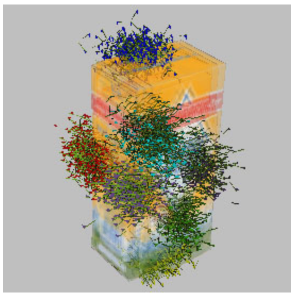
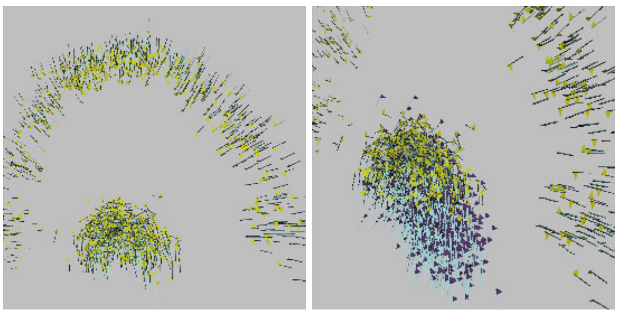


Figure 4: Features

In Figure 4 the seven features we used in this example are visualized. Four edge features and three SIFT features on the front side of the salt box were recognized. That the probability distributions are more widely spread comes from the uncertainty of pose estimation with the stereo camera system of the robot.

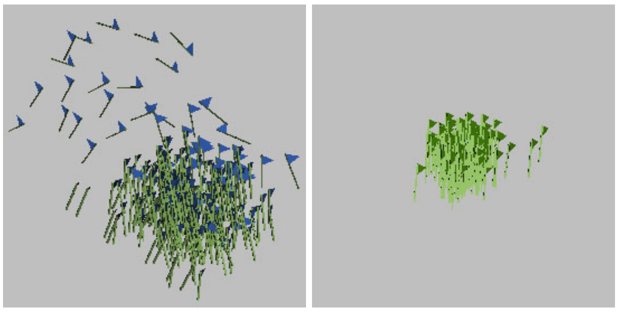
In Figure 5 one sees the object poses resulting from a SIFT and an edge feature. Here we calculated the pose estimation for each feature alone. In the first image we estimated the object poses resulting from the lower front edge feature, visualized through yellow flags in Figure 4.



(a) Object Poses from Edge (b) Object Poses Edge and SIFT

Figure 5: Estimation of Object Pose from Single Features

As we do only know whether it was a short or a long edge, which was detected, but do not know whether the object stands straight up or upside down or which edge it was exactly, we model all possible object poses in Figure 5(a). In combination with the object pose estimation resulting from a SIFT feature, shown in Figure 5(b), where we deduced the object poses from the violet SIFT feature, many object poses are not compatible. That means, under the premise that both features belong to the same object, the fused mixture estimating the object pose already is clearly visible more peaked (see Figure 6).



(a) Fusion of 2 Features (b) Fusion of 7 Features

Figure 6: Estimation of Object Pose after Fusion

The samples in Figure 6(a) are drawn from the fused Mixture we receive out of the yellow and violet Mixtures for the object pose. Figure 6(b) visualizes the result of the algorithm after dealing with all seven feature shown in Figure 4.

Figure 7: Object Pose

Finally in the last figure, the same probability distribution as in Figure 6(b) is shown, which describes the object poses, estimated through fusion and dropping of the density functions we received from the seven feature. The samples drawn from the resulting probability distribution are centered at the estimated object pose. The reason why the salt box surrounding the samples in Figure 7 is diffuse, is that several overlapping salt boxes are visualized at poses drawn from the final distribution.

4 Conclusion and Further Work

In this paper we presented a framework to deal pose uncertainty with 6 DOF. Already with a relatively small number of estimates of features we obtain surprisingly good results. The framework correctly gives small weights to wrong estimates. An aspect that has to be treated in further work is a comparison with large particle sets to assure the correctness and to be able to determine an upper bound of the impreciseness.

References

[1] Muriel Lang. Approximation of probability density functions on the euclidean group parameterized by dual quaternions, January 2011.

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

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

[4] *Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution*, Los Angeles, USA, 2011. Submitted to Robotics: Science and Systems (RSS 2011).

[5] 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.

[6] 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.

[7] *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.

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

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