

Bayesian Orientation Estimation and Local Surface Informativeness for Active Object Pose Estimation

Sebastian Riedel



MASTER'S THESIS

**BAYESIAN ORIENTATION ESTIMATION
AND LOCAL SURFACE
INFORMATIVENESS FOR
ACTIVE OBJECT POSE ESTIMATION**

Freigabe:

Der Bearbeiter:

Unterschriften

Sebastian Riedel



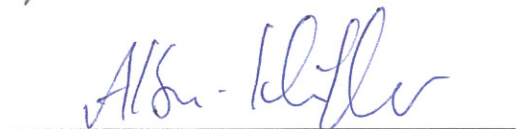
Betreuer:

Simon Kriegel

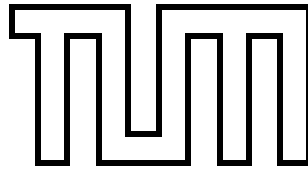


Der Institutsdirektor

Prof. Dr. Alin Albu-Schäffer



Dieser Bericht enthält 102 Seiten, 26 Abbildungen und 12 Tabellen



DEPARTMENT OF INFORMATICS

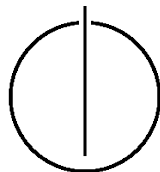
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Bayesian Orientation Estimation and Local Surface
Informativeness for Active Object Pose Estimation

Bayessche Rotationsschätzung und lokale
Oberflächenbedeutsamkeit für aktive Posenschätzung
von Objekten

Author: Sebastian Riedel
Supervisor: Prof. Dr.-Ing. Darius Burschka
Advisor: Dipl.-Ing. Simon Kriegel
Dr.-Inf. Zoltan-Csaba Marton
Date: November 15, 2014



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, November 15, 2014

Sebastian Riedel

Acknowledgments

The successful completion of this thesis would not have been possible without the helpful suggestions, the critical review and the fruitful discussions with my advisors Simon Kriegel and Zoltan-Csaba Marton, and my supervisor Prof. Darius Burschka. In addition, I want to thank Manuel Brucker for helping me with the camera calibration necessary for the acquisition of real test data. I am very thankful for what I have learned throughout this work and enjoyed working within this team and environment very much.

This thesis is dedicated to my family, first and foremost my parents Elfriede and Kurt, who supported me in the best way I can imagine. Furthermore, I would like to thank Irene and Eberhard, dear friends of my mother, who supported me financially throughout my whole studies.

Abstract

This thesis considers the problem of active multi-view pose estimation of known objects from 3d range data and therein two main aspects: 1) the fusion of orientation measurements in order to sequentially estimate an objects rotation from multiple views and 2) the determination of informative object parts and viewing directions in order to facilitate planning of view sequences which lead to accurate and fast converging orientation estimates.

Addressing the first aspect, the Bingham probability distribution over 3d rotations, a parametric probability density function defined on the unit quaternion sphere, is investigated in a black box fusion task based on real data. The experiment shows that the resulting rotation errors are equal to fusion approaches based on pose clustering, a particle filter and a histogram filter while having the advantage of a continuous and parametric probabilistic representation.

To evaluate the informativeness of surface parts and viewing directions of an object with respect to orientation estimation, we present a conceptually simple approach based on the classification of locally computed 3d shape features to viewing directions they could be observed from during a training phase. At first, the applicability of the viewing direction classification to object orientation estimation is investigated. Secondly, the trained classification pipeline is used to determine informative viewing directions and discriminative local surface parts by analyzing the discrepancy between predicted and correct classifications on training data using the Kullback-Leibler divergence as information-theoretic measure of dissimilarity.

Experiments on simulated and real data revealed that the accuracy of the orientation estimation using the proposed method is not yet comparable to state-of-the-art algorithms in the general case of unrestricted viewing directions. The problem was identified as non-robustness of the classification to deviations from the discrete set of training view directions. The evaluation of view and surface part informativeness, however, gives plausible and promising results for building effective view planning criteria.

Glossary

- C inverse regularization strength for logistic regression training, $C \in \mathbb{R}^+$, the smaller, the more regularization. 43, 44, 46, 47, 49
- K number of clusters for K-means clustering. 37, 38, 43, 44, 46, 47, 49
- M number of samples used for sequential Monte Carlo update method. 26, 27, 29, 31, 44
- N_{feat} number of features per view used for object rotation estimation. 40, 41, 43, 44, 46, 49, 51, 53, 57
- N_{sets} number of training point clouds per training pose. 35, 43, 44, 46, 47
- N_{views} number of different training poses for viewing direction based rotation estimation. 35, 37, 38, 43, 44, 46, 47
- V list of component-wise direction vector parameters for a Bingham mixture model. 28
- α list of component weights for a Bingham mixture model. 28, 29, 31, 32, 72, 75
- \mathcal{K} list of component-wise concentration parameters for a Bingham mixture model. 28
- κ single concentration parameter of a Bingham distribution. 29, 31, 32, 40, 43, 44, 46, 49, 51, 53, 72, 75
- BMM Bingham mixture model/distribution. 28, 40
- N_{max} maximum number of components a Bingham mixture model has after applying mixture reduction. 26, 29, 31, 32
- ${}^oT_c^m$ m-th training pose, transformation from camera frame to object frame. 38, 43, 47
- r_f radius for feature estimation on point cloud. 36, 37, 43, 44, 46, 47, 65
- r_n radius for normal estimation on point cloud. 43, 44, 46, 47
- BoW** Bag-of-Words. 12
- FPFH** Fast Point Feature Histogram. 12, 33, 34, 36, 37, 41, 43, 44, 52, 65
- ICP** iterative closest point. 12
- KL** Kullback-Leibler. 25, 57, 58, 60, 61
- LR** logistic regression. 13, 35–37, 43, 47, 57

M+R multiply & reduce. 26, 28, 29, 31, 32, 41, 69

MAP maximum a posteriori. 7, 8, 23, 28, 29, 31, 32, 44, 49, 51–53, 63, 66, 67, 81–83

MI mutual information. 9–11

OvR one vs. the rest. 36

PCL Point Cloud Library. 43

POMDP partially observable Markov decision process. 10, 11

SMC sequential Monte Carlo. 26–29, 31, 32, 41, 44, 69

SVM support vector machine. 12

Contents

Acknowledgments	vii
Abstract	ix
Glossary	xii
1. Introduction	1
1.1. Overview	1
1.2. Conceptual Motivation	2
1.3. Thesis Outline	3
2. Generic Passive and Active Multi-View Pose Estimation	5
2.1. Components of a Multi-View Pose Estimation System	5
2.2. Related Work: View Planning	7
2.3. Related Work: Feature Selection	12
2.4. Summary	14
3. Rotation Estimation using the Bingham Distribution	17
3.1. 3d Rotations	17
3.2. Quaternions	18
3.3. Bingham Distribution	18
3.4. Bingham Mixture Models	22
3.5. Projected Gaussians as Probabilities over Rotations	23
3.6. State Fusion using Bingham Mixture Models	24
3.6.1. Algebraic Fusion	25
3.6.2. Monte Carlo Estimation	26
3.7. Evaluation	27
3.7.1. Evaluation for Gaussian Measurement Model	28
3.7.2. Evaluation for Multimodal Measurement Model	29
3.8. Summary	32
4. Viewing Direction Classification: Application to Orientation Estimation	33
4.1. Method Overview	33
4.2. Viewing Direction Classification	35
4.2.1. Training Setup and Choice of Classifier	35
4.2.2. Choice of Feature	36
4.2.3. Example	37
4.3. Bingham Mixture Measurement Model	39
4.4. Algorithmic Details for Sequential Estimation	40

4.5. Evaluation	42
4.5.1. Parameter Space and Parameter Selection using Simulated Data . . .	42
4.5.2. Evaluation using Real Data	45
4.6. Summary	52
5. Viewing Direction Classification: Application to View Planning	57
5.1. View Informativeness	57
5.2. Model Surface Informativeness	58
5.3. Proof-of-Concept Evaluation of Informativeness Values	63
5.4. Outlook: View Planning Approaches	64
5.5. Summary	68
6. Conclusion	69
 Appendix	 70
A. Complete Rankings for Rotation Fusion Evaluation	71
B. All Sequence Plots for Occlusion Experiment	81
Bibliography	85

1. Introduction

In this chapter, the general motivation and scope of the presented work will be introduced. Relevant, basic terminology will be explained and a short outline of the thesis is given.

1.1. Overview

Recognition and pose estimation of known objects is necessary for many tasks including monitoring and tracking purposes or robotic manipulation of objects. Whereas recognition is the task of deciding which object is present, pose estimation refers to estimating an object's position and orientation in up to three dimensions. If the objects are known in advance, analyzed by the estimation algorithm in an offline training phase and the online application is limited to the a priori known objects, one speaks of model based object recognition and pose estimation. Robotic part handling in industrial applications is a prominent example and commercial use case for such algorithms because industrial manipulation is most often limited to a fixed set of parts known in advance.

Model based recognition and pose estimation have been subject to extensive research since the early 70s (Chin and Dyer [5]) and generally work in two steps. In the offline phase, a representation of the object is built using features derived from training data. In the online phase, incoming sensor data is matched to this representation and by doing so, the desired quantity - for example the object's pose - is measured. The quality of this measurement is affected through several aspects of the measurement process. Aspects involving the sensor directly include for example the inherent loss of 3d information when working with monocular 2d intensity images, loss of color information in 3d range data, limited spatial and temporal resolution, limited field of view and sensor noise. Through the environment, the sensed data can be affected by the scene illumination and occlusions. Lastly, the objects to be recognized and located can share feature characteristics with the environment, among each other or among different views of the same object. This last aspect can lead to classification and pose ambiguities even under perfect environment and noise free sensing conditions.

Aforementioned influences are minimized by designing the extracted features to be more or less invariant to many of these aspects. This way, state-of-the-art algorithms achieve correct recognition results and pose measurements using only a single view of the scene or object. In case of ambiguities or inaccuracies introduced by the environment, the object itself or simply sensor noise, it can be helpful to acquire several sensor measurements and fuse the information obtained by them. Utilizing multiple measurements in general will lead to higher accuracy of the measured quantities. Furthermore, tasks for which the presence of unknown objects is expected often necessitate fusing information from multiple measurements to obtain reliable estimates (for example in object search or scene exploration as explained in Kriegel et al. [19]).

Such multi-view approaches can be divided into passive and active ones. Active multi-view approaches are characterized by some sort of configurability for the sensor data acquisition which is deliberately exploited to obtain optimal conditions for the sequential measurement process. A camera mounted on a robotic manipulator is a simple example for a system where the configurability lies in the choice of the camera pose. Other often considered parameters are camera focus or zoom. From a system's point of view, active multi-view approaches consist of three main components: (1) a component providing measurements given an obtained view, (2) a component fusing measurements of different views and (3) a component deciding how to best configure the data acquisition process for the next measurement. In contrast to this, passive multi-view systems cannot influence the data acquisition process and therefore lack the last component.

The main goal of this thesis is the investigation and development of an active multi-view pose estimation system exploiting the mobility of a 3d depth sensor through mounting it onto a robotic manipulator. While this also involves a hardware and robotic control problem to some extent, the main concern of this work is the software dealing with the perception and planning aspects of such a system. Designing these parts is a complex task as many different aspects come together. This can be showcased when thinking in terms of the aforementioned three components. The first component includes the core computer vision algorithm of the system. It performs feature extraction and often also the matching to the model database. This component is of interest here in so far that it interfaces to the second component via the measurement it provides. The second component is of crucial importance for every multi-view approach in general and in particular for the work done in this thesis. It deals with the probabilistic modeling of the pose space as well as the measurement uncertainty and is investigated in detail. The third component drives the sequential perception-action loop and is a challenging planning problem on the basis of uncertain information. It provides the conceptual motivation for the investigations presented in this thesis and while not being fully evaluated, some insights and proof-of-concept results are presented.

1.2. Conceptual Motivation

To motivate the work presented in this thesis, let us consider the planning aspect of an active pose estimation system a little bit further. Intuitively, the planning is based on some sort of expectation or prediction regarding the measurements which would be obtained for a certain camera configuration. This expectation in turn is based on the more or less known state of the environment and the estimated pose of the object(s). Besides the predictive nature of the problem, the computational complexity and therefore the time necessary to plan the measurement process quickly becomes an issue for practical application.

The problem of computational tractability can be tackled by approximate calculations and by precomputing helpful statistics for the planning based on the a priori known objects. Depending on the amount of precomputation, a planning approach is termed as offline approach (lots of precomputed knowledge) or online approach (no precomputed knowledge). Given for example an object classification task, precomputed knowledge could be provided by means of the most unambiguous viewing direction for every object with respect to a correct classification. This direction would be derived based on the

training images and used in a basic online planning approach by trying to establish this view for the current most likely class. While precomputed statistics can result in large computational savings during online planning, there is an inherent connection between the nature of the precomputed statistics and the flexibility of the planning. In the just mentioned example, partial occlusions as observed in cluttered environments cannot be handled by the planning even if the actual classification is part based and thus can cope with occlusions. The precomputed statistics are on a too coarse level.

One of the conceptual motivations for this thesis was therefore to develop an active pose estimation approach which makes use of finer grained precomputed statistics. In order to plan camera poses in cluttered environments, the precomputed statistics should allow to consider partial occlusions while still enabling to offload some of the computations to an offline analysis phase. The main idea to achieve this behavior is to analyze how informative different features or surface parts of an object are with respect to pose estimation. Given this information the online planning tries to bring such surface parts into the camera's field of view. Because of statistics over surface parts, such an approach would also allow to consider occlusions on a finer level.

Another aspect regarding computational tractability is an efficient probabilistic representation of object poses. This affects the fusion component as well as the planning component. Parametric probability distributions like the Gaussian normal distribution are generally more desirable for efficient measurement fusion than for example sample based representations because they can allow for closed form algebraic solutions. While the translation part of an object's pose behaves nicely in this respect, it is not trivial to define proper probabilistic distributions over 3d rotations.

As the close range visual appearance of objects varies more strongly with the relative rotation between camera and object than with the relative position, object pose estimation is relaxed to object orientation estimation for the presented work. The investigation of a well defined parametric distribution over rotations, the Bingham distribution, and sequential object rotation estimation using this distribution thus became an integral part. Sequential fusion using the Bingham probability distribution is first analyzed for a black box pose estimation algorithm and small involved uncertainties. A second approach to object orientation estimation is then explored using a classification based method. Individual features are classified to viewing directions they could originate from and this way not only the orientation of the object can be estimated but also the aforementioned analysis of surface informativeness can be carried out. Especially for the orientation estimation, the Bingham distribution's unique ability to represent large uncertainties in a parametric way plays an important role.

1.3. Thesis Outline

Chapter 2 will introduce the basic formalism underlying active pose estimation systems. Related work and specific approaches to solve perception planning and feature selection are discussed and evaluated with respect to the outlined motivation. Chapter 3 introduces the Bingham distribution as a probabilistic model for 3d rotations and integral part of the probabilistic representation necessary for the later chapters. Its performance in a passive multi-view fusion task is evaluated against other approaches to orientation fusion, namely

a pose clustering algorithm, a histogram filter (also called discrete Bayes filter) and a particle filter. Chapter 4 introduces a conceptually simple approach to orientation estimation based on feature classification. The approach is again evaluated in a passive (random) multi-view fusion scenario on simulated and real data. Chapter 5 investigates how the previously introduced feature classification can be used to derive a local model of surface informativeness and shows a proof-of-concept for the relevance of these results using a multi-view scenario with simulated occlusion. The findings in this thesis are summarized in chapter 6.

2. Generic Passive and Active Multi-View Pose Estimation

This chapter gives an introduction to the general architecture and relevant components of a multi-view pose estimation system. Relevant design parameters are explained and showcased at various systems described in research literature. Special emphasis is put on systems with an active (decision making) component which drives the sequential pose estimation process. With the established understanding of relevant approaches in the literature, the motivation provided in the introductory chapter is detailed and further put into perspective.

2.1. Components of a Multi-View Pose Estimation System

Dealing with uncertain information lies at the core of every system which uses multiple sources of information. In the general case of our setting, every acquired and processed sensor data (for example an intensity image, a laser scan, etc.) results in a pose measurement z of the object and due to sensor noise and unmet assumptions while processing the sensor data, this measurement will be wrong to some extent. Probability theory is one way to capture and deal with this uncertainty in a principled manner.

Let us define x_t as the system's state vector at time step t . The system's task is to sequentially estimate (x_t, x_{t+1}, \dots) - starting from x_0 - using acquired sensor data. Choosing what exactly is to be estimated and therefore defining the **dimensionality of the state space** $\Omega, x_t \in \Omega$, is an important design parameter. For a classification task, x would be the discrete space of possible class labels, for the following chapters x will be a 3d rotation and in the general case it would be the 6d pose of an object or maybe even the concatenated pose vector of multiple objects. Uncertainty with respect to the current state is formally represented by maintaining not only a single state vector x_t , but a probabilistic belief $bel(X = x_t)$.

By processing new sensor data and gaining more pose measurements z_t , the system's belief should converge to an accurate estimate even though individual measurements are erroneous. In fact, our belief distribution from above is a conditional distribution $bel(X = x_t) = p(X = x_t | z_t, \dots, z_0)$ which represents the state after accumulating the information of all measurements until the current time step. A new measurement z_t is incorporated into the estimate by following Bayes' update rule

$$\underbrace{p(X = x_t | z_t, \dots, z_0)}_{\text{posterior}} \propto \underbrace{p_t(z_t | x_t)}_{\text{measurement model}} \underbrace{p(x_t | z_{t-1}, \dots, z_0)}_{\text{prior}} \quad (2.1)$$

in which we assume the current measurement z_t is conditionally independent of previous measurements given the current state x_t . The measurement model is also often termed

data likelihood function because in practice it is treated with x_t as the variable as z_t is a concrete measurement made at this time step. The prior can be further split up to contain an explicit model of the state dynamics (formally $p(x_t|x_{t-1})$), but since the object is assumed to be static in this thesis and also the discussed related work, we can set the prior equal to the previous time step's posterior.

Equation (2.1) is generally referred to as the measurement update step of a filtering algorithm and its algorithmic implementation is dependent on the **representation and form of the involved belief distributions**. For certain choices of the functional form of the measurement model and prior, algebraic solutions for the posterior distribution can be obtained. The prior which together with the data likelihood leads to a posterior distribution of the same functional form as the prior, is called a conjugate prior for this data likelihood function. Specifically conjugate priors lend themselves to sequential state estimation as given in above equation. The actual implementation of the measurement update step is the information fusion component of our generic multi-view pose estimation system.

A second component is concerned with computing a measurement z_t from acquired sensor data. Formally, it computes a function $z_t = f(\hat{x}_t, a_t, \dots)$ of the true system state \hat{x}_t and some action parameter a_t . The considered action throughout this thesis will be the change in sensor position and orientation, which greatly impacts the acquired sensor data and thus the quality of the computed measurement z_t . The **form and dimensionality** of the measurement space $Z, z_t \in Z$ is a relevant design parameter for the algorithmic implementation of the measurement function f . In many cases, z_t will be the detected 6d pose of an object, but other choices are possible. Eidenberger et al. [10] uses the image coordinates of features as measurements, which results in a more complex but also more powerful measurement model on the information fusion side. For the developed pose estimation in chapter 4, the measurement is a 3d rotation. This component will be termed the measurement component of our generic pose estimation system.

Active state estimation systems are characterized by a third component which intentionally drives the sequential estimation process by choosing a specific action a^* . By definition of our measurement component, this action can influence the resulting measurement, so typically a^* is chosen by reasoning over the expected utility of various actions. The action space $A, a^* \in A$ is a design parameter of the planning component and while the considered action space in this work will be the camera's pose for the next data acquisition (next-best-view planning) the action space could also contain the camera settings (zoom, focus, ...) or even manipulation actions like (re)moving parts or objects. Formally, the planning component computes a function $a^* = g(\text{bel}(x_t), \dots)$ of at least the current belief state. Typically, $g(\dots)$ includes a model of the action's utility as well as the cost associated with performing this action.

To conclude our component overview, figure 2.1 illustrates the resulting perception-action cycle. The next section will give an overview over concrete instantiations of these three components by reviewing related work on active classification and pose estimation approaches.

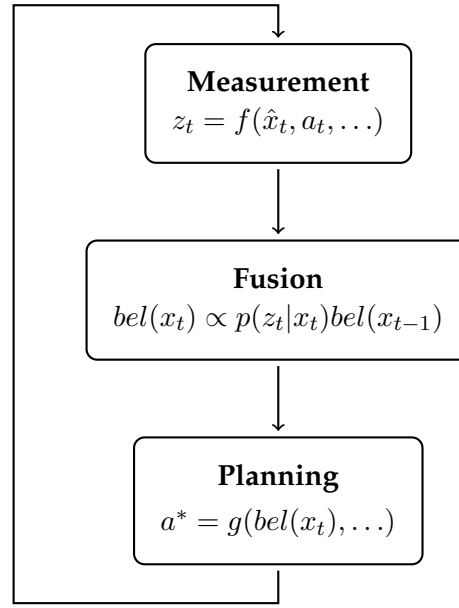


Figure 2.1.: The three standard components of an active state estimation system.

2.2. Related Work: View Planning

An early active vision system was described by Arbel and Ferrie [1] in 2001. They propose an approach for view planning in order to assist object classification. It is based on offline generated entropy maps which suggest viewing directions to allow an unambiguous classification of the object at hand. Class recognition is done via optical flow images obtained by small arc like motions of a camera around the object. Training flow images for every object are used to build a measurement model which is in turn used in a sequential Bayesian scheme as in equation (2.1) to update the belief over class labels. The viewpoint selection strategy is based on the idea of evaluating which training flow image (respectively which training view direction) leads to a correct and maximally unambiguous classification of the object with respect to all other objects. To do so, the information theoretic concept of entropy as a measure of uniformity of a distribution is employed. For every training flow image, the class distribution given the image is obtained, checked for correctness of the maximum a posteriori (MAP) estimate and if so the entropy of the distribution is stored at the image's viewing coordinates (otherwise the maximum entropy value is stored). Intuitively, viewing coordinates for which the entropy is low indicate correct and unambiguously peaked training classifications from this direction for this object. Figure 2.2 taken from Arbel and Ferrie [1] illustrates this behavior. The online application of the computed entropy maps is done by selecting the entropy map for the current MAP class hypothesis, determining a pose estimate of the MAP object (this is also implicitly done using the optical flow images) and then commanding the camera to the minimum entropy pose using the map with respect to the current pose estimate. A comparison of a random with the proposed view selection strategy reveals that the view selection allows for faster and more correct recognition of the objects, especially when the recognition process starts with an ambiguous view. As only discrete distributions are involved, the sequential fusion can

be implemented in an algebraic way. The view planning is heavily based on in advance computed statistics and does not take the current time step's posterior into account except for the MAP estimate.

Another approach based on offline computed discriminative views is described by Sipe and Casasent [34]. During a training phase they build so called Feature Space Trajectories which capture how a globally extracted feature changes in relation to changes of the viewing angle. Areas in the feature space where trajectories from different objects come close to each other indicate ambiguity with respect to object classification. Concerning only a single object, areas where trajectories from far apart viewing perspectives are close indicate ambiguity with respect to the object's pose estimate. Inversely, the most discriminative view for disambiguation of two object classes and the most informative view for the pose estimation of an object are extracted offline and used for planning in a similar fashion than in Arbel and Ferrie [1]. First, they reduce the classification uncertainty by driving the camera towards the most unambiguous view for discriminating between the two most likely class hypothesis. When the object class is known and the pose accuracy is still insufficient, the camera is positioned for the precomputed best view for pose estimation.

While the strength of both described methods lies in the negligible planning overhead in the online phase, the weakness lies in not taking the full state posterior into account. Also, the in advance extracted most unambiguous views are most unambiguous with respect to *all* other classes or object poses. The view which best disambiguates object A from objects B,C,D and E is not necessarily the same view as the best view disambiguating A and from B and C only. While this is a general problem of precomputed knowledge based on one vs. the rest statistics, the particular application during planning adds to the suboptimality of the approach by using only the one or two most likely class or pose hypotheses. As Laporte and Arbel [22] point out, this leads to redundant data acquisition. Lastly, the aforementioned approaches to precomputed statistics are based on globally extracted features and do not take partial occlusions into account.

In 2002, Denzler and Brown [6] described a system for the purpose of object classification, but with a view planning approach based on online information theoretic concepts. Instead of precomputing discriminative views offline, the whole approach is centered around an online evaluation of how much a certain camera action and the resulting observation due to the action is expected to reduce the current state posterior's uncertainty. Formally, this quantity is measured by the mutual information (MI) between the proposed observation and the current state. The MI is defined as the difference between the entropy of the state posterior now and the entropy of the state posterior conditioned on the observation obtained by the action. Evaluating the MI in practice requires integrating over the state and observation space for every action to be considered, which is difficult and computationally costly for continuous state and observation spaces. Even in the case of a simple Gaussian observation likelihood and state posterior, there is no closed form solution for the MI [6]. They thus experiment with a discretization of their continuous spaces and Monte Carlo integration techniques. In both cases, this approach also necessitates a generative model of the observations given a system state and action which they build during a training stage. The experiments conducted involve highly ambiguous objects and in one case very weak observed features (mean gray level over a patch). However, they show that they obtain nearly perfect recognition rates even with such weak features. Further, the MI based planning leads to a drastic reduction in the number of necessary observations in

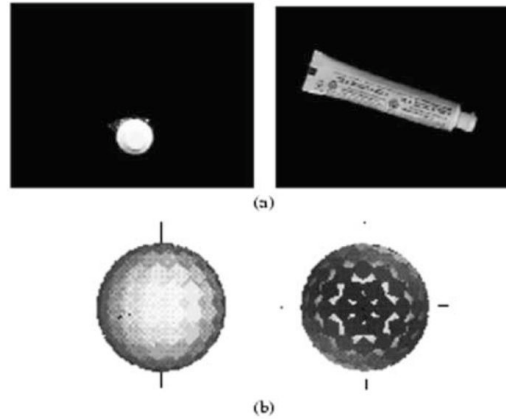


Figure 2.2.: Views of the toothpaste object (top row) and entropy maps (bottom row). The left image shows a view with high entropy (white color on the entropy map) leading to ambiguous classification. The right image shows a view with low entropy leading to unique and correct classification. Source: Arbel and Ferrie [1]

their experiments. Conceptually, this approach results in the optimal action to take with respect to state uncertainty reduction. This does not mean that after every action the uncertainty will be reduced (that depends on the actual observation made), however, in the long-run, convergence of this one step look-ahead strategy can be proven - not necessarily to the true state though [6]. The method can be applied to continuous as well as discrete state and observation spaces, but is fixed to a discrete set of actions. One problem for practical application with higher dimensional state and observation spaces will be the computationally demanding MI computation.

Although the work of Arbel and Ferrie [1] and Denzler and Brown [6] propose interesting and in the case of Denzler and Brown [6] generally optimal strategies for the selection of camera parameters and viewpoints, the system's have only been verified for low dimensional state spaces or treated pose estimation in an unprincipled way because they were designed for an active classification task. For the practical application of recognizing objects and estimating their poses in 3d, view selection and pose estimation go hand in hand as both alter the relative pose between camera and object. A system explicitly modeling the object's orientation in addition to its class is described in Laporte and Arbel [22]. It also introduces a new view selection criterion which is not based on what effect the measurement has on the state posterior (like in the MI criterion), but on "the extent to which an observation is useful in disambiguating two hypotheses [...] based on how probable they are" (Laporte and Arbel [22, p.273]). The pair-wise disambiguation idea is similar to the informative views derived by the Feature Space Trajectories in Sipe and Casasent [34], however here a more probabilistic approach is followed. The view criterion is formalized as a measure of dissimilarity over the measurement distributions for two different state hypothesis but given a fixed action. This dissimilarity measure can be precomputed as no state distribution is involved and the state space is discretized and small enough. On-line application then reduces to weighting the measurement dissimilarities based on how

probable the involved state hypotheses are in the current state posterior. Finally, an action is chosen which, via the dissimilarity in the resulting measurement, disambiguates the most likely hypotheses. For their synthetic experiments, the system's state is comprised of the class label together with two discretized rotation angles specifying pan and tilt of aircraft models. The viewing space is equivalent to the pan and tilt parametrization and consists of the discretized inclination and azimuth angles specifying the camera position on a sphere around the object. Experiments comparing the proposed view selection with a random one and the MI strategy from Denzler and Brown [6] showed that all methods achieve similar pose and recognition accuracy. In comparison to the random strategy, the proposed criterion on average used fewer observations. Compared to the MI criterion, the number of necessary views was similar, but the computation time for the proposed method at least one order of magnitude faster.

The most complete approach described in literature, integrating object classification, 6d continuous pose estimation and view planning, to the best of my knowledge is the work of Grundmann and Eidenberger [9] [10] [11]. Formally, the state space is the joint space of n objects, each with a class label and a 6d pose. By assuming independence between the objects, the distribution factorizes over n single object distributions each defined by a discrete distribution over the class label and a class label conditional distribution over the 6d pose space. Poses in 6d are represented as a concatenation of a 3d position vector and a 3d Rodrigues vector for the rotation. The probabilistic model thereof is a mixture distribution of 6d multivariate Gaussians. Rodrigues vectors specify rotations via an axis-angle parametrization where the angle is given by the length of the vector and the axis by the vector scaled to unit length. The inherent singularity at zero degree rotation, the symmetrical structure of the Rodrigues rotation parametrization and the finite interval $]0, 2\pi]$ for the rotation angle (and therefore the vector length) do not play nicely with the infinite and continuous range over which the 6d Gaussian is defined. While Eidenberger et al. [11] use unconstrained 6d Gaussians for computational efficiency of operations like mixture multiplication, they state that additional steps are required and performed to maintain a proper distribution for the rotational part of the Gaussian. Object recognition and pose estimation is performed using stereo cameras and 3d located SIFT features. The measurement model allows for prediction of feature locations in the image plane given constellations of multiple objects and a camera viewpoint. This way and by building the measurement model not in 6d pose space but in the image coordinate space of the features, occlusions between the objects in the scene can be taken into account. View planning is addressed as decision making problem for an agent in a partially observable environment which needs to select a new observation pose (the planning space is heavily discretized using only in the order of 10 possible camera poses). This leads to the formal framework of a partially observable Markov decision process (POMDP), where an agent tries to maximize the expected future reward by following an optimal action policy which maps the current state belief to an action. In order to derive such a policy, the state and action space, a model of state dynamics given an action, a state observation model and a reward function has to be defined. The reward function defines the immediate reward obtained by taking a certain action in a certain state. For view planning as in Eidenberger et al. [11], the reward function is based on a weighted sum of an uncertainty reduction component similar to the MI criterion, a cost model for the action and a space exploration model rewarding exploratory actions. A policy of maximizing this reward is sought either by considering the reward only for the

next action (1-horizon planning), the next n actions (n-horizon planning) or an infinite series of actions (infinite-horizon planning, typically weighted in a way to account more for short-term reward). Even for simple POMDPs solving for the optimal policy offline can be computationally intractable. In Eidenberger’s case, it is obvious that due to the high dimensional state space over sets of objects, the optimal policy for solving such a POMDP cannot possibly be determined offline. Instead, it is approximately solved online, starting from the current state belief. The different planning strategies (1-horizon, infinite-horizon) and different reward models (with and without exploratory component) are compared to random and incremental (shifting the viewing pose clock-wise) view planning strategies on 200 scenes containing up to 10 cluttered objects. At similar recognition rates, all tested POMDP planning strategies outperform the random and incremental strategy with respect to number of observations as well as cost of movement. Adding an exploratory component to the reward leads to a slightly higher total object recognition rate, however, the increase in pose accuracy for already detected objects is slightly worse compared to non-exploratory driven view sequences. Infinite-horizon planning has no measurable advantage over 1-horizon planning, which can be used to justify a simpler framework based on the direct application of the MI criterion like in Denzler and Brown [6]. Regarding the runtime of the presented method, unfortunately no concrete numbers are given apart from the pose and object detection which takes 0.9 seconds on a 2 GHz Intel multicore processor. The consequent use of parametric distributions, a closed form upper bound approximation for the costly information gain summand of the reward function and the number of real experiments suggests a planning time in the order of seconds.

In summary we have seen offline as well as online approaches to perform better than random approaches to active multi-view object recognition and pose estimation. The benefits of offline (Arbel and Ferrie [1], Sipe and Casasent [34]) and hybrid (Laporte and Arbel [22]) approaches lie in a computationally efficient online application. The practical application of such offline approaches to more complex scenes which involve occlusion, however, has not been demonstrated and by building offline statistics over global features on unobstructed object views these method’s ability to do so is at least questionable. Online approaches based on information theoretic measures of information gain are formally sound but computationally challenging for continuous domains as uncertainty measures like the MI criterion are integrals over the state and observation space. Implementations are based on the discretization of the involved state/observation space, numerical integration techniques (Denzler and Brown [6]) or approximate solutions based on upper bounds (Eidenberger et al. [11]). The system of Eidenberger et al. [11] shows promising real world results in multi object scenarios, but is limited to richly textured objects and has only been demonstrated with a very limited action space (likely due to computational performance reasons). By using descriptive 2d SIFT features, by building a precise measurement model on feature observation level and through the use of parametric probability models, the resulting system is able to explicitly consider occlusions during view planning while still maintaining computational tractability.

From the perspective of view planning, a direction for further research thus lies in finer scale offline computed statistics, which in turn can be used for more flexible online planning. In particular, a measure of informativeness of regions on the object instead of just the informativeness of a viewing direction could allow a flexible online planning method which avoids the computational complexity of purely online information gain

driven methods. The high dimensional and continuous approach in Eidenberger et al. [11] motivates the investigation of parametric probabilistic approaches. Especially the probabilistic treatment of rotations in 3d seems improvable. These aspects and the related work discussed so far further justify the investigations regarding sequential rotation estimation and local surface / feature informativeness pursued in the following chapters.

Another obvious direction for further work would be to extend the work of Eidenberger et al. [11] towards other types of objects (e.g. untextured) using local features on 3d range data instead of intensity based features. However, the different nature of surface geometry in contrast to color and intensity images brings forth some additional difficulties. The local variability of geometry perceived with common sensing approaches like stereo cameras or cheap depth sensors (Microsoft Kinect) is lower than the perceived variability of color information and more prone to noise or missing data. This for example necessitates larger radii for repeatable feature descriptor computation and makes reliable keypoint detection and reference frame estimation more challenging. Many state-of-the-art approaches for object pose estimation based on 3d data therefore often follow a dense correspondence approach instead of keypoint based ones. Dense approaches can work on different levels, for example based on depth data directly (iterative closest point (ICP)) or using dense feature correspondences (Rusu et al. [30]). Dense but randomized methods based on precomputed feature hash tables and generalized voting for object poses work well on 3d features and effectively counteract the higher noise levels present in depth data (Drost et al. [8], Tuzel et al. [37]). Recent work on depth based classification and pose estimation has also shown that dense, 3d feature based approaches can greatly benefit from selecting the most informative regions for processing. In the next section, therefore, related work regarding feature selection strategies for 3d range data is given.

2.3. Related Work: Feature Selection

In the work of Madry et al. [24], a system for category recognition of objects is presented which improves over several other methods by incorporating a measure of feature informativeness. Standard features (they used Fast Point Feature Histogram (FPFH)) describing the local geometry around a point are computed densely over training views of several objects in different categories. The features of all objects combined are clustered into words of similar feature descriptors. A standard Bag-of-Words (BoW) approach to category recognition would now build a model for each category by summarizing all feature to word assignments for features of a category into a histogram over words. As the authors term it, this is a purely quantitative category representation which ignores that for example many features of a bottle are similar to those of a mug because of the round shape. Although the interesting characteristics (the opening of the bottle, the handle and opening of the mug) find their way into the BoW representation, they are likely hidden through the far more often occurring features on the body. The authors therefore propose to train classifiers (they use support vector machines (SVMs)) for every category within every word and suggest to use the classifiers prediction confidence in building a category specific model. To this end, they design a meta-feature by concatenating the most-discriminative (highest prediction confidence) features and use this meta-feature for object category classification. This way, their approach improves the correct categorization rate by 11% with respect to

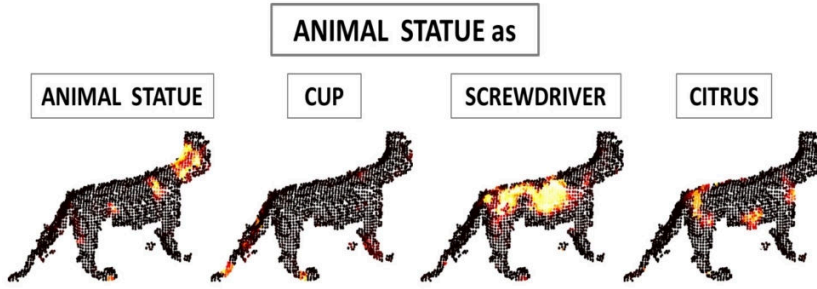


Figure 2.3.: Confidence of features of an animal statue for originating from the animal category and other categories. Whereas neck part is a very informative region for classification to the (correct) animal category, the more round body would characterize a screw driver. Source: Madry et al. [24], ©2013 IEEE

the described BoW based categorization. More interesting for the problem concerned in this work, they obtain a notion of what are the object parts which are most informative for an object category. This regional informativeness with respect to a specific category is proportional to the classification confidence and visualized as heat map plots in figure 2.3 (Source: Madry et al. [24]). One can easily see how this kind of information could be useful for an active vision system by planning views in such a way that the most informative object parts for two competing hypotheses become visible. The implemented classification pipeline in my work is conceptually similar to the one here, except that we train probabilistic logistic regression (LR) classifiers and are interested in informative object parts for orientation estimation rather than classification.

A different approach, specifically targeted towards feature selection for pose estimation is presented in Tuzel et al. [37]. Their pose estimation is an extension of Drost et al. [8]. This is a voting approach based on a hash table which maps point pair features to object poses. During training, point pair features for a model are obtained and the relative pose between the feature frame and the object pose is recorded for the hash table bin the feature descriptor falls into. Online, a point pair of the scene is selected, the model point pairs in the corresponding hash table are retrieved and used to vote for their during training observed object poses. For smooth objects they sample points unconstrained on the surface (surface-to-surface (S2S) features), for industrial object's with many planar regions they sample points on depth edges of the model (boundary-to-boundary (B2B) features). Feature selection is brought into this framework by assigning vote weights to either complete hash table bins or the points on the object model which were used during training. The weight vector is determined using an optimization framework and a dataset of validation 3d scenes on which the pose accuracy with the current weight vector is measured and iteratively improved. The criterion driving this process is finding a weight vector which produces a maximum in pose vote space at the correct pose for all object instances within all scenes. The evaluation of the learned weighting and a baseline uniform weighting reveals a general increase in recognition rate especially for high occlusions and industrial objects, which with uniform weighting have many point pairs leading to ambiguous object poses because of their symmetries and regular structures. Learning a weight vector for model points works slightly worse than learning a weight vector for hash table bins,

which is probably explained by the much larger dimensionality of the problem (weight vector length for bin weighting 5.4K, for model point weighting 39K). However, weighting model points seems interesting from a view planning perspective as the weighting encodes the informativeness of a surface point with respect to pose estimation. While they did not yet target view planning, a complicating aspect would be that the weighting encodes no information regarding the viewing direction the model point needs to be observed from. Also, two model points are needed for a pose estimate and optimally view planning would take this into account by reasoning over the visibility and pairing of points. For their single view experiments on a challenging industrial object dataset, they report improvements of up to 31% in the correct pose recognition rate for learned bin weights in comparison to uniform weights. Besides learned weights, they also experimented with heuristic based weightings, for example weighting a bin inversely proportional to the number of recorded point pairs for that bin. This turned out not to improve the recognition results, because, as they argue, even features which occur often might still be necessary to disambiguate between certain poses.

Despite the approaches for feature selection described in Madry et al. [24] and Tuzel et al. [37], a probabilistic model of the feature distribution over the object's surface would also lend itself towards an analysis of surface point informativeness. In Glover et al. [16], a dense probabilistic model over 3d features is built by clustering the feature descriptors into words and estimating a distribution over feature orientation and position on the object for features belonging to one word. If inverted, observing a certain feature with a local feature frame gives a distribution over possible object poses. An analysis regarding the relevance of a feature for pose estimation thus could be carried out by information theoretic criteria like the entropy of the surface distribution. Eidenberger et al. [11] build a sparse, keypoint based probabilistic model based on stereo matching and clustering similar SIFT features to 3d interest points on the object's surface. The clustering and observations of the same interest point from multiple directions allows for statistics for example over the interest point's location, which is represented as a 3d Gaussian. The sparsity arises naturally from the SIFT keypoint detection and further feature selection is done by throwing away interest points with only a small number of observed feature detections. As explained in the previous section, this model is used in a purely online information gain driven method for view planning.

2.4. Summary

This chapter described the general formalism of active pose estimation systems under the assumption that the object is static. Various approaches to active sequential object classification and pose estimation using the presented formalism have been analyzed with a special emphasis on how they facilitate view planning. Finer grained offline statistics for view planning as well as the selection of discriminative features for pose estimation have been identified as directions for further research.

Encouraged by the performance improvement and simplicity of the approach to feature selection for categorization by Madry et al. [24], a similar approach, but targeting feature selection for orientation estimation, will be presented and applied to orientation estimation in chapter 4 and analyzed with respect to view planning in chapter 5. As necessary

foundation for these later chapters, the next chapter will introduce a parametric probability distribution over 3d rotations and showcase its suitability for sequential orientation fusion.

3. Rotation Estimation using the Bingham Distribution

This chapter introduces the Bingham distribution, a parametric distribution over 3d rotations. The expressiveness of the distribution is illustrated using individual Bingham and Bingham mixture models. A Monte Carlo and an algebraic approach for sequential orientation fusion using Bingham mixture models is derived and evaluated on a standard 3d rotation filtering task. The estimation accuracy of different parametrizations is assessed and the resulting best ranked parametrization compared to other approaches to sequential orientation estimation.

3.1. 3d Rotations

The space of rotations in 3d is inherently different from the space of 3d translations as it is not a vector space (e.g. not commutative meaning when applying a sequence of rotations the ordering within the sequence matters). It is a 3-dimensional manifold (3 degrees of freedom), but no parametrization with only 3 parameters exists which covers this space without singular points [35]. Consider for example Euler angles and specifically the intrinsic Z-X-Z convention. Here, a rotation is specified by three consecutive rotations about the Z-, X- and again Z-axis, where the rotations occur around the consecutively rotated axes. This parametrization is complete in the sense that for every 3d rotation there exist three angles for the rotations Z-X-Z which specify this rotation. However, when setting the rotation angle for X to zero, the remaining two rotation angles which are left to specify (formally two degrees of freedom) produce rotations around the same axis and thus only one degree of freedom is controllable. This is a classical example of a singularity. Over the course of history many parametrizations of the 3d rotation group, the so called special orthogonal group $SO3$, have been presented and important characteristics, for example the uniqueness of the parametrization or the computational complexity of operations (composition, rotating a vector, inversion, renormalization, etc.), have been analyzed. Historically popular parametrizations are e.g. rotation matrices, Euler angles (e.g. roll, pitch, yaw), Rodrigues vectors and unit quaternions.

Singularities or symmetries inherited from a particular parametrization make specifying a proper probability distribution over 3d rotations a non-trivial task. In [21] and [13] the above mentioned parametrizations are analyzed for their suitability with respect to probabilistic data fusion. The quaternion parametrization, which is discussed in the following section, stood out as being very suitable for defining probability distributions over the 3d rotation group. Such a distribution can be defined either using a zero-mean 4-dimensional Gaussian distribution (leading to the Bingham distribution [3] discussed in section 3.3) or a 3-dimensional Gaussian distribution in the local, 3-dimensional tangent space around a specific quaternion rotation as described in [13] and briefly reviewed in section 3.5.

3.2. Quaternions

Quaternions are extended complex numbers $w + \mathbf{i}x + \mathbf{j}y + \mathbf{k}z$ where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ and $\mathbf{ij} = \mathbf{k} = -\mathbf{ji}$ with $w, x, y, z \in \mathbb{R}$. An alternative way of thinking would be to see quaternions as 4-vectors $(w, x, y, z) \in \mathbb{R}^4$ with special and suitably defined addition and multiplication operators [20]. Generally, $w \in \mathbb{R}$ is referred to as the scalar part and $(x, y, z) \in \mathbb{R}^3$ as the vector part.

Every quaternion vector q with unit length, which therefore lies on the unit hypersphere $S^3 \subset \mathbb{R}^4$, represents a rotation in $SO(3)$. The parametrization is unique up to an antipodal symmetry on the sphere S^3 , which means every rotation $r \in SO(3)$ is represented by exactly two unit quaternions q and $-q = (-w, -x, -y, -z)$. Every 3d rotation can be expressed as a rotation about an axis specified by a unit vector $(u_x, u_y, u_z) \in S^2 \subset \mathbb{R}^3$ and a rotation angle θ . This parametrization is called the angle-axis representation of a 3d rotation and is related to the quaternion representation in the following way:

$$q = \cos\left(\frac{\theta}{2}\right) + (\mathbf{i}u_x + \mathbf{j}u_y + \mathbf{k}u_z) \sin\left(\frac{\theta}{2}\right) \quad (3.1)$$

Looking at how the angle-axis representation and the corresponding unit quaternion are related, one can get a first intuition about how different kinds of uncertainties are represented on the quaternion unit sphere S^3 . For example, let us consider a small deviation from the identity rotation in a random direction. The identity rotation is trivially identified using (3.1) and $\theta = \{0, 2\pi\}$ as $q_{id} = (\pm 1, 0, 0, 0)$. A small rotation is obtained by choosing a small rotation angle θ solely. Choosing the rotation axis (u_x, u_y, u_z) uniformly random from S^2 (unit sphere in \mathbb{R}^3) then leads to a small random rotation with uniform preference regarding the direction of rotation. We see from (3.1) that fixing $\theta = a$ to a certain value and changing the axis of rotation only has influence on the vector part of the quaternion. In other words, every rotational deviation from the identity rotation by a fixed θ lies on a circle of the S^3 sphere, created by the intersection of a hyperplane orthogonal to $(1, 0, 0, 0)$ with the sphere S^3 where the axis intersection of the hyperplane with $(1, 0, 0, 0)$ is defined by the amount of rotation through the term $\cos(\frac{\theta}{2})$.

For a different scenario, consider a rotation of arbitrary magnitude $\theta \in [0, 2\pi]$ but around a fixed axis (a_x, a_y, a_z) . From (3.1) we see that changing the angle of rotation θ implies different numerical values for the scalar as well as the vector part of the quaternion. However, the ratio between the elements of the vector part stays the same, since they are only scaled by $\sin(\frac{\theta}{2})$. When gradually changing θ from $\theta = 0$ to $\theta = 2\pi$ while fixing $(u_x, u_y, u_z) = (a_x, a_y, a_z)$ the path taken on the unit sphere S^3 follows a great circle created by the intersection of the sphere with a plane passing through the origin [33]. A great circle is defined by two points on the sphere together with the origin and thus every uncertainty characteristic describing a uniform rotation around a fixed axis specifies a unique great circle by means of the origin, the point $(1, 0, 0, 0)$ for $\theta = 0$ and for example the point $(0, a_x, a_y, a_z)$ for $\theta = \pi$.

3.3. Bingham Distribution

The Bingham distribution was first discussed by Christopher Bingham [3] in 1974 in the context of directional statistics. It is not limited to describe distributions over directions

in 3d (S^2), but can be generalized to arbitrary degrees of freedom respective distributions over n -dimensional hyperspheres S^n , $n \in \mathbb{N}^+$. The distribution is always antipodally symmetric, thus in the case of $n = 3$, it correctly captures the antipodal symmetry of the quaternion parametrization of rotations in 3d. It gained recent popularity through various publications by J. Glover ([14], [16], [15]), who successfully used the distribution for 3d object pose estimation and rotational alignment of point clouds.

The Bingham distribution over quaternions is derived from a 4d zero-mean Gaussian density

$$p(\mathbf{x}; C) = \frac{1}{\sqrt{(2\pi)^4 |C|}} \exp\left(-\frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x}\right) \quad (3.2)$$

where $\mathbf{x} \in \mathbb{R}^4$ and C is a covariance matrix. This standard Gaussian is defined over all \mathbb{R}^4 and by re-normalizing it we can obtain a proper probability distribution over S^3 and thus 3d rotations. Intuitively, the process of renormalization can be thought of as intersecting the unit quaternion sphere with the volumetric Gaussian density in 4d and normalizing the resulting density on the sphere to integrate to one. The Gaussian form of the Bingham distribution is thus

$$p(\mathbf{x}; C) = \frac{1}{F} \exp(\mathbf{x}^T C^{-1} \mathbf{x}) \quad (3.3)$$

with $\mathbf{x} \in S^3$ is now constrained to lie on the unit sphere and F is chosen so that

$$\int_{\mathbf{x} \in S^3} \exp(\mathbf{x}^T C^{-1} \mathbf{x}) = 1 \quad (3.4)$$

The standard form of the Bingham distribution is given by

$$\mathcal{B}(\mathbf{x}; \mathcal{K}, V) := p(\mathbf{x}; \mathcal{K}, V) = \frac{1}{F(\kappa_1, \kappa_2, \kappa_3)} \exp\left(\sum_{i=1}^3 \kappa_i (v_i^T \mathbf{x})^2\right) \quad (3.5)$$

with $\mathcal{K} = (\kappa_1, \kappa_2, \kappa_3)^T$, $\kappa_i \in \mathbb{R}^- \cup \{0\}$ denoting the concentration parameters and $V = [v_1|v_2|v_3]$, $v_i \in S^3$ a set of orthogonal basis vectors. The standard form is based on an eigenvector/eigenvalue decomposition of the Gaussian covariance matrix C . Every covariance matrix can be decomposed into an orthogonal matrix of eigenvectors $Q = [v_1|v_2|v_3|v_4]$ and a diagonal matrix of eigenvalues $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ via principal component decomposition $C = Q\Lambda Q^T$. For convenience we now define $\hat{\mathcal{K}} = \Lambda^{-1} = \text{diag}(\hat{\kappa}_1, \hat{\kappa}_2, \hat{\kappa}_3, \hat{\kappa}_4)$ where $\hat{\kappa}_i = 1/\lambda_i$. Through zero-ing out $\hat{\kappa}_4$, one arrives at above mentioned concentration parameters κ_i as follows.

For definiteness, an ordering constraint is imposed on the eigendecomposition so that $\hat{\kappa}_1 \leq \hat{\kappa}_2 \leq \hat{\kappa}_3 \leq \hat{\kappa}_4$. Noting that any orthogonal transformation is length preserving, the normalization constant is actually only dependent on the magnitudes of $\hat{\kappa}_0, \dots, \hat{\kappa}_4$. Applying the eigendecomposition to equation (3.3) it thus follows

$$p(\mathbf{x}; \hat{\mathcal{K}}, Q) = \frac{1}{F(\hat{\kappa}_1, \hat{\kappa}_2, \hat{\kappa}_3, \hat{\kappa}_4)} \exp(\mathbf{x}^T Q \hat{\mathcal{K}}^{-1} Q^T \mathbf{x}) \quad (3.6)$$

However, this is still an over-parametrization as the re-normalization of the 4d Gaussian to the unit sphere S^3 also brings scale-independence to changes to the eigenvalues

λ_i with it. In [3], Bingham shows that given arbitrary concentration parameters $\hat{\mathcal{K}} = \text{diag}(\hat{\kappa}_1, \hat{\kappa}_2, \hat{\kappa}_3, \hat{\kappa}_4)$ and $\mathcal{K}' = \hat{\mathcal{K}} - \hat{\kappa}_4 I$ with I being the 4d identity matrix, it holds

$$p(\mathbf{x}; \mathcal{K}', Q) = p(\mathbf{x}; \hat{\mathcal{K}}, Q) \quad (3.7)$$

By convention, $\hat{\kappa}_4$ is therefore zeroed out resulting in $\mathcal{K} = (\kappa_1, \kappa_2, \kappa_3)$, $\kappa_i = \hat{\kappa}_i - \hat{\kappa}_4$. Bringing equation (3.6) into the summation form of equation (3.5) and leaving out the last summand due to its concentration coefficient with value zero, one arrives at the standard form of the Bingham distribution.

From the 4d Gaussian perspective, having $\kappa_4 = 0$ implies an eigenvalue of $\lambda_4 = +\infty$ in eigenvector direction v_4 . For concentration parameters $\kappa_i < 0, i \in \{1, 2, 3\}$ the mode of the Bingham distribution will thus always be v_4 (which lies on S^3 and is thus a valid 3d rotation). Generally, the more negative one chooses the three concentration parameters, the more peaked the distribution gets. By choosing the eigenvectors $v_i, i \in \{1, 2, 3\}$ one adjusts the rotation axis in which rotational uncertainty applies. By choosing the concentration parameters, the kind of rotational uncertainty is specified. Two easy interpretable choices are:

- $\kappa_1 = \kappa_2 = \kappa_3 = \alpha \leq 0$: Gaussian like rotational uncertainty. For $\alpha = 0$ the distribution is completely uniform. The more negative the κ_i -s get, the more peaked the distribution gets. Figure 3.1 shows the angular deviation from the mode of the distribution for 68% and 98% of samples drawn of distributions with concentration parameters ranging from $\kappa_i \in [-900, 0]$.
- $\kappa_1 = \kappa_2 \ll \kappa_3 \leq 0$: Increasing the value of κ_3 leads to increased uncertainty of the 4d-Gaussian in the principal component direction v_3 . For κ_3 approaching zero (but not equal to zero), the mode will remain at v_4 but the distribution on the quaternion unit sphere gets increasingly stretched in the direction of v_3 . Setting the concentration parameter κ_3 to zero results in an infinite variance along the direction v_3 for the 4d Gaussian and in terms of the normalized distribution over rotations this results in a density function mimicking a great circle on the unit quaternion sphere. As explained in the previous section, a great circle represents a 3d rotation with fixed axis but uniform rotation angle. When v_3 has the special form of $v_3 = (0, a, b, c)$ this axis around which the uniform rotation occurs is exactly (a, b, c) . In figure 3.2 the rotational uncertainty for different values of $\kappa_3 \in \{-900, -480, 0\}$ and $v_3 = (0, 0, 0, 1)$ is shown and the increase in uncertainty for rotations around $(0, 0, 1)$ is clearly visible. The plots shown are similar in nature to EGI (extended Gaussian image) plots and an intuitive way to visualize distributions over rotations. They work by rotating a base point on the sphere by many sampled rotations of the distribution. At the rotated point's location, a counter is increased. When visualizing the counts via a heat map, this gives a visualization of how the distribution behaves. In the plots in figure 3.2 the point $p = (1, 1, 1)/\|(1, 1, 1)\|$ is rotated by $n = 100000$ sampled rotations. The mode of the distribution is the identity rotation, hence for $\kappa_3 < 0$ the peak is at p . For $\kappa_3 = 0$, we see the expected rotational invariance around the z-axis which is a consequence of choosing $v_3 = (0, 0, 0, 1)$.

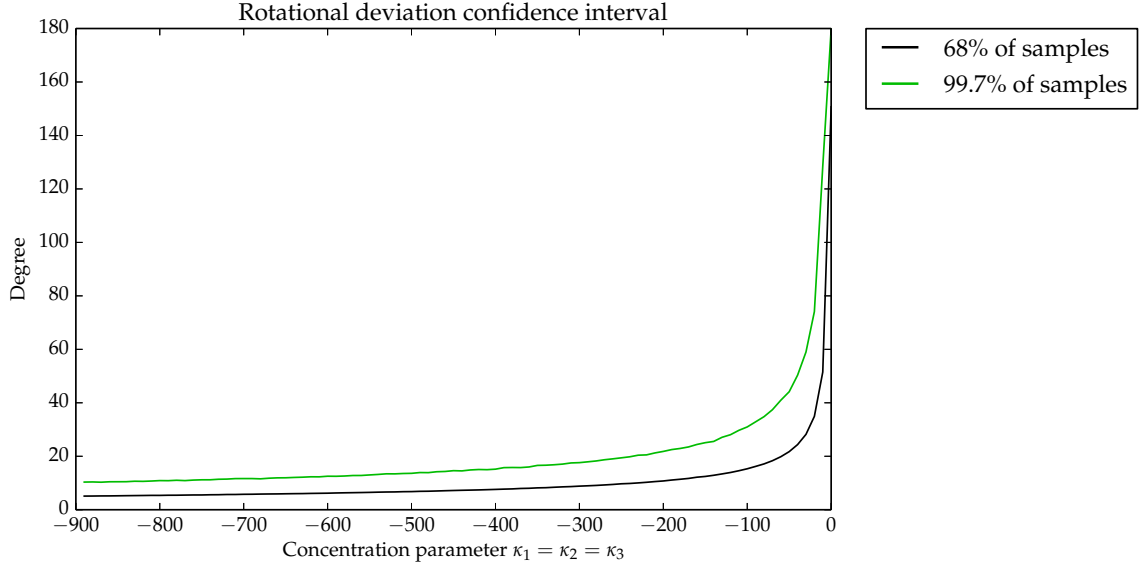


Figure 3.1.: Maximum angular deviation from the mode of the distribution for different concentration parameters (all κ_i set to the same value) and sample percentages.

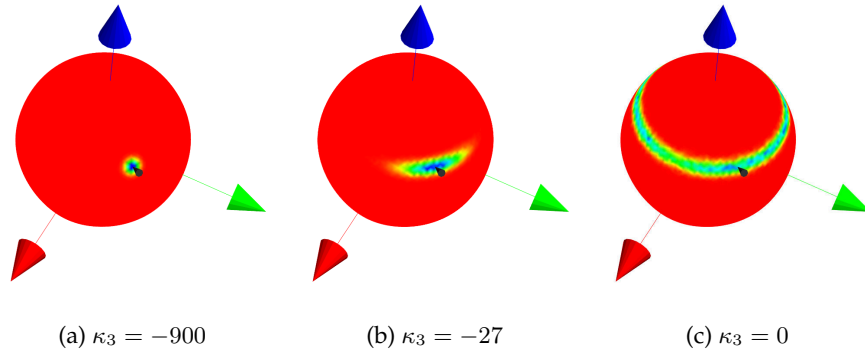


Figure 3.2.: EGI plots showing the effect of different concentration parameters κ_3 with corresponding eigenvector $v_3 = (0, 0, 0, 1)^T$. Increasing κ_3 leads to increased rotational uncertainty around the z-axis, which shows itself in a broader distributed heat map of sampled rotations ($n = 100000$ samples were used). More details on how EGI plots are constructed are in the text.

The major problem with using the Bingham distribution in practice is the normalization constant $F(\kappa_1, \kappa_2, \kappa_3)$ which is expensive to compute [14]. Glover therefore precomputes the normalization constant for a discrete set of concentrations parameters ranging in $\kappa_i \in [-900, 0]$ and interpolates them as necessary. For probabilistic data fusion certain operations on Bingham distributed random variables are of special interest. This includes for example maximum likelihood estimation of the parameters of a Bingham distribution from quaternion samples, sampling a Bingham, composing two Bingham (uncertain quaternion multiplication), rotating a Bingham by a fixed quaternion, merging two Bingham (approximating the sum of two Bingham by a single Bingham), multiplying two Bingham, extracting the mode of a Bingham, computing the entropy of a Bingham and computing the Kullback-Leibler (KL) divergence between two Bingham. Glover derives all above mentioned operations for the Bingham distribution in [14], making use of lookup tables for the normalization constant. Except for the composition of two Bingham and the sampling of the Bingham distribution, all other operations are analytically defined and do not require numerical approximation techniques as long as precomputed normalization constants are used. The composition of two Bingham $q = f \circ g$ where f and g are Bingham distributed and \circ represents quaternion multiplication can only be approximated by a Bingham distribution as Bingham are not closed under composition. Sampling is realized by Glover using a Metropolis-Hastings Markov Chain Monte Carlo (MCMC) approach with the 4d Gaussian distribution as proposal distribution. As the samples from the 4d Gaussian projected to the unit quaternion sphere are good proposals for samples of the actual Bingham distribution, Glover et al. [16] argues that the MCMC sampler converges very quickly to the desired target distribution and that this approach works well in practice.

3.4. Bingham Mixture Models

While individual Bingham distributions are in general unimodal (except one or more concentration parameters are zero), a sum of several Bingham is capable to represent arbitrary probability landscapes over rotations given a sufficient number of components. Formally, a Bingham Mixture Model (BMM) is defined as

$$\text{BMM}(\mathbf{x}; \{\alpha_i\}, \{\mathcal{K}_i\}, \{V_i\}) := \sum_i \alpha_i \mathcal{B}(\mathbf{x}; \mathcal{K}_i, V_i) \quad (3.8)$$

where $\mathcal{B}(\mathbf{x}; \mathcal{K}_i, V_i)$ is Bingham distributed as defined in equation (3.5) and $\sum_i \alpha_i = 1$. Unfortunately, BMMs - just like Gaussian mixture models - lose some of their analytical benefits as operations like maximum likelihood parameter fitting, entropy calculation and KL-divergence calculation are not defined analytically anymore and have to be approximated using for example Monte-Carlo integration techniques. In the context of this thesis, we need to be able to fit BMMs to quaternion samples, multiply two BMMs and extract the maximum mode of a BMM.

For fitting a BMM to a sample set of quaternions $\{q_i\}$, Glover [16] describes a greedy sample consensus method. At every iteration step, M Bingham are fitted to M sample subsets consisting of four randomly drawn quaternions from $\{q_i\}$ (the minimal number to fit a Bingham distribution) using a maximum likelihood approach. The data likelihood

for all of them is evaluated under a capped loss function (maximum error contribution is limited), the best Bingham above a certain likelihood threshold is added to the mixture and all associated quaternion samples are removed from the sample set before the procedure repeats. Adding components to the mixture stops if no more components can be found (threshold on the data likelihood) and a uniform Bingham is added to the mixture to account for the remaining samples.

The multiplication of two BMMs is done by pairwise multiplication of the Bingham distributions of both mixtures. As the result of multiplying two Bingham distributions is again a Bingham and can be done in an algebraic way, multiplication of two BMMs is also well-defined and will be explained further in section 3.6.

Finding the maximum mode of a BMM cannot be done in closed form. An approximation is computed by sampling a high number of rotations from the BMM (typically 50000), evaluating their probability density value under the BMM and selecting the sample with the maximum value. This procedure will be used in later sections whenever the MAP estimate of a BMM belief distribution is computed.

3.5. Projected Gaussians as Probabilities over Rotations

Despite the Bingham distribution, other approaches for specifying probabilities over 3d rotations exist. Feiten et al. [13] propose to use projected Gaussians and/or mixtures of projected Gaussians to represent unimodal and/or multimodal distributions over rotations. A projected Gaussian is a multivariate Gaussian defined in the tangent space of a specific basis rotation $q \in S^3$. In the same fashion as the tangent space for a point on a 3d sphere is a 2d plane (and thus an ordinary 2d vector space), the tangent space for a quaternion rotation on the 4d unit sphere is a 3d vector space. A 3d Gaussian defined in this tangent space defines a distribution over the 4d unit sphere by means of a central projection. Every point in the tangent space maps to two opposite points on the unit sphere through the line specified by the point and the center of the sphere. This correctly captures the antipodal symmetry and thus specifies a proper probability distribution over rotations. Similar as for the Bingham distribution, this distribution requires a renormalization which as derived in [13] can be approximated efficiently.

Comparing the projected Gaussian and the Bingham distribution reveals advantages and disadvantages for both sides. Feiten et al. [13] argue that projected Gaussians are more efficient due to the complicated calculation of the normalization constant for the Bingham distribution. This can be counteracted by a lookup table approach as implemented by Glover and Kaelbling [14]. Furthermore it has to be noted that a projected Gaussian is based on specific base quaternion direction to define the tangent space, which has to be adapted with operations like merging two projected Gaussians and thus also leads to additional computations. A more interesting difference is that the projected Gaussian is closed under composition while the Bingham distribution is not. This enables to propagate uncertainty for example along the links of a robotic arm. For the Bingham distributions, the result of a composition can only be Bingham approximated. A principal advantage of the Bingham distribution, however, is the ability to represent distributions with rotational invariance. A distribution with uniform rotation about a specific axis, for example, resembles a great circle on the quaternion unit sphere and cannot be represented by a

single projected Gaussian whereas for a single Bingham this just means one zero-valued concentration parameter. As Lang [21] points out, the Bingham distribution is more suited towards rotational distributions with high uncertainty. For more concentrated distributions this advantage of course tends to vanish.

For jointly specifying distributions over orientation and position, the projected Gaussian can be extended without much effort as described in [13]. This is done by adding three translational dimensions to the projected Gaussian, resulting in a 6d Gaussian with only the first three dimensions undergoing the central projection. This joint representation allows to capture correlation effects between position and orientation uncertainty and is thus superior to a Bingham based distribution over rotations and a separate 3d Gaussian over the position. A quantitative comparison of the computational efficiency and the representational accuracy between the two approaches has not been conducted yet but might be of interest.

The work in this thesis requires the ability to represent uniform uncertainties around axes which makes the Bingham distribution the distribution of choice here.

3.6. State Fusion using Bingham Mixture Models

The general state estimation problem has been outlined in chapter 2 and follows equation (2.1). Adapted for the case of estimating an object rotation, equation (2.1) can be written as

$$p(q_t | z_t, \dots, z_0) = p(z_t | q_t) p(q_t | z_{t-1}, \dots, z_0) \quad (3.9)$$

where $q_t, z_t \in SO3$ are 3d rotations which are without loss of generality assumed to describe an object's rotation (frame o) in the world frame w ($q_t := {}^w q_{o,t}$ resp. $z_t := {}^w z_{o,t}$). The measurement process underlying $p(z_t | q_t)$ can be described as producing a rotation measurements

$$z_t = w_t \circ q_t \quad (3.10)$$

based on the object's rotation q_t corrupted by a Bingham mixture distributed independent measurement noise $w_t \sim \text{BMM}(\alpha, \mathcal{K}, \mathbf{V})$ with \circ denoting quaternion multiplication. Based on findings of Glover [14], the conditional distribution $p(z_t | q_t)$ is Bingham mixture distributed according to $z_t | q_t \sim \text{BMM}(z_t; \alpha, \mathcal{K}, \mathbf{V} \circ q_t)$ where $\mathbf{V} \circ q_t$ are the eigenvectors of the original Bingham mixture components rotated by the fixed q_t . This mixture is a distribution over z_t given a fixed q_t . To apply this model in practice we can rewrite the distribution to obtain one over q_t given a fixed z_t by reordering the terms in the same way as presented in Glover and Kaelbling [14]

$$\text{BMM}(z_t; \alpha, \mathcal{K}, \mathbf{V} \circ q_t) = \sum_i \alpha_i \left(\frac{1}{F_i} \exp \sum_{j=1}^3 \kappa_i ((v_i \circ q_t)^T z_t)^2 \right) \quad (3.11)$$

$$= \sum_i \alpha_i \left(\frac{1}{F_i} \exp \sum_{j=1}^3 \kappa_i ((v_i^{-1} \circ z_t)^T q_t)^2 \right) \quad (3.12)$$

$$= \text{BMM}(q_t; \alpha, \mathcal{K}, \mathbf{V}^{-1} \circ z_t) \quad (3.13)$$

If we now assume a Bingham mixture prior $p(q_t|z_{t-1}, \dots, z_0) = \text{BMM}(q_t; \beta, \mathcal{K}^*, V^*)$, the posterior is also Bingham mixture distributed by multiplying the prior and measurement BMM. The components of the posterior BMM are built by pairwise multiplication of the prior and measurement mixture components

$$p(q_t|z_t, \dots, z_0) = \sum_i \sum_j \alpha_i \beta_j \mathcal{B}(q_t; \mathcal{K}_i, V_i^{-1} \circ z_t) \mathcal{B}(q_t; \mathcal{K}_j^*, V_j^*) \quad (3.14)$$

As the multiplication of two Bingham is well-defined and carried out by building the Gaussian form of the Bingham, adding their covariance matrices and re-shaping the result into standard Bingham form via an eigenvector/-value decomposition, above equation describes a unique algebraic solution to this sequential estimation problem.

3.6.1. Algebraic Fusion

In the algebraic posterior defined by equation (3.14) we see that the number of components grows rapidly as two mixtures with N and M components result in a posterior mixture with $N*M$ components. For computational reasons, one should try to keep the number of components within mixtures as low as possible but as high as necessary. This brings up the challenge of mixture reduction which means to reduce the number of components while maintaining an acceptable level of representational accuracy with respect to the unreduced mixture. For Gaussian mixtures, mixture reduction has been studied in the past (for example [28] [38] and [31]) and as the Bingham distribution is Gaussian, these methods are in principle applicable to Bingham mixture reduction as well. Special consideration has to be taken only, because all Gaussians within a Bingham mixture have zero mean, that means for example the reduction criterion of Salmond [31] is not applicable as it does depend solely on the mean of the mixture components. As Williams suggests in [38], the representational accuracy between the reduced and the original mixture would ideally be assessed by the KL divergence (an information-theoretic measure of dissimilarity between two probability distributions) and optimized to be as small as possible. However, there is no closed form solution for the KL divergence of two Gaussian (or Bingham) mixtures which makes efficient mixture reduction a non-trivial problem.

For evaluation of the Bingham mixture fusion, the reduction method of Runnals [28], originally developed for Gaussian mixtures, has been implemented and tested for Bingham mixtures. It works by iteratively choosing two components of the mixture and merging them into one component. This is done until a specified number of components is reached. The two components to be merged are chosen by an upper bound criteria derived by Runnals. It states that the the KL divergence of the mixture before the merge (\mathcal{M}_{i+j}) and after the merge (\mathcal{M}_{ij}) of components i and j is smaller than

$$d_{kl}(\mathcal{M}_{i+j}, \mathcal{M}_{ij}) \leq \alpha_i d_{kl}(B_i, B_{ij}) + \alpha_j d_{kl}(B_j, B_{ij}) \quad (3.15)$$

with B_i and B_j denoting the individual original components, B_{ij} denoting the merged component and α_i, α_j are the respective components' weights. As B_i, B_j and B_{ij} are single Gaussians (or Bingham), the KL divergence is analytically computable. Every iteration step the two components leading to a minimal approximated before-after KL divergence are chosen to be merged. An example reduction is illustrated in Figure 3.3 by reducing an eight component mixture gradually down to one component.

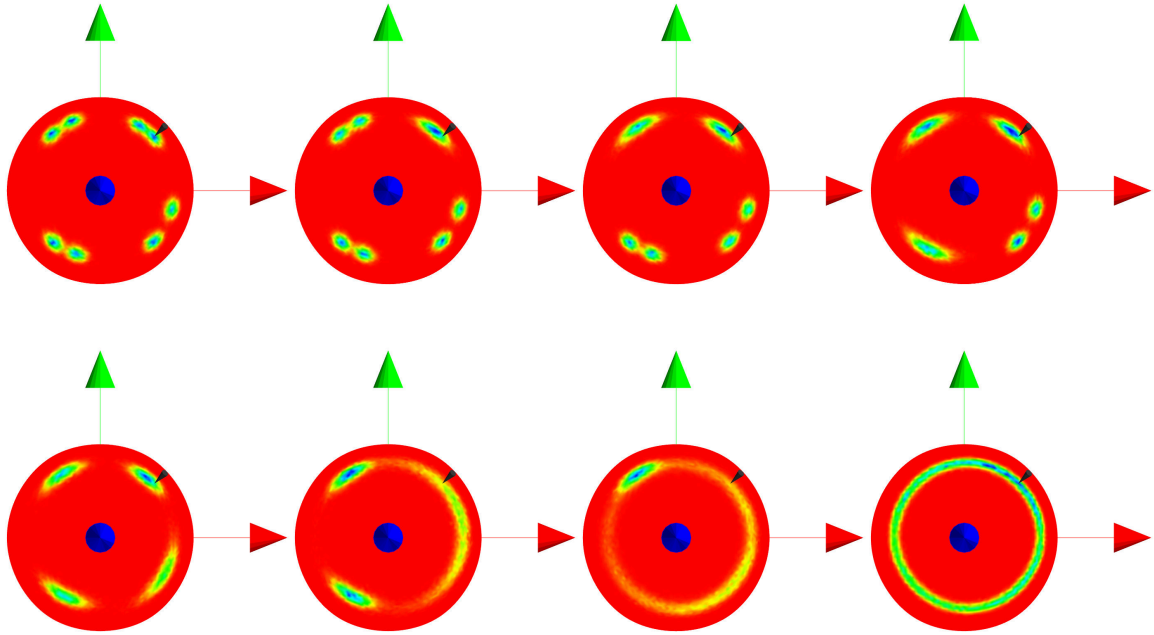


Figure 3.3.: EGI plots for Bingham mixture reduction: The initial Bingham mixture (top, left) has eight components and roughly represents a 4-fold symmetric distribution around the z-axis (two components per fold). From left to right, top to bottom, the mixture gradually gets reduced by one component using Runnals [28] KL-criterion given in equation (3.15). With four components, the mixture still captures the 4-fold symmetry by deciding to merge every fold's two components together. Further reduction leads to a stronger distortion of the original distribution.

The final algorithm for state fusion based on algebraic multiplication thus has two steps. At first, the full posterior is computed through element-wise multiplication as in equation (3.14). In the second step, the full posterior is reduced to a predefined maximum number of allowed components N_{\max} , which is the only parameter of this fusion method. For notational convenience, we denote this fusion algorithm as multiply & reduce (M+R) method.

3.6.2. Monte Carlo Estimation

A different way to compute the posterior distribution was implemented using a sequential Monte Carlo (SMC) approach commonly referred to as particle filter. Instead of representing the posterior distribution in a parametric way (by specifying a density function), it is represented by a set of M equally weighted samples $\{q_t^m\}, m = 1, \dots, M$ distributed according to the posterior distribution $\{q_t^m\} \sim p(q_t|z_0, \dots, z_t)$. This sample set approximately represents the posterior distribution by means of the sample density; regions with higher sample density correspond to regions where $p(q_t|z_0, \dots, z_t)$ has a high value [36].

In the SMC framework, updating the prior (previous time step's posterior) with a new

measurement is done in two steps: an importance weighting of the prior sample set using the current measurement which result in a weighted sample set $\{q_{t-1}^m\} \rightarrow \{(w_m, q_{t-1}^m)\}$ and a resampling step to obtain an equally weighted set representing the new posterior $\{(w_m, q_{t-1}^m)\} \rightarrow \{q_t^m\}$. The importance weighting is done by evaluating the measurement model $p(z_t|q_t)$ for a concrete sample q_{t-1}^m , hence $w^m = p(z_t|q_{t-1}^m)$ [36]. The resampling step is done by simply drawing samples from $\{(w_m, q_{t-1}^m)\}$ with replacement and proportional to their importance weight. It can be proven that in the limit for $M \rightarrow \infty$ the resampled sample set is a valid representation of true posterior distribution. Up to now, although we changed the relative frequency of samples within our sample set, the actual sample values never changed. This is due to the fact that we assumed an identity (static) dynamic model of our state (see chapter 2). To circumvent this issue and to allow the sample set to change "position", a Bingham mixture is fitted to the posterior sample set using the method described in section 3.4 and sampled again in order to obtain the prior sample set for the next time step. The main parameter of the SMC method for state fusion is the number samples M . In all experiments later on $M = 100000$, which provides a good balance between computation time and representational accuracy.

3.7. Evaluation

In order to evaluate how well the sequential rotation estimation methods described in section 3.6 work in practice, test data from a previous experiment of our group has been reused (Marton and Türker [25]). The test data contains object pose detection sequences of three industrial objects (*valve*, *filter* and *control*) using the object recognition and pose estimation method from Kriegel et al. [19]. This pose estimation method works well in practice and returns a single most likely object pose for every object detected. For the three objects used here, the average rotational error of the pose estimate was around four degrees. For the sequences every object was captured on its own in a non-cluttered scene. All sequences contain 20 views of their object from different viewing directions and all views lead to successful detection, hence 20 rotation measurements $z_t, t = 1, \dots, 20$ exist for every object. Whereas the *valve* and *control* object have a clear unique pose, the *filter* object has a strong rotational ambiguity in form of a 4-fold rotational symmetry. It is thus detected with a rotational error of $\pm 90^\circ$ or 180° in 7 out of the 20 views.

Two Bingham mixture measurement models have been defined for evaluation of the test sequences. The first one is the simple uninformed standard model where we assume the rotation measurement results from the correct pose corrupted by small Gaussian-like noise. A measurement model $p(z_t|q_t)$ describing this noise characteristic is created by rotating the correct pose by a Bingham with mode at identity and concentration parameters specifying a small Gaussian-like deviation, specifically

$$\mathcal{K} = \kappa \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \mathcal{V} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$$p(z_t|q_t) = \mathcal{B}(z_t; \mathcal{K}, \mathcal{V} \circ q_t) \quad (3.17)$$

To deal with outliers, a uniform Bingham $\mathcal{B}_{uni}(z_t) = \mathcal{B}(z_t; (0, 0, 0), \mathcal{V})$ is added to the measurement model. This results in the Bingham mixture model

$$p_G(z_t|q_t) = \alpha_1 \mathcal{B}(z_t; \mathcal{K}, \mathcal{V} \circ q_t) + \alpha_2 \mathcal{B}_{uni}(z_t) \quad (3.18)$$

which can be "inverted" to a distribution over q_t by following equations (3.11) to (3.13). The Gaussian measurement model p_G has two parameters: the measurement concentration κ and the outlier ratio $\alpha = (\alpha_1, \alpha_2)$. This measurement model will be used for all three objects.

To test a more complex measurement model, the *filter* object was additionally evaluated with a measurement model taking the 4-fold symmetry around the object's z-axis into account. It is based on a five component Bingham mixture model. The first four components describe rotations of $\{0, +90, -90, 180\}$ degrees of the correct pose q_t around the symmetry axis and the fifth component again a uniform distribution. Construction of the symmetric measurement model p_S is a straightforward extension of equation (3.18) by adding for example the +90 degree fold as $\mathcal{B}_{+90}(z_t; \mathcal{K}, \mathcal{V} \circ q_{+90} \circ q_t)$ where $q_{+90} = \text{q_from_angle_axis}(+90, (0, 0, 1)^T)$. The resulting parameters for the model p_S are again the measurement concentration κ used for the first four components and the component weighting $\alpha = (\alpha_1, \dots, \alpha_5)$.

The metric for comparing different parameter settings for the measurement models $\{p_G, p_S\}$ and state fusion strategies $\{\text{SMC}, \text{M+R}\}$ is based on the average MAP rotation error over the last ten views of the viewing sequences. Given a ground truth rotation q_{gt} (which in our case is constant over all views of a sequence) and the state distribution after measurement $t = 1, \dots, 20$ as $p(q_t|z_1, \dots, z_t) = \text{BMM}_t(q_t, \alpha, \mathcal{K}, \mathcal{V})$ the MAP rotational error θ_t is defined as

$$q_t^* = \text{map_estimate}(\text{BMM}_t(q_t, \alpha, \mathcal{K}, \mathcal{V})) \quad (3.19)$$

$$q_{err,t} = (w, x, y, z) = q_{gt}^{-1} \circ q_t^* \quad (3.20)$$

$$\theta'_t = 2 \arccos(w) \quad (3.21)$$

$$\theta_t = \begin{cases} \theta'_t & \theta'_t \in [0, \pi] \\ |\theta'_t - 2\pi| & \theta'_t \in [\pi, 2\pi] \end{cases} \quad (3.22)$$

The average MAP error $\bar{\theta}_o, o \in \{\text{filter}, \text{control}, \text{valve}\}$ is then defined as

$$\bar{\theta}_o = \frac{1}{10} \sum_{t=11}^{20} \theta_t \quad (3.23)$$

$$\bar{\theta} = \frac{1}{3} \sum_o \bar{\theta}_o \quad (3.24)$$

3.7.1. Evaluation for Gaussian Measurement Model

As a first evaluation, the Gaussian-inspired measurement model p_G was tested with both fusion strategies and all three object sequences. A total of 72 different parametrizations have been tested, 18 using the SMC fusion and 54 using the M+R fusion. The individual

parametrization ranges explored for the different functional components are summarized in table 3.1. All runs were ranked by the average MAP rotational error given in equation (3.24). The full ranking is given in table A.1. A summary of the top ten parametrizations is given in table 3.2, which also includes the maximum MAP error within the last ten views over all three sequences and the percentages of MAP estimates below one, three and five degrees of error for the last ten views. Figure 3.4 shows the MAP error evolution of the best SMC and M+R method compared to the error obtained by several other approaches to orientation fusion evaluated in Marton and Türker [25]. As can be seen, the Bingham mixture approaches yield competitive results to a particle filter, a pose clustering and a histogram filter approach.

Out of the 72 parametrizations, 34 perform very close to the best one with a difference of only 0.21° in the avg. MAP error and 0.64° in the maximum MAP error. Seven of the top 34 runs are based on the SMC fusion, 27 are based on M+R. However, the 27 M+R runs contain only 9 different (α, κ) -parametrizations each of which can be paired with a maximum number of components limit $N_{\max} \in \{1, 5, 10\}$ and still perform nearly optimal. Seven of the 9 (α, κ) -parametrizations for which the M+R fusion work good, also result in near optimal SMC performance. This leads to the conclusion that SMC and M+R fusion obtain good performance with similar parametrization of the Gaussian measurement model and for these parametrizations the number of components the mixture is reduced to (N_{\max}) does not play a great role. Whereas the outlier ratio α does not have an impact on the obtainable performance, the measurement concentration κ does. Increasing the concentration of the measurement distribution by decreasing κ below -120 leads to increasingly worse performance. All of the 34 good performing runs have a concentration $\kappa \in \{-27, -60, -120\}$, which correspond to 68% of the samples drawn from such a distribution having an rotational deviation smaller than $\{29^\circ, 19^\circ, 14^\circ\}$ (cf. figure 3.1) from the mode of the distribution. This is somewhat contradicting to the actual measurement error in the object sequences, as for the *control*, *filter* and *valve* sequence 68% of the measurement errors lie below 4.4° , 6.9° and 4.4° . Thus even a concentration value of $\kappa = -240$ resp. 9.8° for the 68% interval encloses the measurement uncertainty well enough.

Table 3.1.: Explored parameter ranges for Gaussian measurement model evaluation

Functional Component	Parameter	Explored Range
SMC	M	$\{100000\}$
M+R	N_{\max}	$\{1, 5, 10\}$
p_G	α	$\{(.9, .1), (.95, .05), (.99, .01)\}$
	κ	$\{-900, -480, -240, -120, -60, -27\}$

3.7.2. Evaluation for Multimodal Measurement Model

The evaluation of the multimodal 4-fold symmetric measurement model p_S was carried out solely based on the test sequence of the *filter* object, as only this object shows this kind of measurement ambiguity (cf. the raw measurement line in the filter subplot of figure

3. Rotation Estimation using the Bingham Distribution

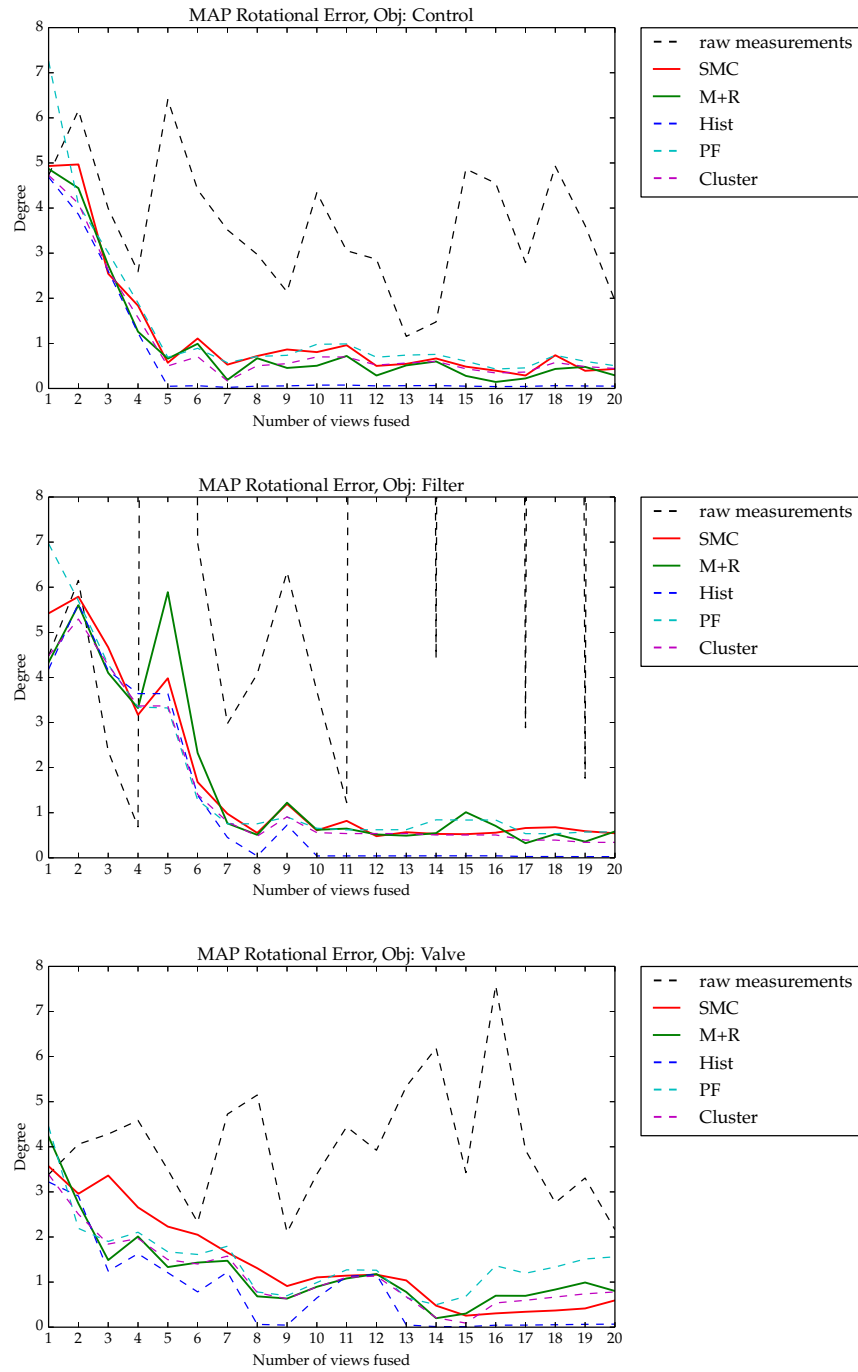


Figure 3.4.: MAP Rotational Error plots for the best SMC vs. the best M+R parameter settings in comparison to a histogram filter (discretized rotation space, Hist), a particle filter (PF) and a pose clustering approach (Cluster).

Table 3.2.: Top 10 ranking of all parametrizations for the fusion evaluation of the Gaussian measurement model. Complete ranking given in table A.1

	avg. MAP	max. MAP	% < 1°	% < 3°	% < 5°	Fusion	N _{max}	α	κ
1	.57°	1.18°	.90	1.00	1.00	M+R	1	[.95, .05]	-60
2	.58°	1.16°	.90	1.00	1.00	SMC		[.90, .10]	-27
3	.59°	1.03°	.97	1.00	1.00	SMC		[.95, .05]	-60
4	.60°	1.42°	.90	1.00	1.00	M+R	1	[.90, .10]	-60
5	.61°	1.21°	.90	1.00	1.00	M+R	1	[.95, .05]	-27
6	.62°	1.30°	.87	1.00	1.00	M+R	1	[.90, .10]	-27
7	.62°	1.15°	.90	1.00	1.00	M+R	1	[.99, .01]	-60
8	.62°	1.31°	.90	1.00	1.00	M+R	5	[.95, .05]	-60
9	.62°	1.31°	.93	1.00	1.00	M+R	1	[.99, .01]	-27
10	.63°	1.36°	.90	1.00	1.00	M+R	10	[.99, .01]	-27
⋮									

3.4). A total of 168 parametrizations has been evaluated exploring combinations of the two fusion methods, with both measurement models p_G and p_S and parametrizations given in table 3.3. The full ranking is again given in the table A.2 whereas a subselection of the ranking is given here in table 3.4.

Table 3.3.: Explored parameter ranges for multimodal measurement model evaluation

Functional Component	Parameter	Explored Range
SMC	M	{100000}
M+R	N _{max}	{1, 5, 10}
p_G	α	{(.9, .1), (.95, .05), (.99, .01)}
	κ	{-900, -480, -240, -120, -60, -27}
p_S	α	{(.25, .22, .22, .22, .1), (.45, .15, .15, .15, .1), (.65, .08, .08, .08, .1), (.85, .02, .02, .02, .1)}
	κ	{-900, -480, -240, -120, -60, -27}

Out of the 20 measurements in the filter sequence, seven are $\pm 90^\circ$ or 180° off, due to the 4-fold symmetry of the filter object. Six of these seven outliers fall into the last ten views of the sequence, over which the avg. MAP error metric is built. This evaluation thus puts special emphasis on how the parametrizations handle this ambiguity. The first 43 parametrizations yield similar performance in terms of average MAP error (best/worst difference: 0.39°) and maximum MAP error (best/worst difference: 1.23°). The 44rd ranked parametrization is the first one having a maximum MAP error of over 2° . Within these top 43 parametrizations, only five are based on the 4-fold symmetric measure-

ment model and all five have very little component weight on the symmetric components ($\alpha_{2...4} = 0.02$), even less than the uniform component weight ($\alpha_5 = 0.1$). The first of these 4-fold parametrizations is ranked 16th, whereas the first 4-fold symmetric parametrization using a significant weight for the symmetric components ($\alpha_{2...4} = 0.15$) is ranked 50th with an avg./max. MAP error of $1.14^\circ / 1.37^\circ$ (cf. table 3.4). The worst parametrization using p_S and the SMC fusion is ranked 85th with an avg./max. MAP error of $5.39^\circ / 10.02^\circ$, while the vast majority of p_S -parametrizations with M+R fusion at some point return an MAP estimate around one of the outlier measurements and therefore obtain much worse MAP averages and maximum errors of up to $\approx 180^\circ$. It can be concluded, that both fusion methods perform better with the simple Gaussian measurement model and in general, if used with a complex measurement model, the SMC fusion has an advantage over the M+R fusion.

Table 3.4.: Selected rankings of parametrizations for the fusion evaluation of the multi-modal measurement model. Complete ranking given in table A.2.

	avg. MAP	max. MAP	% < 1°	% < 3°	Fusion	N_{\max}	Meas.- model	α	κ
1	.41°	.56°	1.00	1.00	SMC		gaussian	[.95, .05]	-120
2	.46°	.88°	1.00	1.00	SMC		gaussian	[.95, .05]	-27
3	.46°	.63°	1.00	1.00	SMC		gaussian	[.90, .10]	-60
⋮									
16	.66°	1.22°	.90	1.00	M+R	10	4fold	[.85, .02, .02, .02, .10]	-60
⋮									
50	1.14°	1.37°	.30	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-27

3.8. Summary

In conclusion, sequential orientation estimation using Bingham mixture models and the two evaluated fusion methods works well in practice and obtains comparable performance to approaches based on a particle filter, pose clustering and a histogram filter. For the 4-fold symmetric object under investigation here, a simple measurement model with Gaussian like uncertainty around the detected orientation and a uniform “outlier” component works better than a more complex measurement model which takes the object’s symmetries into account. Based on the competitiveness of the results to other methods, a Bingham mixture based orientation fusion could serve well as a black box fusion algorithm for sequential estimation of rotations.

4. Viewing Direction Classification: Application to Orientation Estimation

As we have seen in chapter 2, the interplay between the method used for object pose estimation and view planning is critical. Planning is ideally based on predicting the expected measurement for an action. Due to the computational complexity involved, methods using precomputed statistics in order to lower the planning runtime in the online phase have been subject to research and shown to perform comparable to pure online methods.

Many offline computed statistics ([1], [6], [34], partly [22]) are based on ranking the informativeness of viewing directions and thus provide a rather coarse information on how to plan views. A finer grained information would reveal which part of the object or which feature is informative for the pose of the object. With this aim in mind, in this chapter the dense feature scoring method of Madry et al. [24] is adapted to the case of object orientation estimation instead of object categorization. While chapter 5 will give details and proof-of-concept results regarding the application of the presented method to view planning, this chapter focuses on how to obtain an orientation estimate in the first place and how to embed the method into a sequential orientation estimation framework. The probabilistic measurement model is based on the unique ability of the Bingham distribution to represent large uncertainties in a compact parametric way. The accuracy of the resulting orientation estimate is evaluated using random multi-view sequences at the end of the chapter.

4.1. Method Overview

The developed object rotation estimation is based on 3d features calculated from point cloud data acquired with structured light depth cameras like the Microsoft Kinect or Asus Xtion. With the idea of determining discriminative parts of the object with respect to object rotation estimation in order to facilitate this knowledge in view planning algorithms, we followed a similar approach as Madry’s extraction of discriminative features for object classification [24] where we substitute object categories with viewing directions.

The presented approach is based on mapping individual feature descriptors to the viewing directions from which they can be observed from. This mapping is modeled using a probabilistic classifier trained for each object separately and treating the discrete set of viewing directions defined by the training views of the object as class labels. The output of the classifier is a discrete probability distribution over the viewing direction label given a single feature.

In the work presented here, Fast Point Feature Histogram (FPFH) [30] were used as features. These features are computed pointwise and represent the local geometry around a point by summarizing normal and distance statistics in the point’s neighborhood. A single FPFH feature is a 33-dimensional vector consisting of three 11-dimensional histograms

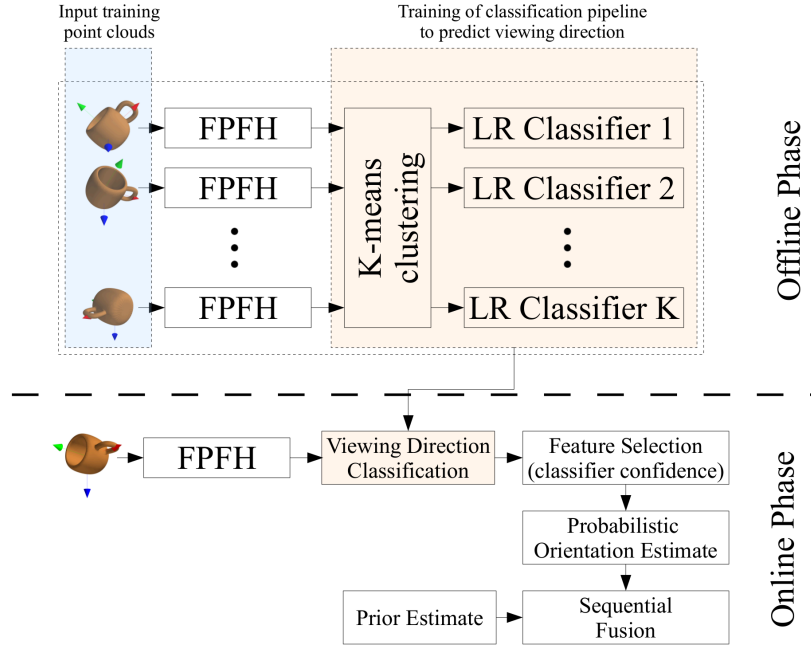


Figure 4.1.: Block diagram of the presented method for orientation estimation. In an offline phase, features (FPFH) are extracted from a set of training point clouds from known viewing directions and a classification pipeline is trained to predict the training view direction given a feature descriptor. Online, features are extracted from the point cloud of the object in unknown orientation and the viewing direction classification is used to obtain a probabilistic estimate of object's orientation which can be used in a sequential orientation estimation framework.

concatenated after each other. As the feature is invariant to rotations in 3d and therefore does not define a feature coordinate frame, the above described viewing direction classification will not specify a unique rotation for a viewing direction, but a set of rotations with rotational invariance around the camera optical axis. A single FPFH feature on its own therefore encodes a large set of possible object rotations. Firstly, by the uncertainty in the viewing direction it could originate from and secondly, by means of the rotational invariance about a specific viewing axis.

A probabilistic measurement model based on the Bingham mixture distribution is used to capture above uncertainties in a principled way. Every component of the mixture encodes one of the possible viewing directions and a single component models the rotational invariance about the camera axis. This measurement model can then be used to sequentially estimate an object's orientation.

A block diagram of the method's basic components is given in figure 4.1 along detailed explanations in the next sections.

4.2. Viewing Direction Classification

4.2.1. Training Setup and Choice of Classifier

The basis for training the viewing direction classifier for an object are point clouds captured from various camera poses around the object. In general, the training data $\mathbf{I} = \{\mathbf{I}_s^m\}, s \in [1, N_{sets}], m \in [1, N_{views}]$ contains N_{sets} segmented point clouds for every of the N_{views} defined camera poses. The notation \mathbf{I}_s will be termed the s -th sample set and contains the s -th point cloud for all N_{views} camera poses.

For every point cloud \mathbf{I}_s^m a feature set F_s^m is computed, where $f_j^{m,s} \in F_s^m$ denotes an individual feature vector. We can now pair every feature vector $f_j^{m,s}$ with its view label m and create the training data for the classifier as feature-label pairs by $\mathbf{X} = \bigcup_s \bigcup_m (f_j^{m,s}, m)$. To give an idea about the size a real dataset has, consider the datasets used in the evaluation with real models in section 4.5. In those datasets, $N_{sets} = 4$ sample sets with $N_{views} = 28$ camera poses have been recorded and used to train the viewing direction classifier. A typical training point cloud contains about ~ 6000 points respective features and hence the final training dataset has around $|\mathbf{X}| = 6000 * 28 * 4 = 672000$ training examples.

The large training data size and the required probabilistic output limits the choice of usable classification methods. Logistic regression (LR) classifiers can be trained very fast, are well studied in large-scale multi-class classification problems such as text classification [18] and give probabilistic output. Therefore, and also because a mature and fast implementation of LR is available (LibLinear [12]), it is the classification method of choice in this work.

LR is a probabilistic model for learning the class probability of a class given a feature vector. For the binary case with only two class labels (LR can be extended to true multi-class classification as well) the posterior class probability for the class one is defined as

$$p(C_1|x) = \sigma(\mathbf{w}^T \Phi(x)) \quad (4.1)$$

with

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (4.2)$$

is the logistic sigmoid function, $x \in \mathbb{R}^K$ is the input feature vector, $\Phi(x) : \mathbb{R}^K \rightarrow \mathbb{R}^L$ is a fixed and possibly non-linear transformation of the feature vector and $\mathbf{w} \in \mathbb{R}^L$ is the model parameter to be learned [4]. The probability for class two is simply $p(C_2|x) = 1 - p(C_1|x)$. For a given training set $\{(x_n, t_n)\}, i = 1, \dots, N$ with labels $t_n \in \{0, +1\}$, one learns the weight vector \mathbf{w} by maximizing the data(set) likelihood

$$p(\mathbf{t}|\mathbf{w}) = \prod_n p(C_1|x_n)^{t_n} \{1 - p(C_1|x_n)\}^{1-t_n} \quad (4.3)$$

with $\mathbf{t} = (t_1, \dots, t_N)^T$. While there is no closed-form solution for maximizing this data likelihood with respect to \mathbf{w} , the optimization problem is convex and hence has a unique global maximum [4]. In practice, the negative log-likelihood $-\ln(p(\mathbf{t}|\mathbf{w}))$ is minimized and usually extended by a regularization term to avoid over-fitting in case of perfectly

separable datasets. The final error function which gets minimized is thus

$$E(\mathbf{w}) = \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w}}_{\text{regularization term}} + \underbrace{C(-\ln(p(\mathbf{t}|\mathbf{w})))}_{\text{data term}} \quad (4.4)$$

$$= \frac{1}{2}\mathbf{w}^T\mathbf{w} - C \sum_n t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n) \quad (4.5)$$

with $y_n = p(C_1|x_n)$ and C is a user specified parameter to balance the data term with the regularization term [4][12]. LibLinear optimizes this function very efficiently using a trust region Newton method [23].

The binary LR classification described above can be extended to multi-class classification with $M > 2$ classes in two ways. The formally correct way would be to describe the true multi-class classification problem and devise an efficient learning method for it. A popular way, however, is to train M one vs. the rest (OvR) binary classifiers and combine their outputs to a probability distribution over all classes by normalizing over the sum of the M OvR class probabilities. This method is obviously practical, because it is fast to implement once the basic binary classification framework is done. In [18], a study between true and OvR multi-class classification using linear Support Vector Classifiers is carried out. The findings are that the practical performance in classification accuracy of both methods is comparable, with a slight advantage for the true multi-class formulation. For the work presented here, an OvR approach is used as it is the standard way LibLinear treats multi-class problems.

In order to achieve higher classification accuracy, we experimented with an additional feature space subdivision based on the approach of [24] and conceptionally equal to [2]. Instead of training one classifier for all features and all classes in the training dataset, the feature space is divided into a set of subspaces and a local classifier is trained for each subspace individually. Every local classifier thus only has to model the structure within a subspace of similar features rather than the structure of the whole feature space. By this means, a relatively simple linear classification model such as logistic regression reaches high classification accuracy. The feature space subdivision is implemented by clustering of the training features \mathbf{X} into K clusters via simple K-means clustering based on the euclidean distance between features. A new feature is evaluated by first examining which cluster (subspace) it falls into and then invoking this cluster's logistic regression classifier. The trained classification pipeline is shown as orange colored set of components in the block diagram in figure 4.1.

4.2.2. Choice of Feature

The described method is not tied to a specific feature, however, in this work, the classification was carried out based on the Fast Point Feature Histogram (FPFH) [30]. The FPFH is a rotation invariant 3d feature calculated from point cloud data without the use of color information. It summarizes the local geometry around a query point $p^* \in \mathbb{R}^3$ by means of three geometrical quantities $(\alpha_i, \phi_i, \theta_i)$ computed pairwise between the query point p^* and neighboring points $p_i, i = 1 \dots k$ within a specified radius r_f of p^* . For every point pair (p^*, p_i) with corresponding 3d normals (n^*, n_i) a so called Darboux *uvw* frame

$(u = n^*; v = (p^* - p_i) \times u; w = u \times v)$ is calculated. The above mentioned quantities are angular variations defined as

$$\alpha_i = v \cdot n_i \quad (4.6)$$

$$\phi_i = (u \cdot (p_i - p^*)) / \|p_i - p^*\| \quad (4.7)$$

$$\theta_i = \arctan(w \cdot n_i, u \cdot n_i) \quad (4.8)$$

where \cdot denotes the dot product. A preliminary Simplified Point Feature Histogram (SPFH) descriptor is built by discretizing the three dimensions (α, ϕ, θ) into typically $d = 11$ bins, building histogram statistics over the computed $(\alpha_i, \phi_i, \theta_i)$ and concatenating the histograms to a vector $\text{SPFH}(p^*) \in \mathbb{R}^{(d+d+d)}$. The final descriptor $\text{FPFH}(p^*) \in \mathbb{R}^{(d+d+d)}$ is calculated via a weighting of nearby SPFHs according to

$$\text{FPFH}(p^*) = \text{SPFH}(p^*) + \frac{1}{k} \sum_i \frac{1}{w_i} \text{SPFH}(p_i) \quad (4.9)$$

where w_i is the distance between the query point p^* and the neighborhood point p_i . This way, geometric information of up to twice the feature estimation radius r_f can enter the final FPFH descriptor. As the dense computation of FPFH features for a point cloud with n points and on average k neighbors is in $O(nk)$, this feature was successfully used in many approaches targeting realtime applications such as object pose estimation [16], road-side environment classification [2] or object category classification [24].

4.2.3. Example

While a full quantitative evaluation of the viewing direction classification is presented later in section 4.5, a short qualitative example is given here in figures 4.2 and 4.3. The dataset and classification pipeline showcased in this example is based on real captured point clouds of a standard mug from $N_{\text{views}} = 28$ training viewing directions. The mug is 9.5cm in height and the feature radius used is 3cm. As the mug has a simple geometry, but also a reflective symmetry as well as surface parts of high ambiguity (the body), it is an ideal object to demonstrate the viewing direction classification.

In figure 4.2 the learned feature subspaces are illustrated using a trained classification pipeline with $K = 5$ and $K = 10$ for the K-means clustering. For $K = 5$, the clustering separates concave body parts (blue) from upper and lower convex body parts (white, black) while features originating from the handle are clustered separately (green). In case of $K = 10$, the clustering is similar for the body parts whereas the handle and body parts close to the handle are subdivided into more clusters.

For each of the feature subspaces, a separate LR-classifier is trained which outputs a discrete probability distribution $p(D = m|f)$, $m \in [1, 28]$ given a single FPFH feature f . In figure 4.3, the resulting viewing direction classification is illustrated using an evaluation point cloud for viewing direction $D = 17$. Color coded on the points is the probability of a feature originating from the shown view, $p(D = 17|f)$. The benefits of the probabilistic classifier are visible in the histograms on the right. The top histogram shows how the classifier captures the reflective symmetry of the mug whereas the bottom histogram shows the general ambiguity of features originating from the body of the mug. Please note the more detailed description in the figure caption.

4.3. Bingham Mixture Measurement Model

To use the viewing direction classification described above for the sequential estimation of an object's rotation (cf. section 3.6), a measurement model $p(z_t|q_t)$ for a measured rotation $z_t \in SO3$ given an assumed rotation $q_t \in SO3$ needs to be defined. This is non-trivial due to two aspects. Firstly, z_t and q_t live in continuous domains while the view classification uses a discrete set of viewing directions. Secondly, the used FPFH features are rotationally invariant thus even a uniquely classified feature does not determine a unique object rotation, but leaves the object's rotation around the camera's optical axis unspecified. We will see that both aspects can be represented without much effort in a measurement model based on a Bingham mixture distribution.

For any object we can define a fixed coordinate frame o whose rigid body transformation wT_o describes the object's pose in the world coordinate frame w . The set of N_{views} training camera poses is specified by their transformation ${}^oT_c^m, m = 1, \dots, N_{views}$ relative to the object. Let us consider a new point cloud measurement I^* with computed features F^* taken from a known camera pose ${}^wT_{c^*}$. Assume one of the resulting features $f^* \in \mathcal{F}^*$ was perfectly classified to the single training view direction $m^* \in [1, N_{views}]$. Without invariance around the viewing direction m^* and in a non-probabilistic view of things, this feature would tell us that the object's rotation is

$$z_t := {}^wq_o = {}^wq_{c^*} \circ ({}^oq_c^{m^*})^{-1} \quad (4.10)$$

with ${}^wq_{c^*} = \text{Rot}({}^wT_{c^*})$ the rotation part of the current camera pose and $({}^oq_c^{m^*})^{-1}$ the training view orientation the feature was classified to.

We can start building a continuous probabilistic representation of this view classification by associating the training view direction m^* with a Bingham distribution describing the object's rotation with respect to the training view as follows

$${}^c b_o^{m^*} \sim \mathcal{B}(\mathcal{K}, ({}^oq_c^{m^*})^{-1} \circ \mathcal{V}) \quad (4.11)$$

$$\mathcal{K} = \begin{pmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{pmatrix} \quad \mathcal{V} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

with \circ denoting quaternion rotation of the directional vectors in \mathcal{V} .

Setting $\kappa_1 = \kappa_2 = \kappa_3 \ll 0$, this Bingham describes a rotational distribution with mode ${}^c q_o^{m^*}$ equal to the object's orientation in the training view m^* and a Gaussian-like rotational deviation around this mode. This is achieved by rotating the Bingham $\mathcal{B}(\mathcal{K}, \mathcal{V})$ into the correct frame by rotating its basis vectors by the fixed training view rotation $({}^oq_c^{m^*})^{-1}$.

To account for the rotational invariance around the camera's view axis, we assume the optical axis of our camera points in positive z-axis direction of the camera frame. By simply setting $\kappa_3 = 0$, which corresponds to the third column in \mathcal{V} and hence the z-axis, we model a uniform rotation about the camera axis as required. Setting the concentration parameters $\kappa_1 = \kappa_2 = \kappa$ big enough, this also effectively bridges the gap between the discrete classification and the continuous probabilistic representation.

By additionally rotating the distribution by the fixed current camera orientation ${}^wq_{c^*}$, we obtain a continuous probabilistic representation with mode equal to (4.10)

$${}^wb_o^{m^*} \sim \mathcal{B}(\mathcal{K}, {}^wq_{c^*} \circ ({}^oq_c^{m^*})^{-1} \circ \mathcal{V}) \quad (4.13)$$

Until now we considered a perfect classification of f to the class m^* . In practice, the classification pipeline will output a distribution $p(D|f)$ over training view labels. The continuous equivalent of this discrete distribution can be described by a mixture model with a single component describing an individual training view rotation as in (4.13) and the mixture weight vector given by $p(D|f)$

$${}^wb_o = \sum_m p(D = m|f) {}^wb_o^m \quad (4.14)$$

$$= \text{BMM}(\alpha = p(D|f), \mathcal{B} = \{{}^wb_o^m\}_{m=1}^M) \quad (4.15)$$

Instead of first building $p(z_t|q_t)$ and then inverting it to a distribution over q_t as in equations (3.11) to (3.13), above BMM directly describes a distribution over the object's poses given a measured feature f and the camera orientation ${}^wq_{c^*}$. Formally we can denote this as

$$p({}^wq_{o,t}|f, {}^wq_{c^*}) := {}^wb_o \quad (4.16)$$

Now, to fuse information from many features $\{f_j\}, j = 1, \dots, N_{feat}$, a simple strategy known as distribution summation [27] works well in practice. The classification distributions for all individual features are summed, renormalized and used as final weight vector for the distribution

$$\alpha_{\text{sum}} = \text{normalize}(\sum_j p(D|f_j)) \quad (4.17)$$

The final inverted measurement model of a set of features can thus be written as

$$p(q_t|z_t) \approx p({}^wq_{o,t}|\{f_j\}, {}^wq_{c^*}) = \text{BMM}(\alpha = \alpha_{\text{sum}}, \mathcal{B} = \{{}^wb_o^m\}_{m=1}^M) \quad (4.18)$$

and is represented by the *probabilistic orientation estimation* component in the block diagram in figure 4.1.

4.4. Algorithmic Details for Sequential Estimation

In section 3.6, it was explained how sequential fusion using Bingham mixture models as prior and measurement likelihood works using a SMC and M+R method. This also carries over to using the measurement model defined in equation (4.18) with some additional assumptions regarding the preprocessing and a specialty arising from the camera-axis invariance of the measurement model.

In the previous section we treated the rotational invariance of the used features through modeling the measurement distribution to be uniform around the camera's optical axis. Using a only a single view, this means the orientation of the object can only be detected up to an invariance around the camera axis. For a unique orientation estimate, the sequential estimation requires at least two views from two non-parallel viewing directions.

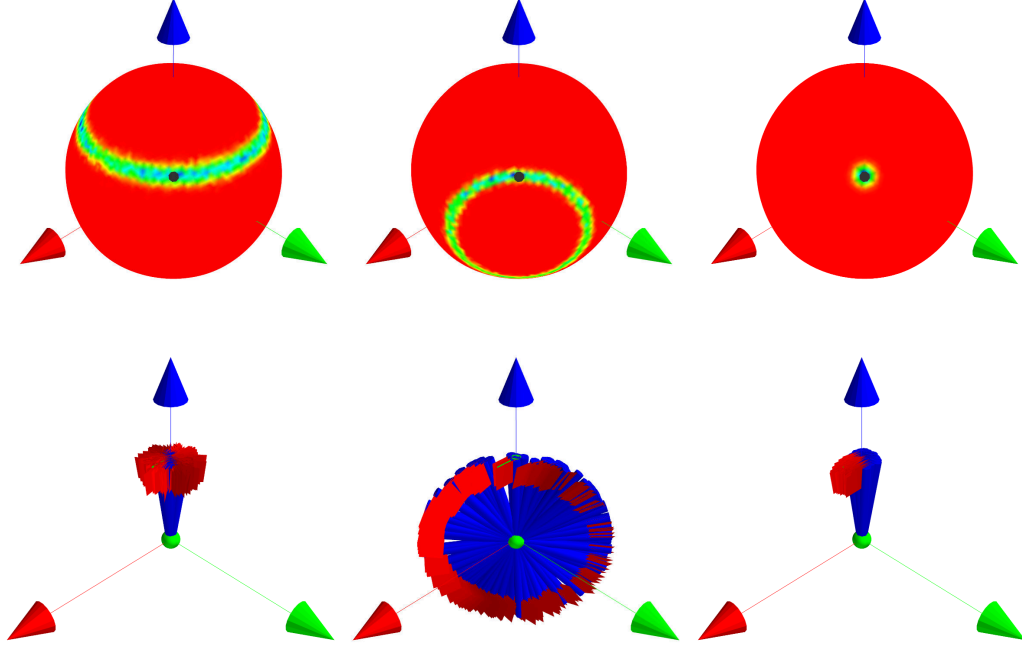


Figure 4.4.: Synthetic experiment for the fusion of two camera-axis invariant orientation measurements. All shown distributions are represented by a single Bingham distribution. Left column: first synthetic measurement distribution obtained by a view along axis $(0, 0, -1)$ illustrated as EGI plot and flag plot. Middle column: second synthetic measurement distribution obtained by a view along axis $(-\frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{2}, 0)$. Right column: fusion (multiplication) of the two measurement Bingham distributions which results in a unique orientation estimate correctly centered around the identity rotation.

This behavior is illustrated in figure 4.4 for two synthetic measurements. Each individual measurement results in a camera-axis invariant orientation distribution which shows as a ring on the EGI plot. Sequential fusion (multiplication) of the Bingham distributions results in a unique orientation estimate. In addition to the EGI-plots, the bottom row shows corresponding flag plots. A single flag encodes an orientation in a unique way using the convention that the blue pole points into the positive z-direction (blue axis) and the flag points into the positive x-direction (red axis). The flag plots are then generated by sampling 50 rotations from the underlying distribution and rotating an identity orientation flag by this rotation.

Furthermore, to apply equation (4.18) in practice, the preprocessing pipeline needs to output a set of features $\{f_j\}, j = 1, \dots, N_{feat}$ originating from the object of interest which is visible in the input point cloud I^* . In order to simplify the feature extraction and because the focus of this work is the sequential estimation of an object's rotation, aspects regarding object recognition (what object are we looking at?) and segmentation (which point cloud data belongs to the object of interest?) are not taken into account and solved by giving

appropriate knowledge into the preprocessing pipeline. Data segmentation is carried out at first and either unnecessary (simulated data) or done using a bounding box filter around the a priori known position of the object (real data). For the obtained point cloud segment $I_o \subset I^*$, feature extraction is performed with the parameter settings (normal and FPFH estimation radius) the used classifier was trained on. After feature extraction, up to N_{feat} features are selected to enter the measurement model equation of (4.18). The selection criterion is based on the classifier's confidence for a particular feature, which is calculated by means of the discrete entropy of the classification distribution

$$H(p(D|f)) = - \sum_m p(D = m|f) \log(p(D = m|f)) \quad (4.19)$$

The entropy $H(p)$ reaches its maximum value H_{max} for a uniform distribution and decreases the more peaked the distribution is. It reaches its minimum value for a distribution where $p(D = m|f) = 1$ for a particular m and $p(D = m'|f) = 0, \forall m' \neq m$. By ranking all extracted features of the segmented point cloud, the N_{feat} features with lowest entropy are selected to enter the measurement model described in equation (4.18).

4.5. Evaluation

4.5.1. Parameter Space and Parameter Selection using Simulated Data

In order to evaluate the proposed method for object rotation estimation, the influence of the following parameters has been analyzed using simulated and real test data.

Table 4.1.: Parameters of View Classification based Object Rotation Estimation

Parameter	Description
N_{views}	number of different training view poses
${}^oT_c^m$	$m = 1, \dots, N_{views}$, training view poses
N_{sets}	number of training point clouds per training pose
r_n	normal estimation radius
r_f	FPFH feature estimation radius
K	number of clusters for K-means feature space subdivision
C	$C \in \mathbb{R}^+$, inverse regularization strength for LR training
κ	measurement concentration for Bingham Mixture model
N_{feat}	maximum number of features used for rotation estimation per view

At first, a number of tests using simulated training and evaluation data has been performed. The purpose of these tests was to provide a proof-of-concept evaluation as well as to narrow down the whole parameter set to a couple of interesting and/or data dependent parameters for tests with real data. Parameter-wise, the focus of the simulation tests were on those parameters which have direct influence on the training data, namely N_{views} , ${}^oT_c^m$, N_{sets} , r_n and r_f . The simulated tests were performed using two 3d models roughly 11cm in height - a mug and a cartoon character, see figure 4.5 - and the simulation framework within the Point Cloud Library (PCL) [29]. This simulation is based on rendering a 3d

object model using virtual camera settings (focal length, view frustum, resolution) equal to those of structured light depth sensors like the Microsoft Kinect or Asus Xtion. The noise model applied is a simple Gaussian distributed noise with fixed standard deviation of 0.15cm on the depth measurement of each pixel followed by the typical depth value quantization. It does not include a depth varying noise model nor the typical artifacts such as missing data at edges or on surfaces almost parallel to the camera's optical axis.

For the simulated as well as the real data evaluation, the training view poses ${}^oT_c^m$ for the camera were generated to be approximately uniformly distributed on a viewing sphere around the object and pointing towards the center of the sphere (which equals the object's center of mass). The number of poses N_{views} was indirectly determined by the subdivision level of the pose generation algorithm. Simulated tests were carried out using $N_{views} \in \{12, 48\}$ and a view radius of 1.0m. The average difference in rotation between a generated pose and the closest neighboring pose (in terms of rotation difference) is $68.0^\circ \pm 12.7^\circ$ for the pose set with $N_{views} = 12$ and $31.1^\circ \pm 1.4^\circ$ for the pose set with $N_{views} = 48$. A qualitative impression of the pose density for $N_{views} = 48$ is given in figure 4.5. The training datasets have been generated with $N_{sets} = 3$ sample sets in order to give the classifier a chance to generalize over a particular view's feature noise characteristic.

For the feature computation, the normal estimation radius r_n and the FPFH computation radius r_f are the relevant parameters. For useful statistics over the estimated normals, it is common practice to choose $r_f \geq r_n$. For the simulated data $(r_n, r_f) \in \{(.01m, .01m), (.02m, .03m)\}$, were explored. By these means, four training datasets based on simulated data have been created combining the two possibilities for $N_{views} \in \{12, 48\}$ with the two feature estimation parameter sets.

The parameters involving the classification pipeline (K and C) and the parameters involving the rotation estimation (κ and N_{feat}) were jointly evaluated. The classification pipeline has been trained using $K \in \{1, 5, 10\}$ and $C = 1.0$ on above four training datasets. The pose estimation parameters have been varied according to $\kappa \in \{-900, -27, -18\}$ and $N_{feat} \in \{100, 300, 1000\}$.

The simulation evaluation then has been carried out based on 20 random view sequences with 20 views per sequence. The random view sequences have been generated by selecting camera poses uniformly on a viewing sphere but with the same radius as used for training data generation. This means that the evaluation camera poses generally do not coincide with the training camera poses, which is realistic given an online application of the algorithm but also poses a significant challenge for the classification pipeline as we will see in the later evaluation on real data. The same 20 sequences have been used for all evaluations to ensure comparability. As fusion algorithm, the SMC based fusion with $M = 100000$ samples was employed as it seemed to work better with measurement models with many components.

Without a further quantitative analysis but by inspection of MAP error evolution plots similar to figure 3.4, it showed that the evaluation runs based on $N_{views} = 12$ or $(r_n, r_f) = (.01m, .01m)$ do not lead to a converging rotation estimate over most of the 20 sequences and these parameter values could be discarded for further evaluation. For the classifiers based on $N_{views} = 48$ and $(r_n, r_f) = (.02m, .03m)$, the remaining 27 parametrizations for K, κ and N_{feat} are ranked with respect to the MAP error averaged over the last 10 views of the sequence and over all 20 sequences. This is the same error metric as in equation (3.24), except now we average over 20 random sequences instead of three objects. Evaluation for

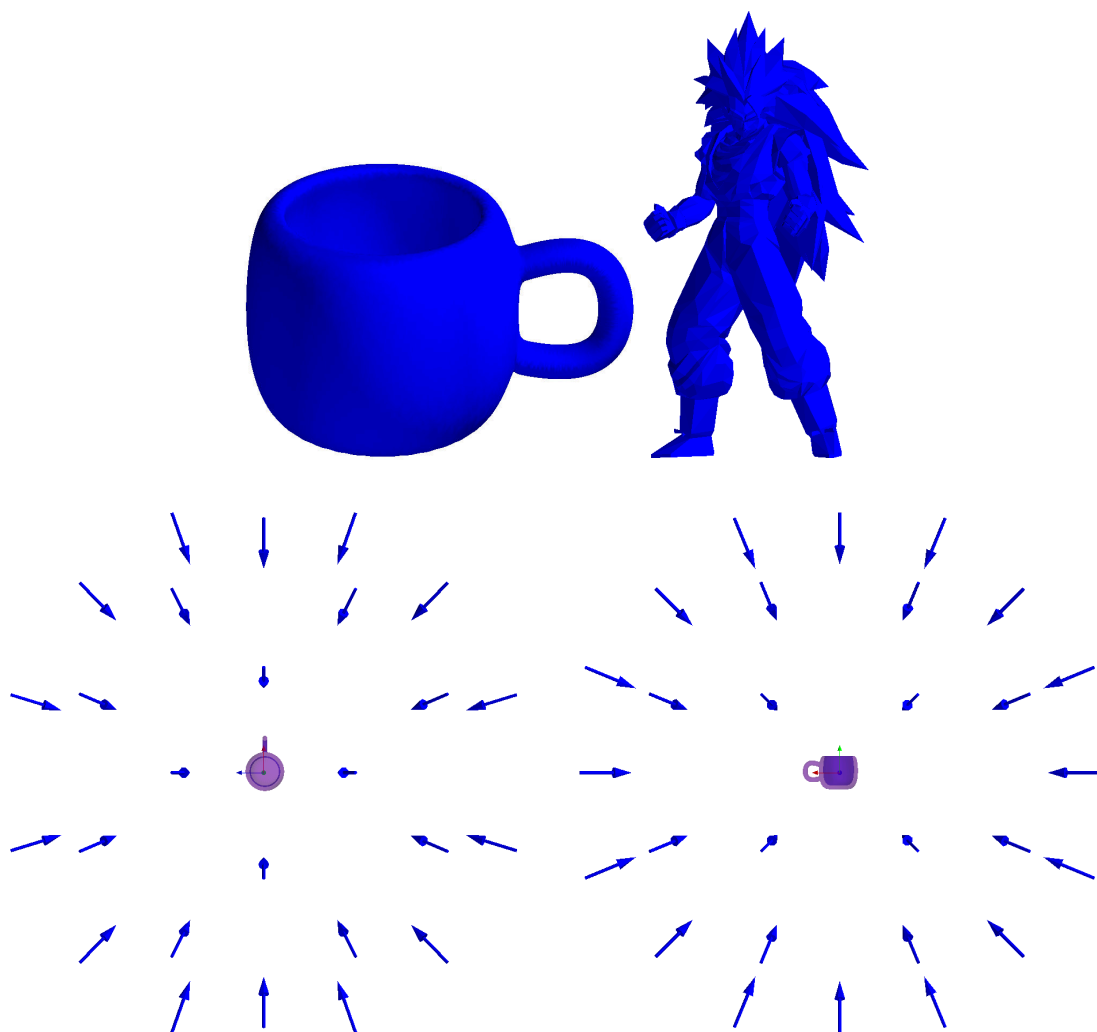


Figure 4.5.: Top row: mug and cartoon character models used for simulation evaluation.
Bottom row: 48 training directions, top and side view.

the mug and cartoon model has been performed independently and the rankings are given in figure 4.6. As expected, a higher number of K-means clusters and taking more features into account in the measurement model in general leads to a smaller average MAP error. Regarding the concentration parameter, higher uncertainties $\kappa \in \{-18, -27\}$ tend to work better. It's interesting to note, that the model's geometric complexity, which manifests itself in a more diverse feature vector set, has measurable influence on the overall obtainable orientation error. The top ranked cartoon model parametrization obtains an average error of 3.61° whereas the best parametrization for the mug model is on average more than 2° worse.

4.5.2. Evaluation using Real Data

Data Acquisition

In order to evaluate the rotation estimation on a real dataset, training and evaluation scans of two objects - a mug and a bunny (cf. figure 4.7) - have been captured. For data acquisition an Asus Xtion camera mounted onto a Kuka KR-16 robot arm has been used. Through a camera-to-robot calibration, the Kuka arm can position the camera accurately at the desired locations ${}^oT_c^m$. Training and evaluation data has been captured by putting the object in a known position on a flat table surface. Due to limitations of the robot's workspace the viewing sphere radius for the training and evaluation poses was set to 0.65m.

Training camera poses have been generated as in figure 4.5. Evaluation camera poses were generated with one subdivision level more, where none of the evaluation camera poses is identical to one of the training camera poses. As it is difficult to obtain real scans from all directions on a viewing sphere, we limited training and evaluation to poses in the upper half sphere above a table surface. This finally leads to $N_{views} = 28$ training poses and 100 evaluation poses which are illustrated in figure 4.7. For every training pose, $N_{sets} = 4$ scans have been captured for classifier training and two additional ones as a separate validation set. For the evaluation camera pose set, two scans have been captured. Using knowledge about the object's position and size, the data belonging to the object of interest has been segmented from training as well as test scans by filtering out the planar table surface and then applying a bounding box filter. Feature extraction has been performed on the resulting segments using $r_n = 0.02\text{m}$ and $r_f = 0.03\text{m}$.

Classifier Training

For the classifier training on real data, the parameter ranges $K \in \{1, 5, 10\}$ for the number of K-means clusters and $C \in \{.1, 1.0, 10.0\}$ for the logistic regression regularization have been explored by means of a 4-fold cross validation. The 4 folds are defined by the four sample sets acquired ($N_{sets} = 4$) and no further randomization or test/training split was performed. The cross validation metric used is the logistic loss also known as cross entropy loss. It takes the probabilistic output of the LR classifier into account and is defined as

$$\text{logloss}(t, p) = \sum_x t(x) \log(p(x)) \quad (4.20)$$

for discrete probability distributions $t(x)$ and $p(x)$ over a space $x \in X$ where $t(x)$ represents the ground truth distribution and $p(x)$ is the predicted distribution. In the classifi-

4. Viewing Direction Classification: Application to Orientation Estimation

	avg. MAP [°]	K	κ	N_{feat}		avg. MAP [°]	K	κ	N_{feat}
1	3.61°	10	-18	1000	1	5.74°	10	-27	300
2	3.82°	10	-27	1000	2	6.20°	10	-27	1000
3	4.17°	5	-27	1000	3	6.20°	5	-27	1000
4	4.18°	10	-27	300	4	6.73°	10	-18	300
5	4.20°	10	-18	300	5	7.07°	10	-120	1000
6	4.28°	5	-18	1000	6	7.12°	10	-18	1000
7	4.48°	5	-18	300	7	7.26°	10	-27	100
8	4.51°	5	-27	300	8	7.29°	5	-18	1000
9	5.32°	10	-27	100	9	7.40°	10	-120	300
10	5.62°	10	-18	100	10	7.64°	10	-120	100
11	6.49°	10	-120	1000	11	7.67°	5	-120	1000
12	6.51°	10	-120	300	12	8.40°	10	-18	100
13	6.61°	5	-27	100	13	10.03°	5	-120	300
14	6.72°	10	-120	100	14	10.85°	1	-120	300
15	6.73°	5	-18	100	15	11.28°	1	-120	1000
16	6.90°	5	-120	1000	16	11.55°	1	-18	300
17	7.08°	5	-120	300	17	12.13°	1	-27	1000
18	7.98°	5	-120	100	18	12.46°	5	-27	300
19	9.57°	1	-27	1000	19	13.06°	1	-27	300
20	9.79°	1	-18	1000	20	14.01°	5	-18	300
21	9.79°	1	-120	1000	21	14.27°	1	-18	1000
22	10.43°	1	-120	300	22	14.71°	1	-18	100
23	11.74°	1	-27	300	23	14.80°	5	-120	100
24	12.54°	1	-18	300	24	19.47°	5	-18	100
25	15.16°	1	-27	100	25	19.58°	1	-27	100
26	15.66°	1	-120	100	26	20.28°	1	-120	100
27	16.88°	1	-18	100	27	20.71°	5	-27	100

(a) cartoon character
(b) mug

Figure 4.6.: Ranking of all parametrizations for the cartoon and mug model. Fixed parameters: $N_{views} = 48$, $N_{sets} = 3$, $(r_n, r_f) = (.02m, .03m)$ and $C = 1.0$

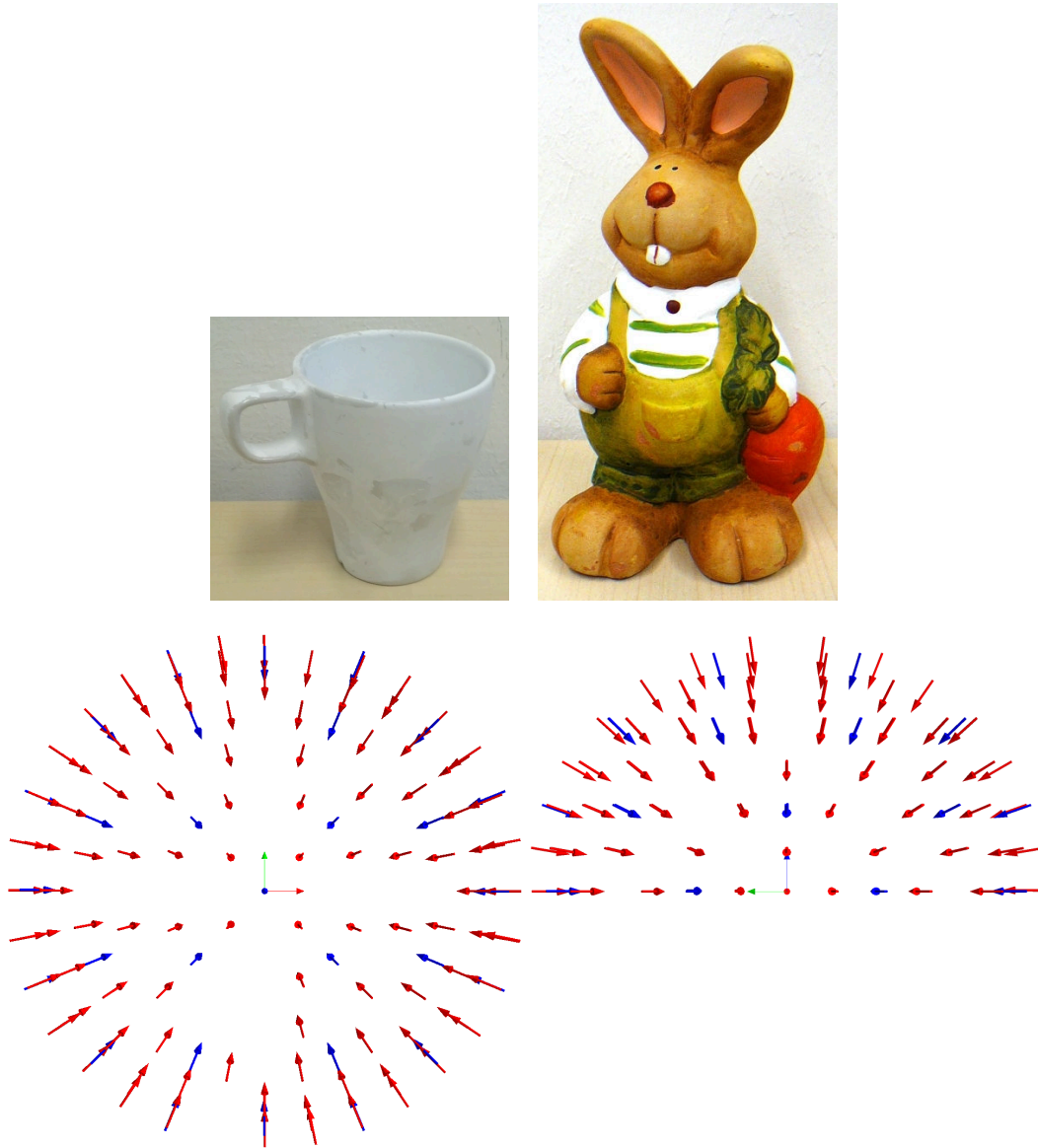


Figure 4.7.: Top row: images of the used real world objects refereed to as mug and bunny object. Bottom row: training (blue) and evaluation (red) dataset poses for the experiments with real data; on the left, top view of the camera poses and on the right a side view. The object is in the center of the view sphere. Note that evaluation and training viewing directions do not coincide on purpose.

avg. logloss	std. dev.	K	C	avg. log-loss	std. dev.	K	C
-1.57886	0.43376	10	1.0	-1.99539	0.16346	10	0.1
-1.58922	0.26503	10	0.1	-2.05744	0.16180	5	0.1
-1.65085	0.53596	10	10.0	-2.12633	0.33440	10	1.0
-1.84854	0.24731	5	0.1	-2.13468	0.25056	5	1.0
-1.86035	0.36980	5	1.0	-2.24397	0.31133	5	10.0
-1.88594	0.43127	5	10.0	-2.30537	0.42968	10	10.0
-2.34703	0.33612	1	1.0	-2.55143	0.14739	1	0.1
-2.34828	0.27285	1	0.1	-2.59923	0.19464	1	1.0
-2.34982	0.34380	1	10.0	-2.60467	0.19130	1	10.0

(a) bunny
(b) mug

Figure 4.8.: Cross validation scores for explored parameter ranges.

cation scenario, $t(x)$ is defined as 1 for the correct class label and 0 everywhere else and therefore the logistic loss collapses to the log-probability of the true class label. For a set of predictions and ground truth labels the logistic loss averaged over the set is a measure of the classifiers performance. Its ideal value would be zero and the worse the performance the more negative the value gets. For the two class case, the logistic loss is equal to the data likelihood function of the logistic regression model itself, but without the regularization (cf. equation (4.3)). Figure 4.8 summarizes the cross validation results. Similar to what we have seen in the evaluation on simulated data, more K-mean clusters work in general better. The bunny model allows overall better classification results due to its greater variance in surface geometry which leads to less ambiguous classifications. This can be seen in the logistic loss score in figure 4.8 as well as the confusion matrix plots in figure 4.9. For the confusion plots the best parametrization has been selected and evaluated on previously unseen data - the two additional sample sets recorded. Although the confusion plot for the mug reveal many misclassifications (off diagonal entries), this does not necessarily have to lead to bad orientation estimates in general. While the confusion plots are based on the single most likely class, the orientation estimation takes the complete predicted distribution into account and thus as long as the view ambiguity of a feature is correctly represented in the output distribution it will not lead the orientation estimation into a wrong direction. For the subsequent evaluation of the sequential rotation estimation, the best scoring classifier parametrization according to the cross validation was used.

Rotation Estimation Results

Two experiments targeting the convergence and correctness of the rotation estimation have been carried out. Similar to the evaluation on simulated data, they are based on assessing the MAP error on 20 random sequences with 20 views per sequence. The first set of sequences was created by randomly choosing views out of the 100 evaluation camera poses and will be termed S_{100} , the second set was created by using training view poses (S_{28}).

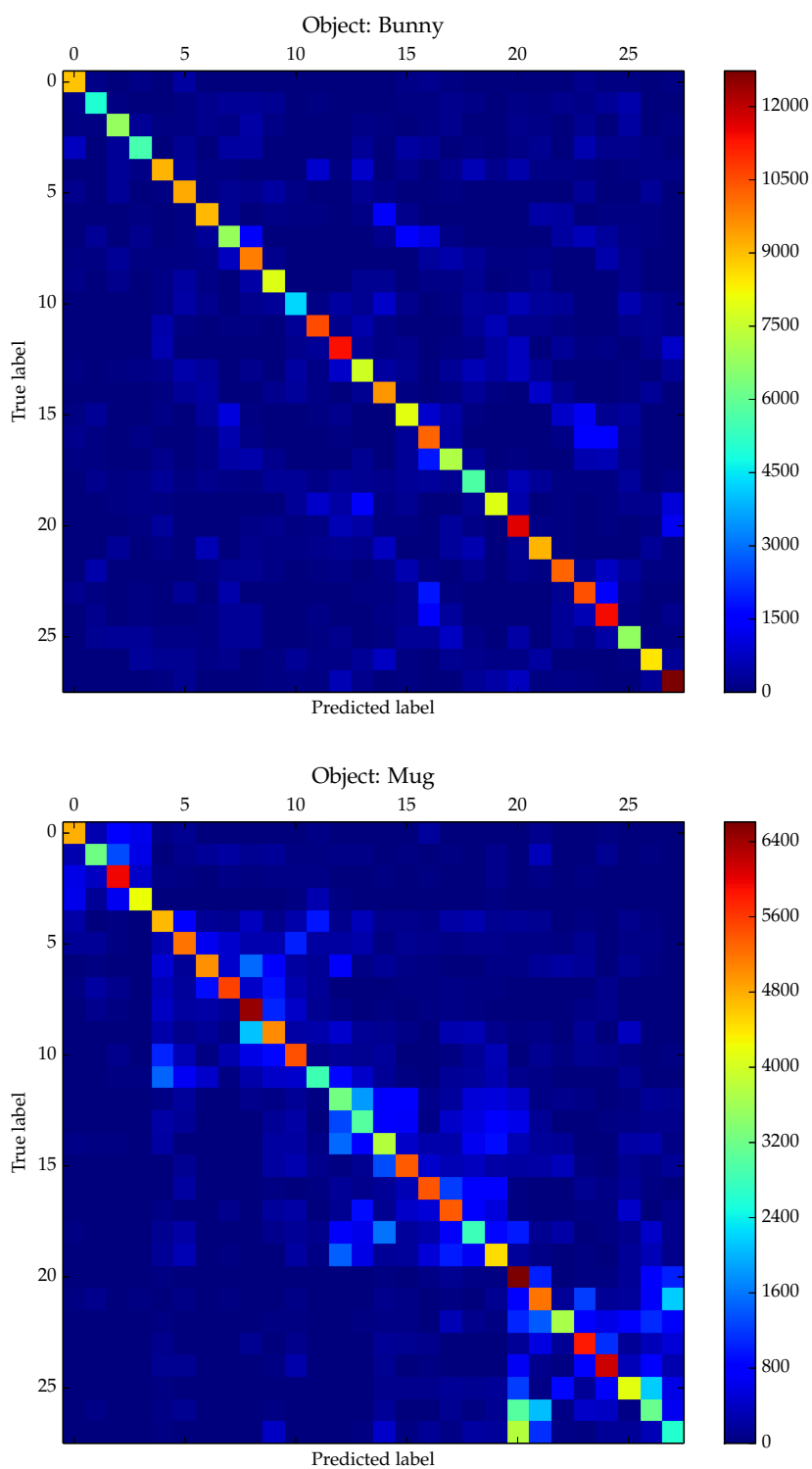


Figure 4.9.: Confusion matrices for the best classifier parametrization for each object model. Predicted labels are the 28 training view directions.

Whereas the first set of sequences thus resembles an online application in which the views of the object generally will not coincide with training views, the second set is a best case scenario by providing the classification pipeline with online data from the viewpoints it was trained on. Such viewpoints could potentially be chosen in a real online application via a next-best-view planning approach, but this remains to be proven. For the S_{28} sequences, the two captured sample sets which were not used for classifier training have been used for the evaluation. Hence all evaluations are based on data not previously used for training.

The evaluation on the S_{100} sequences was carried out with orientation estimation parameters ranging in $\kappa \in \{-27, -60, -120\}$ and $N_{feat} \in \{300, 1000\}$. The obtained performance measured via the average MAP error over all sequences and the last 10 views is given in figure 4.10. For the best ranked parametrizations, the MAP error evolution for all 20 sequences is plotted in figure 4.11. The results obtained on real data are similar to the previous results with simulated data. Rotation estimation for the geometrically rich bunny object converges to estimates with an average of error 4.74° with the best parametrization, whereas the mug object with a largely ambiguous surface area (the body) on average performs much worse (average error of 9.08° with the best parametrization). As none of the camera poses of the S_{100} sequences are camera poses actually used for training the classifiers, it is interesting to see how much this degrades the estimation performance. To assess this, an exemplary evaluation using $N_{feat} = 300$ and $\kappa = -27$ has been carried out using the S_{28} sequences, which contain data unseen by the classification pipeline but captured exactly from the training views. The resulting MAP error plots are given in figure 4.12. As can be seen qualitatively, the average MAP error is significantly lower compared figure 4.11. The MAP error averaged over the last 10 views and all 20 sequences for the bunny object is 0.67° , for the mug object it is 0.78° . The large difference in the remaining error after convergence between on- and off-training direction sequences is explainable by the discrete nature of the viewing direction classification. For views obtained from the training directions, the classification is stable and mostly correct as the S_{28} sequences show. Views obtained in between training directions should ideally result in a classification distribution with probabilistic weight partitioned over nearby training directions. Together with the modeled uncertainty of the training directions via the Bingham distribution, this would result in correct continuous rotation estimates even for non-training directions. In practice, however, the results suggest that the classification does not partition the probabilities as wanted but instead most probabilistic weight falls into one of the nearby views. As the distance between neighboring views among the 28 training views is roughly 31° , this can result in a theoretical MAP error of up to $\sim 15.5^\circ$ and explains why some of random sequences of the S_{100} sequences converge to estimates with a remaining error of over 10° .

Despite the average MAP error numbers just given, the behavior of the sequential estimation especially during the first couple views as seen in figure 4.11 and 4.12 is representative of the inner workings of the presented method. After view zero, no matter how perfect this view was recognized by the classifier, the plots show rotation errors of almost 180° for the S_{100} as well as the S_{28} experiments. These errors stem from the inherent rotational invariance around the camera axis which was introduced into the measurement model as the used features are rotational invariant and no further feature frame is estimated or used. In theory, at least one additional view (so two views in total), ideally orthogonal to the first one, is necessary to resolve this ambiguity. This resolving behavior is clearly seen

at view 2 in the S_{28} sequences for both objects, where the MAP error drops below 10° for most random sequences. In case of the S_{100} sequences, a similar behavior can be observed although much more uncertainty is left during the first views because the view classification for off-training direction views is more ambiguous, partly wrong or partitioned in a biased way over nearby training directions. The higher classification ambiguity with off-training directions shows as generally slower convergence behavior compared to the S_{28} sequences and wrong classifications, which lead to intermediate increases in the MAP errors of some random view sequences, occur more often for the S_{100} sequences as well. The MAP error evolution plot for the mug (figure 4.11) shows one particularly bad sequence which does not converge until the 14th view. A closer investigation here revealed that for this sequence, in views 0, 2, 4, 5 and 10 (of which 0, 2 and 5 coincidentally were the same viewing direction) many of the features were classified with high certainty to a completely wrong training view direction which thus misleads the orientation estimation. The larger tendency for wrong feature classifications for the mug model compared to the cartoon model is visible in many sequences during the first eight views of the S_{100} set as the MAP error for many mug sequences tend to jump up and down a few times before convergence.

The difference in estimation behavior for the on-training direction sequences in S_{28} compared to the off-sequences in S_{100} suggest that deviations from the training directions have a great impact on the classification pipeline. Although the used FPFH features are computed locally and in principle rotation and scale invariant, it seems that the object's self occlusion and thus missing or additional surface points play an important role for the feature computation and thus affect the classification performance. On the one hand, self occlusion allows for characteristic features exploitable by the classifier, on the other hand, these patterns sometimes seem to vary rather strongly even with slight changes in the viewing direction. Figure 4.13 illustrates this behavior for the mug object and the best ranked classification pipeline in figure 4.8. For every point of the displayed scans, the respective feature has been classified and the classification probability $p(D = m|f)$ for a specific training view direction m is visualized color coded. Whereas the left column shows scans obtained from exactly the training view direction m , the right column shows a scan from the closest evaluation viewing direction. The scan pairs in one row have been transformed in the same reference frame, hence the small view dependent variation in the scans is easily visible. Due to the small change in viewing direction, some of the probabilistic mass should be partitioned away from the direction visualized on the left to other training directions close to the evaluation direction. This effect can be seen in the top row where the largely red area around the handle on the left image gets lower probabilities on the right image, for example red regions shift to orange and orange regions to green. We also see some topological changes in the point clouds due to the shift in viewing direction. For example, the rim on the left side of the mug is closed from the training perspective but open on the evaluation point cloud. This leads to spurious regions with high probability for the training direction where the original training point clouds did not show any significant probability. Such regions disturb the orientation estimation. The bottom row shows a more severe case of non-robustness against slight changes in viewing direction. The training point cloud shows clear regions of high probability mass where the handle touches the mug body which don't carry over to the close-by evaluation view. An investigation for this view pair revealed that most of the probability mass for the evaluation

avg. MAP [°]	κ	N_{feat}	avg. MAP [°]	κ	N_{feat}
4.74	-27	1000	9.08	-120	1000
5.58	-60	1000	9.27	-60	300
6.15	-120	1000	9.56	-60	1000
6.70	-27	300	10.38	-27	300
7.17	-60	300	12.33	-27	1000
7.79	-120	300	14.82	-120	300

(a) bunny
(b) mug

Figure 4.10.: Ranking for pose estimation evaluation on the S_{100} sequence set.

view is concentrated at a viewing direction on the other side of the view sphere which will significantly disturb the orientation estimation. The evaluation view in this bottom row illustration turns out to be the 0th, 2nd and 5th of the outlier mug run mentioned before, which explains the long lasting phase with a rotation error of almost 180° .

4.6. Summary

Summing up this chapter, a method for estimating the orientation of known objects in 3d based on depth information has been presented. A conceptually simple feature-to-viewing direction classification together with an appropriate probabilistic measurement model based on Bingham mixture models allows to estimate an object's rotation with real world orientation errors below 10° after fusion of of about 10 views of the object. This accuracy is not yet comparable to state-of-the-art single view methods, however, promising results with errors below one degree have been achieved when restricting the relative orientation between object and camera to training view directions. Further, the local and dense nature of the approach without the necessity to match features to previously extracted interest points makes the approach interesting to scenarios with occlusions and applicable to objects with simple geometric shapes such as a mug. The implicit view-related object model built by training a viewing direction classifier will be investigated in the next chapter and is another interesting aspect of the presented method.

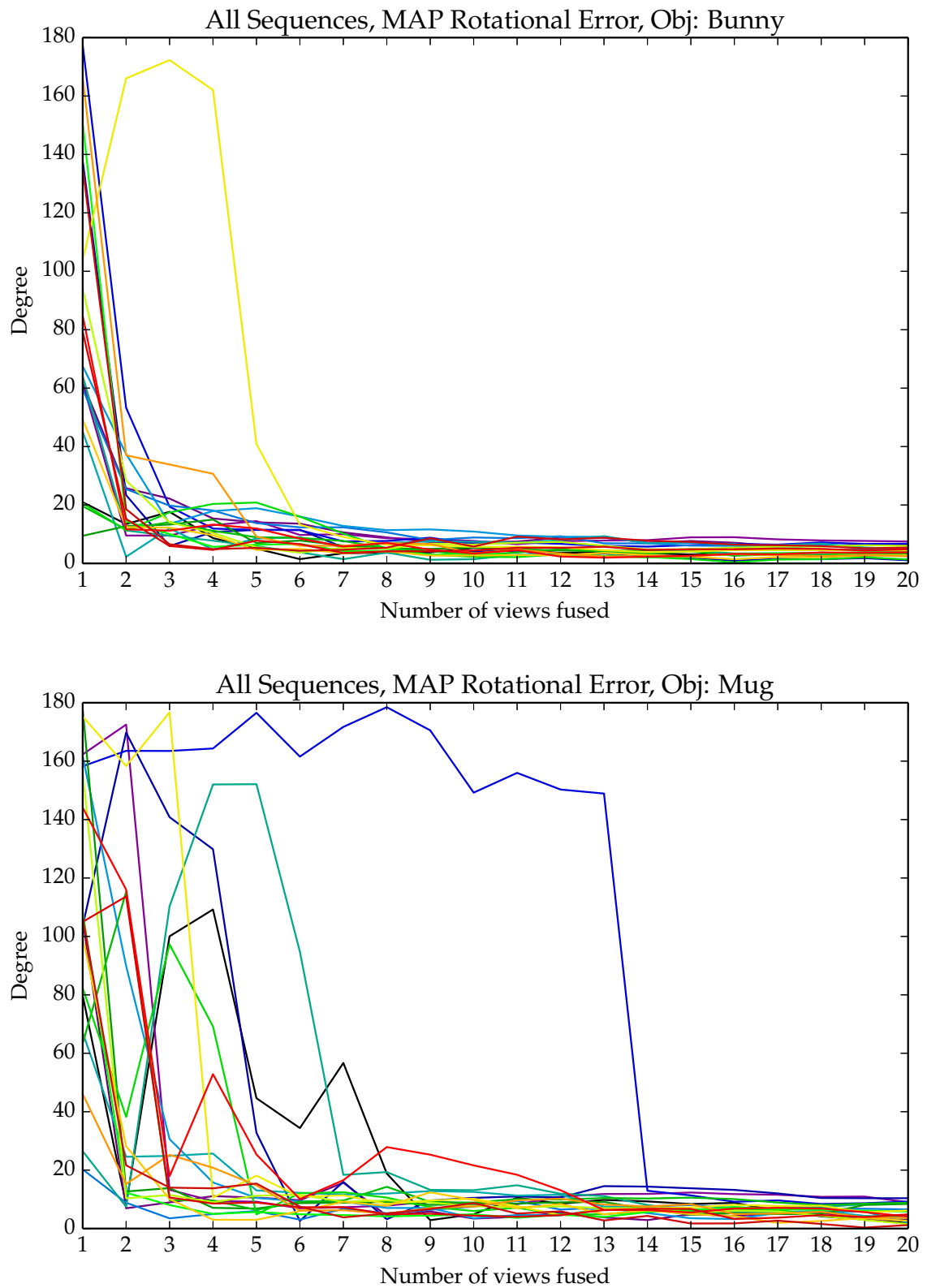


Figure 4.11.: MAP Rotational Error plots for 20 random sequences with 20 views.

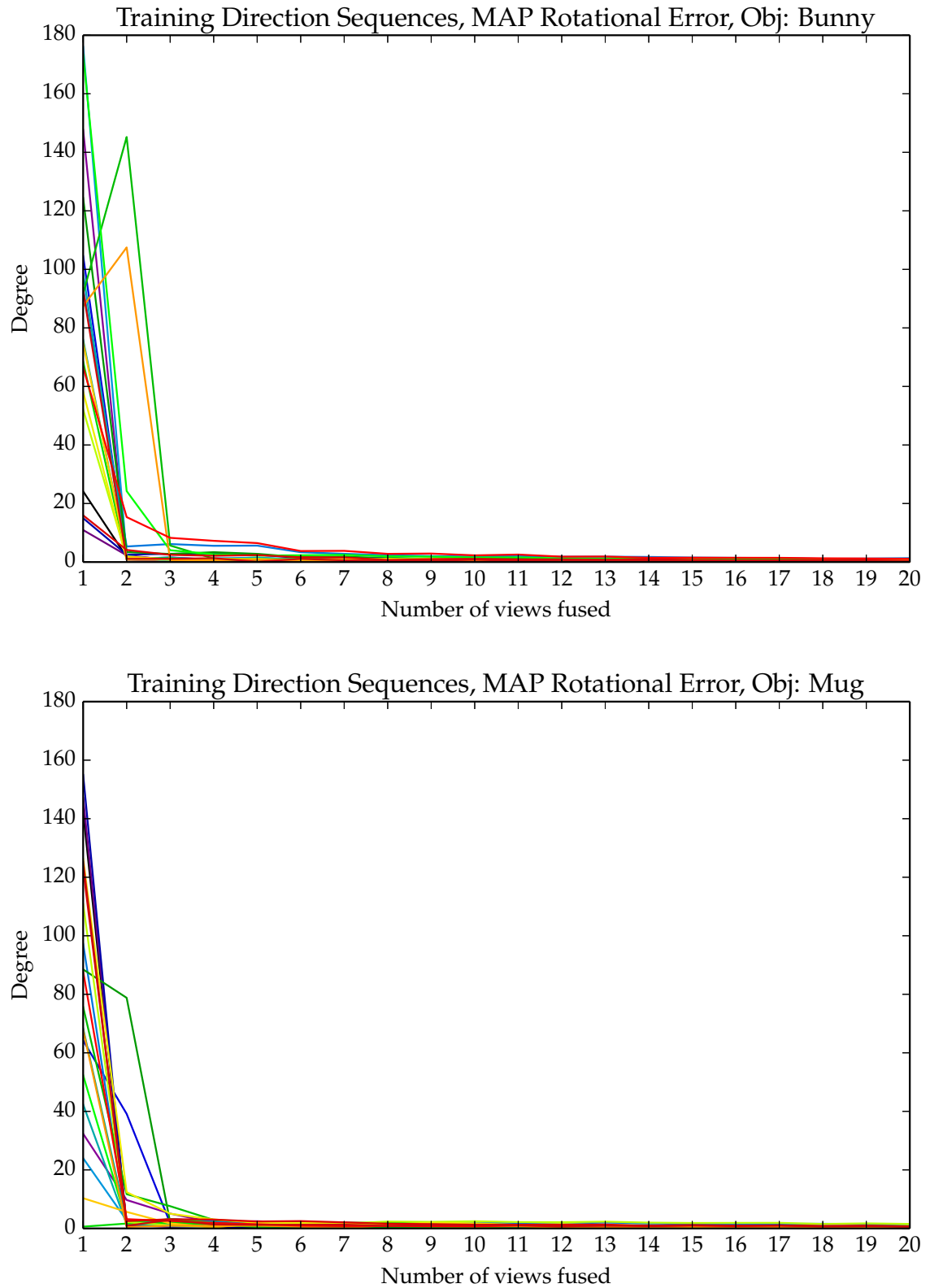


Figure 4.12.: MAP Rotational Error plots for 20 random sequences constrained to viewing directions present in the training data.

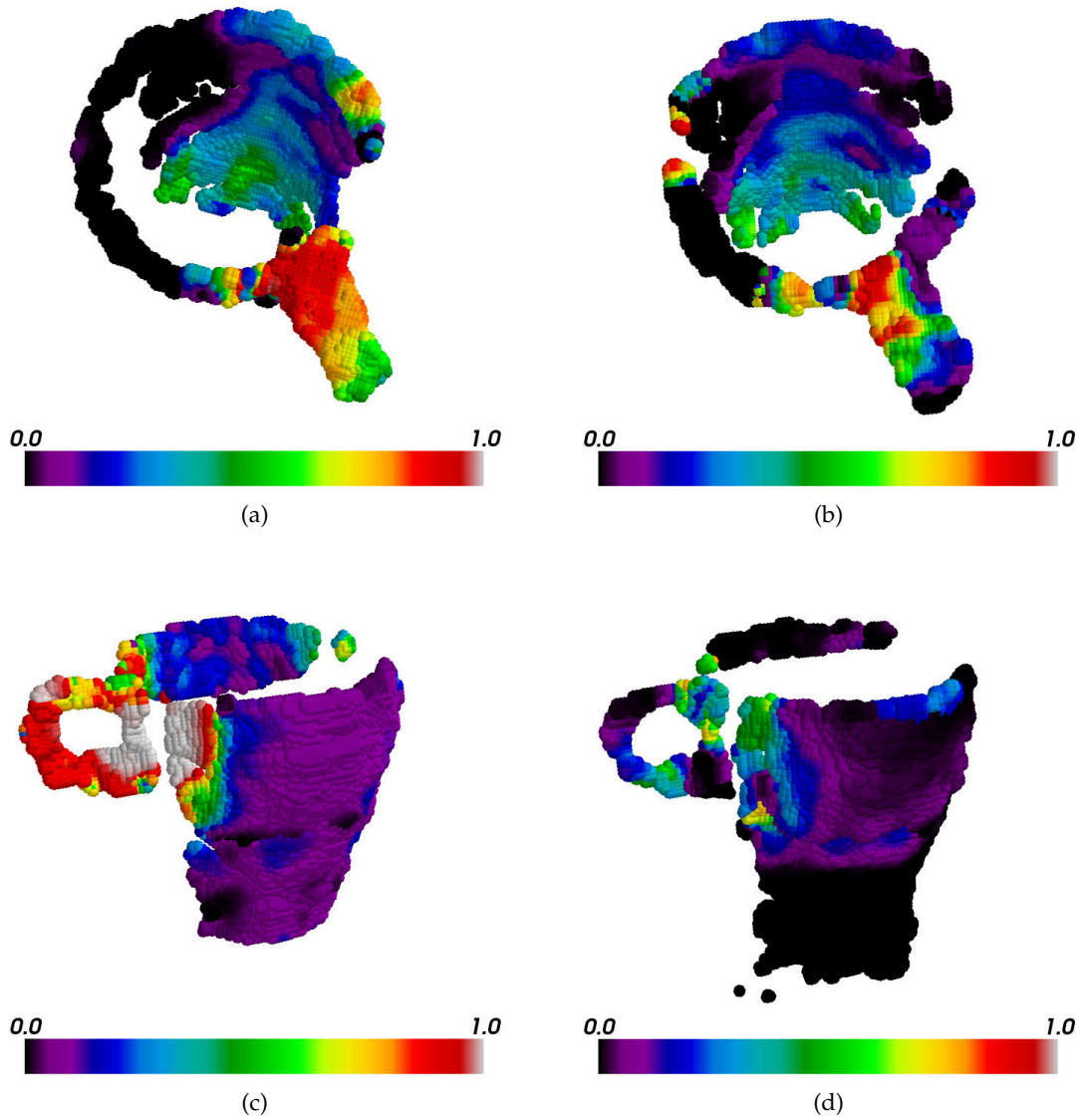


Figure 4.13.: Left column: point clouds obtained from training directions $m = 1$ (a)) and $m = 18$ (c)). Right column: point clouds for the respective closest evaluation direction. Each row's clouds are rotated into a common frame. Color coded is the classification probability $p(D = m|f)$; in a) and b) $p(D = 1|f)$, in c) and d) $p(D = 18|f)$. Per row, regions which obtain high probabilities on the left point cloud should obtain almost as high probabilities on the right point cloud. This indicates that the classification is stable also for deviations from training view directions as some of the probabilistic weight is shifted to nearby training directions. While the top row's scan pair exhibits this behavior, the bottom row's pair fails to do so.

5. Viewing Direction Classification: Application to View Planning

In the previous chapter an approach to orientation estimation based on viewing direction classification was presented. The viewing direction classification is motivated by the idea, that this classification allows an insight into which viewing directions and also which object parts are significant for the pose estimation of the object. This chapter now investigates this for the simulated datasets and shows how a local model of informativeness is obtained using the previously described classification framework. In the last section suggestions for possible next-best-view criteria are derived from the gained insight.

5.1. View Informativeness

As described in detail in chapter 4, the classification pipeline is trained to predict the training view direction a feature is observed from. As the used LR classifier outputs a probability over training view directions $p(D = m|f)$ given a single feature f , the classification pipeline implicitly encodes a model of view-related surface ambiguity. The broader and more uniform a feature's view distribution is, the less it tells us about how the object is oriented with respect to the camera.

We can exploit the classifier's model of surface ambiguity by evaluating which training view directions have features which identify the view correctly. The training view which results in the most unambiguous classification in this sense is the most informative for estimating the object's orientation. For every training direction m' separately, the training views are analyzed by first extracting features with the same settings as used by the online applied classifier. The features are ranked according to the entropy of their classification distribution and the top N_{feat} features are selected to estimate the view's informativeness. This basic feature ranking and selection procedure is the same as performed before orientation estimation and described in section 4.4. The selected feature's view distributions are summed and a measure of correctness is obtained by calculating the discrete KL divergence between the correct distribution p^* and the extracted summed distribution p_{sum} . The KL divergence measures the difference of the extracted distribution from the theoretically correct distribution and is defined as

$$d_{KL}(p^*||p_{sum}) = \sum_m p^*(m) \ln \frac{p^*(m)}{p_{sum}(m)} \quad (5.1)$$

$$= \ln \frac{1.0}{p_{sum}(m')} \quad (5.2)$$

where the correct distribution is defined as 1.0 for the training view direction of concern m' and 0.0 everywhere else. The simplification in the second equality is thus possible due

to the form of p^* . We can see that the KL divergence is zero for a perfect summed feature distribution with $p_{sum}(m') = 1.0$ and goes to positive infinity as $p_{sum}(m')$ approaches zero. For an expected informativeness ranking of a viewing direction, the KL divergences of all training point clouds for that direction are averaged and ranked in ascending order. The resulting ranking is illustrated in figure 5.1 using simulated data for the cartoon and mug model. For interpretation purposes, some of the viewing directions are illustrated by means of a rendering of the object observed from that viewing direction.

For the mug model one can observe that the best viewing directions lie on the plane defining the reflective symmetry of the mug whereas the least informative views show large parts of the body of the mug. This is intuitively correct as features on the body of the mug can be observed from many directions. Furthermore, as the features are rotational invariant and the classification is based on individual features and therefore local information, most views, even if they show the handle, are ambiguous as the reflective symmetry cannot be resolved. The most unambiguous views are therefore correctly identified as the ones on the reflection plane which additionally show large parts of the handle or the inside of the mug.

For the cartoon model there is no clear intuitive ranking of views from a human perspective, but we will shed light on which parts of an object are informative and thus the reason for this ordering within the next section.

5.2. Model Surface Informativeness

The view ranking in the previous section was based on accumulating information of several features of a view into a summed distribution and assessing the correctness of this distribution. This way we obtained information about the informativeness of a viewing direction. In this section, we accumulate information of features within a small neighborhood of a point on the object's surface and thus assess how informative a surface point is.

For the method presented here, we assume the availability of a set of points $S = \{s_0, \dots, s_N\}$, $s_i \in \mathbb{R}^3$ representing the complete surface of the object. For the cartoon and mug object in this evaluation, we have 3d models and thus the set S was generated by sampling the surface of the 3d model with uniform density using the stratified sampling approach described in Nehab and Shilane [26] and Doria [7]. Our objective is now to score every surface points s_i by means of how informative the point is or more precisely, how informative features originating from that point are. As a surface point may be visible from more than one viewing direction, at first a scoring matrix $K \in \mathbb{R}^{N \times M}$ is computed which ranks the N surface points separately with respect to the M training view directions. For a given surface point s_i and viewing direction m , the nearest neighbor points $NN_m(s_i)$ within a radius of 0.5cm are obtained in the training point clouds for that viewing direction. The corresponding features $NNF_m(s_i)$ in the training point clouds are extracted and the average KL divergence between the correct and the predicted view distributions is calculated and stored

$$K[i, m] = \frac{1}{|NNF_m(s_i)|} \sum_{f \in NNF_m(s_i)} d_{KL}(p^* || p(D|f)) \quad (5.3)$$

For further reference, the average KL divergence stored at $K[i, m]$ will be termed view-

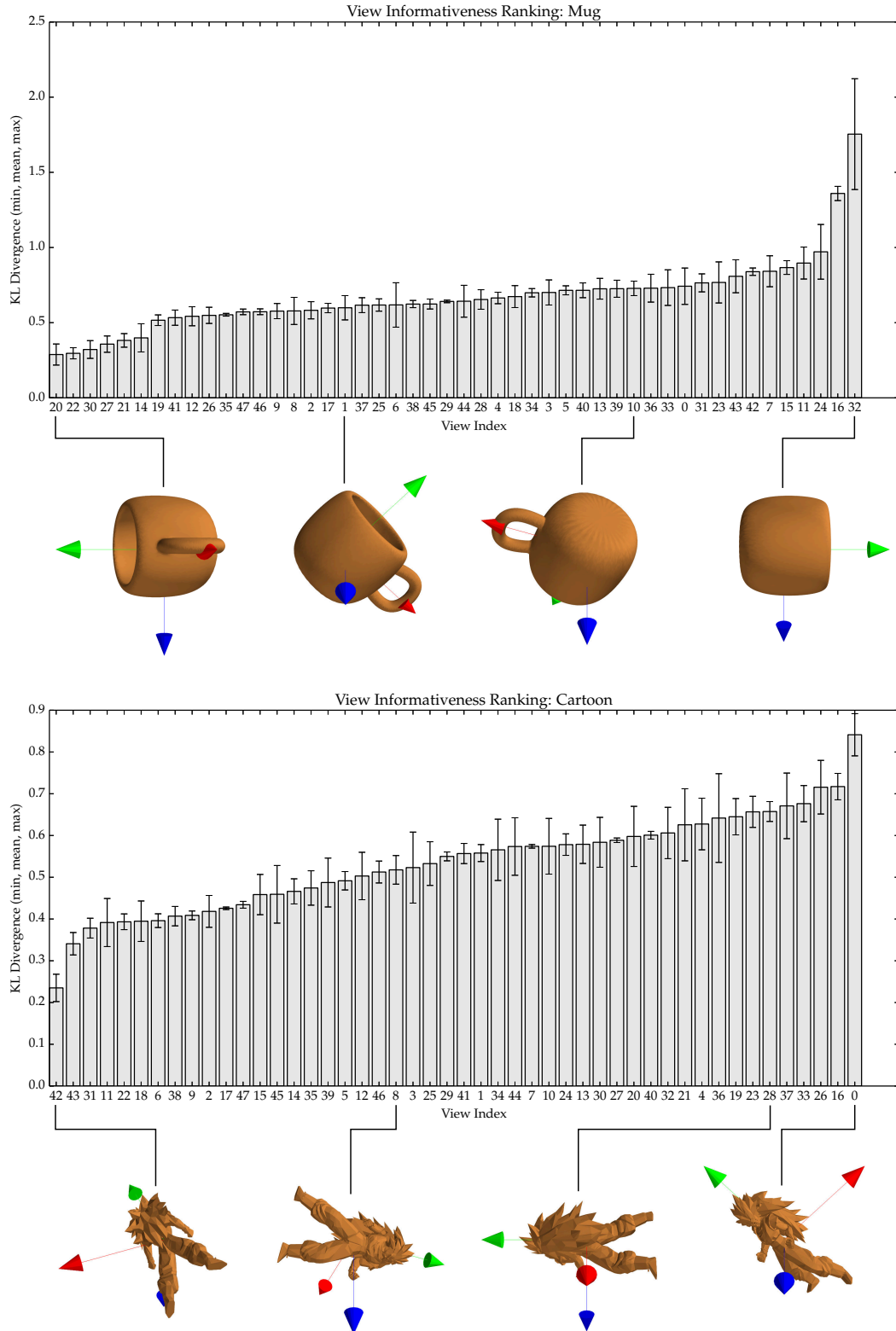


Figure 5.1.: Training directions ranked by average view KL divergence for mug and cartoon model. Below the bar charts, selected views are illustrated by renderings of the object from those views. Note, the y-axis limits are different for both plots showing.

conditional score of surface point i to viewing direction m . If a surface point is not observable from a direction and hence no nearest neighbors could be found, the score is set to -1 to indicate this. The scoring matrix K thus encodes the visibility and view-conditional informativeness of every surface point by assessing the average prediction correctness of features computed at these locations. A global, not view-conditional measure of a points informativeness is obtained by averaging the point's scores for all viewing directions it was observed from.

In figure 5.2 the view-conditional scores for a set of views - the same views as in figure 5.1 - are illustrated through a heatmap visualization. In other words, every pair of rendered and heatmap images visualizes a specific column of the score matrix K for that object. The colormap was chosen so that the color white corresponds to a KL divergence of zero and black corresponds to the median KL divergence of the complete scoring matrix (ignoring the -1 for non-visibility). This way, the same colormap is used for all shown views of an object and the heatmaps can be compared to each other. White color indicates, that features at this surface point are reliably recognized as originating from the viewing perspective shown. The views presented are ordered left to right by the overall view ranking extracted in the previous section and thus we clearly see which object parts make the most informative view (most left) better than the least informative view (most right). For the mug, our intuition that the handle is more informative than the body is now quantitatively proved. For the cartoon object, it seems that the concave regions within the character's hair as well as the rear part make the best view so informative.

Another interesting aspect is revealed when taking a closer look at the two left-most views of the mug. The upper handle part is colored white in both views which might seem contradictory at first as this means that features originating from the same physical region can be reliably classified to more than one view. This behavior can be explained by remembering that features are computed over geometry within a certain radius (here 3cm) and thus encode the view-specific self-occlusion of the object, which turns out to be very descriptive.

In figure 5.3, the global KL score is visualized by means of averaging a surface point's view-conditional score over all viewing directions. The colormap is scaled on a per object basis to show white for the lowest observed KL score and black for the highest observed score (first column for each object) or the median of the observed scores (second column of each object). This measure and the illustration show where distinctive features on the object's surface can be expected, independent of the viewing direction. For the mug model, again, regions on and around the handle are generally distinctive. For the cartoon model, a general observation is the higher average classification correctness of features and surface points. This is visible in the value range of the min-to-max colorbars and the feature distribution over that range as well as by comparing the total average KL divergence over all surface points. For the cartoon character the average KL divergence over all model points is 1.69 versus 2.15 for the mug model. Via the equation (5.2) this results in an average probabilistic weight for the correct viewing direction of 18.4% for the cartoon character versus 11.6% for the mug model. Regions of high informativeness for the cartoon character appear to be within the character's hair, the hands and the rear. Also, pointed surface regions like the feet and hair tips show high ambiguity or wrong classification which is probably due to unstable normal estimation in those areas.

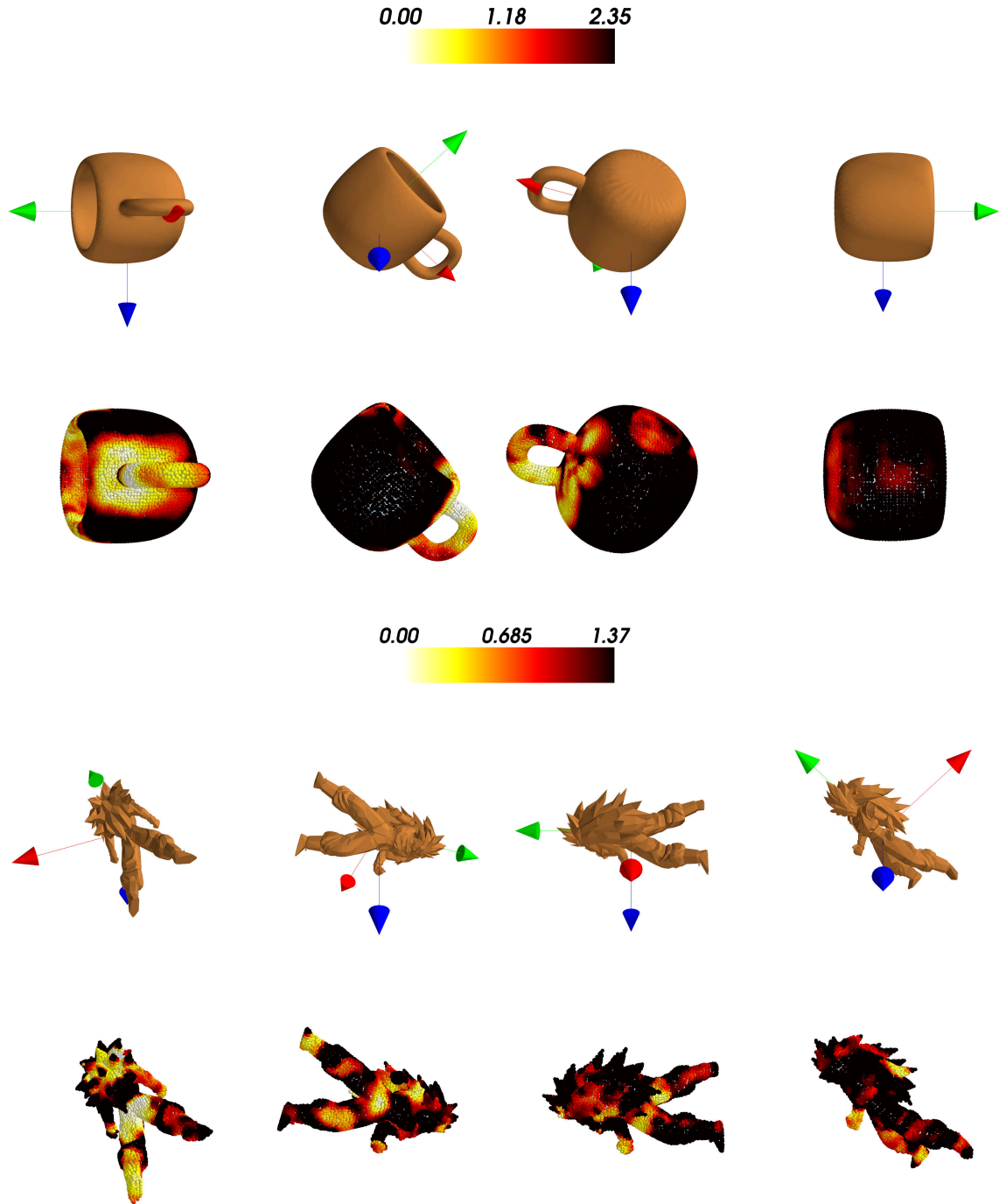


Figure 5.2.: View-conditional KL scores for chosen example viewing directions. For comparability, the same viewing directions as shown in figure 5.1 are shown. Corresponding image pairs show a plain rendering of the object obtained from the viewing direction and below a heatmap visualization of the view-conditional KL scores, with white representing a KL value of zero (low classification ambiguity) and black representing the median view-conditional KL value of the score matrix K .

5. Viewing Direction Classification: Application to View Planning

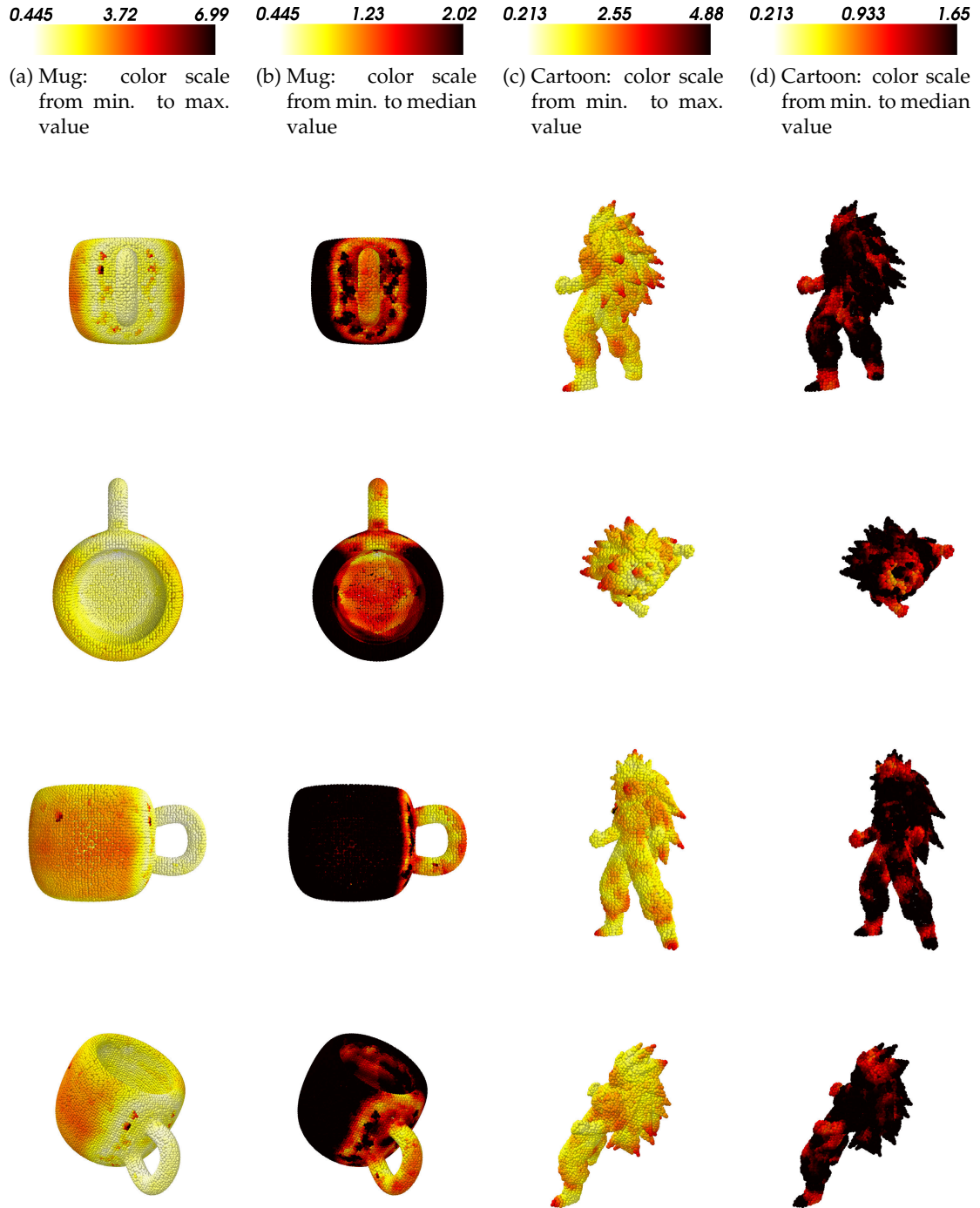


Figure 5.3.: Global KL score obtained by averaging view-conditional scores over all viewing directions for every surface point. For each object a view along x-axis, y-axis, z-axis and an isometric view are given. The lower the KL score for a point, the more informative are features computed from there.

5.3. Proof-of-Concept Evaluation of Informativeness Values

In order to show the practical value of the extracted model surface informativeness, a proof-of-concept experiment using the simulated mug model under varying degrees of occlusion was conducted. The visible part of the mug was manually chosen to consist of regions of high informativeness according to the findings in the previous section and figure 5.3. The overall experimental setup is identical to the evaluation in section 4.5 apart from the added occlusions. 20 random view sequences with 20 views per sequence have been generated and the development of the MAP rotational error after each view is analyzed.

To simulate the occlusion of parts of the mug which have been found to be uninformative (mainly the body of the mug), two points on the mug handle were selected manually, one on the upper side of the handle and one on the lower side, which together with a visibility radius r_{vis} around those points define the fixed observable region of the mug. The simulation pipeline proceeds by first generating a complete point cloud of the mug as seen from a given viewing direction and then selecting the sub-cloud within the distance r_{vis} around the two selected points as final simulation output. Normal estimation and feature computation is then done on the extracted sub-cloud. The experiment was performed six times with visibility radii in $r_{vis} \in [3\text{cm}, 4\text{cm}, 5\text{cm}, 6\text{cm}, 7\text{cm}, 8\text{cm}]$. The visible part of the mug for the different radii is illustrated in figure 5.4. For radii of 6cm and larger, the visible part includes surface regions on the rim and the inside of the mug, which allow a unique orientation estimate in contrast to radii smaller than 6cm.

For all experiments, the best parametrization in table 4.6 was used for the orientation estimation and classification pipeline (this means a feature estimation radius of 3cm). A selected subset of the resulting MAP error plots is given in figure 5.5. A summarizing comparison between all visibility radii is given in figure 5.6 by displaying the median MAP error over all 20 sequences for the different visibility settings. As observable in the median plots, the sequential estimation converges slower as the visible surface region gets smaller. For visibility radii of 8cm, 7cm and 6cm the error after convergence is comparable to the baseline experiment with no occlusion (visibility radius 'All'). Starting with a visibility radius of 5cm and smaller, the error of convergence gets significantly larger. For the 5cm setting, this is largely due to an ambiguity between the upright (mug opening in positive z-direction) and the flipped orientation (opening in negative z-direction) which arises because the handle is symmetric and the visible surface area does not include the rim and inner surface of the mug (cf. figure 5.4). This is clearly visible in the sequence plot for the 5cm setting in figure 5.5, especially in the EGI plot to the right where the flip in z-axis direction becomes apparent. Due the orientation representation as Bingham mixture model, an interesting question here is whether the flip-ambiguity is present in the orientation estimate as two separate mixture components. An investigation for the 5cm case revealed, that this is, however, not the case and random sequences either converge to a unimodal distribution with the mode close to the flipped or the non-flipped orientation.

Overall, the results presented show the robustness of the orientation estimation to occlusions of up to 81.9% respective $r_{vis} \geq 6\text{cm}$. This is achieved due to the local and correspondence-less nature of the viewing direction classification. It also shows that model surface informativeness ranking extracts surface areas relevant for orientation estimation as the increase in orientation error when occluding presumably uninformative parts is small. Therefore, it seems valuable to actively plan views in such a way that regions of

high estimated informativeness are visible.

5.4. Outlook: View Planning Approaches

Given the assessment of robustness towards occlusions of uninformative parts and the described assessment of view and surface point informativeness, the next step towards an active orientation estimation system is to derive and evaluate possible next-best-view planning approaches. This chapter introduces thoughts and ideas for next-best-view criteria, but no principled evaluation has been performed yet and thus no quantitative analysis can be presented. Experiences with some early experimental view planning implementations using simulated data are given where possible.

From the generic active state estimation model presented in chapter 2 we know that the pose (or orientation) measurement process has implications for the planning approach as it defines what data and how the data is processed into a measurement. For the presented approach to orientation estimation based on viewing direction classification, there are mainly three aspects which have to be considered in the planning approach:

- **rotational invariance of the orientation estimate with respect to the camera optical axis:** This is a principal drawback and complicates view planning in two ways. Firstly, this leads to high uncertainty about the object's orientation, especially during the first couple views. This makes planning more challenging as a large uncertainty for the object's current orientation also means large variety in the expected measurements for every possible next viewing direction. If the measurement prediction is not precise enough, this might lead to no significant difference for the expected information gain for different viewing directions and thus reduce the planned to a practically random view selection strategy. Especially for the second view in any sequence, the camera axis invariance introduces a limitation for the planning. Even if the first view of the object led to a perfect viewing direction recognition, the orientation invariance around the camera axis makes the second view unoptimizable. This becomes intuitively clear when one tries to decide for a direction in which to rotate around the object as no direction can be preferred due to the invariance. The only statement which can be made at this point is that the second view should be as orthogonal as possible to the first view in order to resolve the camera axis ambiguity. The second aspect comes into play when considering the use of precomputed statistics like the view or model surface informativeness. These rankings are based on the informativeness of the discrete distribution over viewing directions and hence do not include the additional camera axis invariance which will be part of the actual measurement.
- **local, individual processing of features and known regions of high and low informativeness:** The local processing of features without the need to match features and find correspondences makes view planning easier as one does not have to reason over the simultaneous visibility of a pair or a triplet of features. Further, the local ranking of model surface points with respect to their classification performance is a useful indicator of what surface parts should be observed and can be used to guide the planning while considering environment occlusions.

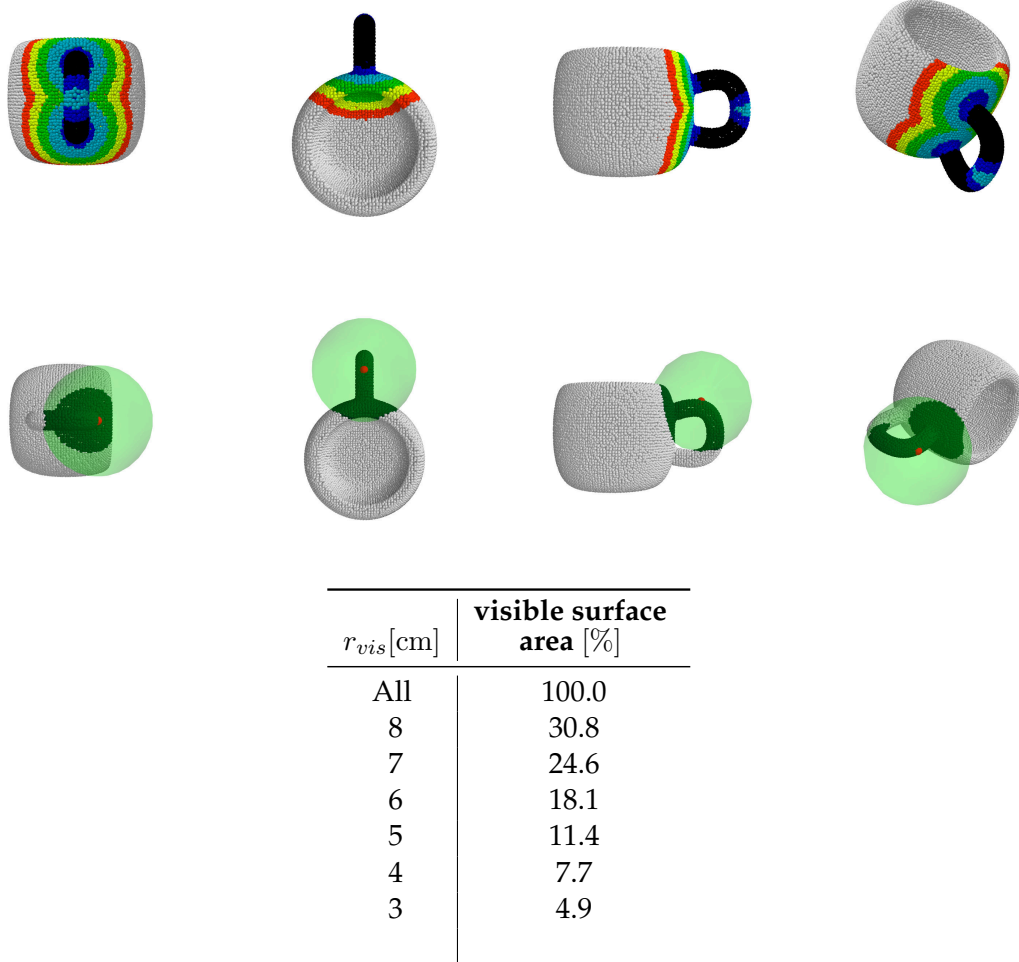


Figure 5.4.: Top row: mug surface visibility for different visibility radii $r_{vis} \in [3, 4, 5, 6, 7, 8]$ in centimeters. The 3cm radius is depicted as black region with the larger radii corresponding to blue, cyan, green, yellow and red regions with progressively more surface area. Bottom row: maximum support region for the feature descriptor computation. For an example surface point marked in red and the feature estimation radius of $r_f = 3$ cm used in all occlusion experiments, the maximum region of influence is illustrated via the green sphere centered at the point and by the coloring of the mug surface. Note that by the way FPFH features are computed, information from a radius up to $2r_f$ can influence the feature descriptor (cf. section 4.2.2 and reference [30]). The visualized sphere and the highlighted surface region is thus based on the maximum influence radius of 6cm.

5. Viewing Direction Classification: Application to View Planning

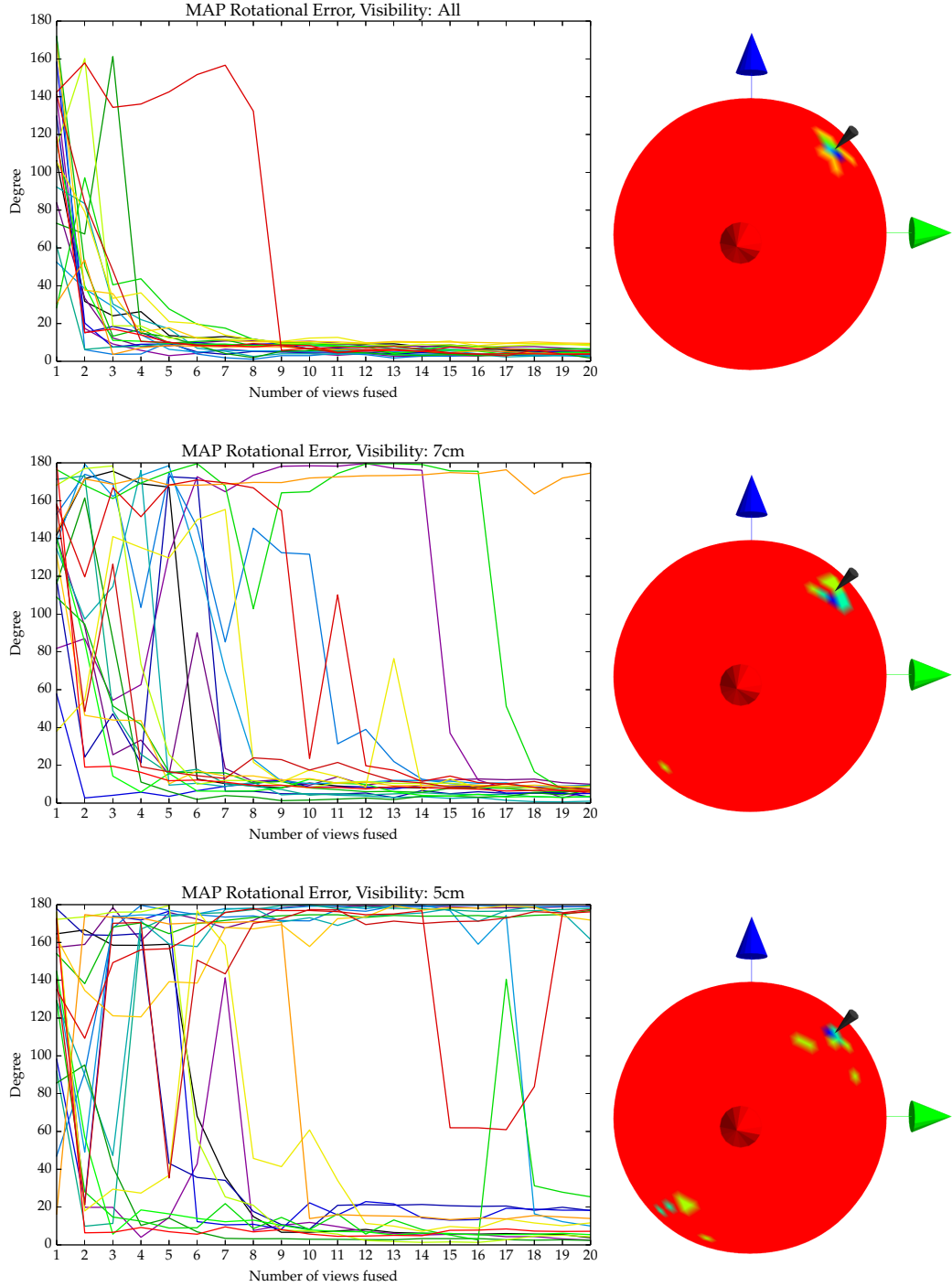


Figure 5.5.: Left: MAP error sequence plots for visibility radii $r_{vis} \in [+\infty, 7\text{cm}, 5\text{cm}]$. Right: EGI plots of the MAP rotations after the last view. For more explanations, see the text. The plots for all other tested visibility radii are given in the appendix in figures [B.1](#), [B.2](#) and [B.3](#).

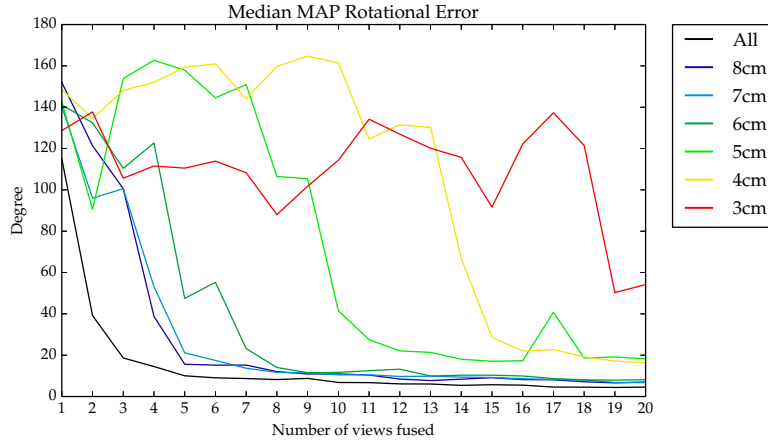


Figure 5.6.: Median MAP error over all sequences for all visibility settings.

- **varying robustness to deviations from training view directions:** This aspect is central to view planning in the sense that it makes the difference between convergence errors in the single digit range (use of non-training view directions) and errors smaller than one degree (use of training view directions). The difference in estimation performance for both settings was discussed in depth in the previous chapter and suggests that planning should try to establish training view directions.

Based on above insights, the subsequent paragraph will motivate ideas for dealing with these aspects and ways to achieve view planning:

- **Planning for training directions:** To plan a view from or close to training viewing directions, one needs to consider the current orientation estimate in some way. The simplest way to do so is to consider the current MAP orientation of the object and restrict the next view to be parallel to a training view direction for the MAP orientation. A more probabilistic way would be to sample the current object orientation distribution and derive viewing directions which minimize the angular distance to the training viewing directions observed via the sampled object orientations.
- **Planning for informative surface regions:** Informative views could be established by a voting scheme based on the precomputed statistics. One way would be to sample several object orientations from the current estimate and let informative regions vote for viewing directions they can be observed from. The voting space could be a view sphere around the object or a discretized 3d space whose voxels accumulate the vote weights. When votes originate from model points one could account for the visibility of this point's neighborhood (depending on the feature estimation radius) by checking for occluding geometry in the line of sight between the camera position being voted for and the model point. A more coarse voting could be based on the view statistics in section 5.1 rather than model points.
- **Planning for effective viewing directions considering the camera-axis invariance:** In addition to considering informative viewing directions, we also need to consider

the effect of the camera axis invariance inherent to a viewing direction. In order to account for this, the viewing axes of a planned view sequence should be chosen to be as orthogonal as possible to resolve this ambiguity. In the voting and scoring framework outlined in the previous point, this could be integrated for example by weighting a proposed viewing direction by the angular difference between it and the closest already visited viewing direction.

Early view planning experiments using the simulation environment were conducted using an informative surface voting as described in the middle bullet point above. These experiments did not lead to improvements over choosing camera positions at random and for some settings even deteriorated the estimation performance as views were not scattered around the view sphere as in the random setting but concentrated to similar viewing directions.

5.5. Summary

This chapter demonstrated how the viewing direction classification as part of the orientation estimation presented in chapter 4 can be used to extract view-related as well as model surface related informativeness values. Using surface areas which were estimated to be of high informativeness, the general robustness of the orientation estimation to simulated occlusions has been demonstrated in a proof-of-concept experiment for the mug model. The findings in this and the previous chapter justify further investigations into how to actively plan for viewing directions for which some directions of future work have been outlined.

6. Conclusion

The initial goal of this thesis was the investigation and development of an active multi-view pose estimation system. Throughout the literature review, two ingredients for such a system stood out to be of major interest and thus ended up being the main concern of the presented work: firstly, the probabilistic representation of 3d rotations and the sequential fusion of object rotation measurements; secondly, the nature and flexibility of in advance computed, object related knowledge with an application to online next-best-view planning.

To address the first issue, the Bingham distribution, a parametric probability density function over 3d rotations represented as unit quaternions, has been investigated. In a first experiment, its applicability to sequential fusion of orientation measurements has been evaluated by fusing orientation measurements of view sequences for three objects obtained by a block box pose estimation algorithm. Sequential fusion using Bingham mixture models was compared to a particle filter, a discrete Bayes filter (histogram filter) and a pose clustering approach on the same data. For the Bingham mixture fusion an approximate algebraic solution to the fusion problem via a mixture multiply & reduce (M+R) approach as well as a sequential Monte Carlo (SMC) approach has been implemented and evaluated with a simple two component Bingham mixture measurement model (one Gaussian-like component + one uniform 'outlier' component) and a more complex five component mixture model tailored towards the specific symmetries of one of the used objects. The evaluation showed that the sequential fusion using the simple two component measurement model with either SMC or M+R implementation achieves competitive results compared to the three comparison approaches. The more complex Bingham measurement model gives no advantage over the two-component model, but revealed that the SMC implementation is more suited for dealing with multi component and multi-modal measurement models.

For the second mentioned issue, a key point of interest and distinction with respect to other published approaches was to build a fine grained model of object surface informativeness related to the goal of orientation estimation. The underlying motivation was to precompute which object parts are essential and most informative for estimating an object's orientation. Based on this knowledge, one could derive view planning approaches to achieve accurate and fast converging orientation estimation. Due to the interplay between the orientation measurement process and the planning process, an orientation estimation based on a feature-to-viewing direction classification was developed. A classification pipeline is trained to predict the relative orientation (viewing direction) between the camera and a known object by means of independently processed and locally computed features. Due to the rotational invariance of the used features, this orientation can only be determined up to the rotation around the camera optical axis. This behavior is correctly modeled using a Bingham mixture distribution as probabilistic measurement model. In simulated as well as real experiments it was then shown that sequential orientation estimation using the proposed viewing direction classification is possible, but can only compare

to the accuracy of state-of-the-art object pose estimation algorithms when the relative orientation between object and camera is restricted to training view directions. In the general case with arbitrary deviations from training view directions, errors in the range of zero to nine degrees are possible even after fusion of ten or more views.

In a next investigation, the implicit model of feature ambiguity inherent to the classification pipeline was leveraged to build a model of viewing direction and object surface informativeness which is directly coupled to the presented orientation estimation. The experiments show plausible results for a geometrically simple mug object whose orientation relevant parts are determined to be on and around the handle. A simulated experiment in which only the informative part of the mug (manually selected area close to the handle) is observable and the rest is occluded shows that the orientation estimation is robust to occlusion and achieves orientation errors comparable to the baseline experiment with no occlusion. This highlights the relevance of the extracted regions of high informativeness.

A principled investigation of next-best-view criteria and the evaluation of a fully integrated active orientation estimation approach is at an early stage and no quantitative results could be presented on this matter. Exploring and evaluating different next-best-view criteria would thus be one direction of further research for which insights and suggestions were given in the outlook section 5.4.

A general disadvantage of the presented approach and therefore direction for possible future work is the orientation measurement invariance around the camera optical axis. This invariance, introduced by the rotationally invariant 3d features, complicates view planning by: 1) leading to high estimation uncertainties in the first couple views in general, 2) preventing effective planning for the second view in a sequence and 3) introducing a semantic gap between the actual measurement uncertainty of a viewing direction (includes rotational invariance) and the pre-computed informativeness measures (do not include rotational invariance). A direction for further research is therefore to experiment with different features and/or introducing unique local feature frames as in Salti et al. [32] to get rid of the uncertainty around the camera axis.

Another interesting direction of further research would be to make use of negative information, for example the absence of expected features, for the orientation estimation. Grundmann [17] implements such a system for their SIFT features based pose estimation by reasoning over measured/unmeasured and expected/unexpected features. To achieve this, the presented method would need to be extended to model the relationship between feature descriptors and their location on the surface of the object in addition to the current modeled relationship between features and the viewing directions they can be observed from. The work in [16] could be a starting point for further research in this direction as they propose an approach based on modeling the surface distribution of densely computed 3d features.

A. Complete Rankings for Rotation Fusion Evaluation

On the subsequent pages, the complete rankings for all parametrizations evaluated in section 3.7 are given.

Table A.1.: Complete ranking of all parametrizations for the fusion evaluation of the Gaussian measurement model in table 3.2

	avg. MAP [°]	max. MAP [°]	% < 1°	% < 3°	% < 5°	Fusion	# max. comps.	Meas.- model	α	κ
1	.57°	1.18°	.90	1.00	1.00	M+R	1	gaussian	[.95, .05]	-60
2	.58°	1.16°	.90	1.00	1.00	SMC		gaussian	[.90, .10]	-27
3	.59°	1.03°	.97	1.00	1.00	SMC		gaussian	[.95, .05]	-60
4	.60°	1.42°	.90	1.00	1.00	M+R	1	gaussian	[.90, .10]	-60
5	.61°	1.21°	.90	1.00	1.00	M+R	1	gaussian	[.95, .05]	-27
6	.62°	1.30°	.87	1.00	1.00	M+R	1	gaussian	[.90, .10]	-27
7	.62°	1.15°	.90	1.00	1.00	M+R	1	gaussian	[.99, .01]	-60
8	.62°	1.31°	.90	1.00	1.00	M+R	5	gaussian	[.95, .05]	-60
9	.62°	1.31°	.93	1.00	1.00	M+R	1	gaussian	[.99, .01]	-27
10	.63°	1.36°	.90	1.00	1.00	M+R	10	gaussian	[.99, .01]	-27
11	.63°	1.37°	.90	1.00	1.00	M+R	5	gaussian	[.90, .10]	-60
12	.63°	1.21°	.90	1.00	1.00	M+R	5	gaussian	[.99, .01]	-60
13	.63°	1.22°	.90	1.00	1.00	M+R	10	gaussian	[.99, .01]	-60
14	.63°	1.14°	.93	1.00	1.00	SMC		gaussian	[.99, .01]	-60
15	.64°	1.26°	.93	1.00	1.00	SMC		gaussian	[.90, .10]	-60
16	.64°	1.28°	.90	1.00	1.00	M+R	10	gaussian	[.90, .10]	-60
17	.64°	1.33°	.87	1.00	1.00	M+R	10	gaussian	[.90, .10]	-27
18	.64°	1.53°	.93	1.00	1.00	M+R	5	gaussian	[.90, .10]	-27
19	.64°	1.43°	.90	1.00	1.00	M+R	5	gaussian	[.99, .01]	-27
20	.65°	1.25°	.80	1.00	1.00	M+R	10	gaussian	[.95, .05]	-27
21	.65°	1.26°	.93	1.00	1.00	SMC		gaussian	[.95, .05]	-27
22	.66°	1.42°	.87	1.00	1.00	M+R	5	gaussian	[.95, .05]	-27
23	.66°	1.42°	.77	1.00	1.00	SMC		gaussian	[.95, .05]	-120
24	.67°	1.19°	.90	1.00	1.00	M+R	10	gaussian	[.95, .05]	-60
25	.69°	1.55°	.77	1.00	1.00	M+R	1	gaussian	[.99, .01]	-120
26	.69°	1.40°	.73	1.00	1.00	M+R	1	gaussian	[.95, .05]	-120

Continued on next page

27	.72°	1.42°	.77	1.00	1.00	SMC		gaussian	[.99, .01]	-120
28	.72°	1.35°	.73	1.00	1.00	M+R	10	gaussian	[.90, .10]	-120
29	.72°	1.30°	.77	1.00	1.00	M+R	5	gaussian	[.90, .10]	-120
30	.74°	1.35°	.73	1.00	1.00	M+R	10	gaussian	[.95, .05]	-120
31	.76°	1.42°	.73	1.00	1.00	M+R	5	gaussian	[.95, .05]	-120
32	.76°	1.52°	.73	1.00	1.00	M+R	5	gaussian	[.99, .01]	-120
33	.78°	1.51°	.73	1.00	1.00	M+R	10	gaussian	[.99, .01]	-120
34	.78°	1.67°	.63	1.00	1.00	M+R	1	gaussian	[.90, .10]	-120
35	.97°	2.43°	.57	1.00	1.00	SMC		gaussian	[.95, .05]	-240
36	.98°	2.33°	.60	1.00	1.00	SMC		gaussian	[.90, .10]	-240
37	1.01°	2.35°	.60	1.00	1.00	SMC		gaussian	[.99, .01]	-240
38	1.19°	2.97°	.50	1.00	1.00	M+R	10	gaussian	[.90, .10]	-240
39	1.21°	3.13°	.50	.97	1.00	M+R	5	gaussian	[.90, .10]	-240
40	1.26°	3.51°	.50	.93	1.00	M+R	10	gaussian	[.95, .05]	-240
41	1.30°	3.90°	.43	.97	1.00	M+R	5	gaussian	[.95, .05]	-240
42	1.40°	4.75°	.47	.90	1.00	M+R	5	gaussian	[.99, .01]	-240
43	1.41°	4.61°	.43	.93	1.00	M+R	10	gaussian	[.99, .01]	-240
44	1.43°	3.50°	.50	.97	1.00	SMC		gaussian	[.99, .01]	-480
45	1.43°	3.48°	.50	.97	1.00	SMC		gaussian	[.90, .10]	-480
46	1.44°	3.59°	.50	.97	1.00	SMC		gaussian	[.95, .05]	-480
47	1.48°	5.62°	.40	.93	.97	M+R	1	gaussian	[.99, .01]	-240
48	1.63°	6.69°	.43	.90	.97	M+R	1	gaussian	[.95, .05]	-240
49	1.80°	4.67°	.30	.90	1.00	SMC		gaussian	[.90, .10]	-900
50	1.81°	4.71°	.33	.90	1.00	SMC		gaussian	[.95, .05]	-900
51	1.83°	4.72°	.33	.87	1.00	SMC		gaussian	[.99, .01]	-900
52	2.61°	27.62°	.43	.83	.90	M+R	1	gaussian	[.90, .10]	-240
53	10.05°	142.07°	.63	.90	.93	SMC		gaussian	[.90, .10]	-120
54	22.52°	179.91°	.77	.87	.87	SMC		gaussian	[.99, .01]	-27
55	29.03°	178.97°	.33	.70	.80	M+R	10	gaussian	[.90, .10]	-480
56	29.22°	178.55°	.37	.73	.80	M+R	5	gaussian	[.90, .10]	-480

Continued on next page

Continued from previous page									
57	29.87°	178.47°	.30	.73	.80	M+R	1	gaussian	[.90, .10] -480
58	30.15°	178.40°	.30	.73	.80	M+R	1	gaussian	[.95, .05] -480
59	32.80°	178.28°	.37	.73	.77	M+R	1	gaussian	[.99, .01] -480
60	34.24°	179.60°	.20	.63	.73	M+R	5	gaussian	[.90, .10] -900
61	34.46°	178.21°	.17	.63	.80	M+R	1	gaussian	[.95, .05] -900
62	35.03°	178.78°	.17	.63	.73	M+R	1	gaussian	[.99, .01] -900
63	35.44°	177.94°	.17	.67	.80	M+R	1	gaussian	[.90, .10] -900
64	38.42°	178.84°	.33	.70	.73	M+R	5	gaussian	[.95, .05] -480
65	38.83°	179.81°	.33	.70	.73	M+R	10	gaussian	[.99, .01] -480
66	38.88°	179.66°	.30	.70	.73	M+R	5	gaussian	[.99, .01] -480
67	38.99°	179.16°	.33	.70	.73	M+R	10	gaussian	[.95, .05] -480
68	39.40°	179.94°	.23	.63	.73	M+R	10	gaussian	[.90, .10] -900
69	42.44°	179.97°	.20	.60	.70	M+R	10	gaussian	[.95, .05] -900
70	42.55°	179.96°	.17	.60	.70	M+R	5	gaussian	[.95, .05] -900
71	42.69°	179.99°	.17	.60	.70	M+R	10	gaussian	[.99, .01] -900
72	42.77°	179.87°	.17	.60	.70	M+R	5	gaussian	[.99, .01] -900
Concluded									

Table A.2.: Complete ranking of all parametrizations for the multimodal measurement model in table 3.4

	avg. MAP [°]	max. MAP [°]	% < 1°	% < 3°	% < 5°	Fusion	# max. comps.	Meas.- model	α	k
1	.41°	.56°	1.00	1.00	1.00	SMC		gaussian	[.95, .05]	-120
2	.46°	.88°	1.00	1.00	1.00	SMC		gaussian	[.95, .05]	-27
3	.46°	.63°	1.00	1.00	1.00	SMC		gaussian	[.90, .10]	-60
4	.50°	.69°	1.00	1.00	1.00	SMC		gaussian	[.99, .01]	-27
5	.51°	.78°	1.00	1.00	1.00	SMC		gaussian	[.95, .05]	-60
6	.51°	.72°	1.00	1.00	1.00	SMC		gaussian	[.90, .10]	-27
7	.53°	.67°	1.00	1.00	1.00	SMC		gaussian	[.99, .01]	-120
8	.54°	.67°	1.00	1.00	1.00	SMC		gaussian	[.99, .01]	-60
9	.57°	1.28°	.90	1.00	1.00	M+R	1	gaussian	[.90, .10]	-27
10	.57°	.65°	1.00	1.00	1.00	SMC		gaussian	[.90, .10]	-120
11	.61°	1.05°	.80	1.00	1.00	SMC		gaussian	[.95, .05]	-240
12	.62°	.99°	1.00	1.00	1.00	M+R	1	gaussian	[.95, .05]	-27
13	.65°	1.19°	.90	1.00	1.00	M+R	5	gaussian	[.90, .10]	-60
14	.66°	1.06°	.90	1.00	1.00	M+R	1	gaussian	[.99, .01]	-60
15	.66°	1.05°	.90	1.00	1.00	M+R	1	gaussian	[.95, .05]	-60
16	.66°	1.22°	.90	1.00	1.00	M+R	10	4fold	[.85, .02, .02, .02, .10]	-60
17	.66°	1.04°	.80	1.00	1.00	M+R	5	4fold	[.85, .02, .02, .02, .10]	-60
18	.66°	1.03°	.90	1.00	1.00	M+R	10	gaussian	[.99, .01]	-60
19	.67°	1.26°	.90	1.00	1.00	M+R	1	gaussian	[.95, .05]	-120
20	.67°	1.33°	.90	1.00	1.00	M+R	1	gaussian	[.99, .01]	-27
21	.67°	.97°	1.00	1.00	1.00	M+R	10	4fold	[.85, .02, .02, .02, .10]	-27
22	.69°	1.25°	.90	1.00	1.00	M+R	5	gaussian	[.95, .05]	-120
23	.69°	1.14°	.70	1.00	1.00	SMC		gaussian	[.99, .01]	-240
24	.69°	.97°	1.00	1.00	1.00	M+R	10	gaussian	[.95, .05]	-60
25	.69°	1.24°	.90	1.00	1.00	M+R	1	gaussian	[.99, .01]	-120
26	.69°	1.15°	.90	1.00	1.00	M+R	10	gaussian	[.90, .10]	-60

Continued on next page

A. Complete Rankings for Rotation Fusion Evaluation

27	.69°	1.20°	.90	1.00	1.00	M+R	5	gaussian	[.90, .10]	-120
28	.74°	1.21°	.70	1.00	1.00	M+R	1	gaussian	[.90, .10]	-60
29	.74°	1.14°	.70	1.00	1.00	M+R	5	gaussian	[.90, .10]	-27
30	.74°	1.31°	.90	1.00	1.00	M+R	5	gaussian	[.99, .01]	-27
31	.74°	1.24°	.70	1.00	1.00	SMC		gaussian	[.90, .10]	-240
32	.74°	1.17°	.90	1.00	1.00	M+R	5	gaussian	[.95, .05]	-27
33	.74°	1.40°	.90	1.00	1.00	M+R	10	gaussian	[.95, .05]	-120
34	.75°	1.29°	.90	1.00	1.00	M+R	5	gaussian	[.95, .05]	-60
35	.75°	1.29°	.80	1.00	1.00	M+R	5	gaussian	[.99, .01]	-60
36	.75°	1.55°	.90	1.00	1.00	M+R	10	gaussian	[.99, .01]	-120
37	.76°	1.40°	.80	1.00	1.00	M+R	10	gaussian	[.90, .10]	-120
38	.77°	1.41°	.90	1.00	1.00	M+R	5	gaussian	[.99, .01]	-120
39	.78°	1.35°	.80	1.00	1.00	M+R	10	gaussian	[.99, .01]	-27
40	.78°	1.16°	.80	1.00	1.00	M+R	10	gaussian	[.95, .05]	-27
41	.79°	1.79°	.90	1.00	1.00	M+R	10	4fold	[.85, .02, .02, .02, .10]	-120
42	.80°	1.64°	.80	1.00	1.00	M+R	5	4fold	[.85, .02, .02, .02, .10]	-120
43	.80°	1.21°	.80	1.00	1.00	M+R	10	gaussian	[.90, .10]	-27
44	.83°	2.19°	.90	1.00	1.00	M+R	5	4fold	[.85, .02, .02, .02, .10]	-27
45	.92°	1.60°	.60	1.00	1.00	M+R	1	gaussian	[.90, .10]	-120
46	.96°	1.17°	.50	1.00	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-27
47	1.02°	1.91°	.70	1.00	1.00	SMC		gaussian	[.99, .01]	-480
48	1.04°	1.88°	.60	1.00	1.00	SMC		gaussian	[.95, .05]	-480
49	1.08°	1.85°	.60	1.00	1.00	SMC		gaussian	[.90, .10]	-480
50	1.14°	1.37°	.30	1.00	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-27
51	1.15°	1.39°	.20	1.00	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-27
52	1.31°	2.55°	.60	1.00	1.00	SMC		gaussian	[.95, .05]	-900
53	1.31°	2.47°	.50	1.00	1.00	SMC		gaussian	[.99, .01]	-900
54	1.32°	1.67°	.10	1.00	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-60
55	1.33°	2.55°	.50	1.00	1.00	SMC		gaussian	[.90, .10]	-900
56	1.35°	2.76°	.60	1.00	1.00	M+R	10	gaussian	[.90, .10]	-240

Continued on next page

Continued from previous page

57	1.35°	1.54°	.10	1.00	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-60
58	1.39°	1.60°	.10	1.00	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-60
59	1.44°	2.93°	.50	1.00	1.00	M+R	5	gaussian	[.90, .10]	-240
60	1.47°	3.59°	.50	.90	1.00	M+R	10	gaussian	[.95, .05]	-240
61	1.51°	2.04°	.10	1.00	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-120
62	1.55°	2.11°	.10	1.00	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-120
63	1.56°	1.95°	.10	1.00	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-120
64	1.59°	3.65°	.50	.90	1.00	M+R	5	gaussian	[.95, .05]	-240
65	1.84°	4.84°	.50	.80	1.00	M+R	10	gaussian	[.99, .01]	-240
66	1.84°	2.43°	.10	1.00	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-240
67	1.87°	2.46°	.10	1.00	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-240
68	1.87°	2.42°	.10	1.00	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-240
69	1.87°	4.53°	.40	.70	1.00	M+R	5	gaussian	[.99, .01]	-240
70	2.07°	5.73°	.30	.80	.90	M+R	1	gaussian	[.99, .01]	-240
71	2.11°	3.05°	.10	.90	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-480
72	2.15°	3.01°	.10	.90	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-480
73	2.15°	2.99°	.10	1.00	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-480
74	2.31°	8.46°	.50	.70	.90	M+R	10	4fold	[.85, .02, .02, .02, .10]	-240
75	2.37°	3.62°	.20	.70	1.00	SMC		4fold	[.85, .02, .02, .02, .10]	-900
76	2.39°	3.57°	.20	.60	1.00	SMC		4fold	[.65, .08, .08, .08, .10]	-900
77	2.47°	3.80°	.20	.60	1.00	SMC		4fold	[.45, .15, .15, .15, .10]	-900
78	2.65°	6.60°	.20	.70	.90	M+R	1	gaussian	[.95, .05]	-240
79	2.73°	8.48°	.30	.70	.80	M+R	5	4fold	[.85, .02, .02, .02, .10]	-240
80	2.73°	5.18°	.00	.70	.90	SMC		4fold	[.25, .22, .22, .22, .10]	-27
81	2.79°	4.26°	.00	.50	1.00	SMC		4fold	[.25, .22, .22, .22, .10]	-120
82	3.26°	4.73°	.10	.40	1.00	SMC		4fold	[.25, .22, .22, .22, .10]	-240
83	3.39°	5.18°	.10	.30	.90	SMC		4fold	[.25, .22, .22, .22, .10]	-60
84	4.58°	9.31°	.00	.30	.60	SMC		4fold	[.25, .22, .22, .22, .10]	-480
85	5.39°	10.02°	.00	.20	.60	SMC		4fold	[.25, .22, .22, .22, .10]	-900
86	5.73°	28.25°	.20	.50	.70	M+R	1	gaussian	[.90, .10]	-240

Continued on next page

A. Complete Rankings for Rotation Fusion Evaluation

87	23.28°	97.04°	.00	.20	.30	M+R	5	4fold	[.25, .22, .22, .22, .10]	-120
88	36.41°	179.65°	.70	.80	.80	M+R	10	4fold	[.65, .08, .08, .08, .10]	-27
89	54.90°	179.73°	.20	.50	.70	M+R	10	4fold	[.65, .08, .08, .08, .10]	-60
90	56.54°	179.99°	.00	.40	.50	M+R	5	4fold	[.65, .08, .08, .08, .10]	-27
91	61.41°	173.63°	.00	.30	.40	M+R	5	4fold	[.25, .22, .22, .22, .10]	-60
92	71.90°	179.33°	.20	.40	.50	M+R	10	4fold	[.65, .08, .08, .08, .10]	-240
93	72.34°	179.22°	.30	.40	.60	M+R	10	4fold	[.65, .08, .08, .08, .10]	-120
94	72.78°	176.63°	.00	.10	.20	M+R	5	4fold	[.25, .22, .22, .22, .10]	-27
95	84.04°	179.07°	.20	.30	.40	M+R	10	gaussian	[.90, .10]	-480
96	84.55°	178.60°	.20	.30	.40	M+R	5	gaussian	[.90, .10]	-480
97	86.50°	178.63°	.10	.30	.40	M+R	1	gaussian	[.90, .10]	-480
98	87.35°	100.21°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-120
99	87.52°	178.97°	.10	.30	.40	M+R	1	gaussian	[.95, .05]	-480
100	87.62°	178.34°	.10	.30	.30	M+R	5	4fold	[.65, .08, .08, .08, .10]	-60
101	87.91°	105.22°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-900
102	88.02°	133.13°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-60
103	88.49°	106.99°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-480
104	90.15°	179.71°	.10	.30	.50	M+R	10	4fold	[.45, .15, .15, .15, .10]	-240
105	90.33°	179.41°	.10	.30	.30	M+R	10	4fold	[.45, .15, .15, .15, .10]	-60
106	90.38°	179.33°	.00	.10	.20	M+R	10	4fold	[.25, .22, .22, .22, .10]	-900
107	90.76°	179.65°	.10	.20	.30	M+R	5	4fold	[.65, .08, .08, .08, .10]	-480
108	93.37°	130.55°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-27
109	94.57°	179.43°	.00	.20	.20	M+R	5	4fold	[.25, .22, .22, .22, .10]	-900
110	95.07°	178.32°	.10	.30	.30	M+R	1	gaussian	[.99, .01]	-480
111	95.65°	157.79°	.00	.00	.00	M+R	1	4fold	[.25, .22, .22, .22, .10]	-240
112	95.77°	178.68°	.00	.10	.10	M+R	10	4fold	[.45, .15, .15, .15, .10]	-27
113	98.32°	179.46°	.20	.30	.30	M+R	10	4fold	[.85, .02, .02, .02, .10]	-900
114	98.85°	179.76°	.20	.20	.20	M+R	5	gaussian	[.90, .10]	-900
115	99.47°	178.85°	.10	.20	.40	M+R	1	gaussian	[.95, .05]	-900
116	99.50°	155.48°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-27

Continued on next page

Continued from previous page

117	99.79°	139.26°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-900
118	100.02°	143.41°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-240
119	100.10°	138.78°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-480
120	100.82°	178.37°	.10	.20	.20	M+R	1	gaussian	[.99, .01]	-900
121	101.98°	179.82°	.10	.10	.20	M+R	10	4fold	[.25, .22, .22, .22, .10]	-480
122	103.11°	179.53°	.10	.30	.40	M+R	1	gaussian	[.90, .10]	-900
123	103.86°	159.14°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-120
124	104.75°	158.92°	.00	.00	.00	M+R	1	4fold	[.45, .15, .15, .15, .10]	-60
125	105.59°	169.13°	.00	.00	.00	M+R	1	4fold	[.65, .08, .08, .08, .10]	-240
126	105.68°	171.09°	.00	.00	.00	M+R	1	4fold	[.65, .08, .08, .08, .10]	-120
127	105.99°	169.32°	.00	.00	.00	M+R	1	4fold	[.65, .08, .08, .08, .10]	-900
128	106.09°	170.79°	.00	.00	.00	M+R	1	4fold	[.65, .08, .08, .08, .10]	-480
129	106.10°	176.56°	.10	.10	.10	M+R	1	4fold	[.65, .08, .08, .08, .10]	-60
130	106.70°	179.78°	.00	.30	.30	M+R	5	4fold	[.65, .08, .08, .08, .10]	-240
131	107.05°	179.88°	.10	.20	.40	M+R	5	4fold	[.45, .15, .15, .15, .10]	-900
132	107.25°	178.62°	.00	.00	.00	M+R	1	4fold	[.65, .08, .08, .08, .10]	-27
133	107.33°	179.55°	.20	.20	.40	M+R	5	4fold	[.65, .08, .08, .08, .10]	-900
134	107.33°	178.15°	.20	.20	.20	M+R	1	4fold	[.85, .02, .02, .02, .10]	-120
135	107.41°	178.67°	.10	.30	.30	M+R	1	4fold	[.85, .02, .02, .02, .10]	-900
136	107.41°	179.32°	.00	.20	.20	M+R	5	4fold	[.45, .15, .15, .15, .10]	-60
137	107.46°	179.22°	.10	.20	.30	M+R	1	4fold	[.85, .02, .02, .02, .10]	-240
138	107.49°	179.46°	.10	.40	.40	M+R	5	4fold	[.45, .15, .15, .15, .10]	-240
139	107.50°	179.09°	.00	.20	.20	M+R	5	4fold	[.25, .22, .22, .22, .10]	-480
140	107.55°	179.20°	.10	.20	.20	M+R	1	4fold	[.85, .02, .02, .02, .10]	-27
141	107.58°	178.87°	.10	.20	.30	M+R	5	4fold	[.45, .15, .15, .15, .10]	-120
142	107.60°	178.24°	.00	.30	.40	M+R	10	4fold	[.45, .15, .15, .15, .10]	-480
143	107.68°	179.46°	.00	.30	.40	M+R	10	4fold	[.45, .15, .15, .15, .10]	-900
144	107.69°	179.75°	.10	.30	.40	M+R	5	4fold	[.45, .15, .15, .15, .10]	-480
145	107.71°	179.70°	.10	.20	.40	M+R	1	4fold	[.85, .02, .02, .02, .10]	-480
146	107.71°	179.47°	.10	.20	.20	M+R	1	4fold	[.85, .02, .02, .02, .10]	-60

Continued on next page

A. Complete Rankings for Rotation Fusion Evaluation

Continued from previous page											Concluded
147	107.99°	179.86°	.00	.20	.30	M+R	5	4fold	[.45, .15, .15, .15, .10]	-27	
148	108.08°	179.78°	.10	.20	.40	M+R	10	4fold	[.45, .15, .15, .15, .10]	-120	
149	108.15°	179.59°	.00	.20	.20	M+R	5	4fold	[.65, .08, .08, .08, .10]	-120	
150	108.16°	179.50°	.10	.30	.30	M+R	10	4fold	[.65, .08, .08, .08, .10]	-480	
151	108.37°	179.99°	.20	.20	.30	M+R	10	4fold	[.65, .08, .08, .08, .10]	-900	
152	110.40°	179.69°	.10	.20	.20	M+R	5	4fold	[.85, .02, .02, .02, .10]	-480	
153	112.15°	179.03°	.20	.20	.20	M+R	5	gaussian	[.95, .05]	-480	
154	113.26°	179.73°	.20	.20	.20	M+R	10	gaussian	[.99, .01]	-480	
155	113.31°	179.64°	.20	.20	.20	M+R	5	gaussian	[.99, .01]	-480	
156	113.84°	179.38°	.20	.20	.20	M+R	10	gaussian	[.95, .05]	-480	
157	114.15°	179.88°	.20	.20	.20	M+R	10	gaussian	[.90, .10]	-900	
158	117.20°	178.91°	.00	.10	.20	M+R	10	4fold	[.25, .22, .22, .22, .10]	-240	
159	118.21°	179.13°	.20	.20	.20	M+R	10	4fold	[.85, .02, .02, .02, .10]	-480	
160	123.28°	179.95°	.10	.10	.10	M+R	10	gaussian	[.95, .05]	-900	
161	123.44°	179.97°	.10	.10	.10	M+R	5	gaussian	[.95, .05]	-900	
162	123.96°	179.93°	.10	.10	.10	M+R	10	gaussian	[.99, .01]	-900	
163	124.39°	179.99°	.10	.10	.10	M+R	5	gaussian	[.99, .01]	-900	
164	131.39°	179.93°	.10	.10	.10	M+R	5	4fold	[.85, .02, .02, .02, .10]	-900	
165	133.87°	178.91°	.00	.00	.00	M+R	5	4fold	[.25, .22, .22, .22, .10]	-240	
166	141.16°	178.87°	.00	.00	.00	M+R	10	4fold	[.25, .22, .22, .22, .10]	-27	
167	142.09°	179.87°	.00	.10	.10	M+R	10	4fold	[.25, .22, .22, .22, .10]	-60	
168	143.02°	179.93°	.00	.10	.10	M+R	10	4fold	[.25, .22, .22, .22, .10]	-120	
											Concluded

B. All Sequence Plots for Occlusion Experiment

On the subsequent pages, the complete sequence plots for the proof-of-concept evaluation in section 5.3 are given.

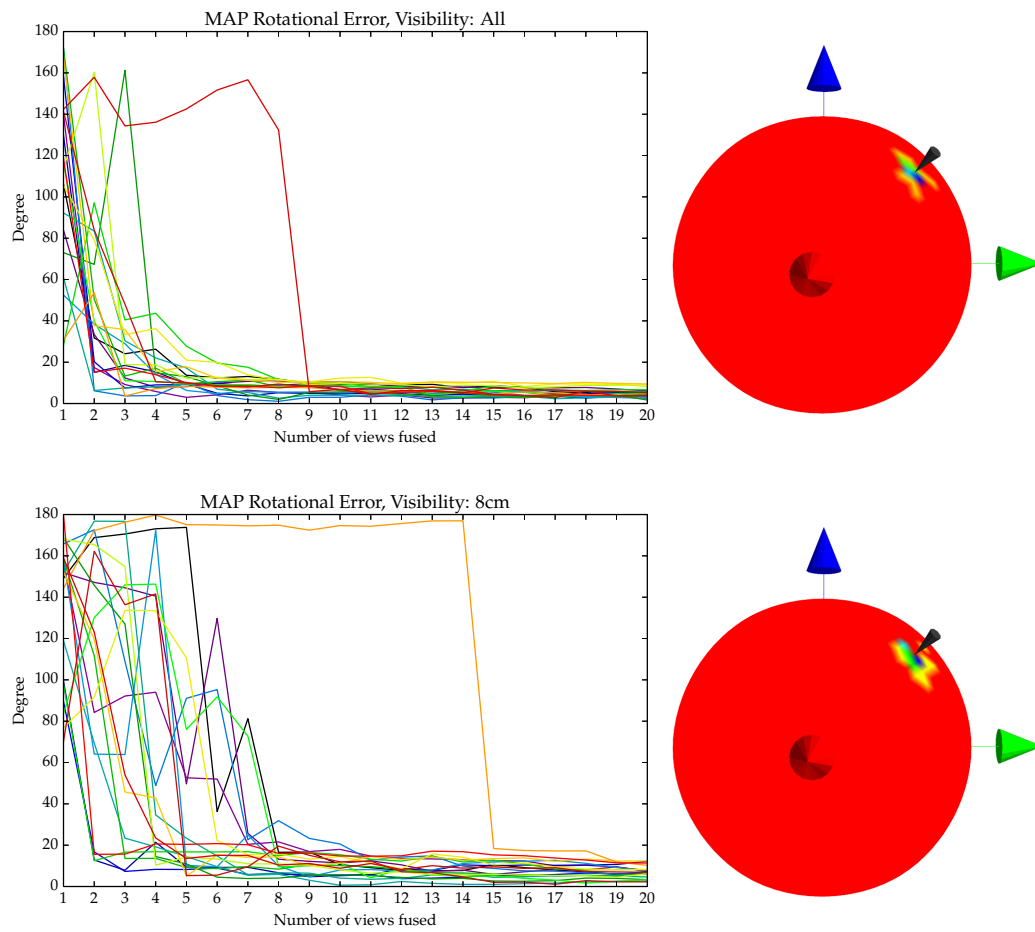


Figure B.1.: Left: MAP error sequence plots for visibility radii $r_{vis} \in [+\infty, 8\text{cm}]$. Right: EGI plots of the MAP rotations after the last view.

B. All Sequence Plots for Occlusion Experiment

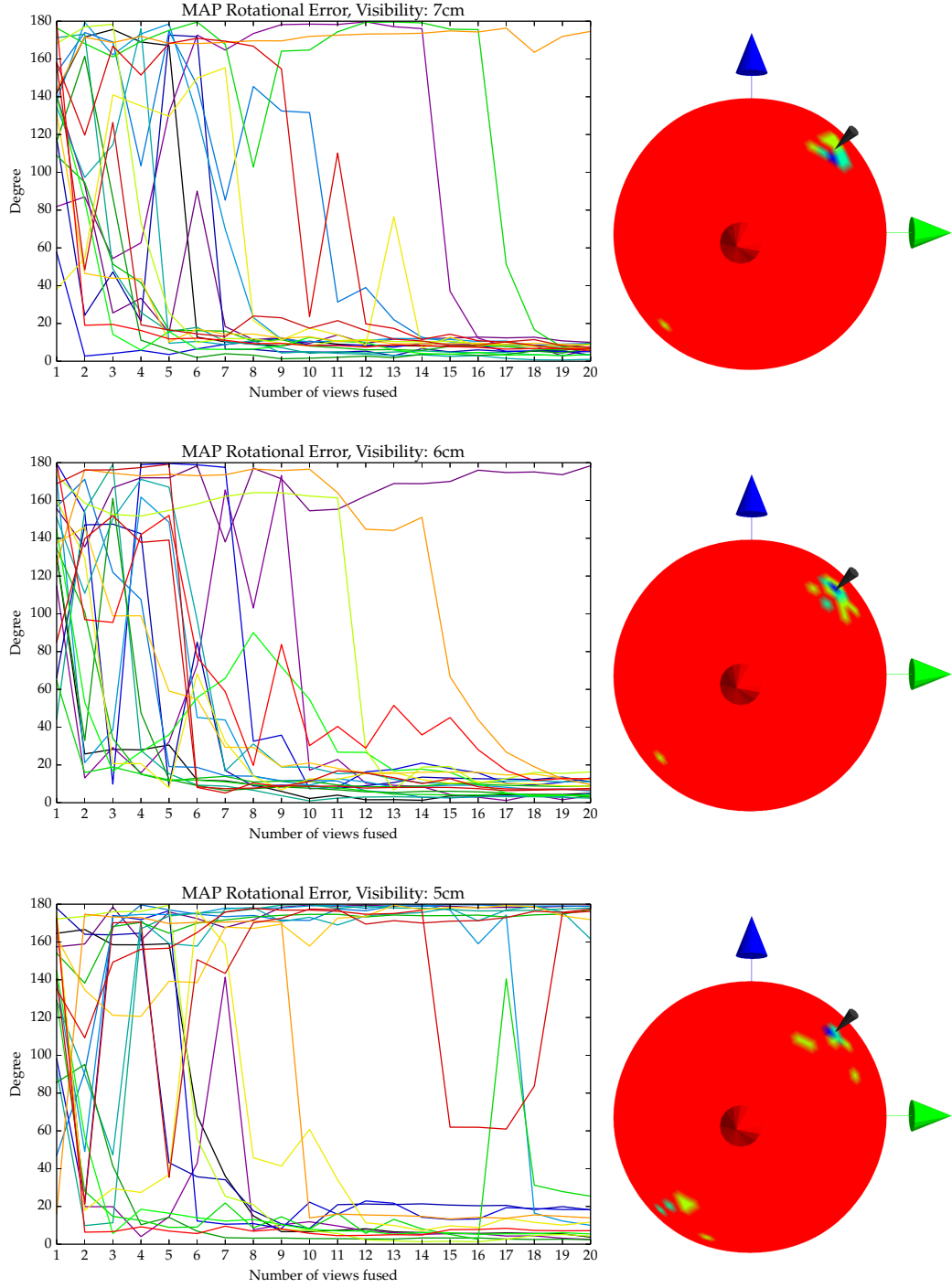


Figure B.2.: Left: MAP error sequence plots for visibility radii $r_{vis} \in [7\text{cm}, 6\text{cm}, 5\text{cm}]$. Right: EGI plots of the MAP rotations after the last view.

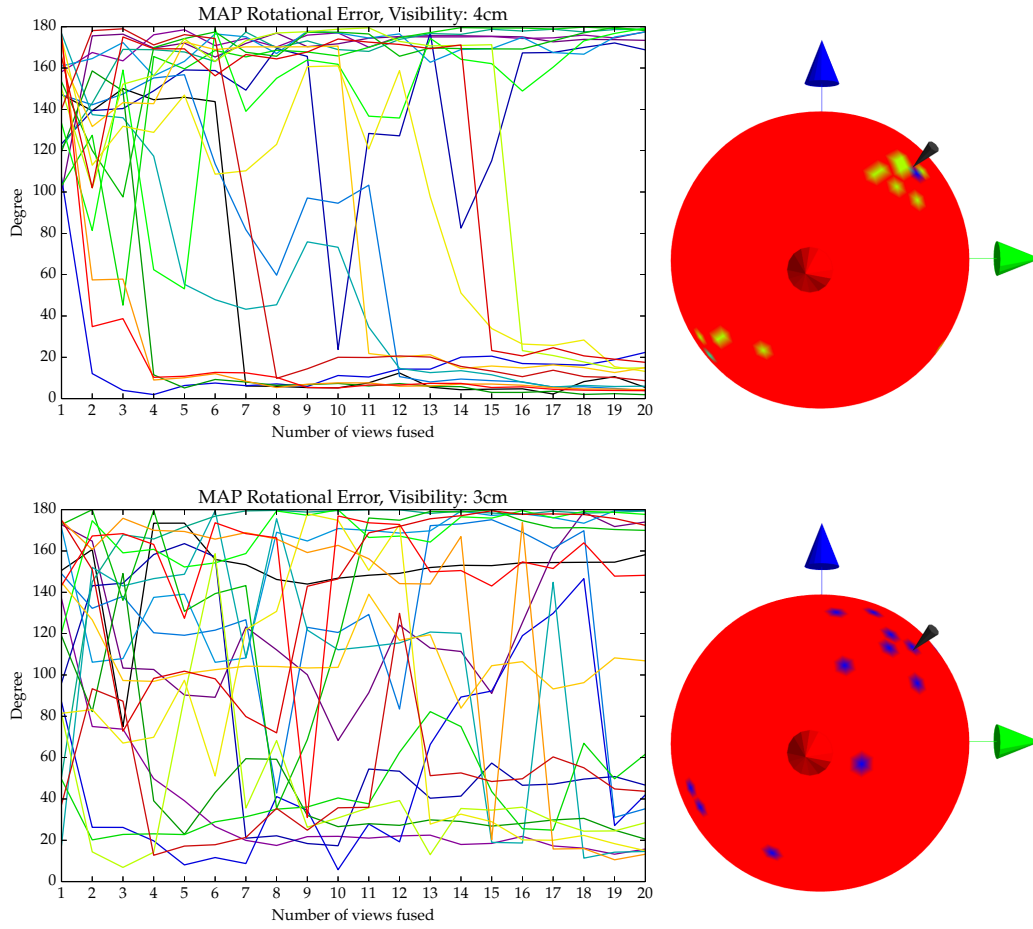


Figure B.3.: Left: MAP error sequence plots for visibility radii $r_{vis} \in [4\text{cm}, 3\text{cm}]$. Right: EGI plots of the MAP rotations after the last view.

Bibliography

- [1] Tal Arbel and Frank P Ferrie. Entropy-based gaze planning. *Image and Vision Computing*, 19(11):779–786, 2001.
- [2] Jens Behley, Volker Steinhage, and Armin B. Cremers. Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4391–4398, 2012. ISBN 9781467314053.
- [3] Christopher Bingham. An Antipodally Symmetric Distribution on the Sphere. *The Annals of Statistics*, 1974. URL <http://www.jstor.org/stable/2958339>.
- [4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 9780387310732. URL <http://www.library.wisc.edu/selectedtocs/bg0137.pdf>.
- [5] Roland T. Chin and Charles R. Dyer. Model-Based Recognition in Robot Vision. *ACM Computing Surveys (CSUR)*, 18(1), 1986.
- [6] Joachim Denzler and Christopher M. Brown. Information theoretic sensor data selection for active object recognition and state estimation. *Pattern Analysis and Machine Intelligence, 2002 IEEE Transactions on*, 24(2):145–157, 2002.
- [7] David Doria. Stratified Mesh Sampling for VTK. *The VTK Journal*, March, March, 2010.
- [8] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *Computer Vision and Pattern Recognition, 2010 IEEE Computer Society Conference on*, pages 998–1005. Ieee, June 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5540108. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5540108>.
- [9] Robert Eidenberger, Thilo Grundmann, Wendelin Feiten, and Raoul Zoellner. Fast parametric viewpoint estimation for active object detection. In *Multisensor Fusion and Integration for Intelligent Systems, 2008 IEEE International Conference on*, pages 309–314. IEEE, 2008.
- [10] Robert Eidenberger, Thilo Grundmann, and Raoul Zoellner. Probabilistic action planning for active scene modeling in continuous high-dimensional domains. In *Robotics and Automation, 2009 IEEE International Conference on*, pages 2412–2417. Ieee, May 2009. ISBN 978-1-4244-2788-8. doi: 10.1109/ROBOT.2009.5152598. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152598>.

- [11] Robert Eidenberger, Thilo Grundmann, Martin Schneider, Wendelin Feiten, Michael Fiebert, Georg v. Wichert, and Gisbert Lawitzky. Scene Analysis for Service Robots. In *Towards Service Robots for Everyday Environments*, pages 181–213. 2012. URL <http://www.springerlink.com/index/N9212641455202J2.pdf>.
- [12] Rong-en Fan, Kai-Wen Chang, Cho-Jui Hsieh, Xiang-rui Wang, and Chih-jen Lin. LI-BLINEAR : A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9(2008):1871–1874, 2008.
- [13] Wendelin Feiten, Muriel Lang, and Sandra Hirche. Rigid motion estimation using mixtures of projected Gaussians. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1465–1472. IEEE, 2013.
- [14] Jared Glover and Leslie Pack Kaelbling. Tracking 3-D Rotations with the Quaternion Bingham Filter. Technical report, MIT, 2013. URL <http://dspace.mit.edu/handle/1721.1/78248>.
- [15] Jared Glover and Sanja Popovic. Bingham Procrustean Alignment for Object Detection in Clutter. In *Intelligent Robots and Systems, 2013 IEEE International Conference on*, April 2013. URL <http://arxiv.org/abs/1304.7399>.
- [16] Jared Glover, Gary Bradski, and Radu Bogdan Rusu. Monte Carlo pose estimation with quaternion kernels and the bingham distribution. *Robotics: Science and Systems*, 2012.
- [17] Thilo Grundmann. *Scene Analysis for Service Robots*. PhD thesis, Albert-Ludwigs-Universität Freiburg, 2012.
- [18] S. Sathiya Keerthi, S. Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A sequential dual method for large scale multi-class linear SVMs. In *Knowledge Discovery and Data Mining, 2008 ACM SIGKDD International Conference on*, pages 408–416, 2008. ISBN 9781605581934. URL <http://dl.acm.org/citation.cfm?id=1401942>.
- [19] Simon Kriegel, Manuel Brucker, Zoltan-Csaba Marton, Tim Bodenmüller, and Michael Suppa. Combining Object Modeling and Recognition for Active Scene Exploration. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.
- [20] Jack B. Kuipers. *Quaternions and Rotation Sequences*, volume 66. Princeton University Press, 1999.
- [21] Muriel Lang. Approximation of Probability Density Functions on the Euclidean Group Parametrized by Dual Quaternions, 2011.
- [22] Catherine Laporte and Tal Arbel. Efficient Discriminant Viewpoint Selection for Active Bayesian Recognition. *International Journal of Computer Vision*, 68(3):267–287, May 2006. ISSN 0920-5691. doi: 10.1007/s11263-005-4436-9. URL <http://link.springer.com/10.1007/s11263-005-4436-9>.

-
- [23] Chih-jen Lin, Ruby C. Weng, and S. Sathiya Keerthi. Trust Region Newton Method for Large-Scale Logistic Regression. *Journal of Machine Learning Research*, 9:627–650, 2008.
- [24] Marianna Madry, Heydar Maboudi Afkham, Carl Henrik Ek, Stefan Carlsson, and Danica Kragic. Extracting essential local object characteristics for 3D object categorization. *Intelligent Robots and Systems, 2013 IEEE/RSJ International Conference on*, pages 2240–2247, November 2013. doi: 10.1109/IROS.2013.6696670. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6696670>.
- [25] Zoltan-Csaba Marton and Serkan Türker. On Bayesian Inference for Embodied Perception of Object Poses. In *Intelligent Robots and Systems, 2013 IEEE/RSJ International Conference on, Workshop on Metrics of Embodied Learning Processes in Robots and Animals, Oberpfaffenhofen*, 2013. German Aerospace Center (DLR).
- [26] Diego Nehab and Philip Shilane. Stratified Point Sampling of 3D Models. In *Point-Based Graphics, IEEE/Eurographics Symposium on*, 2004.
- [27] Lior Rokach. *Pattern Classification using Ensemble Methods*. World Scientific, 2010.
- [28] Andrew R. Runnalls. Kullback-Leibler Approach to Gaussian Mixture Reduction. *Aerospace and Electronic Systems, 2007 IEEE Transactions on*, 43(3):989–999, July 2007. ISSN 0018-9251. doi: 10.1109/TAES.2007.4383588. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4383588>.
- [29] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point Cloud Library (PCL). In *Robotics and Automation, 2011 IEEE International Conference on*, pages 1–4, 2011.
- [30] RB Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Robotics and Automation, 2009 IEEE International Conference on*, pages 3212–3217, 2009. ISBN 9781424427895. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5152473.
- [31] David J. Salmond. Mixture reduction algorithms for target tracking in clutter. In *Proc. SPIE 1305, Signal and Data Processing of Small Targets*, volume 1305, pages 434–445, 1990.
- [32] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique Signatures of Histograms for Surface and Texture Description. *Computer Vision and Image Understanding*, 125:251–264, August 2014. ISSN 10773142. doi: 10.1016/j.cviu.2014.04.011. URL <http://linkinghub.elsevier.com/retrieve/pii/S1077314214000988>.
- [33] Ken Shoemake. Quaternions. Technical report, Department of Computer and Information Science University of Pennsylvania, Philadelphia, 1994.
- [34] Michael A. Sipe and David Casasent. Feature Space Trajectory Methods for Active Computer Vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1634–1643, December 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.

1114854. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1114854>.
- [35] John Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 49(4), 1964. URL <http://epubs.siam.org/doi/abs/10.1137/1006093>.
- [36] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [37] Oncel Tuzel, Ming-yu Liu, Yuichi Taguchi, and Arvind Raghunathan. Learning to Rank 3D Features. In *Computer Vision (ECCV), 2014 European Conference on*, pages 520–535. Springer, 2014.
- [38] Jason L. Williams. Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter, 2003. URL <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA415317>.