
A Neurodynamic Model for Haptic Spatiotemporal Integration

Author:
M.Sc. Claudius Strub
September, 2016

PHD THESIS

A Neurodynamic Model for Haptic Spatiotemporal Integration

Dissertation zur Erlangung des Grades eines Doktor-Ingenieurs der Fakultät
für Elektrotechnik und Informationstechnik an der Ruhr-Universität Bochum

Author:

M.Sc. Claudius Strub
(born in Potsdam, Germany)

INSTITUTE FOR NEURAL COMPUTATION
RUHR-UNIVERSITÄT BOCHUM

AND

DEPARTMENT OF COMPUTATIONAL NEUROSCIENCE
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

September, 2016

Abstract

In this thesis the problem of spatiotemporal integration of haptic information during manual object exploration is analysed and a biologically plausible neurodynamic model for this process is proposed. Integrating haptic information over time and space is required in order to build a representation of the object shape. Haptic exploration of objects may occur in one of two settings, the first is a pure mapping problem, where the object pose with respect to the body remains fixed. In the second setting, the object pose may change during the exploration, leading to a simultaneous localization and mapping problem (SLAM). This thesis focuses on haptic spatiotemporal integration in this second, SLAM scenario. In particular, the aim is to contribute to the understanding of how the nervous system solves this task by proposing a biologically plausible model. This model is designed as a dynamical system with dynamic neural fields (DNF) and is evaluated in experiments with a robotic hand rotating and mapping several objects. The capabilities and limitations of the model are discussed, together with the implications. In a side track of this thesis, the autonomous adaptation of DNFs with intrinsic plasticity is introduced.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Identification	1
1.2.1	Implicit vs. Explicit Representation	2
1.2.2	SLAM of an Object	3
1.2.3	Haptic SLAM of an Object	4
1.3	The Approach: Why Neural Dynamics?	5
2	Background Information	9
2.1	Haptic Shape Processing in the Primate Nervous System	9
2.2	Haptic Shape Processing in Robotics	13
2.3	Simultaneous Localization and Mapping	15
2.4	Dynamic Neural Fields	19
2.5	Intrinsic Plasticity	24
2.6	Natural Gradient	25
3	Experimental Setup	29
3.1	Objects	29
3.2	Shunk Dexterous Hand 2	31
3.3	Coordinate Systems	32
3.4	Feature Extraction	33
3.4.1	Position	33
3.4.2	Orientation	33
3.4.3	Curvature	35
3.5	Object Exploration	37
3.5.1	Rotational movements from kinesthetic teaching	37
3.5.2	Generating a rotational behaviour	38
3.5.3	A Note on Translation	39
3.6	The Kinematics of Object Manipulation	39
3.7	Recording Datasets	42
3.8	Post-processing of the Curvature Feature	44
3.9	Implementation	47
3.10	Challenges of the Setup	47

4	Dynamic Neural Fields with Intrinsic Plasticity	49
4.1	Derivation	50
4.2	Evaluation	51
4.2.1	Low Amplitude	52
4.2.2	High Amplitude	53
4.2.3	High Shift	54
4.3	Discussion	56
5	A Neurodynamic Model for Haptic Spatiotemporal Integration	58
5.1	Feature Representation and Classification	61
5.1.1	Edge Detection	63
5.1.2	Surface Detection	63
5.1.3	Detection Parameters	65
5.2	Object Mapping	65
5.3	Object Localization	67
5.3.1	Orientation from Surfaces	67
5.3.2	Position from Edges	70
5.3.3	Localization Parameters	73
5.4	Pose Correction	73
6	Results	76
6.1	Object Mapping	76
6.1.1	Qualitative Evaluation of the Mapping	76
6.1.2	Quantitative Evaluation of the Mapping	84
6.2	Object Localization	94
6.2.1	Object Position Estimate	94
6.2.2	Pose Corrected Tactile Data	95
7	Discussion	99
7.1	The Model in General	99
7.2	Related Work on Primates	102
7.3	Related Work in Robotics	103
7.4	Scientific Contribution	106
7.5	Conclusion	107
8	Bibliography	109
A	Appendix: DNF with IP	119
B	Appendix: Neurodynamic Model	121

Chapter 1

Introduction

1.1 Motivation

Humans and primates in general have highly sophisticated manipulation skills with their hands. These hands are not only a complex and interwoven composition of muscles, bones, joints and tendons – implying the necessity of an intelligent motor control, but are also a sensory masterpiece. Deformable tissue, fingerprints, fingernails and skin influence the stimulation of several types of mechanoreceptors, giving the organism rich feedback about the things touched. Humans rely on their haptic feedback system when manipulating objects and when using them as tools [JOHANSSON & FLANAGAN 2009, AUGURELLE et al. 2003]. Furthermore, after manually exploring objects by manipulating these with haptic feedback alone, humans are capable to discriminate and recognise objects [DAVIDSON et al. 1974, KLATZKY et al. 1985, NORMAN et al. 2008]. These capabilities imply that humans and primates in general perform spatiotemporal integration of haptic sensations, i.e. they are able to combine several haptic sensations from different locations and different points in time into a single percept of the object. This raises the question of how in particular they do this. How in particular does the brain realize this spatiotemporal integration, how robust is it, how precise? The answers to these questions are currently unknown. This thesis contributes to the field of research by giving a theoretical analysis of the problem and a biologically plausible implementation by a neurodynamic model for the process of haptic spatiotemporal integration.

In the next section the problem of haptic spatio-temporal integration will be discussed and specified. For interested readers, details and background information relevant for this section are given in the next chapter Ch. 2. Finally a motivation for a neurodynamic approach to this problem concludes this introductory chapter.

1.2 Problem Identification

Haptic sensing gives information of structure in space which is inherently three dimensional. Extracting knowledge about this three dimensional structure out of the (possibly) high dimensional haptic information can in principle be done in an implicit or explicit way, as will be discussed in the following subsection. In the implicit approach, the tactile information is directly utilized for the task dependent behaviour,

never having an explicit representation of the three dimensional structure. To the contrary, in the explicit approach the three dimensional structure is explicitly extracted and represented (i.e. learned) as an intermediate step for goal directed behaviour (e.g. recognition, discrimination, manipulation).

1.2.1 Implicit vs. Explicit Representation

With an implicit representation the learning problem is viewed as a sequence learning task, where a sequence of pairs containing tactile stimuli and proprioceptive information is used to learn an desired output value. More precisely, the problem is formulated as learning a function f that maps a sequence of tactile and proprioceptive inputs $(Z(1:t), U(1:t))$ to an action from an arbitrary action space A . This action space might be a discrimination, a recognition or be in the space of motor commands. Hence, the goal is to learn the function:

$$f(Z(1:t), U(1:t)) \rightarrow \text{action} \in A$$

with $Z(t)$ holding the tactile information for time-step t and $U(t)$ proprioceptive information for time-step t . If the action depends on the object shape or pose, these must be inferred from the past history by integrating proprioceptive and tactile information.

However, if the object pose changes as the object is manipulated during the movements where tactile information is acquired, things get complicated. A change in the object's pose corresponds to a change in the relationship between the tactile stimuli $Z(t)$ and the according proprioceptive information $U(t)$. When viewing $(Z(t), U(t))$ tuples as input to the learner, a change in object pose would correspond to a change in hidden underlying parameters controlling the structure of the input. This corresponds to distinguishing the changes in the input into parts due to changes of hidden parameters underlying the input (object pose) and parts due to a static mapping defined by the object shape. As this input parametrization is unknown to the learner, it has to be additionally learned implicitly. This makes haptic information seem very irregular and the learning very hard when viewed as a sequence learning problem.

Additionally to this theoretical problem, there is evidence from neurobiology, psychophysics and developmental psychology that indicates explicit object shape representations in primates (see Sec. 2.1 in the background information chapter for further details).

This brings us to the alternative approach to the problem: an explicit representation. In the above formulation the coupling of proprioceptive $U(t)$ and tactile $Z(t)$ information by the object shape and pose is implicit. The explicit formulation would be to first split up the learning of the action:

$$f(m(t), X(t)) \rightarrow \text{action} \in A \tag{1.2.1}$$

$$P(m(t), X(t) \mid Z(1:t), U(1:t)), \tag{1.2.2}$$

where $m(t)$ and $X(t)$ are the object shape (geometry) and pose (translation + rotation) estimates at time-step t , respectively. $P(\cdot)$ denotes a conditional probability distribution. The object shape and pose are formulated as a conditional probability

distribution here because they have to be estimated from the haptic data. Thus $P(\cdot)$ in (1.2.2) is the probability of an object shape $m(t)$ (also termed “map”) and its pose $X(t)$, given all the observed tactile data $Z(1:t)$ and proprioceptive information $U(1:t)$ (i.e. joint angles). Using the object shape and pose expressions of (1.2.2), the relationship of tactile to proprioceptive information can be made explicit:

$$P(Z(t) \mid m(t), X(t), U(t)) \quad (1.2.3)$$

In expression (1.2.3) the probability of sensing the current tactile information is conditionally dependent on the estimate of the object location $X(t)$ and shape $m(t)$ as well as the current proprioceptive information $U(t)$. The problem of computing expression (1.2.2) is well known and in the robotic’s community and is referred to as the SLAM problem, *Simultaneous Localization and Mapping*. The slight difference with respect to the conventional formulation of the SLAM problem in this case is that localization and mapping is with respect to the object and not with respect to the environment. This conceptual difference is explained in the following subsection.

1.2.2 SLAM of an Object

The object based analogy to the SLAM formulation the mobile robot navigation literature (see Sec. 2.3 in the background information chapter for further details) can be viewed as following:

In Object manipulation, the role that a mobile robot has in the SLAM for navigation is now taken by a “mobile” robot gripper. The role of the environment which has to be mapped in SLAM for navigation now is an unknown object. Hence, every interaction of the robot with the object is viewed as a manipulation of the robot pose with respect to the object coordinate system. In mobile robotics the odometry of the robot is used to predict changes in the robots pose from motor commands. In object manipulation this role is taken by a forward model predicting the consequences of the robot-object interaction by taking the object kinematics into account. Although the location of the robot gripper is always known in the robot coordinate system, it is not known in the object coordinate system. Hence, the task is to build a map of the unknown (static) object while simultaneously localizing the robot with respect to the object.

However, in the context of manipulating an unknown object there is an SLAM problem with respect to the object, if and only if all of the following conditions are met:

- (I) A “mobile” object and the possibility of interaction. The agent can change the orientation and / or position (i.e. the pose) of an object, i.e. manipulate or interact with it.
- (II) The object shape is unknown. There is no prior information about the particular object available i.e. no shape, friction, weight, etc.
- (III) Inability to directly measure the object location. The position and orientation of the object can not be measured directly, but must be inferred from temporal sequences of sensory measurements or matching these with an object map.

- (IV) Inability to directly build a map of the object. The object is only partial observable by sensory measurements and mapping the object requires manipulation of it, i.e. interacting with it. Hence it is not possible to directly build a complete map of the object without using actuators to change the object pose.
- (V) Uncertainty in interaction outcomes *and* in sensory measurements. Note, that uncertainty in the interaction outcome is not the same as uncertainty in the sensory measurements, as the first leads to an integration of errors by the forward / motion model which in turn, induces a statistical dependency on sensory measurement errors and thus causes a drift of the map [THRUN *et al.* 2002] [MONTEMERLO & THRUN 2007]. In contrast, measurement noise alone merely affects the map precision and may be compensated by successive measurements, as the errors of each measurement are uncorrelated.

1.2.3 Haptic SLAM of an Object

A special case of SLAM of an object, discussed in the previous subsection, is when the only sensory modality used for this is haptics. For solving the SLAM problem the key are growing correlations between measured features (“landmarks”) in the map. These correlations are induced by measuring multiple landmarks in a single measurement step, which are then correlated by the same location error (see Sec. 2.3 for further details). Successive measurements of the same landmark enables localization with respect to these, decreasing the uncertainty in the pose estimate and further increasing the landmark correlations. As correlations between the mapped landmarks grow, the map converges.

In haptic SLAM the major problem is the constraint of only being able to measure very small spatial extents of the object at a time. The contact area of each sensor with the object is rather small for rigid, non-deformable sensors and objects, therefore the contact may only be characterised by features at one object location. In the mobile navigation context, this would correspond to exploring and mapping the environment by only using the bumpers of a robot as sensors. The low spatial extent of a tactile contact often only allows for features of limited complexity to be extracted for each contact made. Examples might be the position of contact, its classification into edges, corners and surfaces, and their orientation. At the very best an estimate of the object curvature can be made. The problem with these features is, that they are often repetitive and ambiguous in space, making them hard to use for mapping and localization in space. This makes it very hard to decide, whether a measured feature corresponds to a feature previously mapped or not (correspondence problem) [Fox *et al.* 2012].

Furthermore, as the feature (vector) of a tactile contact describes the contact attributes at the corresponding location on the object, this implies that most contact features (object landmarks) are only measured once. Features would only be re-measured when re-touching the same object location (loop closure), limiting information for object localization to these incidents. In contrast, with range sensors localization can be performed in the vicinity of mapped features.

Last, the major insight into solving the SLAM problem are the growing correlations between mapped features due to simultaneous measurement of multiple features, which lead to a convergence of the map. Multiple tactile contacts are needed, e.g. from

multiple fingers, in order to simultaneously measure features from multiple locations of the object at a time, which also makes the data association easier.

Summing up, SLAM with haptic sensors is hard because the small spatial extent of sensory information makes the localization hard, the low feature complexity makes the correspondence problem hard and the low number of simultaneously measured features fails to induce growing correlations between mapped features. These challenges make most of the approaches to SLAM in mobile navigation not directly applicable to the equivalent problem in haptic learning.

1.3 The Approach: Why Neural Dynamics?

From the previous section it becomes apparent, that the task of spatiotemporal integration of haptic sensations during object manipulation is a non-trivial task. Nevertheless, the primate nervous system is capable of successfully building object shape representations which may be used for task specific actions. The exact mechanisms of this process of haptic spatiotemporal integration are unknown (see Sec. 2.1). However, modelling this process gives insight into how the primate brain could perform this. The particular challenges, necessary assumptions and the resulting capabilities and limitations give rise to new scientific questions for experimental research. Hence, modelling a solution helps to better understand the underlying problem and thus aids the research on the mechanisms of neural processing of haptic information.

The aim of building a biological plausible model requires the information processing to be consistent with neural principles of computation. Choosing the right level of neuronal description may reveal structures of the process, enabling to identify constituents and to perform a functional analysis. Since the process of spatiotemporal integration is inherently continuous in time and space, a dynamical system is in particular suited as a process model. The dynamics are constrained by neuronal computational principles in order to meet the requirement of biological plausibility. Modelling the information processing with a dynamical system implies an abstraction from the particular neural substrate and its structure. This corresponds to a mean field approach where information is encoded in population activities which are continuous in time and space. This is in particular motivated from studies showing, that information encoding of object shape in neurons of the visual cortex (V4) may be interpreted as a population coding of curvature over object space, e.g. see [PASUPATHY & CONNOR 2002]. Modelling neural information processing on this abstract yet biologically plausible level is done within the *dynamic field theory* (DFT), which will be introduced in the methods section Sec. 2.4.

Note, that information processing with dynamical systems in closed loops is contrasted by many classical machine learning approaches, which often times focus on learning input-output mappings. Traditional neural networks may be trained either by batch-, online-, or incremental / sequential learning algorithms. While batch learning is often the most efficient learning method, it requires distinct phases of data acquisition and storage, learning, and exploitation – which is biologically implausible. In online learning, each training example is immediately processed by the learning algorithm and exploitation may take place at any time. However, most online learning algorithms require statistically independently and identically distributed inputs –

which is often times violated in the context of sensory processing during space- and time continuous behaviours. In incremental online learning (also termed sequential learning) no assumption on the independence of the input data is imposed. This type of learning is also referred to as *autonomous learning*, where data acquisition, learning, and exploitation are highly interwoven. For being biologically plausible, the dynamic process model needs to implement spatiotemporal integration in an autonomous, incremental and online fashion.

Outline of the thesis

In this thesis the argument for an explicit shape representation is made, built from haptic information during manual interaction with an object. A neurodynamic model is proposed that enables building such representations and evaluated in robotic experiments. This model is part of an architecture, comprising a object exploration with a robot hand, processing the haptic sensory information, estimating changes in the object pose and the localization and mapping by the model. Figure 1.3.1 shows an overview of this architecture, as proposed in this theses. The Fig. 1.3.1 also gives an visual overview of the according sections in the third and fifth chapters of this thesis. The *second chapter* of this thesis gives background information on the state of the art in haptic processing, on the SLAM problem in general and on the used methods and hardware. The *third chapter* introduces the used experimental setup. This includes the description of a forward model (object kinematics) for estimating changes in the ob-

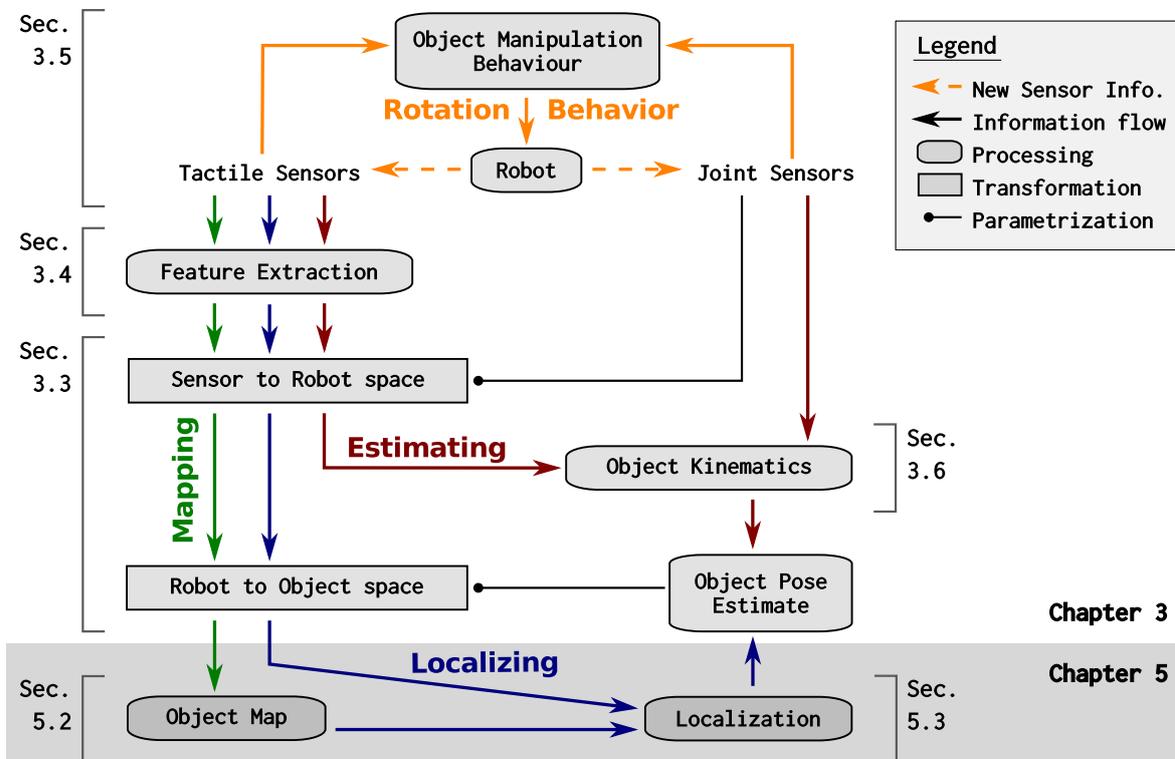


Figure 1.3.1: Overview of the architecture. The two boxes at the very bottom (gray overlay) comprise the neurodynamic model described in chapter Chp. 5. The remainder of the figure is described in the experimental setup Chp. 3.

ject pose from robot actions and explains how datasets of robot object manipulations were recorded for evaluation the model. Furthermore the used features and coordinate transformations are described. The *fourth chapter* introduces dynamic neural fields with intrinsic plasticity, including a brief evaluation and discussion. The *fifth chapter* is the core of this thesis, where a neurodynamic model is introduced for spatiotemporal integration of haptic information. This model performs the object mapping as well as the object localization with respect to the map, based on tactile features (highlighted with a gray overlay in Figure 1.3.1). In *chapter six* the proposed model is evaluated and the results are presented, based on the recorded robotic datasets. Finally, in *chapter seven* the model itself is discussed and set into the context of related work. The thesis is concluded by highlighting the scientific contribution and final remarks. The appendix mainly features additional figures of the results for the sake of completeness.

Chapter 2

Background Information

This chapter shall give background information, as far as it is relevant for the proposed work in this thesis. First, the current knowledge on haptic shape processing and representation in primates is reviewed, followed by exemplary approaches to haptic shape representation in robotics. Here the earlier work of e.g. [EL SADDIK et al. 2011, VAN ERP et al. 2010] is followed in defining *haptics* as an overall term including tactile and proprioceptive perception. The term *tactile* is used to denote any type of pressure sensors, *proprioceptive* to refer to any type of body posture sensors, e.g. measuring joint angles (also termed kinesthetic).

Second, a very brief and coarse introduction to the problem of simultaneous mapping and localization is given, without any details on particular solutions. In the scope of this thesis, only some pointers to publications in this field can be given, as there is a wealth of literature, books and in particular research papers on this topic. This chapter is concluded by sections introducing the used methods of dynamic neural fields, intrinsic plasticity and the natural gradient.

2.1 Haptic Shape Processing in the Primate Nervous System

This section will give an overview of haptic shape processing in the nervous system, followed by a section on approaches of haptic shape processing in robotics. The proposed neurodynamic model is inspired from what is known about haptic sensing, processing and shape representations in humans and primates (i.e. macaques). Here a brief review of the extensive literature shall be given. After mentioning haptic sensing, the processing of haptic information is described, focusing on four brain regions in particular: the primary and secondary somatosensory cortices, the intraparietal sulcus and the lateral occipital complex.

Haptic Sensors

Besides the proprioceptive information from joint angles, muscle lengths, etc. the hands and in particular the fingertips give a rich sensory feedback from tactile contacts [LOUW et al. 2000, JOHANSSON & FLANAGAN 2009, SAAL & BENSMAIA 2014]. The sensed information includes temperature, texture, curvature, edge orientation and higher or-

der moments, stiffness and temporal information such as vibrations [GOODWIN et al. 1997, JOHNSON 2002, HSIAO & GOMEZ-RAMIREZ 2011, PRUSZYNSKI & JOHANSSON 2014]. The dense distribution of mechanoreceptors in combination with the deformable tissue results in complex spatio-temporal patterns of force distributions with respect to amplitudes as well as force directions [JOHANSSON & FLANAGAN 2009]. Sensing vibrations on multiple frequency bands enables rapid object slip detection necessary for stable grasping and gives feedback of vibrations that are transmitted from the working end of tools [HSIAO & GOMEZ-RAMIREZ 2011]. For a more detailed introduction to the mechanoreceptors and their specific functions, see [JOHNSON 2002].

Tactile information from sensory neurons is first processed in subcortical brain areas as the brainstem and the thalamus [HSIAO & GOMEZ-RAMIREZ 2011]. From there, the information is processed in several regions of the primary somatosensory cortex (SI), described next.

SI: primary somatosensory cortex

In SI the tactile information is processed while segregating signals from different types of mechanoreceptors. Local, two dimensional features are computed, which include curvature and orientation of edges [YAU et al. 2013] and are largely comparable to features observed in the primary visual cortex (V1) [YAU et al. 2009, MASSON et al. 2015]. Additionally, global features are computed comprising information from several digits [HSIAO & GOMEZ-RAMIREZ 2011] which are fused with proprioceptive information into features in three dimensional space, i.e. adding a positional information [HSIAO 2008, AZAÑÓN et al. 2010]. Furthermore, there is evidence indicating that SI also serves as working memory for tactile features [HARRIS et al. 2002].

From here it is increasingly unclear which information is processed where. In [JAMES et al. 2007] two distinct pathways are proposed for further processing: one containing textural and material information, termed “microgeometry” and a separate pathway containing information about the form and geometry (“macrogeometry”). This distinction is similar to the distinction of the ventral and dorsal pathways in visual processing [KRAVITZ et al. 2013, KRAVITZ et al. 2011]. However, a clear distinction of pathways is disputed [CICHY et al. 2011, THEYS et al. 2015, DIJKERMAN & DE HAAN 2007](in particular the open peer commentary of the last article).

It has been shown, that information from SI is passed to the superior temporal gyrus (STG) for temporal tactile processing in the auditory cortex [BOLOGNINI et al. 2010]. Another human brain region critical for spatio-temporal integration of tactile information is the inferior parietal lobule (IPL) [KITADA et al. 2003] and in the supra-marginal gyrus (ASM) in particular [BODEGÅRD et al. 2001]. Furthermore, there is a high multisensory interaction on multiple processing stages, e.g. SI receives input from the visual cortex V1 and is directly connected to V2 and the medial temporal cortex (MT) [HSIAO & GOMEZ-RAMIREZ 2011]. SI also projects information to one of the two major multisensory processing areas: the intraparietal sulcus (IPS).

IPS / AIP: intraparietal sulcus / anterior intraparietal area

A major brain region involved in object shape representation is the human intraparietal sulcus (IPS), which is thought of to correspond to the monkey anterior intraparietal area (AIP) [OLIVER et al. 2009]. The IPS receives information from SI [BODEGÅRD et al.

2001] and is involved in haptic and visual processing (e.g. mental rotations, shape discrimination) [JAMES et al. 2007] and is in particular relevant for integrating sensory information from several fingers [KITADA et al. 2003]. Additionally, the IPS is relevant for temporally integrating tactile information and is crucial for determining whether information is fused into a single object representation or if multiple representations are constructed [KITADA et al. 2003]. Information from the IPS are passed on to anterior supramarginal gyrus (ASM), another area where object shape might be integrated into a holistic object [BODEGÅRD et al. 2001].

Recent evidence from intracellular recordings in macaques showed that the shape coding in the AIP is based on selectivity for small line fragments (similar as neurons in V2 and V4) which is sensitive to location. Neuronal preference tuning is very dependent on location and thus the neural activation on a population level may be entirely different for small changes in object position [ROMERO et al. 2014]. In combination with low response latencies to stimuli, it is not clear where in a processing hierarchy the AIP is located, in particular as the boundaries to other brain areas, e.g. the anterior lateral intraparietal (LIP) are unclear [ROMERO et al. 2014]. In [ROMERO et al. 2013] the encoding of aspect ratio and orientation was shown for populations of AIP neurons. The shape representation in AIP is known to project to the ventral premotor cortex (F5) and is a crucial step with respect to grasping objects [THEYS et al. 2013, SRIVASTAVA et al. 2009].

SII: secondary somatosensory cortex

From SI information is passed to the secondary somatosensory cortex (SII) which additionally might also independently synthesize tactile features as e.g. curvature [YAU et al. 2013]. SII seems also to integrate sensory information from multiple digits of both hands and shows position invariant features [HSIAO 2008]. The role of the tactile information processing in SII is mostly assumed to be of integrating geometric features into a holistic object shape representation relevant for object recognition [MIQUÉE et al. 2008] and conscious detection of somatosensory stimuli as e.g. object size and shape [REED et al. 2004, HSIAO & GOMEZ-RAMIREZ 2011].

Information from SII is (amongst others) projected to V4 [HSIAO & GOMEZ-RAMIREZ 2011], which is thought to have a similar shape encoding [YAU et al. 2009]. In V4, and thus presumably in SII, the boundary of the object shape is encoded in the population activity of the neurons [PASUPATHY & CONNOR 2002]. In general, vision and haptic pathways for constructing an object representation are highly interleaved, multi sensory processing and integration takes place at a variety of stages [JAMES et al. 2007, HSIAO & GOMEZ-RAMIREZ 2011, SNOW et al. 2013, MIQUÉE et al. 2008, THEYS et al. 2013]. This is supplemented by experiments in psychophysics showing that metric knowledge of objects is transferred from vision to haptics and vice versa, also indicating shared, multisensory object representation(s) [WALLRAVEN et al. 2014]. From SII there are also projections to the second major multisensory shape processing areas: the lateral occipital complex (LOC).

LOC / IT: lateral occipital complex / inferotemporal lobe

The lateral occipital complex (LOC) which presumably corresponds to the inferotemporal lobe (IT) in macaques, receives information amongst others from SII [HSIAO &

GOMEZ-RAMIREZ 2011] and from the IPS / AIP [SRIVASTAVA et al. 2009]. The LOC is a multisensory brain region critical for geometric object shape representation [JANSSEN et al. 2000, JAMES et al. 2007, HUNG et al. 2012, YAMANE et al. 2008, OLIVER et al. 2009] and interacts with the hippocampus which might store object representations [HSIAO & GOMEZ-RAMIREZ 2011]. It is assumed that the LOC is an convergence point for visual and haptic shape representation [AMEDI et al. 2002, MASSON et al. 2015] which not only contains information of the geometry (ventral “what” pathway) but also information about the location of objects (dorsal “where” pathway) [CICHY et al. 2011]. The information processed in the human LOC is presumably more about the object shape than on the level of specific features or contours, as indicated by fMRI studies [GRILL-SPECTOR et al. 2001]. Despite robust involvement of the LOC in haptic object recognition tasks, the LOC is not absolutely necessary for this, as a brain lesion study showed [SNOW et al. 2015]. The subject with lesions in the occipitotemporal lobe was unable to identify familiar objects by vision, but had unimpaired performance in haptic object recognition.

Furthermore, intracellular recordings in the macaque IT reveal that shape representation is a composite of neurons encoding the position, orientation and curvature of local surfaces. This representation is invariant with respect to the position, size and orientation of the object. The neurons in IT not only encode surface features, but also the skeleton of object parts, i.e. the medial axis [YAMANE et al. 2008, HUNG et al. 2012].

Differences between IT and AIP

In [THEYS et al. 2013] the shape representation in the macaque brain regions AIP and F5a is analysed, highlighting that IT holds a more detailed object representation, utilized for object categorization, while AIP / F5a has a rather coarse object representation used for grasping. The AIP region is said to be more boundary-coded, i.e. having little information about surfaces. The region F5a is assumed to transform the AIP representation into an intermediate representation used by the primary motor cortex (M1). Another study comparing these brain regions revealed that AIP has a more metric shape representation than IT, although IT is more sensitive to discontinuities, i.e. sharp edges [SRIVASTAVA et al. 2009]. However, it is not straight forward how the brain regions in the macaque brain link to the human brain and if the results are transferable [THEYS et al. 2013, BODEGÅRD et al. 2001]. This is partly due to the fact, that most of the studies on information representation in the AIP and IT are done with intracellular neural recordings, while the research on the human analogous brain regions is mostly restricted to fMRI.

Summary

Summing up, the processing of haptic information into object shape representations is a nontrivial process which is highly distributed and multimodal. This is reflected in fMRI studies of haptic shape exploration which lead to activations in multiple brain regions across the whole brain, involving prefrontal, parietal, occipital and temporal regions [MIQUÉE et al. 2008]. The exact mechanisms of object shape representation can therefore not yet be determined in particular as there are several distributed representations of object shape in the brain. The object shape is rather represented in

a conglomerate of task and function specific shape representations (e.g. for grasping, affections, decision making and reward, motor feedback) interacting with one another, partly sharing the same neural substrates [HSIAO & GOMEZ-RAMIREZ 2011, NORMAN et al. 2008]. However the evidence from multiple stages (in particular SI, AIP and IT) of (amongst others) haptic sensory processing show prominent haptic features similar to the ones in the primary visual cortex. In particular these tactile features are zero-, 1st, and 2nd order moments, which correspond to position, orientation, and curvature, respectively. Furthermore, there is strong neurobiological evidence that at least some object representations are invariant to the pose, which holds for vision [HUNG et al. 2012] as well as for haptics [HSIAO 2008]. This separation of an objects shape representation from its pose representation is also indicated by developmental psychology studies [SMITH 2009, SPELKE et al. 2010]. In [SPELKE et al. 2010] it is pointed to the possibility that the pose is represented in the hippocampus, e.g. by place cells. Altogether, there is a lot of research on where and how haptic information are processed and on spatial integration in the brain. However, how in particular *spatiotemporal* integration is achieved currently remains unknown.

2.2 Haptic Shape Processing in Robotics

In this section a brief review of the current state of the art of haptic sensing in robotics shall be given.

Sensors

There are multiple approaches for tactile sensors in robotics, some general types include capacitive-, piezoresistive-, piezoelectric-, inductive-, optoelectric- and strain gauges tactile sensors, for details see [TIWANA et al. 2012, GIRÃO et al. 2013, KAPPASSOV et al. 2015]. Measuring the joint positions in robotic systems can be done directly with built in encoders.

In the robotics community there is no common convention on how tactile data should be processed or how shape should be represented.

Tactile feedback is often only considered for determining whether the manipulator has contact to the object and to control the force of the grasp [PLATT et al. 2011, POPOVIĆ et al. 2010]. Additionally, tactile feedback is used for slip detection by analysing vibration frequencies [SCHÜRSMANN et al. 2012]. Research in the fields of robot grasping and manipulation where haptic feedback is used beyond direct motor feedback frequently circumvents the SLAM problem by either fixing the object pose, using prior knowledge (e.g. of the shape) or by incorporating other sensors (e.g. vision) to estimate and track the pose. Thus, research focusing on haptic feedback has mainly considered: (1) learning the shape and geometry of a rigidly mounted object, e.g. [MEIER et al. 2011, DRAGIEV et al. 2011]; (2) localizing objects where the shape is assumed to be known, e.g. [PEZZEMENTI et al. 2011, CHALON et al. 2013, KOVAL et al. 2013, LUO et al. 2015].

In order to learn the object geometry with haptics, a *spatiotemporal* integration into a shape representation is required. In principal, shape processing and representations greatly differ in their complexity. Here only some examples for different levels of object representation complexity shall be given.

Point cloud and occupancy grid representations

On the raw-, sensory near side are the group of point cloud and occupancy grid approaches. Here the tactile contacts are either directly stored as 3D contact points or more sophisticated methods for preventing the linear growth of required memory are used. An example for an elaborate point cloud approach is given in [MEIER et al. 2011]. The authors probe contacts with fixated, three dimensional objects and use a kd-tree of tactile contact positions as object shape representation. For preventing the tree from linear growth in the number of nodes, each node does not store the raw contact position but is a Kalman filter representing the mean contact position with the according covariance. New tactile contacts are fused to the nearest neighbour filter or, if supra threshold, a new Kalman filter is added to the kd-tree. An example for the occupancy grid approaches is given in [PEZZEMENTI et al. 2011], where a fixated, two dimensional object is sampled with a tactile sensor array. The environment is initially represented by an empty two-dimensional grid. For each sensor measurement, the positional information is related to the according cell in the grid and the grid value is increased. As there are successive tactile measurements, more and more cells in the grid are activated and thus ensemble the object representation.

Feature based representations

A second group of shape representations can be categorized as feature based. Technically speaking, the computation of a 3D position from a tactile sensor array activation could already be regarded as feature computation (like in the previous group). However, in this group the focus is primarily on the extraction of features for encoding shape information. For example, in [NAVARRO et al. 2012] the information from tactile sensors first undergoes a linear dimension reduction with principal component analysis (PCA) and is then fed into a nonlinear dimension reduction by a self organizing map (SOM). The resulting SOM coordinates after learning of an object are stored together with joint configurations (also encoded in a SOM) in a bag of keypoints descriptor for a further classification by a multi layer perceptron (MLP).

Parametric representations

Last, the category of parametric object shape representations shall be mentioned. The idea here is to use the tactile information to learn a parametric model of the shape. An example is given in [DRAGIEV et al. 2011], where the authors map a fixated three dimensional object by successive probing with a tactile sensor. For each sensor measurement the three dimensional position and the according surface normal vector of a tactile contact is computed and stored together with all previously recorded features. The necessity of storage of all datapoints technically disqualifies this approach as a parametric one, however, this is only due to the inability of online (incremental) learning of the particular parametric model. The stored features are used to (repeatedly) train a Gaussian process, modelling implicit surfaces of the object for shape estimating and grasping. Other examples for parametric object representations are [BIERBAUM et al. 2008] (superquadric functions) and [FAION et al. 2015] (transformed plane curves).

2.3 Simultaneous Localization and Mapping

The classic example for simultaneous mapping and localization (SLAM) is a robot which is driving through an office building with a laser scanner. After each movement the robot has to determine its new position (localization) by comparing laser measurements of the office environment with respect to its map of the office. Simultaneously, after each measurement of the office with the laser scanner, it has to fuse the laser measurements into the office map dependent on its current position. If there is an error in the localization, i.e. the robot assumes it is at a wrong location, this error will propagate to the map as the new measurements are fused at wrong locations. Errors in the map will lead to successive wrong localizations, and so on. The robot will get lost as the robot map of the office will increasingly deviate from the true office environment.

More generally speaking: an agent which senses parts of an unknown environment and acts upon it with the intention of achieving an environment related goal not directly measurable by the sensors is confronted with simultaneously localizing and mapping.

This SLAM becomes a problem as soon as there is uncertainty in the sensing *and* acting with the environment, as in any “real world” scenario. The uncertainty comes from two sources: uncertainty in sensory measurements and uncertainty in the interaction outcomes with the environment. The first transfers directly to the mapping, the other to the localization. Ignoring one of these uncertainties will lead to a divergence of the map (and location) from the ground truth and furthermore will lead to an inconsistent map. Therefore these two problems cannot be coped with separately, but must be solved simultaneously: Continuous localization with respect to an uncertain map and mapping with respect to an uncertain location.

The correspondence problem

Solving SLAM involves to solve the data association problem, also termed correspondence problem: do (parts of) the currently sensed data correspond to previously mapped data? If so, then they may be used to reduce the uncertainty in the location (localization). If not, they should be added to the map (mapping), where the uncertainty of the measurements is systematically influenced by the uncertainty in localization. This systematic impact of the localization error to the mapping leads the map to drift away from the true environment, as the map is used for the next localization. This correspondence problem becomes increasingly hard to decide, as

- the amount of uncertainty in location or sensory measurements rises.
- the amount of uncertainty in anticipated action outcome rises.
- sensory data decreases in frequency (spatial or temporal), i.e. is sparse.
- sensory data increases in ambiguity, i.e. from complex features (unique) to binary features.

The formal SLAM definition

The SLAM problem of localizing a mobile robot with respect to a (static) environment and simultaneously building a map of it has received great attention in the robotic navigation community and a rich amount of research has been conducted on how to solve this issue [THRUN et al. 2002, DURRANT-WHYTE & BAILEY 2006, BAILEY & DURRANT-WHYTE 2006].

In [DURRANT-WHYTE & BAILEY 2006], solving the SLAM problem is mathematically described as computing:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (2.3.1)$$

where x_k is the state x at time k , m is the map conditioned on all the past sensory measurements $Z_{0:k}$; and $U_{0:k}$ is the sequence of all the past actions of the agent which started at state x_0 . The $P(\cdot)$ denotes that the solution is a probability distribution. The chosen action u of the agent is assumed to be known with certainty.

Estimating the solution of Eq. (2.3.1) is done in three steps: First, a probabilistic action model is defined for predicting the outcomes of an action:

$$P(x_k | x_{k-1}, u_k) \quad (2.3.2)$$

describes the probability of being in state x_k conditioned on having been in state x_{k-1} and applying action u_k . Incorporating the predicted action outcome is then done by computing:

$$P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1} \quad (2.3.3)$$

Second, a probabilistic observation model is defined for prediction and rating of sensory measurements:

$$P(z_k | x_k, m) \quad (2.3.4)$$

describes the probability of measuring z_k conditioned on being in state x_k and having a map m . Thus, computing Eq. (2.3.4) requires to solve the correspondence problem. This implies that successive measurements z are conditionally independent of one another, given the location x and the map m . The fusing of the sensor measurements into the joint SLAM equation is then given by:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k | x_k, m) P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k | Z_{0:k-1}, U_{0:k})} \quad (2.3.5)$$

This is now the joint estimate of the current agent state x and the map m , computed by estimating the state based on the last state and a performed action, and corrected by observed measurements.

Approaches to the SLAM problem

Now, the general ideas for solving the SLAM problem and the associated correspondence problem shall be introduced. There are several approaches to the **correspondence problem**, which are often combined. Typically first a validation gating [BAR-SHALOM 1987, BAILEY et al. 2006] is applied to measured features. This checks the distance of a measured feature to the previously mapped features with respect to the

estimated location and rejects a data association for unlikely (large) distances. The distance measure usually incorporates the estimated variance / covariance of the sensors. All measured features within the validation gate remain to be associated with features in the map. The second important ingredient for robust data association is either measuring multiple features at a time (laser scan) or having “rich” feature measurements (e.g. camera image) as both enhance the reliability of an association [BAILEY & DURRANT-WHYTE 2006]. Last, the option of maintaining multiple hypothesis, i.e. multiple possible maps is a possibility to revert wrong associations. However, this is usually only done in particle filter approaches [BAILEY & DURRANT-WHYTE 2006].

The core insight to cope with the uncertainties in SLAM is to exploit the **growing correlations between measured features** [BAILEY & DURRANT-WHYTE 2006, SMITH & CHEESEMAN 1986, DURRANT-WHYTE 1988]. Multiple features which are simultaneously measured will be correlated in the map and correlation increases for each measurement of these features in successive action sequence steps, as the knowledge of the relative locations of the features with respect to one another is improved. Thus, the correlation between two features in the map grows monotonically for each simultaneous measurement of these two features. This increase of correlations creates a network of correlated features (i.e. a map), including the agent itself. Due to this correlation, reductions in positional uncertainty of one feature also propagates to all correlated features, which in turn reduces the next localization uncertainty, reducing the next measured feature location uncertainty, ... and so on. This property leads to the contraction of the probability distribution width for each feature m_j of the map m and thus, to an effective solution to the SLAM problem.

There are many implementations of SLAM solutions, most of them representing and computing with uncertainties in measurements (Eq. (2.3.1)) and in the location estimates [THRUN et al. 2002]. While this is considered to be mandatory for an optimal solution to SLAM, it requires an explicit formulation of a motion model (Eq. (2.3.2)) and an observation model (Eq. (2.3.4)). Two major approaches have been established, the extended Kalman filter (EKF) and the particle filter (FastSLAM) approach, see [MONTEMERLO & THRUN 2003, DURRANT-WHYTE & BAILEY 2006, BAILEY & DURRANT-WHYTE 2006] for details.

However, there have also been several biologically motivated approaches to the SLAM problem, not modelling the distribution probabilities explicitly. Here three examples are mentioned, just to give a brief impression.

For example in [LI et al. 2012] the authors represent the map of the environment in a dynamical system (shunting short term memory), similar to a DNF. Into this map the relative locations from a laser scan are fused. A scan matching algorithm then determines the maximum likelihood robot location for new data after a movement, conditioned on the predicted pose location. The exponential fading of the map entries serves as a feature for coping with dynamic environments.

In [MILFORD et al. 2004] a dynamic neural field (DNF) is used for representing the robots pose. Additionally, there are view cells which encode distance, relative bearing and color of detected objects in the robot environment. The connections between the activated pose and the currently activated view cell is learned by Hebbian learning. As the view cells project back into the pose DNF, re-localization is possible. The critical part here is the tuning of parameters of when to re-localize (e.g. ambiguous objects).

Furthermore, map drifts are only partially avoided.

In [ANTONELO & SCHRAUWEN 2012] a combination of a reservoir network (a nonlinear recurrent neural network at the edge of stability), slow feature analysis (extracting components of signals that change slowly in time) and independent component analysis (transforming a signal in statistically independent components) are used to autonomously build an implicit mapping of the environment without any motion model. Only eight infra-red sensors are used. However, the motion of the robot is required to be smooth and continuous for enabling the computation of slow features.

2.4 Dynamic Neural Fields

In the remainder of this chapter, the used methods are introduced, starting with dynamic neural fields (DNF). When the aim is to process information in a biologically plausible way, this raises the question on what level the processing should be biologically plausible [SCHÖNER 2008]. There are multiple approaches of modelling information processing in the nervous system, on different levels of abstraction. These range from detailed cellular models of neurons, to simplified spiking and non-spiking neuron models, to abstract descriptions of information processing in neural networks by mean field approaches. Dynamic neural fields are on the rather abstract end of this spectrum, encoding neural information in time and space continuous population activity. The processing of this information is then described by a differential equation which incorporates lateral interactions and external inputs into the temporal development of the system state. DNFs have been proposed in [AMARI 1977, WILSON & COWAN 1973, GROSSBERG 1988] where also a neural inspired derivation and an analysis of the dynamics is presented. Research has been conducted to model various processes with DNFs, such as object recognition [FAUBEL & SCHÖNER 2008], scene representation [ZIBNER et al. 2011], sequence generation [RICHTER et al. 2012], saccadic eye movements [SANDAMIRSKAYA & STORCK 2015] and spatial language [RICHTER et al. 2014].

Neural Dynamics

DNFs are dynamical systems which model the processing of information in recurrent neural networks, where the input information is explicitly encoded in a population representation (space encoding). This input may be a perceptual feature, location in space, or a motor control variable. The input feature is encoded along a feature dimension x , and the activation $u(x, t)$ at position x encodes the confidence that the feature has value x . This enables to encode multiple possible values of the feature as well as uncertainty. In contrast, traditional artificial neural networks encode a feature value in only one neuronal unit, where the value of the feature is represented by the activity of the unit (rate coding).

The equation for the DNFs used in the proposed model is described by Eq. (2.4.1), which defines the rate of change in activation $u(x, t)$ of the field:

$$\tau \dot{u}(x, t) = -u(x, t) + h + S(x, t) + \int \omega(|x - x'|) f(u(x', t)) dx'. \quad (2.4.1)$$

In Eq. (2.4.1), $u(x, t)$ is the activation of the DNF at time step t and position x . The position x describes a feature dimension and may be multi-dimensional: $\vec{x} \in \mathbb{R}^n$. Typically the dimensionality of x is in the range of $[0, 4]$. In particular, in this thesis one and two dimensional fields are used.

The term $-u(x, t)$ stabilizes an attractor for the activation function at values, defined by the last three terms in the equation. The time constant τ determines how fast activation $u(x, t)$ relaxes to the attractor. The negative resting level h ensures that the DNF produces no output without any external input $S(x, t)$, driving the DNF. The convolution term models lateral interactions between locations of activity in the

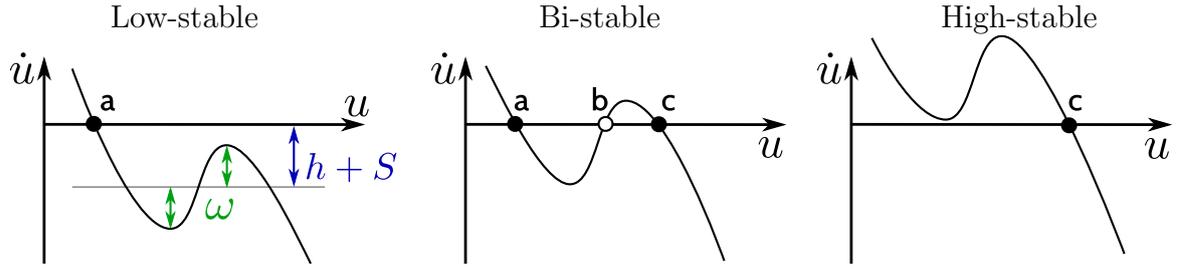


Figure 2.4.1: Phase plot for different regimes in the simplified DNF equation. Black dots indicate stable fixed points, empty circles unstable fixed point attractors.

DNF, shaped by the interaction kernel:

$$\omega(|x - x'|) = c_{exc} \exp \left[-\frac{(x - x')^2}{2\sigma_{exc}^2} \right] - c_{inh} \exp \left[-\frac{(x - x')^2}{2\sigma_{inh}^2} \right], \quad (2.4.2)$$

with a short-range excitation (strength c_{exc} , width σ_{exc}) and a long-range inhibition (strength c_{inh} , width σ_{inh}). A sigmoidal non-linearity, $f(u(x, t)) = \frac{1}{1 + \exp[-\beta u(x, t)]}$ defines the output of the DNF with which the DNF impacts on other dynamics in the model, as well as on its own dynamics through the lateral interactions.

The $-u(x, t)$ in Eq. (2.4.1) guarantees the existence of at least one stable fixed point (attractor). Dependent on the parametrization of the lateral interaction ω , the DNF may undergo saddle-node bifurcations. The phase plot depicted in Fig. 2.4.1 (“Low-state”) qualitatively shows Eq. (2.4.1), when the state x is zero-dimensional (i.e. a point). The black dot denotes a stable fixed point (attractor), while empty circles denote unstable fixed points (repeller). Loosely speaking, the resting level h together with the input S of the DNF shift the function up and down, while ω determines the non-linearity of the function. For example, assume a system as in the phase plot in Fig. 2.4.1, where the system state is stable at (a) (Low-stable). For an appropriate parametrization of the DNF, input S may cause a bifurcation, creating two new fixed points, a stable (c) and an unstable one (b) (labelled with “Bi-stable”). If the input further increases, a second bifurcation occurs, where the unstable fixed point (b) collides with the stable fixed point (a). Now the system state at the former fixed point (a) has lost its stability and the system will converge to the remaining stable fixed point (c) (“High-state”). This second bifurcation, where the current system state loses stability is termed *detection instability*. If the input now decreases again, it will induce a bifurcation, leading back to the bi-stable regime, however the system will remain at the stable fixed point (c) (hysteresis). Only if the input decreases enough to induce the second bifurcation, where the stable fixed point (c) collides with the unstable fixed point (b), the system will return to the stable fixed point (a) (i.e. from right to left in Fig. 2.4.1). Thus, the lateral interactions by the kernel ω stabilize the system in its state, even though the input may fluctuate.

For very strong lateral interactions ω , these may become sufficient to sustain the activity, even in the absence of external input. The dynamic system then has a line attractor as long as the interaction kernel $\omega(|x - x'|)$ is symmetric. For asymmetric kernels the system may enter into a limit cycle.

Memory Trace

Memory traces (MT) are described by dynamical systems without lateral interactions, which preserve memories of past activities in a DNF. The dynamics of a MT is given by Eq. (2.4.3) [SANDAMIRSKAYA 2013]:

$$\tau \dot{P}(x, t) = \lambda_{build}(-P(x, t) + u(x, t))f(u(x, t)) - \lambda_{decay}P(x, t)(1 - f(u(x, t))). \quad (2.4.3)$$

Here, $P(x, t)$ is the strength of the memory trace at position x of the DNF activity $u(x, t)$ and the sigmoided DNF output $f(u(x, t))$. The rates λ_{build} and λ_{decay} regulate the build-up and decay of the memory trace. The build-up of the memory trace is active at positions with a high positive activation in the DNF $f(u(x, t))$, the decay is active at positions with a low DNF activation.

Projections

DNFs with different dimensions may be connected with each other, therefore these connections are defined here. In order to pass activity from one DNF to a DNF of different dimensionality, several types of projections are used. These types of projections are illustrated in Fig. 2.4.2.

1D→2D For projecting the activity in a DNF with one feature dimension to a DNF with two feature dimensions, the DNF with one dimension projects its output activity $f(u(x, t))$ along the second feature dimension of the other DNF. Thus, the activity is “copied” along the new feature dimension.

2D→1D For projecting activity from a DNF with two feature dimensions to a DNF with one feature dimension there are several possible options. The most intuitive is to perform a marginalization of the DNF activity along one of the feature dimensions. This corresponds to integrating the activity along one dimension. Alternatively, the maximum activity in the dimension along which the activity is projected (i.e. the dimension which is marginalized) can be used to obtain a projection which remains in the same activity range, i.e. $[0, 1]$. Furthermore, a projection along a rotated dimension can be made, typically a rotation of 45° is used. This is in particular useful,

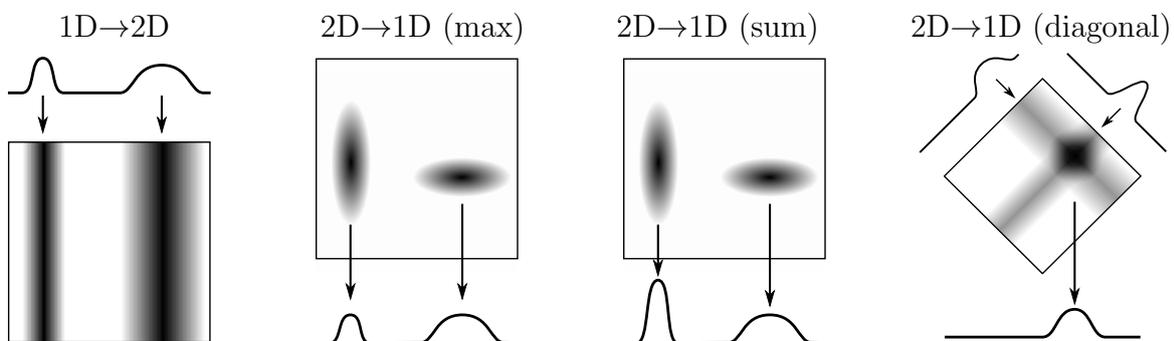


Figure 2.4.2: The types of projections used to connect DNFs with feature spaces of different dimensionality.

when the activity in the DNF feature dimensions is obtained from a superposition of two projections from DNFs with one dimensional feature spaces (see the sketch at the right of Fig. 2.4.2). This “diagonal projection” results in an activity peak in a relative feature dimension, encoding the activity of one of the two initial DNF feature spaces (one dimensional) relative to the other. This procedure implements a coordinate transformation and is described in detail in [SCHNEEGANS & SCHÖNER 2012].

In principal, all of these projection techniques can be generalized to DNFs with higher dimensional feature spaces. However, in this thesis only DNFs with one or two feature dimensions are used.

Discretization

The DNF equation is continuous in time- and space. The discretized version of the DNF equation with the Euler method for the temporal discretization, and the rectangle method for the spatial discretization is equivalent to the standard recurrent neural network equation:

$$u(k+1) = (1 - \Delta t)u(k) + \Delta t (W_{\text{inp}}S(k) + W_{\text{rec}}f(u(k))). \quad (2.4.4)$$

The Euler step width Δt defines the temporal resolution of the discretized time variable $\hat{k} = k\Delta t$. For small Δt the dynamics approximate the time continuous case, for $\Delta t = 1$ the standard discrete case for neural networks is obtained. The input $S(k)$ and the state (i.e. activation) $u(k)$ are now vectors (for one dimensional feature dimensions) or matrices (for two dimensional feature spaces), holding all the midpoints of the sampled rectangles along the feature and network state dimensions, respectively. Characteristic for a DNF are the particular structures of the input weight matrix W_{inp} and the recurrent weight matrix W_{rec} . These matrices are both square band matrices, typically symmetric. The input weight matrix W_{inp} is nonnegative with the values on the diagonal defined by a Gaussian distribution with mean of zero and width σ . The recurrent weight matrix W_{rec} is a square band matrix which typically has a central positive band enclosed by two negative bands, defined by a Mexican-hat distribution. The parameters of the distributions are typically tuned manually.

An framework for simulation of DNFs is described in the next subsection.

cedar Software Framework

A software framework for simulating dynamical systems, in particular DNFs was implemented in [LOMP et al. 2013] and is termed **cedar**¹.

The software is available for linux and windows and enables to design neurodynamic models with a graphical user interface where all parameters can be tuned interactively. A screenshot of the cedar graphical user interface is shown in Fig. 2.4.3.

The DNFs simulated in cedar have an additive activation noise to the state $u(x, t)$ with zero mean and the sigma is set to $\sigma_{\text{noise}} = 0.01$. For computational optimization, global interactions in the DNF activity like e.g. global inhibition are not realized by defining kernels of large width, i.e. interaction range. Instead, these are implemented

¹Source code available from: <http://cedar.ini.rub.de/>

by integrating the activity in the entire DNF and multiplying it by a parameter termed “global inhibition”, which is added to the kernel function output. The according equation will be defined in Eq. (5.0.4) in Chp. 5. While this mechanism increases the computational efficiency without any functional relevance, it leads to an additional, distinct parameter.

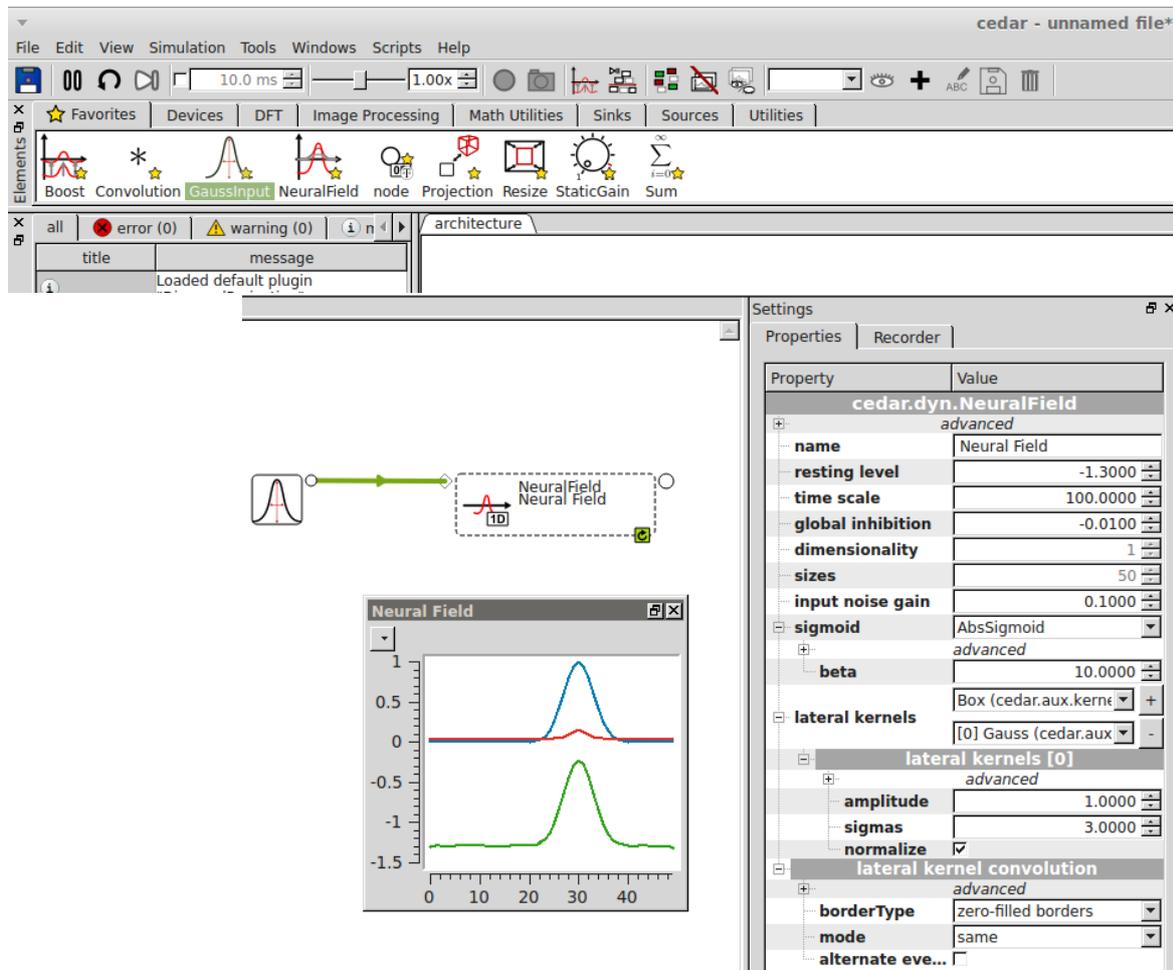


Figure 2.4.3: Parts of a screenshot of the cedar GUI. On the top the simulation controls and a selection of components are visible. In the lower part of the screenshot an example with a Gaussian input and a DNF with one dimensional feature space is shown. On the right the panel with the parameters is shown, which can be adapted while the simulation is running.

2.5 Intrinsic Plasticity

In this section the computational method of intrinsic plasticity is introduced. The description in this section and the following about the natural gradient are adapted from my master thesis [STRUB 2012].

Neurons in biological organisms have a large spectrum of plasticity mechanisms, implementing a board range of functions. One functional class of neuronal plasticity mechanisms is termed “homeostatic plasticity”, optimizing the information processing within a neuron by taking into account its metabolic costs. For reviews on forms and types of homeostatic plasticities see [DESAI 2003, POZO & GODA 2010, TURRIGIANO 2011, TURRIGIANO 2012].

Non-synaptic, i.e. intrinsic forms of homeostatic plasticity are termed “intrinsic homeostatic plasticity” (IP), which adapt the intrinsic excitability of a neuron [FRICK & JOHNSTON 2005, SCHULZ 2006]. Plasticity of excitability has also been discovered in compartmentalised dendritic structures of biological neuron’s [FRICK & JOHNSTON 2005, LOSONCZY et al. 2008, MAKARA et al. 2009]. Recent reviews on the roles of dendritic structures and their forms of plasticity with respect to the neural information processing and storage are given in [REMY et al. 2010, BRANCO & HÄUSSER 2010].

In the context of artificial neural networks IP has been introduced as a mechanism which modifies the excitability of a neuron in order to achieve a specified output distribution for a given input distribution [STEMMLER & KOCH 1999, TRIESCH 2005]. This is done by manipulating the parameters of a transfer function, transforming the internal neural state to an output. A commonly used transfer function is the Fermi function, also called logistic function, defined in Eq. (2.5.1).

$$\phi_{a,b}(x) = (1 + \exp(-ax - b))^{-1} \quad (2.5.1)$$

The a, b are termed gain and bias of the function ϕ and x is the input which is transferred to the output space.

The IP learning rule for adapting the parameters of the transfer function is achieved by minimising the Kullback-Leibler-divergence (KLD) [KULLBACK & LEIBLER 1951], such that the output distribution of a neuron is close to an exponential distribution. For Fermi functions (Eq. (2.5.1)) and the exponential as a target distribution this has been done in [TRIESCH 2005], as will be described in the following. For neurons using the tanh as transfer function, see [SCHRAUWEN et al. 2008].

$$\begin{aligned} L_{\text{KL}}(\phi || \text{exp}) &= \mathbb{E}_x[L_{\text{KL}}(\phi_{a,b}(x), \vec{\theta} || \text{exp}_{\mu}, \mu)] \\ &= \int f_{\phi}(\phi_{a,b}(x) | \vec{\theta}) \log \left(\frac{f_{\phi}(\phi_{a,b}(x) | \vec{\theta})}{f_{\text{exp}}(\phi_{a,b}(x) | \vec{\theta}, \mu)} \right) dx \\ \vec{\theta} &= (a, b)^T, \end{aligned} \quad (2.5.2)$$

In Eq. (2.5.2) the KLD is formulated with the probability distribution function f_{ϕ} of the outputs of the Fermi transfer function ϕ , with respect to the exponential target distribution f_{exp} . The output distribution of the Fermi function ϕ is conditioned on the parameters $\vec{\theta}$, while the parameter of the exponential distribution is the mean μ .

Minimising the KLD is done by taking the derivative with respect to $\vec{\theta}$, which leads to the learning rules for adaptation of the parameters $\vec{\theta}$ (i.e. the gain a and bias b) with the learn rate η :

$$\Delta b = \eta \left(1 - \left(2 + \frac{1}{\mu} \right) \phi_{a,b}(x) + \frac{1}{\mu} \phi_{a,b}(x)^2 \right) \quad (2.5.3)$$

$$\Delta a = \frac{\eta}{a} + x \Delta b \quad (2.5.4)$$

Besides this online adaptation rule introduced in [TRIESCH 2005], a batch version of IP was derived in [NEUMANN & STEIL 2011]. The application of IP has repeatedly been reported to improve performances in reservoir computing – a particular form of computing with transients in dynamical systems – [STEIL 2007a, STEIL 2007b, SCHRAUWEN et al. 2008, WARDERMANN & STEIL 2007] as well as increasing the robustness with respect to the parameter initialisation [NEUMANN & STEIL 2011]. There have been a number of variations of IP learning with respect to the target distribution, for further information see [VERSTRAETEN et al. 2007, SCHRAUWEN et al. 2008, BOEDECKER et al. 2009b, BOEDECKER et al. 2009a]. Furthermore, combinations of IP with other forms of plasticity have been investigated, e.g. with Hebbian learning, which leads to identification of independent components in the input [TRIESCH 2007].

Finally, it should be noted, that it has previously been reported that IP leads to instability of recurrent neural networks (RNN). In [MARKOVIĆ & GROS 2010] the authors claim that introduction of IP in RNN leads to the destruction of attractor stability, resulting in spontaneous and continuously ongoing activity. For networks without and with very small input amplitudes oscillatory, chaotic and bursting behaviour have been reported [MARKOVIĆ & GROS 2012]. The result of RNN destabilisation by IP has also been confirmed in spiking neural networks [LAZAR et al. 2007]. These destabilizing effects on the dynamics are relevant for applying IP in dynamic neural fields as proposed in Ch. 4, which will be discussed there.

2.6 Natural Gradient

The concept of a natural gradient was introduced in [AMARI 1998]. It has been shown, that the metric structure of the parametric space of neural networks has a Riemannian character [AMARI 1998]. Thus the relationship between the distance of two sets of parameters and the distance in the output space of the neural network is non-linear. Adapting the conventional gradient with respect to the Riemannian metric corrects for this non-linearity, such that the distance of two parameter sets linearly transfers to the output space. This change of the gradient is termed *natural gradient* and leads to a substantial performance increase in the convergence rate for IP (discussed in the previous section) [NEUMANN & STEIL 2012, NEUMANN et al. 2013]. Therefore, the natural gradient will be described in this section together with its application to the IP gradient. The remainder of this section is adapted from my master thesis [STRUB 2012].

The used notation is the following: parameters θ in parameter space Θ determine a mapping $f(\cdot|\theta)$ in the function space of all possible functions \mathcal{F} . The function $f(\cdot|\theta)$ maps input \cdot from \mathbb{R}^{IN} \rightarrow \mathbb{R}^{OUT} where IN is the number of input dimensions and OUT the

number of output dimensions. A loss-function $L(\theta) = L(f(\cdot|\theta); g^*(\cdot))$ in the function space of all possible loss-functions \mathcal{L} defines a distance of a mapping $f(\cdot|\theta)$ to a target mapping $g^*(\cdot)$. For the sake of simplicity, it is assumed that the target mapping $g^*(\cdot)$ is in the set of possible functions \mathcal{F} , i.e. $\exists \theta^* : g^*(\cdot) = f(\cdot|\theta^*) \in \mathcal{F}$ with target parameters θ^* . If $g^*(\cdot)$ is not in \mathcal{F} , the optimal parameters and thus the optimal mapping are given by $\theta^* = \arg \min_{\theta} L(\theta)$.

There is an important difference in considering distances between parameters θ in Θ or in $L(\theta)$. Ideally the loss-function $L(\theta) = L(f(\cdot|\theta); f(\cdot|\theta^*))$ represents the Euclidean distance of a parameter θ to the optimal parameter θ^* . This is the case if the mapping $f(\cdot|\theta)$ is the identity \mathbb{I} and thus, the loss-function is defined directly on the parameter space. Then the gradient on $L(\theta)$ would point to the optimum θ^* in parameter space. However a non-linear mapping $f(\cdot|\theta)$ and the particular choice of $L(\theta)$ induce a non-linear distance measure (metric) in Θ . This implies that the direction of steepest descent in $L(\theta)$ is not necessarily the direction with the strongest decrease in the Euclidean distance to the optimal parameter θ^* . If this distortion is not taken into account, gradient descent on $L(\theta)$ will lead to suboptimal trajectories in Θ , as illustrated on the right hand side in Fig. 2.6.1.

It has previously been noted that the metric in Θ of artificial neural networks with sigmoid activation functions is not Euclidean but Riemannian [AMARI 1998]. Therefore the task is to find a non-linear metric in Θ , such that the distance $d_F(\theta_1; \theta_2)$, defined as the geodesic distance of θ_1 and θ_2 with respect to a Riemannian metric tensor

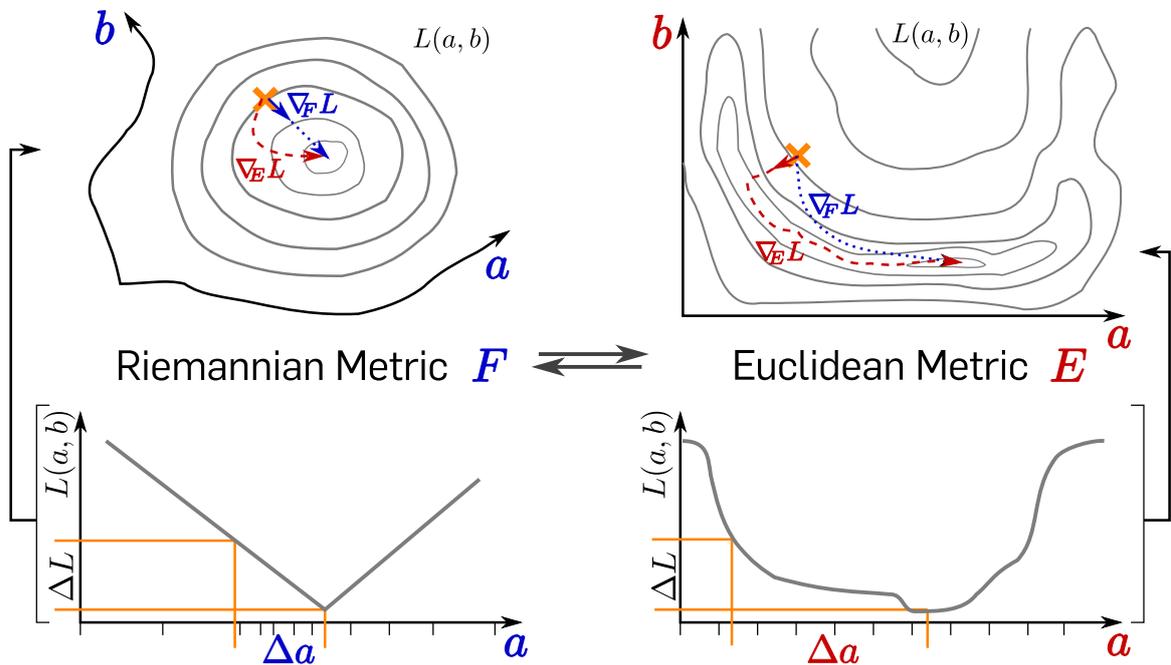


Figure 2.6.1: Loss-function L visualised with Riemannian metric (left) and Euclidean metric (right) in parameter space. Top: gradient descent with the Euclidean metric (∇_E red dashed lines) and natural gradient descent computed with the Riemannian metric tensor F (∇_F blue dotted lines). The loss-function $L(a, b)$ is schematically illustrated by the contour lines in the top plots. Bottom: a slice out of the top sketches for fixed b and the loss-function L on the y -axis. This figure is adapted from [STRUB 2012].

$F(\theta)$ corresponds to an Euclidean distance measure $d_E(L(\theta_1); L(\theta_2))$ on $L(\theta)$. As this Riemannian metric directly links distances in Θ with distances in $L(\theta)$, it ensures that the steepest descent direction of the loss-function coincides with the direction in which the error in the parameters decreases the fastest. Hence $L(\theta)$ encodes the distance of the current to the optimal parameters, with respect to the Riemannian metric tensor². This is visualised on the left hand side in Fig. 2.6.1 where the potential field of the loss-function is normalised to a cone by incorporating the Riemannian metric. Gradient descent on $L(\theta)$ then directly points to the optimum in parameter space, thus coinciding with the desired gradient in parameter space, resulting in a minimal trajectory length. As the incorporation of the Riemannian metric normalises the slope of the loss-function, the gradient magnitude is isotonic throughout parameter space.

Using gradient descent with the Riemannian metric is termed *natural gradient* [AMARI 1998] and does not alter the optimum but gives an increased convergence rate. Furthermore, it leads to an asymptotically Fisher-efficient estimator, i.e. the NG reaches the Cramer-Rao bound for unbiased parameter estimators [AMARI & DOUGLAS 1998].

A natural gradient-based parameter adaptation for IP termed NIP has been derived in [NEUMANN & STEIL 2012]:

$$\begin{aligned}\vec{\theta} &= (a, b)^T \\ \Delta\vec{\theta} &= -\eta_{\text{IP}} \left(F(\vec{\theta}) + \epsilon\mathbb{I} \right)^{-1} \nabla_E L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta})\end{aligned}\quad (2.6.1)$$

$$\begin{aligned}&= -\eta_{\text{IP}} \nabla_F L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta}) \\ F(\vec{\theta}) &= \mathbb{E}_x \left[\nabla_E L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta}) \cdot \nabla_E^T L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta}) \right]\end{aligned}\quad (2.6.2)$$

The Matrix $F(\vec{\theta})$ is the Fisher information, i.e. the Riemannian metric tensor and L_{KL} the KLD for neuron output $\phi_{a,b}(x)$ and parameters $\vec{\theta}$. As the needed expectation value of the gradient with respect to the input in Eq. (2.6.2) is not available in an online framework, the tensor is estimated online by:

$$\hat{F}_{t+1} = (1 - \lambda)\hat{F}_t + \lambda \nabla_E L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta}) \cdot \nabla_E^T L_{\text{KL}}(\phi_{a,b}(x)|\vec{\theta}), \quad (2.6.3)$$

with λ realising a low pass filter with exponential decay which is set to 0.01. For computational efficacy the inversion of the tensor F in Eq. (2.6.1) can be circumvented by directly estimating the inverse tensor F^{-1} as described in [PARK et al. 2000].

Using NIP gives a good approximation of the gradient direction in parameter space as the experiments in the evaluation in Sec. 4 confirm.

²Riemannian metric tensor field, to be exact.

Chapter 3

Experimental Setup

This chapter will give an introduction to the experimental setting of the research in this thesis. Two fingers of the robotic hand SDH2 (Sec. 3.2) are used to rotate different n-gon objects (an extrusion of a convex polygon with n sides) supported by a table surface. The goal of the experiments is then to autonomously build a representation of the object shape in an online and incremental fashion, while simultaneously estimating errors in the moment-to-moment localization of tactile features on the object surface. How this may be achieved will be described in detail in Sec. 5, while this chapter deals with the according experimental setup. This chapter gives

- an overview of the used objects and the used robot hand,
- the definition of several coordinate systems,
- a description of the feature extraction from tactile sensor responses,
- the objects exploration via rotational manipulations with the robot SDH2,
- the introduction of a forward model for predicting the object kinematics from robot manipulations,
- the procedure of recording datasets for further processing by the neurodynamic model,
- and finally, the encountered challenges in this setup.

3.1 Objects

For the experiments, seven cylinder based n-gons were custom made from aluminium, shown in Fig. 3.1.1.

The objects for the haptic spatiotemporal integration experiments consist of three sets. The first set comprises two objects with a small (outer) diameter of 3.0cm, one eight-sided and a six-sided object. The second set consists of three objects with a large (outer) diameter of 4.0cm, one cylindrical, one eight-sided and a six-sided object. The third set contained two objects with a large (outer) diameter of 4.0cm which, unlike the previous two object sets, both did not have a repetitive pattern of surfaces. Instead

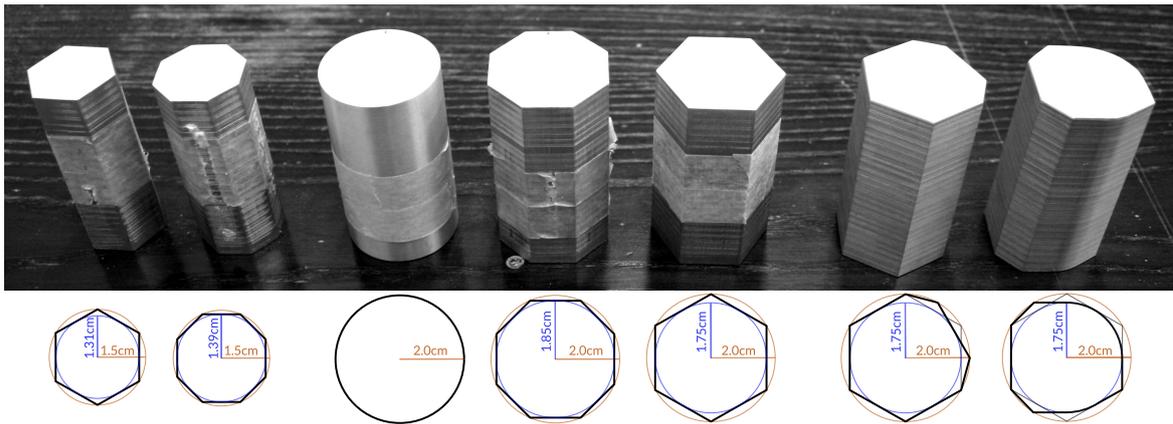


Figure 3.1.1: Objects: the top row shows the custom made objects. The bottom row shows the object shape from a top-view (black) including the outer diameter (orange), the inner diameter (blue) and the underlying shape (gray) in the case of the asymmetric objects. Note that in the image the tape of the two asymmetric objects on the right is still missing.

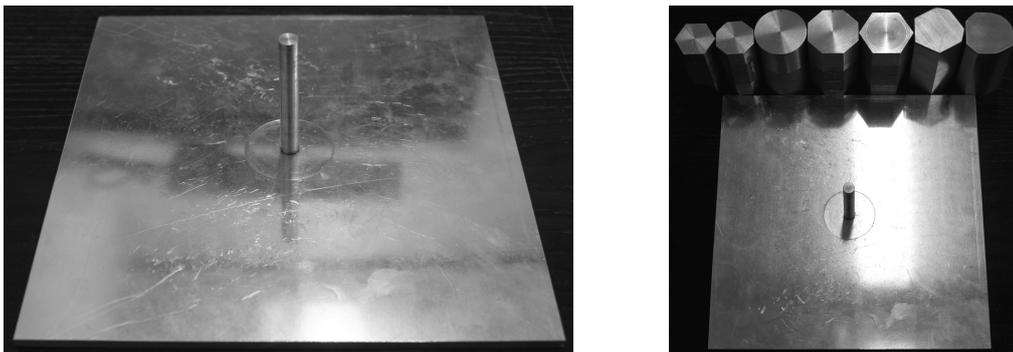


Figure 3.1.2: The ground plate with a bolt to fixate the position of objects.

the location and orientation of surfaces is asymmetric (see the two objects on the most right of Fig. 3.1.1). The first asymmetric object has seven surfaces but only five edges, due to smooth radial blending between neighbouring surfaces, as visible on the most right in Fig. 3.1.1. The second asymmetric object has seven surfaces with seven edges in between. From the seven surfaces of the asymmetric objects two surfaces have comparably small spatial extent, shown in the last and second to the last on the right in Fig. 3.1.1.

All objects had a height of 7.0cm. In order to facilitate the grip of the robot fingers during object manipulation, the surface of the objects was roughened and partially equipped with a double-faced (weakly) adhesive tape, see the photos in Fig. 3.1.1. All the objects had a 6.0cm deep hole of 0.8cm diameter in the bottom, by which they could be attached to a steel axis to prevent tipping of the objects and to control the object Position. For this, an aluminium plate with a bolt in the middle (6.0cm height and 0.75cm diameter) was manufactured, as shown in Fig. 3.1.2.

3.2 Shunk Dexterous Hand 2

For the experiments a Shunk dexterous hand in the second version (SDH2) with seven degrees of freedom (DoF) was used to manipulate objects. In Fig. 3.2.1 a picture of the SDH2 is shown. The SDH2 has three fingers, each with two phalanges connected by a middle joint and a co-linear joint near the palm. Two of these fingers are attached to the palm by an additional joint, enabling a coupled change in the orientation of the two fingers. The third finger is directly attached to the palm, leaving it with only two DoF to control the proximal and distal phalanx. A fingertip like, curved ending of the distal phalanges including the sensor surface enhances the grasping capabilities of the SDH2.

Each joint is equipped with a sensor to measure the current joint angle. Furthermore, each of the six phalanges is equipped with a tactile sensor array (DSA) with 250 kPa pressure measuring range. The tactile sensors on the distal phalanges have a resolution of 6×13 pixels (texels) with a narrow section in the fingertips of only 4 texels. The tactile sensors on the proximal phalanges have a throughout resolution of 6×14 texels. All of the tactile sensors have a sampling rate of 230 frames per second (fps).

The SDH2 comes mounted on a wooden base by default, but may be attached to robotic arms as e.g. the KUKA LBR 4 lightweight robot arm.

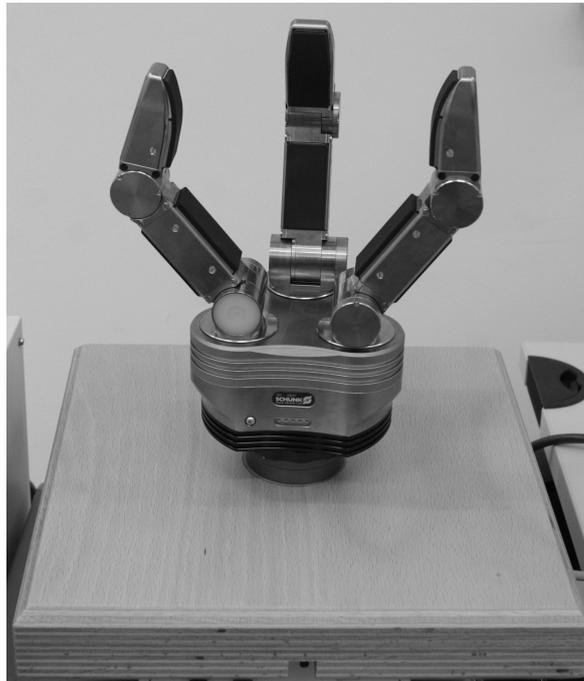


Figure 3.2.1: A picture of the Shunk SDH2

3.3 Coordinate Systems

When referring to tactile contacts from the SDH2 or extracted features from these sensory data (as will be described in Sec. 3.4), several coordinate systems are distinguished. These are termed the sensory-, robot- and the object-space. In Fig. 3.3.1 an illustration of these coordinate systems is given.

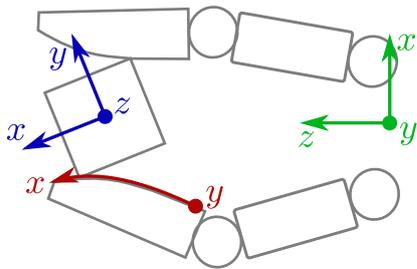


Figure 3.3.1: Overview of the three coordinate systems used. Red: sensor space, green: robot space, blue: object space. The dot in a coordinate system indicates an upright axis pointing downwards. See text for the relationship between these systems.

Sensor Space

The sensor coordinate system is a two dimensional space rooted on the according tactile sensor, thus a specification with respect to the tactile sensor must be made. As a sensor coordinate system is directly linked to the tactile sensory array, it is only defined with a restricted spacial extension.

Robot Space

When the kinematics of the robot is known and additional information from proprioceptive sensors are available, e.g. joint angle measurements, coordinates from the sensory space can be transformed into three dimensional robot space. The system of the robot coordinate system is fixated on the robot hand and the sensory information of the sensor space now lies within two dimensional manifolds. While the robot space is essentially three dimensional, the third dimension (height) is neglected in the proposed setup, as the fingers and thus the sensors only move within a two dimensional plane. In the following the z axis of the robot-centred coordinate system (green) in Fig. 3.3.1 will be termed **distal-proximal axis (DPA)**, while the x -axis is termed **lateral axis (LA)**.

Object Space

If the pose (i.e. position and orientation) of the object in robot space is known, i.e. with respect to the robot hand, information from the robot space can be transformed into the object space. Here, the coordinate system is rooted on the object. Just as in the case of robot space, the object space is essentially three dimensional, but as the object exploration only takes place in a two dimensional, planar “slice” of the object, the third dimension is neglected here as well.

In the terms of the SLAM literature, this corresponds to the map coordinate system of the object, i.e. the object representation. Building and maintaining this object representation requires to transform contact information into object space.

3.4 Feature Extraction

For further processing of tactile sensory information, several features are extracted. The specific types of features are inspired from what is known about haptic shape processing in humans and primates, briefly summarized in Sec. 2.1. Prominent features in both – the visual and haptic systems – are zero-, 1st, and 2nd order moments, which correspond to position, orientation, and curvature of the tactile contact, respectively. In accordance with the findings in biological systems, these three features are used as inputs to the neurodynamic model. The features are computed from tactile sensor information in sensor space, and are then transformed into robot space using the forward kinematics of the robot. Since there could be multiple tactile contacts of a sensor with an object, the tactile responses are first clustered, such that neighbouring active texels are appointed to the same cluster, i.e. contact area. This clustering was performed with the openCV cluster algorithm `cvblob1`. For each detected contact area three features are computed, described in the following subsections.

3.4.1 Position

The position of a contact area is defined as the cluster centroid as illustrated in Fig. 3.4.1.

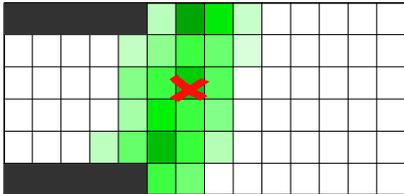


Figure 3.4.1: Sketch of a tactile sensor array of a distal phalanges of the sdh2 with a single contact area and its centroid. Active texels are illustrated in green color coding and the centroid is marked by a red cross.

The cluster centroid is computed by

$$\vec{c} = \frac{1}{\sum_{i=0}^n p_i} \sum_{i=0}^n p_i \vec{x}_i$$

with n the number of texels belonging to the contact area, p_i the pressure value of texel i and \vec{x}_i the position of texel i .

As this centroid is weighted with the pressure levels of each texel, the contact position can be determined with a higher resolution and robustness compared to using the texel coordinates of the maximum pressure level.

3.4.2 Orientation

For computation of the orientation feature the detected cluster of texels is approximated with a two dimensional Gaussian. The orientation of a contact area is then computed by determining the angle of the eigenvector with the largest eigenvalue. The eigenvector with the larger eigenvalue corresponds to the major semiaxis of the ellipse, defined by the covariance matrix of the Gaussian approximation. This is shown in

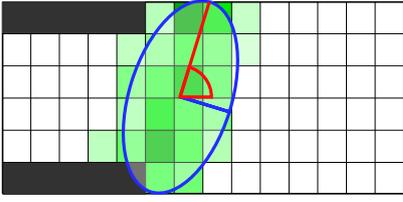


Figure 3.4.2: Sketch of a tactile sensor array with a single contact area and its orientation. Active texels are illustrated in green color coding and the orientation is marked by the red major axis of the blue ellipse.

Fig. 3.4.2, where the covariance matrix is illustrated by an ellipse denoting an equidistant line from the Gaussian kernel.

The computation of the covariance matrix Σ is given by:

$$\Sigma = \frac{1}{n-1} \sum_{i=0}^n (\vec{x}_i - \vec{\bar{x}}) \cdot (\vec{x}_i - \vec{\bar{x}})^T$$

with n again denoting the number of texels belonging to the cluster, \vec{x}_i denoting the position of texel i and $\vec{\bar{x}}$ denoting the arithmetic mean of all the cluster texel positions.

The eigenvalues e can be determined by using the quadratic formula to solve the characteristic polynomial of Σ , which can be written as:

$$e_{i \in [1,2]} = \frac{Tr(\Sigma)}{2} \pm \sqrt{\left(\frac{Tr(\Sigma)}{2}\right)^2 - Det(\Sigma)}$$

where $e_{i \in [1,2]}$ are the two eigenvalues, $Tr(\Sigma)$ is the trace of Σ and $Det(\Sigma)$ is the determinant of Σ .

The according eigenvectors can be computed by exploiting the Cayley-Hamilton theorem, resulting in the equation:

$$A_{1/2} = \Sigma - e_{1/2} \mathbb{I}$$

where $A_{1/2}$ is a matrix which is composed of the according eigenvector and its multiplicatives (column wise, i.e. $A_1 = [\vec{e}_1, a \cdot \vec{e}_1]$ with \vec{e}_1 the first eigenvector) and \mathbb{I} is the identity matrix.

Note, the used experimental setup practically constrains the orientation of a contact to one direction only. This is due to fact, that the shape of the object does not vary along its height, i.e. is purely defined by its footprint.

The contact orientation of contacts with object edges, i.e. long and narrow contact areas on the tactile sensor surface is well defined. On the other hand, the orientation of circular contact areas, e.g. when making contact with flat object surfaces or corners, is more limited in its information content. However, when the two dimensional sensor surface is transformed into three dimensional robot space the kinematics determine the orientation by the sensor geometry and pose, as shown in Fig. 3.4.3.

Here, the normal vector of the sensor surface at the contact position is given by the robot kinematics. This normal vector determines the orientation of surfaces in three dimensional space, while the orientation of the contact area in sensor space then determines the orientation of edges in three dimensional space.

Note, that the normal of the object surface does not necessarily coincide with the normal of the sensor surface, due to the rigid fingers and sensor surfaces.

¹Source code available from: <https://code.google.com/archive/p/cvblob/>

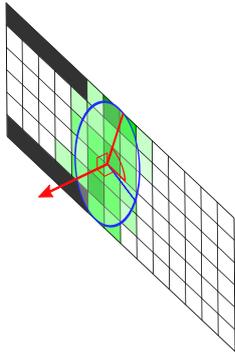


Figure 3.4.3: Sketch of a tactile sensor array with a single contact area and its orientation in three dimensional robot space. Active texels are illustrated in green color coding and the orientation is marked by the red major axis of the blue ellipse and the red Sensor surface normal vector.

3.4.3 Curvature

The curvature of a contact is computed by the ratio of the smaller eigenvalue λ of the covariance of the tactile contact area to the larger one (EvR).

$$\text{EvR} = \frac{\lambda_{\text{small}}}{\lambda_{\text{large}}} \in [0, 1] \quad (3.4.1)$$

The eigenvalues are denoted in Fig. 3.4.4 as the norm of the major and minor semiaxis of the ellipse.

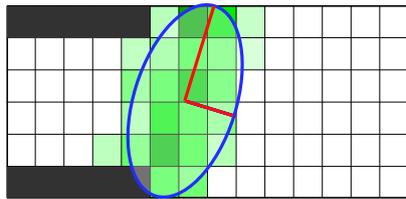


Figure 3.4.4: Sketch of a tactile sensor array with a single contact area and its curvature. Active texels are illustrated in green color coding and the curvature is computed from the ratio of the minor to the major axis (marked in red) of the blue ellipse.

For a circular contact area, the eigenvalues are approximately equal, hence the Eigenvalue ratio (EvR) is near one. For contacts with edges, leading to long and narrow contact areas, the EvR decreases approaching a singularity at zero. However, the non-zero spacial extension of the texels prevents the smaller eigenvalue from reaching zero. In principle, this approach to estimate the curvature of contact surfaces is limited by the tactile sensor resolution and the extent to which the sensor surface is compressible, which is typically a foam cellular plastic or rubber material.

Alternative Approach for Curvature Estimation

Estimating the curvature of the object shape at the tactile contact can additionally be done by estimating the distribution of sensor contact normal orientations *in object space* within a temporal window. This approach requires a continuous movement of the sensor surface along the object (i.e. maintaining contact) with a constant change in the finger orientation. This is achieved e.g. by continuous rotational movement of the object with the robot fingers. The second requirement is that the sensor surface must be curved, as is the case with the fingertips of the SDH2. The curved sensor surface enables several sensor measurements along a flat object surface, each having the same contact normal orientation in object space. In contrast, a flat sensor surface would lead to only one (very large) contact when sensor surface and object surface are

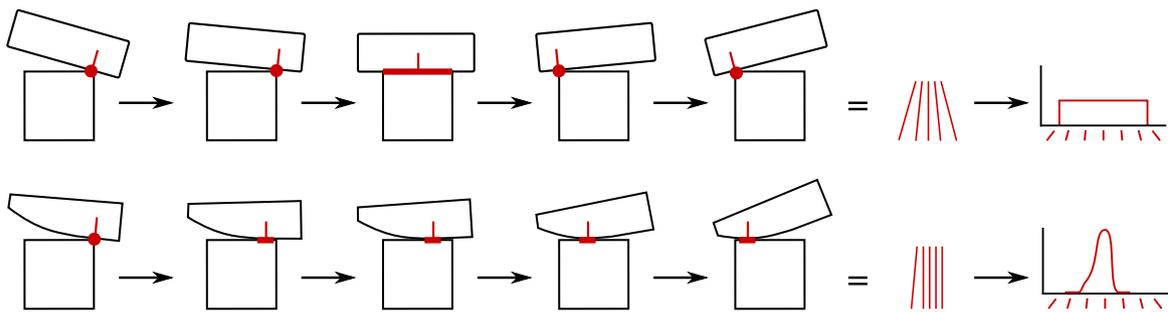


Figure 3.4.5: Contact orientation normal for flat (top) and curved sensors (bottom) touching a flat object surface. From the left to the right: first a sequence of tactile contacts of the sensor with an object surface is depicted, resulting in a set of measured contact normal orientations. The distribution of the contact normal orientations is sketched on the most right.

exactly collinear. This is sketched in Fig. 3.4.5, the top row shows the sensor contacts and their normals (in red) for a finger with a flat sensor surface. The bottom row shows the according information for fingers with curved sensor surfaces. On the right of the figure the contact orientation normals are plotted for the sequence of contacts, respectively. This illustrates the benefit of curved sensor surfaces, as rotating contacts with flat objects result in multiple contact normals with the same orientation. When the finger rotates along an object surface the contact position moves along the sensor surface, while the sensor normal orientation at the contact position remains stable. Hence, the mass of the normal orientation distribution in object space will accumulate at the according object surface orientation. In contrast, when rotating around object edges the contact position remains stable on the sensor surface, while the according sensor normal orientation changes. Thus, the orientation distribution will be broader as the robot finger performs the rotational movement, as depicted on the most right in Fig. 3.4.5.

In publications of previous versions of the model [STRUB et al. 2014b] and [STRUB et al. 2014a] this distribution estimate was explicitly used as a curvature feature. The advantage of estimating curvature from the orientation distribution is its robustness with respect to sensor noise, as only the contact position in sensor space has to be determined in order to know the surface normal for temporal integration. On the other hand, the delay in estimating contact curvature due to the temporal integration and the necessity of an approximately constant change of the sensor surface orientation are strong limitations. Furthermore, the evidence from biology suggest that curvature is directly computed from tactile neuron responses in the skin and no transformation into an object coordinate system with additional temporal integration is required (see Sec. 2.1 for further details). Therefore the curvature information from integrated contact normal orientations is not used as a direct input to the revised version of the model here. However, the principle of temporally integrating the normal orientations of detected surfaces is achieved within the model by a short term memory trace. Additionally, the temporal integration is also facilitated indirectly by the kernel parametrization of the surface detection DNFs, as will be described in Sec. 5.1.

3.5 Object Exploration

In our experimental setup a Shunk Dexterous Hand 2 is used and configured such that only two of the three fingers are used, each having two degrees of freedom (i.e. controlled joints). The two fingers each consist of a proximal and a distal phalanx, both equipped with a tactile sensor (see Sec. 3.2 for further details). Figure 3.5.1 shows the robotic setup used in the experiments (on the right side) as well as the manipulation behaviour used in these (left). For implementing the rotation behaviour, first the mapping from joint angles to a parametric space and then the generation of the behaviour based on these parameters will be described in the following two subsections.

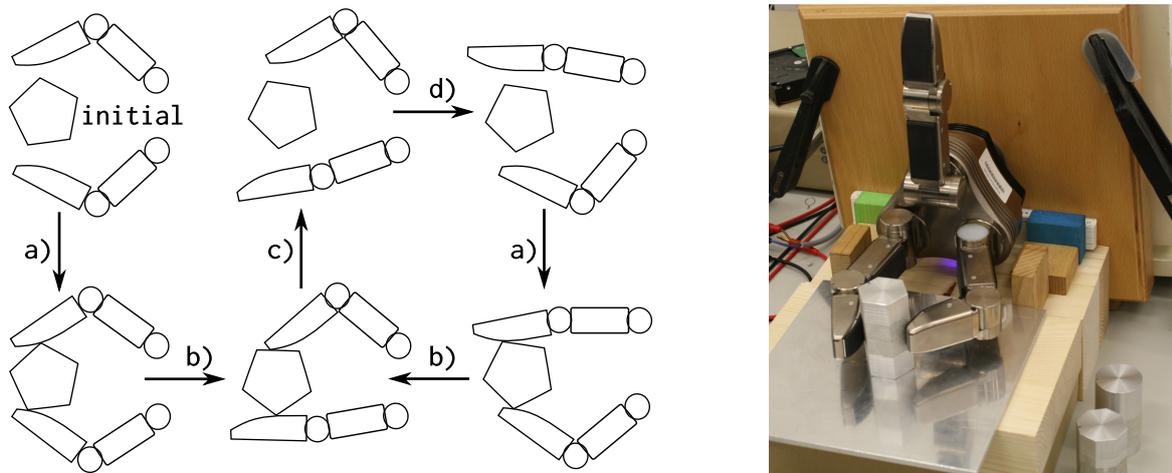


Figure 3.5.1: Left: sketch of the rotation behaviour starting from the initial pose. a) approach, b) rotate, c) release, d) reverse; Right: picture of the experimental setup

3.5.1 Rotational movements from kinesthetic teaching

Learning of Robot movements can be achieved by kinesthetic teaching, which is a form of “learning from demonstration” [ARGALL et al. 2009]. In this paradigm, the robot is manually brought into a desired configuration (usually configurations are sampled along a desired trajectory) and a ‘snapshot’ of the entire system, i.e. all the joint angles and values of other proprioceptive sensors is taken and stored. After performing several iterations of this sampling process, the collected snapshots serve as data points (“examples”) for a learning algorithm in order to extract underlying structures and generalize to new data points. The rotational movements of the SDH2 to rotate cylindrical objects supported by a table was learned by kinesthetic teaching combined with a Principal Component Analysis (PCA) on the retrieved data points (i.e. joint angles).

The two fingers of the SDH2 were manually moved such that an object rotation behaviour was achieved while recording the joint angles, as sketched in Fig. 3.5.1-left. For this four joints of the two used fingers are manipulated and the contact with the object is manually enforced. With this procedure, 55 data points consisting of four joint angles each were recorded, using n-gon objects with different diameters (see Sec. 3.1 for further details). For further processing, the mean position of each

joint in the data was calculated and subtracted in order to have mean-free data. Analysing the principal components of the data revealed, that all data points in the four dimensional joint space reside within a two dimensional subspace. The first of these two dimensions corresponds to the rotation of the grasp, i.e. opposing finger movements, and the other with the grasp diameter. The remaining two dimensions did not hold any major information and were subsequently dropped. Accordingly, the mapping between joint-space and parametric space was determined by Singular Value Decomposition (SVD):

$$M_{\text{joint}} = U \cdot S \cdot V^*,$$

where M_{joint} is the matrix of all the mean-corrected data points in joint space ($M_{\text{joint}} \in \mathbb{R}^{n \times 4}$), the orthogonal matrix U corresponds to the data points projected on the principal component vectors, S is a diagonal matrix holding the singular values (i.e. a scaling), the orthogonal matrix V is composed of the orthonormal principal component vectors of M_{joint} (i.e. PCA basis vectors) and V^* denotes the conjugate transpose of V . Thus, the mapping of parametric- to joint-space is given by:

$$M_{\text{joint}} \approx M_{\text{parametric}} \cdot S \cdot V^*,$$

where $V^* = V^T$ and $M_{\text{parametric}} \in \mathbb{R}^{n \times 2}$ holds the first two parameters (i.e. grasp angle and radius) of the according data point in M_{joint} . Since only the first two principal components are used, $(S \cdot V^*)$ is cropped to $\mathbb{R}^{2 \times 4}$ dimensions. At last, the mean of each joint from the recorded data has to be added to the according column of M_{joint} before it is sent to the SDH2 controller.

Likewise, the inverse mapping from joint-space to the parameter-space is given by:

$$M_{\text{parametric}} \approx M_{\text{joint}} \cdot V \cdot S^+,$$

where S^+ is the Moore-Penrose pseudoinverse of S and the matrix $(V \cdot S^+)$ is cropped to $\mathbb{R}^{4 \times 2}$ dimensions. The two dimensions of $M_{\text{parametric}}$ then correspond to grasp-angle and grasp-radius, respectively, and were used to directly control the SDH2 rotation behaviour during the experiments.

3.5.2 Generating a rotational behaviour

This section describes the rotation behaviour performed by the SDH2, which leads to the rotation of the n-gon objects in the experiments. In Fig. 3.5.1 on the left side the the different stages of the rotational behaviour are illustrated. The starting pose is a position where the two used fingers are positioned in a symmetric way with the grasp-angle parameter $\in [0.0, 1.0]$ is set to 0.5, as in step a) of the figure. The grasp-radius is set such that there is no contact to the object located in the centre between the two distal phalanges of the fingers. First, in the *approaching phase* depicted as a) on the left part of Fig. 3.5.1, the fingers are moved towards each other until tactile feedback signals sufficient contact with the object by continuously reducing the grasp-radius. The contact of both fingers triggers the *rotation phase* (b) in Fig. 3.5.1), were an opposing movement parallel to the object's surface of each finger while controlling the contact force with the object is performed. This finger movement is achieved by continuously increasing (or decreasing) the grasp-angle and controlling for the desired

contact pressure by adapting the grasp-radius accordingly. Thus the grasp rotation speed is constant in the joint angle space of the SDH2, but not in the actual angular rotation of the object for which the grasp-radius is also relevant. When one of the tactile contacts reaches the edge of a tactile sensor array (i.e. the edge of the corresponding phalanx), the *release phase* is entered (c) in Fig. 3.5.1). Here, the fingers are moved apart from each other by increasing the grasp-radius by a predefined value (here 2.0). When the desired grasp-radius is reached, the *reverse phase* resets the grasp-angle to a predefined value opposing to the step direction, i.e. if the grasp-angle is increased each step, the grasp-angle is reset to a value near zero and is set to a value near one if it is decreased each step (d) in Fig. 3.5.1). From there, the behaviour enters the approaching phase again and the cycle begins anew. The rotation behaviour was implemented via a simple C++ routine controlling the movement parameters grasp-angle and grasp-radius.

3.5.3 A Note on Translation

In the used setup, the induced object motion is primarily in one degree of freedom: rotation along its z-axis. Nevertheless, there is an initial error in the estimate of the object's position which the model needs to compensate for as well as small translations due to object motions from grasping and releasing the object with the robot SDH2.

The object translation in the lateral axis (i.e. between the two fingers) is necessarily constrained by the fingers, in particular when the object is grasped. Translations on the proximal-distal-axis (DPA) may become increasingly large for successive object release and re-grasping, as the rotation behaviour does not detect nor compensate for these. For too large translations along the DPA, the object eventually slips out of the grasp and the rotation behaviour stops. In order to prevent this, the translation along the DPA was mechanically constrained by limiting the movement capabilities of the aluminium plate (see Sec. 3.1) to an area where object rotation could be performed successfully. This was approximately a range of $\pm 1.0\text{cm}$ along the DPA. The rotation behaviour typically lead to a drift of the object position towards the proximal end of this working space (i.e. near to the palm).

3.6 The Kinematics of Object Manipulation

When the robot hand performs a manipulation of a grasped object, proprioceptive sensors (e.g. joint angles) can be combined with tactile sensory information in order to estimate the rotation and translation induced by this manipulation to the object, assuming that the robot kinematics is known. This estimation of changes in orientation and position of the object is done by a forward model, based on changes in joint angles and hand-object contact positions retrieved from the tactile sensors. For this, an initial estimation of the object position is needed. In the context of this work the initial grasp determines the starting orientation (i.e. zero) and position (i.e. the centre of the contact points). The assumption that the objects are convex implies that there is only one contact point for each finger with the object. Hence, there are two contact points at each considered point in time, one for each of the two used fingers. Before

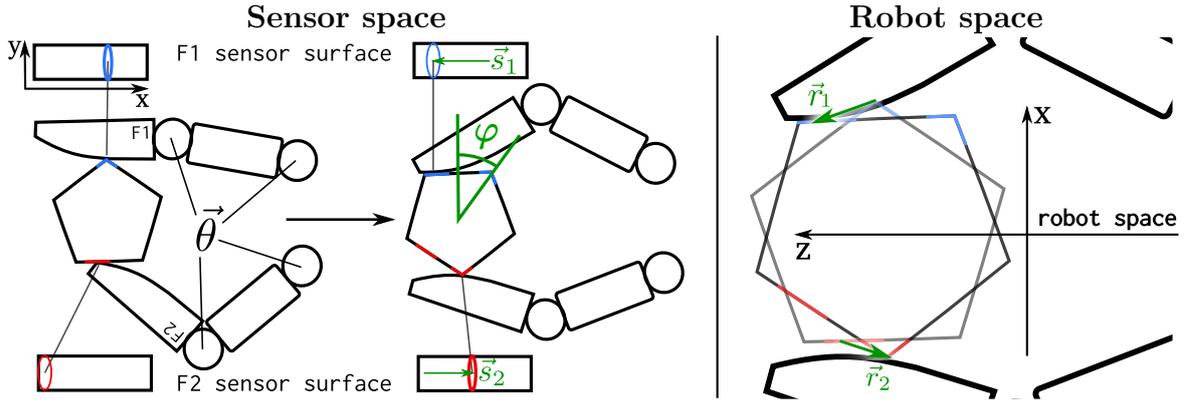


Figure 3.6.1: Left: Contact movement in sensor space and robot space from two snapshots in time during the rotation of an object with two fingers of the SDH2. Sensor space: In the Top row the tactile sensor surface of finger F1 with its corresponding pressure activation (blue ellipse) is shown, for finger F2 the same is shown in the bottom row (red). The joint angles are stored in θ , the object rotation is φ ; Robot space: the movement of contact points in robot space derived by transforming the contact points from the sensor space via the kinematics, for finger F1 (\vec{r}_1) and finger F2 (\vec{r}_2), respectively.

and after each small movement of the robot hand these two contact points are recorded and their change gives insight into the movement of the object.

First, the contact points in the sensory coordinate frame (sensor space) are transformed into a robot centred coordinate frame (robot space) using the known kinematics and the joint angle information. Second, the movement of contact points in two-dimensional sensor and robot space are combined to determine the rotation and translation of the object. In Fig. 3.6.1 the movement of contact points from two consecutive points in time is depicted for both fingers in sensor and robot space, respectively. In order to compute the rotation and translation of the object from these contact point movements, first the processing in robot space will be discussed, followed by including information from sensor space.

The contact movement vectors in robot space \vec{r} of both fingers are decomposed into a component tangential to the sensor surface and an orthogonal part. For computing the part of the movement vector which is tangential to the sensor surface (i.e. \approx object surface) several methods are feasible. Here the tangential part is computed via trigonometric functions starting with the elevation angle of the sensor surface normal at the contact point. In Fig. 3.6.2 the according angles are depicted.

The angle β_1 is computed by $\beta_1 = \text{atan2}(r_x, r_z)$, γ_1 is computed by $\gamma_1 = \frac{3}{4}\pi + \delta_1 - \beta_1$ and then r_{\parallel} is given by $r_{\parallel} = \sin \gamma_1 \|\vec{r}_1\|$. The orthogonal part of the movement vector \vec{r} is analogously given by $r_{\perp} = \cos \gamma_1 \|\vec{r}_1\|$ and contributes to the translational part of the object movement.

In order to compute the rotation of the object, the contact movement in sensor space has to be included. The distance each contact has moved on the sensor surface is computed from tactile data incorporating the known sensor geometry and is then added to the tangential part of the contact movement in robot space. This is necessary as rotating e.g. a cylindrical object does not necessary lead to a movement of contact points in robot space, although the contact points do move in sensory space. Thus,

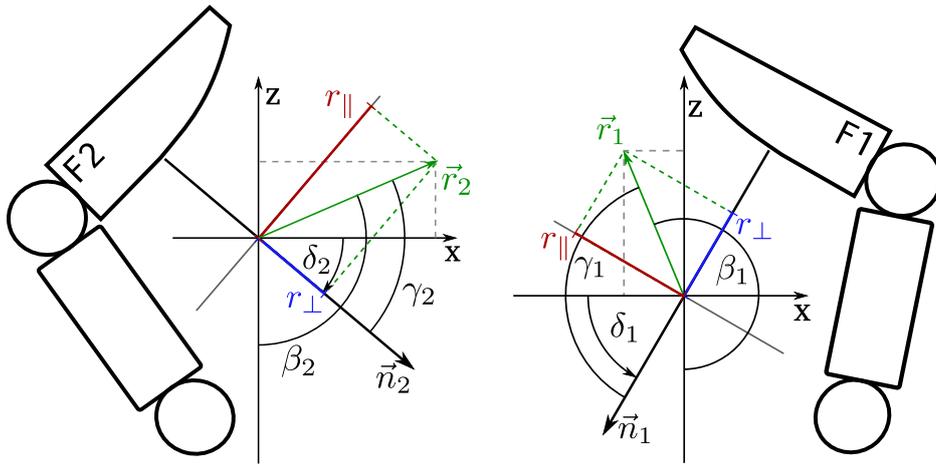


Figure 3.6.2: Computing the tangential parts of the movement vectors \vec{r}_1 and \vec{r}_2 . The normal vector \vec{n} is given by the angle δ , relative to the coordinate axis x . See text for further description.

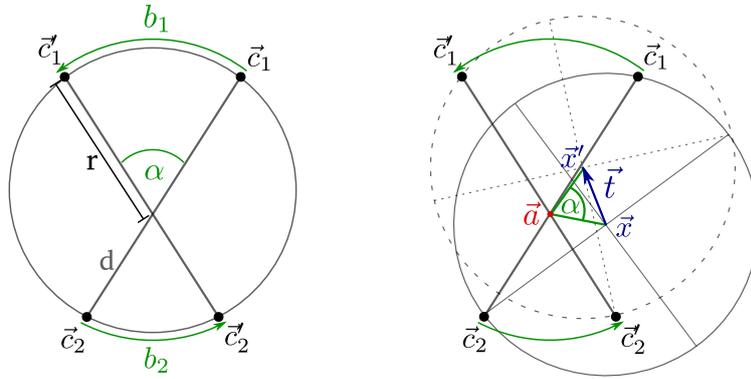


Figure 3.6.3: Left: sketch of the rotation induced by the movement of the contact points $\vec{c}_{1/2}$. The grasp diameter d is assumed to be constant throughout the movement, r is the radius by which the contact point of finger one is rotated by angle α . Right: sketch showing the translation \vec{t} induced by the rotation with the rotation centre given by \vec{a} , of the object centre \vec{x} .

finger movement and contact movement on the finger (i.e. sensor) might cancel each other in robot space.

The sum of the contact movements $b = s + r_{\parallel}$ is considered as a circular arc, where s is in sensor space and r_{\parallel} is in robot space. The underlying assumption is that the diameter of the grasp, i.e. the distance d between the contact point s is constant during small movements. On the left part of Fig. 3.6.3 the circular arcs b are shown for both fingers, respectively. The radius r of the rotation is computed by $r = d \left(\frac{b_1}{b_1 + b_2} \right)$. Then the rotation angle α is given by $\alpha = b_1/r$ (α is measured in radian).

Hence, the object is rotated by the angle α , which holds for small object manipulations and approximately cylindrical objects. Note, that the distance between \vec{c} and \vec{c}' is not b . The contact vectors \vec{c} and \vec{c}' may be identical in robot space, but through the contact movement in sensor space b can be non-zero.

The rotation of the object induces a translation of the object centre, as illustrated

on the right in Fig. 3.6.3. The translation vector \vec{t} is computed by first determining the centre of rotation: $\vec{a} = \vec{c}_1 + \frac{\vec{c}_2 - \vec{c}_1}{d} \cdot r$. Then the object centre \vec{x} is rotated by the angle α around the rotation centre \vec{a} , such that the new object centre is computed by $\vec{x}' = R(\alpha)(\vec{x} - \vec{a}) + \vec{a}$, where $R(\alpha)$ is the two dimensional rotation matrix and \vec{x}' is the translated object centre. Thus, the total object translation is composed of the orthogonal parts of the contact movement vectors (orthogonal to the sensor surfaces) of both fingers $\vec{n}_1 \cdot r_{1\perp} + \vec{n}_2 \cdot r_{2\perp}$ added to the translation vector resulting from the object rotation \vec{t} .

Note, that in order to incorporate the translation due to rotation of the object, its current centre has to be known. This is in contrast to the contributions from contact movements orthogonal to the sensor surface, as these indicate a change of object pose independent of the current pose.

Altogether, provided that the forward model has access to the current position estimate, it can predict the change of the object orientation and position from changes in joint angles and tactile sensory information. However, these predicted object pose changes are error prone, due to slippage of the object, movements while only one of the fingers has contacts (especially when object is grasped / released) and accumulating and potentially magnifying past errors. Therefore, an additional mechanisms of detecting and correcting the pose estimate of the object is required.

3.7 Recording Datasets

For testing and evaluating the proposed model, datasets were recorded while the robot SDH2 performed the rotation behaviour described in the previous section Sec. 3.5. Five datasets were recorded from each of the objects: the six and eight sided objects from the small set and the large set, and the two asymmetric objects. The round cylindrical object was only used for training the rotational behaviour. Each dataset consists of recordings during an estimated total object rotation of four times 360 degrees. Hence, 30 datasets were collected, in which the tactile patterns and joint angles were sampled with approximately 2-3 Hz and the features described in Sec. 3.4 were computed and stored. All datasets consist of 1020 up to 1500 samples. The objects were mounted on the plate as described in Sec. 3.1 to prevent them from tipping. Object translations along the DPA were manually constrained in the experimental setup to a maximum of ± 1 cm, i.e. object movements of a maximum of 25% of its diameter are possible during the rotations. Due to slight differences in the realization of the translation limitation the translational movements for the large six and eight sided object was larger than for the objects with small diameter and the asymmetric objects. This is visible when comparing the raw data in the appendix section Fig. B.0.23 and the following figures.

An exemplary subset of two rotations is visualized in Fig. 3.7.1 from the first dataset of the large six sided object, where finger one corresponds to the upper finger of the sketch in Fig. 3.5.1 in Sec. 3.5.

It is clearly visible that there is a drift in the object rotation estimate as the data points do not align for consecutive full rotations of the object (note the temporal color-coding in Fig. 3.7.1). Furthermore the impact of the error in the position estimate is visible in the “bump-like” shifts between two consecutive surfaces. These distortions of the object corners are a systematic effect resulting from the rotation of the contact

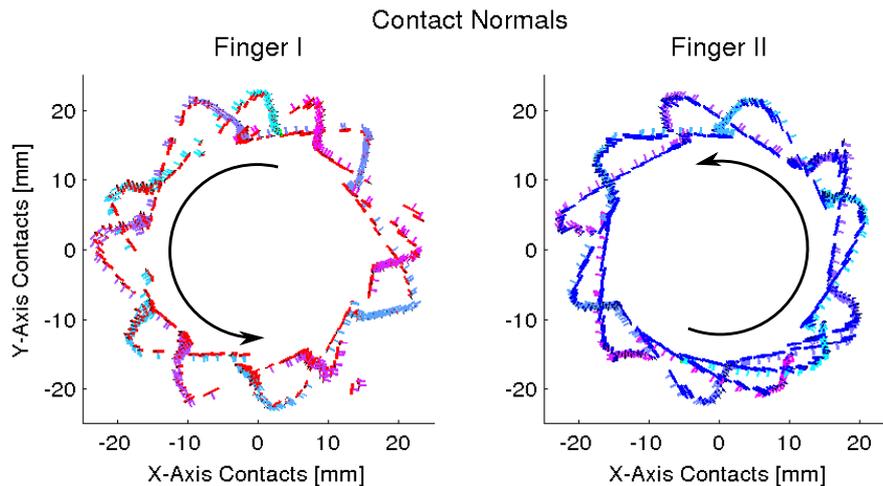


Figure 3.7.1: Raw, uncorrected data of the last two full rotations for a six-sided object. The position and orientation of the tactile contacts in the object coordinate system is shown. The color-code shows the temporal order from start (teal) to end (pink) of the two rotations, the object rotation direction is indicated by the arrows.

points with respect to a wrong estimate of the object centre when transforming these from the robot to the object space. An illustration of this effect is given in Fig. 3.7.2, showing the data for finger one. The errors in the object rotation have been compensated by the proposed model which will be introduced in the following subsections. Here, the effects of errors only in the position estimate is shown, dependent on the direction of deviation. The blue object estimate is further away from the palm while the red one is closer to it. Note, that the the wrong position estimate in Fig. 3.7.2 is fixed during the object rotation, while normally the wrong position estimate would be rotated around the rotation centre for each object rotation step. Due to the assumption of convex objects and practical constraints of the SDH2, the object centre along the lateral axis (roughly orthogonal to the finger surfaces) is limited to positions approximately in the middle of the contact points.

Therefore, the position estimate will always converge to the centre of the object rotation, as the object is continuously rotated. In order to prevent this degeneration, the translation estimate resulting from the object rotation of the forward model is ignored. Thus, the forward model only updates the orientation estimate of the object. The position estimate is only corrected by the model.

Nevertheless, if the objects true position deviates from the estimated position, the characteristic distortions showed in Fig. 3.7.2 in object space will occur, depending on the direction of the error. For an position error closer to the palm, the edges of the object tend to drift towards the object centre, while for an position estimate error further away from the palm, the edges drift away from the object centre.

In the appendix the raw data of all datasets are shown in figures Fig. B.0.23 and Fig. B.0.24. It can be seen that the amount of accumulated orientation estimate errors during rotation has large variations across the datasets. For evaluation the datasets were looped, i.e. after the last sample of a dataset was represented to the model, the dataset was repeatedly represented, starting with the first sample. Note, that there is

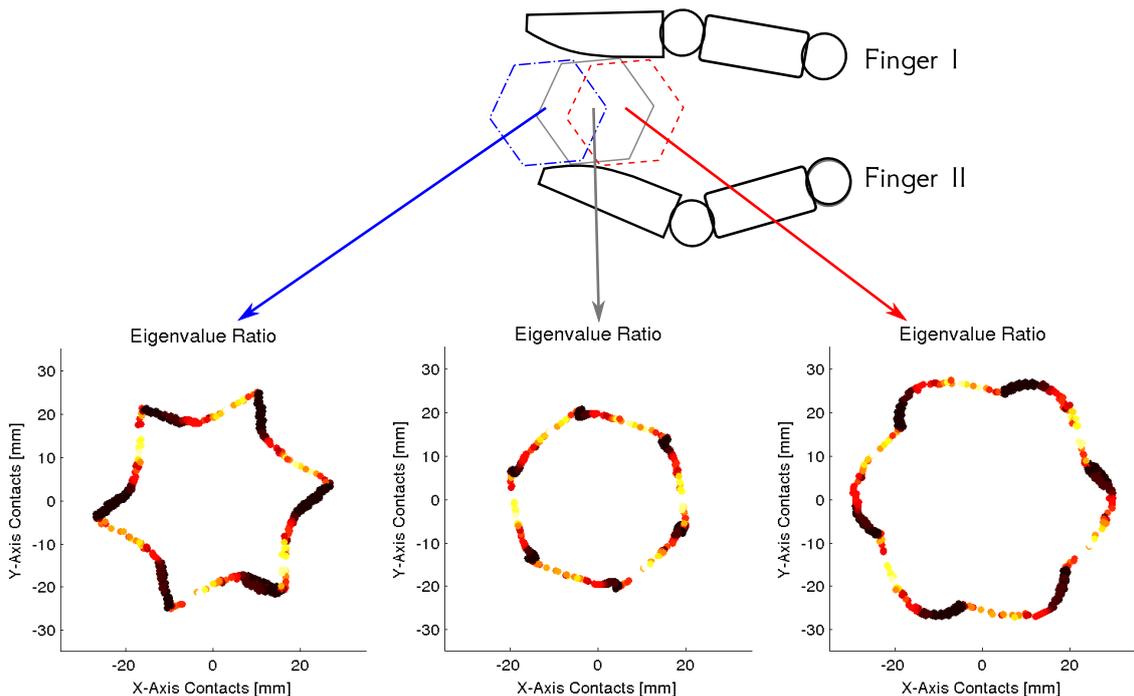


Figure 3.7.2: Impact of errors in the object position estimate. The middle object between the robot fingers (depicted in grey) is the true position, the red and blue objects are two error prone position estimates. The bottom row of plots shows the according contact points in object-space (data from finger one), accumulated during one full rotation of the object with a clamped wrong position estimate, but corrections for errors in the rotation estimate. The color of the contact points encodes the EvR: black denotes a low value, i.e. edge; bright values a high value, i.e. surface.

no smooth transition between the loops and thus this re-representation of the dataset implies a re-localization of the object. The consequences of this effect will be discussed in the results chapter of the model, Chp. 6.

3.8 Post-processing of the Curvature Feature

The curvature feature based on the EvR is postprocessed with a non-linear normalization. The non-linearity is chosen such that it amplifies the distance of the “surface values”, i.e. high values, from the “edge values”, i.e. low values. The curvature of the sensor surface increases the number of sensor measurements of flat object surfaces but nevertheless, the majority of all sensor contact measurements result from contacts with object edges. A method termed intrinsic plasticity (IP, introduced in Sec. 2.5) is chosen as a normalization method in order to incorporate this observation. Here the “surfacedness” is the information passed by the abstract neuron (i.e. a non-linear function), which is then maximised with IP by setting an exponential distribution of surfacedness responses as a desired output distribution. This implies the assumption of frequent low amplitude EvRs (i.e. edges) and infrequent high EvRs (i.e. surfaces). The quantitative impact of IP on the distribution of eigenvalue ratios is shown in Fig. 3.8.1. It is visible that the EvRs computed from the tactile sensor information

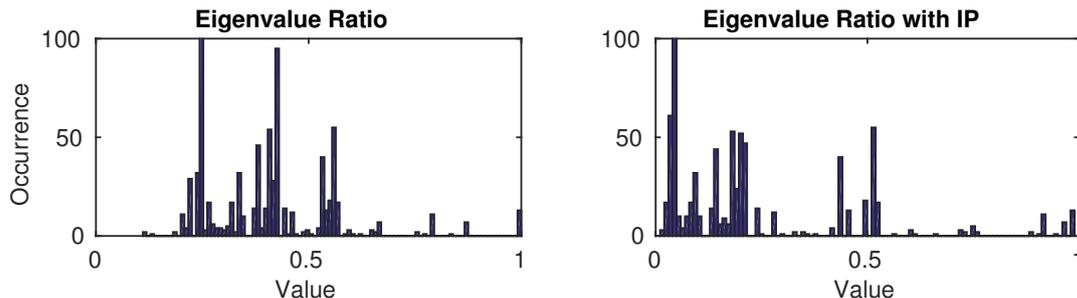


Figure 3.8.1: Impact of IP: example of the distribution of the eigenvalue ratio before and after the application of IP. The data is from dataset #10, finger one of the SDH2 - see Sec. 3.7 for details.

do not fill the range $[0, 1]$ but are clustered around the values 0.25, 0.45 and 0.55. Normalization with IP results in a shift of the first two clusters towards zero and a separation from the last cluster.

A qualitative analysis of the normalization with IP is shown in Fig. 3.8.2. Here the curvature (EvR) is visualized according to its orientation in object space over time, dark color indicating flat surfaces (EvR near one). In the first row the EvR computed from Eq. (3.4.1) is shown, in the second the EvR normalized with IP and a desired output mean of $\mu = 0.2$. The top plot shows that there are regularly high EvR measurements with similar orientation (the shallow ends of the “concatenated sigmoids”). This corresponds to the effect of accumulating tactile contacts with similar normal orientations when rotating along object surfaces, i.e. high EvR.

In the second plot, it is clearly visible that the distinction of surfaces is increased by the normalization with IP and the remaining feature values mostly cluster into small groups. These clusters show the error in the object pose estimate, as successive clusters do not align in their orientation value but drift over time.

For the evaluation of the model the curvature feature (EvR) was normalized online with IP (Eq. (2.5.3) in Sec. 2.5). The initial IP parameters were set to the mean batch solution ([NEUMANN & STEIL 2011]) over the datasets. The optimal IP parameters retrieved from the batch solution are shown for each dataset of the large six and eight sided object in Fig. 3.8.3.

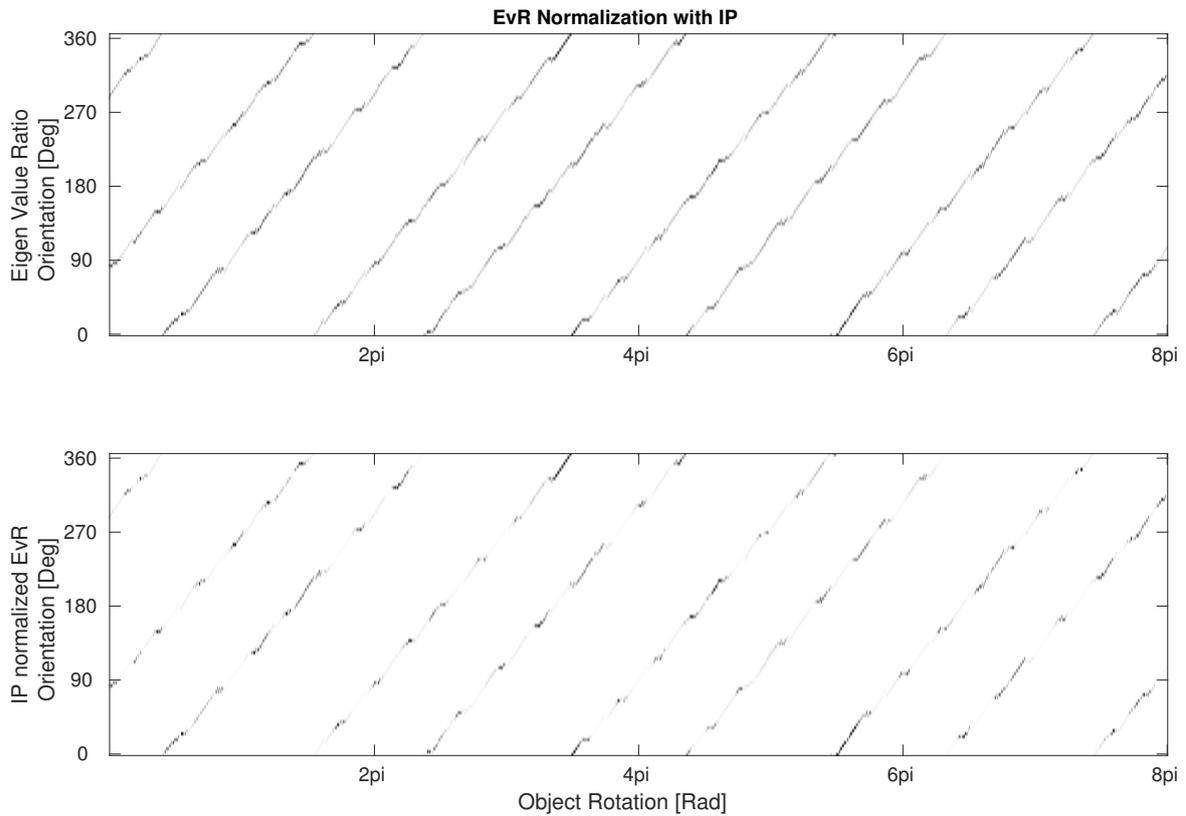


Figure 3.8.2: Surface detection in Dataset # 10 (large six sided object). Top: The ratio of the two eigenvalues (EvR) of the contact area. High (dark) values indicate a surface, low values an edge. Bottom: The EvR normalized via intrinsic plasticity (IP). See text for discussion.

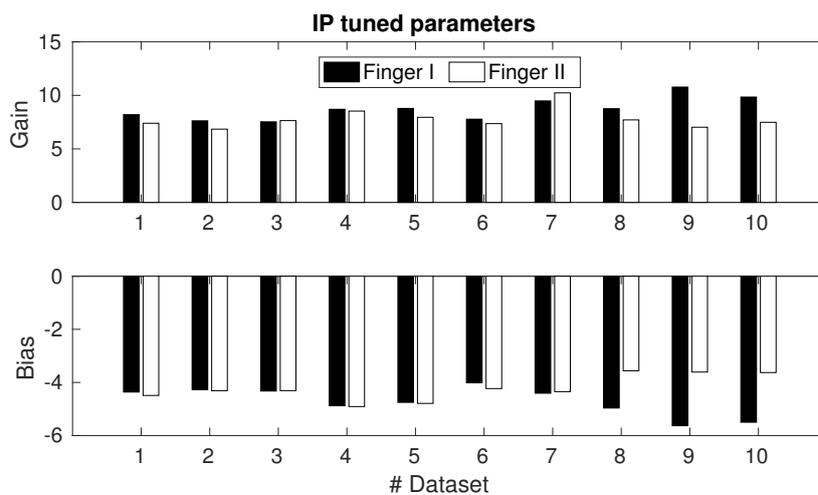


Figure 3.8.3: IP parameters from batch learning: the IP gain and bias for the recorded datasets of the large six and eight sided objects.

3.9 Implementation

The implementation of the robot SDH2 rotation behaviour (Sec. 3.5) together with the feature extraction in Sec. 3.4 were implemented in C++ libraries. The interface between the cedar software framework (Sec. 2.4) and the C++ libraries was implemented via a cedar plug-in. The neurodynamic model, including all the transformations between object and robot space was implemented using cedar. This implementation is capable of building the object shape representation from previously recorded data from rotational object manipulations (Sec. 3.7), just as processing haptic data from real-time robot object manipulations. All DNFs in cedar have an additive activation noise to the state $u(x, t)$ with zero mean and a sigma of σ_{noise} of 0.01. During the processing of haptic data, i.e. spatio-temporal integration of haptic sensory information, the activation of relevant DNFs and MTs is recorded and stored on the hard drive.

For evaluation of the recorded activations in the model while processing haptic data is then done in Matlab, e.g. all the plots in Sec. 6.

3.10 Challenges of the Setup

In this concluding section of the experimental setup chapter some remarks on the challenges and problems of the used setup shall be mentioned. Due to only having two joints per finger, there are not only forces orthogonal to the object surface, but also tangential components. These lead to an uncontrolled movement when the object is released, which cannot be detected nor prevented in the proposed setup. Together with the unintended slight shift and rotation of the object when the grasp is closed again, these are the main sources of noise in the pose estimate. In general, these unintended movements should be systematic and indeed, there is a strong tendency of systematically underestimating the object rotation. Errors in the translation estimate are partly due to not perfectly centred objects during the initial grasps and partly due to a systematic shift of the objects towards the proximal side of the robot palm. This leads to a systematic offset between the true object position and the position estimate, as the object has only minor translations as soon as the object plate reaches the maximally possible translation during the manipulation (limited by mechanical constraints).

Furthermore the tangential shear forces lead to “ghost contacts”, activation of tactile sensor elements due to changes in conductivity of the material covering the sensor surface. The conductivity of the cellular plastic is a function of the deformation of the material, which is usually due to mechanical compression from contact with the environment. However, as the shear forces increase the tactile sensor reports pressure in areas with no mechanical contact to the environment (i.e. the object). This may lead to a deformation of the sensed pressure patterns, e.g. contact with an object edge may lead to a circular pressure pattern, or even to the formation pressure areas distinct from the area of contact. In the former case the EvR will be wrong, reporting contact to flat surfaces while there is actually only a very sharp contact to an object edge. This happens occasionally and is a source for false detection of surfaces (especially in the datasets #4, #5 and #24). In the latter case, two contact areas will be reported by the tactile sensor, while there is only mechanical contact at one location, requiring

to select one of the pressure areas for further processing. In this case the heuristic to select the larger contact area was chosen.

Chapter 4

Dynamic Neural Fields with Intrinsic Plasticity

In this section a novel mechanism for adaptation of a DNF with IP (described in Sec. 2.5) is described and discussed. The proposed adaptation is based on a coupled gain and bias for the entire DNF. This is motivated from accumulating biological evidence of neuronal mechanisms that perform a multiplicative normalization of entire populations of neurons, for a review see [CARANDINI & HEEGER 2012]. The existence of global, network-wide activity regulation in addition to single neuron and synaptic adaptations is also proposed in [SLOMOWITZ et al. 2015] based on recordings of cultured hippocampal networks. In particular, the existence of a coupled gain and bias among neuronal populations has been proposed as an explanation for results from large-scale recordings in the primary visual cortex (V1) [LIN et al. 2015].

In this thesis, the adaptation of a coupled gain and bias for the population of neurons in a DNF is proposed as a homeostatic mechanism to control the activation of the field. Qualitatively speaking, a global sensitivity and resting level for all neurons of a population is autonomously adapted in order to regulate the overall activity of the population. This adaptation ensures that the information encoding in the DNF remains stable for changes in the input mean or amplitude, which will be relevant in the model proposed in the next chapter Ch. 5. While this regulation is supported by research from computational neuroscience, the coupled adaptation also ensures that the encoding of the input remains stable. In population coding, the value of a feature is encoded by the activity of neurons in the population which represent this value. If there are any feature values which are represented in the neuronal population but do not actually occur (frequently), the according neurons will only sporadically be active. If the gain and bias of each neuron in the population would be adapted individually, every neuron would be tuned such that it achieves the desired target distribution. This would have a destructive impact on the information encoding, as the input weights of the neurons are not altered and thus the tuning curve of the neuron cannot be changed. For a neuron representing feature values which are not required, i.e. do not occur in the input, the gain parameter of the neuron would be adapted by IP in order to amplify noise, e.g. intrinsic or background noise. Therefore, the population output would converge to an uniform distribution of feature values in the population code, independent of the input distribution – i.e. destroying the encoding.

In the following subsections the derivation of DNF with IP will be outlined and then evaluated.

4.1 Derivation

For implementation of IP in a DNF the field equation needs to be slightly reformulated. The standard formula of a DNF is given in Eq. (2.4.1) in Sec. 2.4:

$$\tau \dot{u}(x, t) = -u(x, t) + h + S(x, t) + \int f(u(x', t)) \omega(|x - x'|) dx'.$$

Here, $u(x, t)$ is the activation at time t and position x , h is the negative resting level, $S(x, t)$ is the input and $f(\cdot)$ is the logistic function. The lateral interaction is determined by the convolution with a kernel $\omega(\cdot)$.

Here, the logistic transfer function $f(\cdot)$ is used in the parametric version:

$$f_{a,b}(x) = \frac{1}{1 + \exp(-ax + b)} \quad (4.1.1)$$

where a is the gain of the function controlling the steepness of the sigmoid and b is the bias controlling the position of the sigmoid. The bias corresponds to a gain dependent resting level $b = ah$ which replaces the former resting level h in Eq. (2.4.1).

Furthermore, three design choices have to be made for deriving the IP learning rules:

- (I) Define a scalar measure z for the input of the field.
- (II) Define a scalar measure y for the output of the field.
- (III) Chose the desired target output distribution.

Concerning the first two points, the *output measure of the field* $y(t)$ is defined to be the maximum output of the neural field

$$y(t) = \max_x \left(f_{a,b}(u(x, t)) \right). \quad (4.1.2)$$

and accordingly, the *input measure to the field* $z(t)$ is then given by the field activation at the position of the maximum output

$$z(t) = u \left(\arg \max_x \left(f_{a,b}(u(x, t)) \right), t \right). \quad (4.1.3)$$

Hence, the input for IP is a composition of the actual field input and lateral field interactions, i.e. recurrent components. The main advantage of this measure is that it does not alter the output range, e.g. if the field output activity is in the range of $(0, 1)^n$, the $\max(\cdot)$ is too. This prevents an additional normalization / parameter tuning.

Two alternative definitions would be the integrated (i.e. summed) or the mean of the field output activity. In contrast to the maximum, these are sensitive to the

field size with respect to the peak size, i.e. the lateral interaction kernel width and strength. The mean activity of the field strongly depends on the field size, while the integrated activity is dependent on the kernel width. Hence, both of these alternative measures require a tuning of the target distribution parameters with respect to the particular DNF parametrization and are therefore neglected.

The *target output distribution* of y is set to the exponential distribution with the mean μ set to 0.2, implying a sparseness constraint on the field output (see Sec. 2.5 for details). The exponential distribution is particularly suitable as the DNF output is desired to be near zero for the majority of inputs (i.e. most of the time) and output activity is only required for matched features, as will be described in Sec. 5.3. With these design choices, the learning rules for adapting the gain a and bias b are given by

$$\frac{\Delta a}{\Delta t} = \frac{\eta}{a} + \frac{\Delta b}{\Delta t} z(t) \quad (4.1.4)$$

$$\frac{\Delta b}{\Delta t} = \eta \left(1 - \left(2 + \frac{1}{\mu} \right) y(t) + \frac{1}{\mu} y(t)^2 \right) \quad (4.1.5)$$

derived in [TRIESCH 2005] (see Sec. 2.5 for further details). The learn rate η is set to 0.001.

To prevent IP from overestimating the variance of the input signal, the gradient direction of Δa and Δb is corrected by the metric tensor imposing a Riemannian structure in parameter space, i.e. the *natural gradient* is computed as described in [NEUMANN et al. 2013] (see Sec.2.6 for further details). The tensor decay parameter λ in Eq. (2.6.3) is set to $(1/1000)\tau$ where τ is the time constant of the DNF equation in Eq. (2.4.1). The regularization parameter ϵ in Eq. (2.6.1) of the tensor inversion is set to 0.0001.

4.2 Evaluation

It is expected that introducing IP to a DNF leads to an autonomous adaptation of the resting level and input gain such that the information encoding in the DNF is an optimal representation of the input, i.e. a maximum of input information with a minimum of DNF activity. For evaluating the DNF with IP, an input time series was constructed from the recorded datasets (see Sec. 3) in order to mimic real tactile data. A one dimensional population code is generated from the two features contact normal orientation and eigenvalue ratio (EvR), as depicted in Fig. 4.2.1. At each recorded time step the EvR is the amplitude at the position of the contact normal orientation for both fingers, respectively. The EvR is smoothed with a Gaussian filter across the contact normal orientation dimension. The contact normal orientation is then sampled with $360^\circ/100$ degree resolution, leading to a 100 dimensional time series. Finally, the time series was looped in order to present it for arbitrarily long periods. Filtering along the temporal dimension is not necessary, as the features were sampled at a sufficient rate to ensure smooth transitions. This time series is fed into a one dimensional DNF with IP, sampled at 100 points (i.e. a size of [1,100]). The setup is run with presenting the input time series based on recorded data in realtime (3fps) and the DNF with IP has a τ of 100ms and is updated with an Euler step width of 10ms.

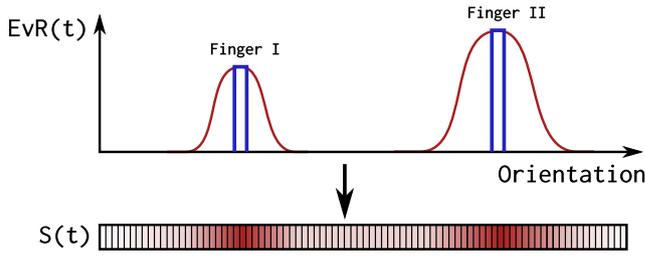


Figure 4.2.1: Sketch of the input encoding used for evaluation of DNFs with IP. Top: the original EvRs (blue) and the Gaussian filter across the normal orientation (red). Bottom: sampling the high dimensional vector for the input time series $S(t)$.

With this setup the following cases are evaluated:

- (I) input with very low amplitude
- (II) input with very high amplitude
- (III) input with very high bias

In all cases the DNF with IP is run until the parameter adaptation by IP did not change qualitatively, i.e. converged. The results of this evaluation are shown in the following figures (e.g. Fig. 4.2.2).

These figures show the input distribution (A) and the final output distribution (B) (i.e. after the learning experiment was stopped). Furthermore, the output distribution is plotted (y axis) over time (x axis) with a 5min sliding time window to estimate the distribution in (C). The size of the time window was adjusted such that it contained one full input period (approximately 5:15min with 3.33fps) representative for the input distribution. For an enhanced visualization the output distribution over time is plotted on a logarithmic color scale in (D). The development of the gain over time is plotted in (E), and the bias in (F). The correlation of the input with the output of the DNF is shown in (G), computed for a sliding time window located at t :

$$\text{corr}(t) = \frac{\sum_{t_i=t-l}^t (z(t_i) - \bar{z})(y(t_i) - \bar{y})}{\sqrt{\sum_{t_i=t-l}^t (z(t_i) - \bar{z})^2 \sum_{t_i=t-l}^t (y(t_i) - \bar{y})^2}} \quad (4.2.1)$$

$$\bar{z} = \frac{1}{l} \sum_{t_i=t-l}^t z(t) \quad (4.2.2)$$

The length of the time window l is set to 5min, just as for the computation of the sliding output distribution plots mentioned above. In the following the results of the evaluation with different input amplitudes and initial bias settings are presented.

4.2.1 Low Amplitude

The results of the test with input of low amplitude between $[0, 0.2]$ are shown in Fig. 4.2.2. The DNF is initialized with a resting level (i.e. bias) of -5 and a gain of 1. The output distribution plots in the middle (C) and bottom (D) rows of the left column show that initially the DNF has no output activity as the activity induced by

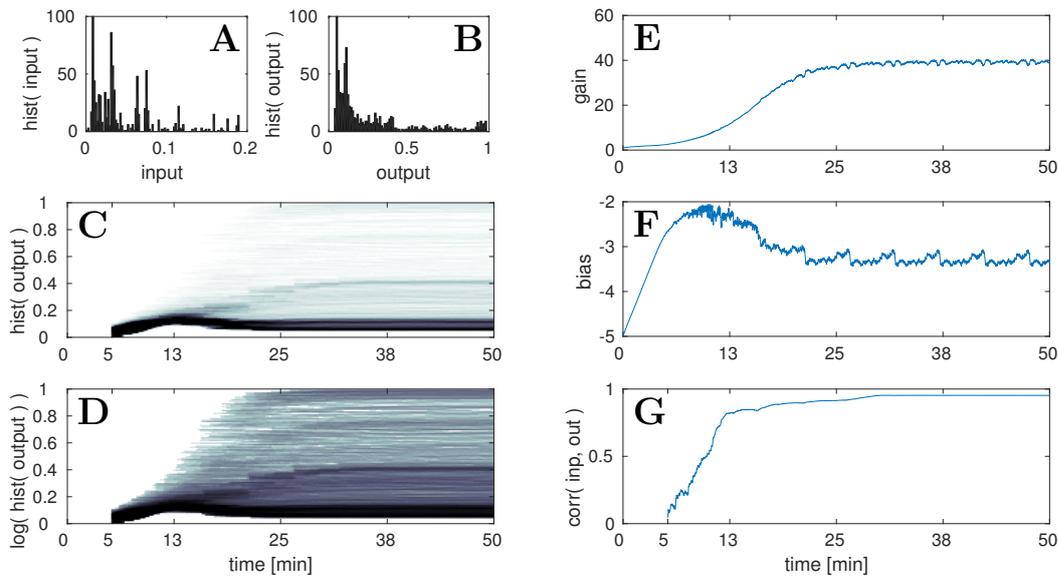


Figure 4.2.2: IP in a DNF with **low amplitude** input, see text for description.

the input is too low. The gain (E) is continuously increased by IP until it converges around the value 38 (see the top right plot in Fig. 4.2.2). Simultaneously the resting level (F) (i.e. bias) is shifted towards -2 and is corrected to -3 when the gain amplifies the signal strong enough to induce some output activity around the 13th minute, visible in the left (C) and right (F) plots of the middle row. The 13th minute is also the point, where the input-output correlation (G) reaches nearly one, as input activity now influences the output of the DNF, shown in the bottom right plot of Fig. 4.2.2. The gain and bias converge approximately in the 30th minute, where the input-output correlation reaches its final maximum. Note, that the correlation is near one as the output with the maximal amplitude is chosen for each step in time, making it very likely to use input output pairs in the near-to linear regime of the transfer function. The final output distribution in the second plot of the top row (B) is similar to the exponential distribution.

4.2.2 High Amplitude

In Fig. 4.2.3 the impact of IP for inputs with very high amplitude $[0, 40]$ is shown. The DNF is again initialized with a bias of -5 and a gain of 1. In the output distribution plots (C-D) clearly show that the DNF output initially is in the saturated regime of the transfer function. As gain (E) and bias (F) converge to the final values at around the 30th minute the output distribution is changed towards an exponential distribution, visible in (C) and (D). The correlation (G) continuously increases with a final boost just before the IP parameters converge.

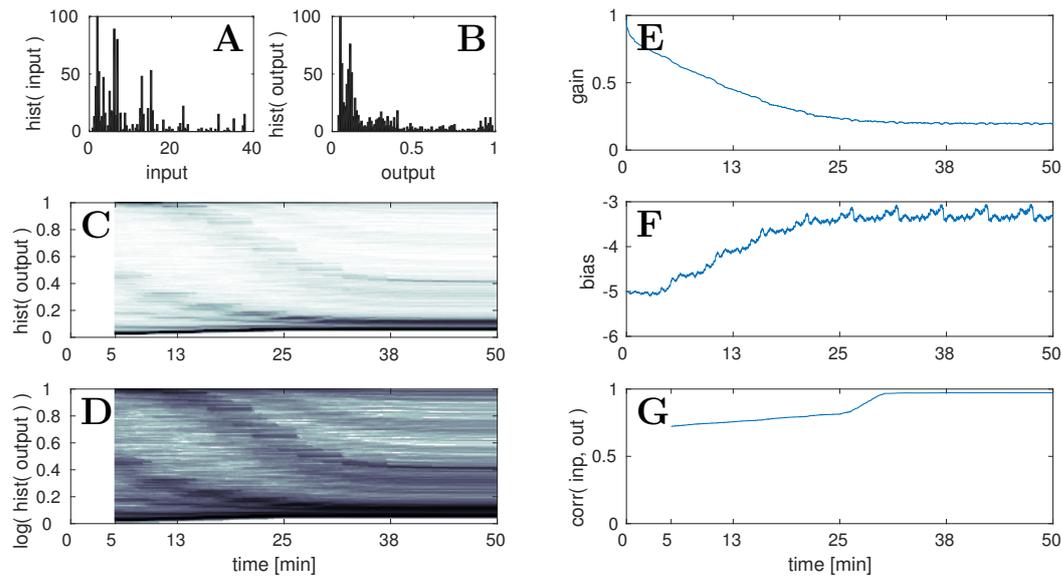


Figure 4.2.3: IP in a DNF with **high amplitude** input, see text for description.

4.2.3 High Shift

In this subsection the evaluation of IP for inputs with moderate amplitudes but a high offset (i.e. shift) to the range of $[-25, -15]$ is shown. The DNF is initialized as above with a bias of -5 and a gain of 1 . Additional to the procedure in the previous two subsections the impact of the natural gradient is analysed in Fig. 4.2.4. In the following, first the IP adaptation with the natural gradient (NG) will be described as before and then the case of adapting IP without the NG is compared. In the top three rows of the left column in Fig. 4.2.4 the input (A) and output distributions (B-D) are shown, analogous to the previous sections. While the output of the DNF is zero until ca. the 29th minute, the gain is only initially decreased, visible in the blue gain plot (I) in the 4th row of Fig. 4.2.4. The decrease of the gain for inputs with high bias (i.e. shifts) is a variance overestimation problem of the IP algorithm [NEUMANN et al. 2013]. The input variance, i.e. the deviation of the input signal from zero, can be reduced by lowering the gain, thereby reducing the error of the output distribution with respect to the target exponential distribution. The standard gradient descent of IP learning therefore drastically lowers the gain in order to increase it again when the bias has been adapted such that the input mean is compensated, visible in the orange graph in (I). Here the computation of the natural gradient, i.e. the transformation of the gradient from the Euclidean space into the Riemannian space prevents the reduction of the gain to nearly zero and only leads to a slight input variance overestimation, visible in the small increase of the gain around the 35th minute, shown in the glue graph in (I). At this point in time the bias reaches a regime where the input produces output activity in the DNF and the gain starts to converge to the final value, reached at approximately the 40th minute (J).

In this setup of inputs with a high bias, IP without the natural gradient correction leads to an “overestimation” of the signal variance. This becomes apparent in the

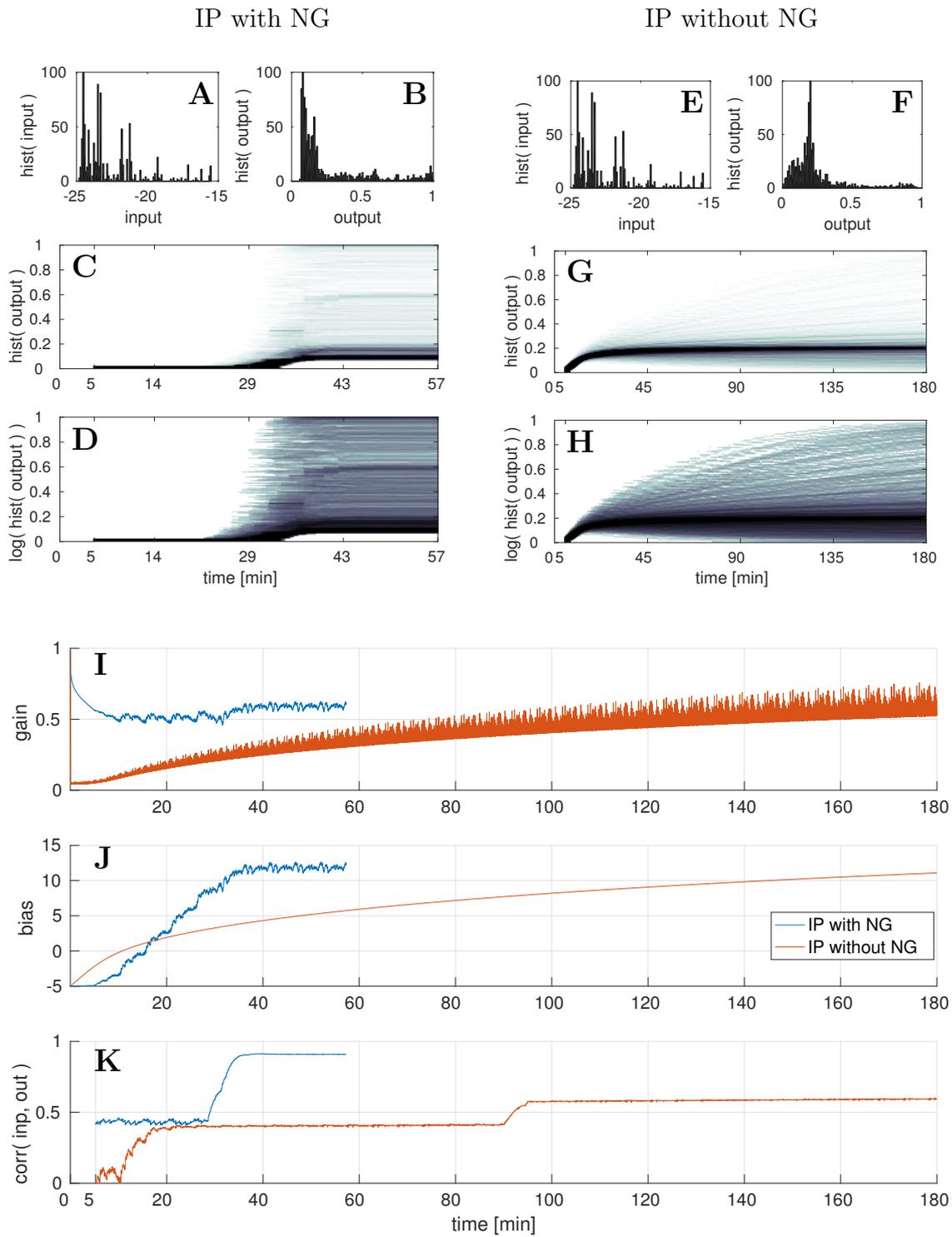


Figure 4.2.4: IP in a DNF with shifted input. Top, left three rows show input (A) and output (B-D) distribution for IP with NG. The top three rows on the right (E-H) show the same plot for IP without NG. The bottom three rows show the parameter adaptation (I-J) and input-output correlation (K) for both procedures. See text for further description.

bottom three rows (I-K) in Fig. 4.2.4. In contrast to the case with NG the gain is decreased near to zero and only slowly converges back to a higher value as the bias increases. Although the two learning algorithms have the same learning rate of $\eta = 0.001$, the learning with NG is much faster. Furthermore, it is noticeable that the input output correlation without the NG learning is much lower (K). This is due to the still relative high learning rate for IP which also leads to the high gain fluctuations visible in (I) in Fig. 4.2.4.

Altogether, the use of the NG leads to a significantly faster convergence with less fluctuations in the parameter adaptation.

4.3 Discussion

In this subsection the results of applying intrinsic plasticity into a DNF is discussed. Defining scalar measures of the global field input and output enables a procedure for global adaptation of DNF parameters via IP, analogues to IP in single neuron models. Altogether, the results show that the proposed implementation of IP in DNFs works as expected, leading to an autonomous gain and bias (i.e. resting level) adaptation.

In the case of deprived input, i.e. no input at all, the gain will increase continuously until system noise (if present) is sufficiently amplified in order to cause output activity. As the DNF dynamics reacts inertial to input, white system noise causes outputs in DNFs only for very high noise amplitudes. Adaptation of the gain in the absence of input thus leads to increasingly high gain values (results not shown).

Choosing to use the maximum output activity of the DNF as field output for IP makes the adaptation of gain and bias robust with respect to the question of whether multiple simultaneous output peaks are desired or not. The possibility of multiple peaks remains a question of the kernel parametrization (lateral interaction) and a matter of global inhibition. In contrast, choosing the total output activity of a DNF as field output would make the output distribution more sensitive to the simultaneous occurrence of multiple peaks.

Concerning the impact of IP on the stability of the DNF dynamics, it should be noted that IP drives the dynamics towards the detection instability, i.e. to the edge of stability. There the distribution of the defined output measure is closest to the exponential distribution, when the DNF only reaches the detection instability for a subset of input amplitudes. When there is no input at all, IP will lead to the destruction of the stable fixed point attractor in which the system state currently is. This in turn leads to a oscillation between the two stable fixed point attractors, i.e. drive the system permanently through the detection instability. Hence, IP prevents the DNF from operating in a self-stabilizing regime where the recurrent interaction is sufficient for self excitation in the absence of input. This also holds for DNFs with parameters initialized in this self-stabilizing regime (results not shown). Therefore, IP enforces the DNF to operate in a regime where the system state regularly runs through the detection instability. The overall dynamics may be shaped by choosing the target output distribution for IP. For example, the Kumaraswamy's double bounded distribution with $a + b = 1.0$ could be an interesting candidate, if the DNF should spend more time in the activated state.

As already mentioned in Sec. 4.1, the introduction of IP for population codes on

a local, single neuron based fashion is not straight forward. It is only for a neuron encoding actual relevant feature values (i.e. having any input at all) that an adaptation of intrinsic excitability makes sense. In order to realize an individual adaptation of gains and biases of single neurons, there therefore needs to be an additional mechanism in place to adapt the input weights of each neuron, i.e. to tune the neuron response to represent a new feature value. This would correspond to an adaptive feature resolution with fovea-like effects for feature value regimes with high probability. The originally proposed algorithm of the self organizing maps (SOM) would be an example of such an input weight tuning of a DNF [KOHONEN 1982]. The problem here is the reliance on stochastic, uncorrelated input required for training and maintaining the SOM, which renders the SOM algorithm inapplicable in the proposed setup with highly correlated input to the DNF.

Summarizing this section on intrinsic plasticity in dynamic neural fields, the adaptation of the DNF resting level and sensitivity works well in the tested cases. It ensures an highly input sensitive operating regime for the DNF dynamics, independent of the input amplitude and mean. This unsupervised adaptation of a DNF is in particular relevant for architectures where DNFs receive inputs from multiple sources with unknown distributions.

Chapter 5

A Neurodynamic Model for Haptic Spatiotemporal Integration

This chapter describes a neurodynamic model for the process of haptic spatio-temporal integration, i.e. the localization and simultaneous mapping of an object with tactile sensors while manipulating it. Hence, the proposed model performs the tasks of object mapping and object localization, as depicted in the two boxes at the bottom of Fig. 1.3.1. In particular, the model makes use of biologically plausible tactile features (Sec. 3.4) and is formulated using dynamic neural fields, introduced in Sec. 2.4. Based on evidence in developmental psychology studies (e.g. [SMITH 2009, SPELKE et al. 2010]) the object shape representation is separated from the object’s pose representation, i.e. the model uses an explicit object representation. This is also supported by neurobiological work indicating that object representations are invariant to its pose, which holds for vision [HUNG et al. 2012] as well as for haptics [HSIAO 2008]. For more information on haptic processing and shape representation, see Sec. 2.1. Inspired from these findings an initial version of the proposed model was published in [STRUB et al. 2014b], focusing on the estimation of object orientation. This model was extended in [STRUB et al. 2014a] in order to additionally estimate the object position along one translational axis. Since then, the model has undergone a major revision as will be described in this chapter, and was reimplemented in cedar (Sec. 2.4).

The model is analysed and evaluated with data from robotic experiments (described in Chp. 3), as it is very complicated to simulate tactile sensors that give an adequate reproduction of the complexity of real sensory information from robotic object manipulations. The perceived tactile features in robot space are transformed into object space by shifting and rotating the features’ spatial representation according to the current estimation of the object’s pose. These features in object space are then used as inputs to the model, implemented in DNFs.

The theory of DNF has been described in Sec. 2.4. However, a brief recall of the relevant equations used in the model is given here for convenience. In the according sections of this chapter the parameters of the equations are given for each DNF. The

DNF equation used in the model is:

$$\tau\Delta u(x, t) = -u(x, t) + S(x, t) + \sum_{x_i \in X} f(u(x_i, t))\omega(|x - x_i|) \quad (5.0.1)$$

$$S(x, t) = \sum_{i=0}^n w_i I_i(x, t) \quad (5.0.2)$$

$$f(x) = \frac{1}{1 + \exp(-ax + b)} \quad (5.0.3)$$

$$\omega(x) = \sum_{j=0}^k \pi_j \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma_j}\right)^2\right) - \pi_{\text{glob}} \sum_{x_i \in X} u(x_i, t) \quad (5.0.4)$$

The resting level of the DNF is now encoded in the bias parameter b . The DNF has an input from n inputs I , weighted by the factors w_i . The lateral interaction is defined by the kernel ω , which is a linear combination of k Gaussian kernels with widths of σ_j and weights π_j . Note, that the weights may be negative to represent lateral inhibition. To enhance computational efficacy, a global inhibition parameter π_{glob} is introduced to implement an inhibitory kernel acting on the entire field. The MT equation used in the model is given by:

$$\Delta P(x, t) = \frac{1}{\tau_{\text{build}}}(u(x, t) - P(x, t))f(u(x, t)) - \frac{1}{\tau_{\text{decay}}}P(x, t)(1 - f(u(x, t))) \quad (5.0.5)$$

with the parameters τ_{build} and τ_{decay} , in addition to the gain and bias parameters of $f(x)$.

In the model the tactile features are classified into flat surfaces and edges, which are processed in two distinct pathways. Each of these pathways has its own object map, represented in the activity of a memory trace. The current detected features are matched with this map and the differences are used to correct the estimated object rotation and translation. While the object shape (map) is explicitly represented in the activity of memory traces, the object pose (location) is not represented in the model. The representation of the object orientation and position estimate are implemented as linear integrators, integrating the estimated changes in orientation and position from the forward model and the pose corrections from the model.

An overview of the model is given in Fig. 5.0.1, where the plots show the activation $u(x, t)$ for a temporal snapshot t .

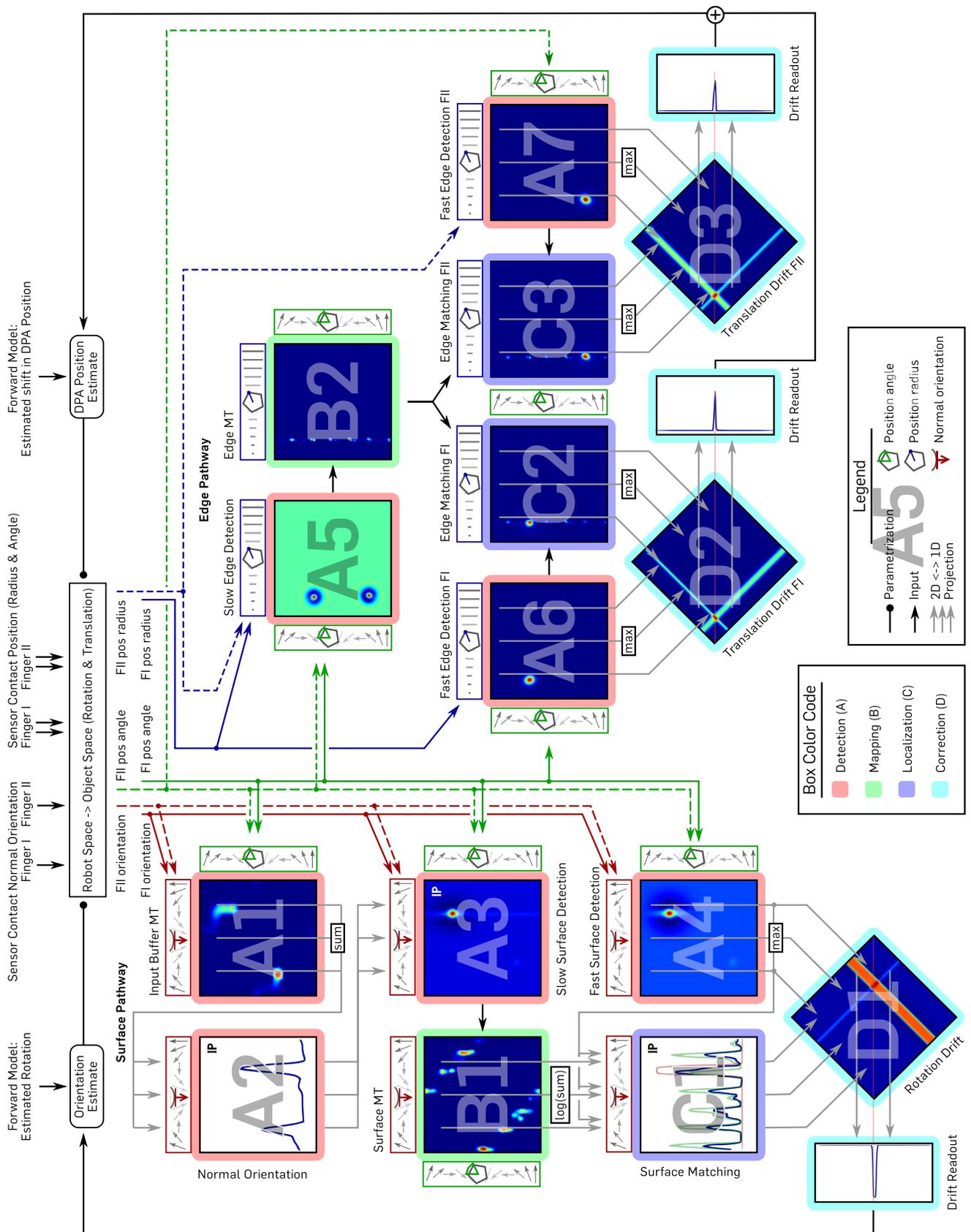


Figure 5.0.1: Overview of the neurodynamic model, see text for description.

The DNFs and MTs in Fig. 5.0.1 are marked with colored boxes and labels, indicating the functional category. Red boxes and labels (A) correspond to detection, green (B) to mapping, blue (C) to matching (i.e. localizing) and cyan (D) to correction functions. The flow of information is depicted with arrows, where red arrows correspond to orientation features, green to position angle and blue to the positional radius feature dimension. For all feature dimensions the according amplitude is given by the eigenvalue ratio (EvR). Arrows with solid lines correspond to information from finger I and dashed arrows to information from finger II. The different stages of the model will be described in detail in subsequent subsections.

The remainder of this chapter is structured as follows: First the tactile feature representation and the detection of edges and surfaces is explained in Sec. 5.1, which correspond to the seven boxes marked with a red border and labels starting with A in Fig. 5.0.1. Next, the object mapping is described in Sec. 5.2, which corresponds to the two boxes marked with green borders (and B labels) in the figure. In Sec. 5.3 the core of the neurodynamic model is described, the localization of the object which is done by matching the current tactile features with the object map, visualized by the three boxes with blue borders and C labels in the figure. Finally, in Sec. 5.4 the mechanism for correcting the current object pose estimate is described, corresponding to the boxes marked with cyan borders and D labels in the bottom row of Fig. 5.0.1.

5.1 Feature Representation and Classification

The neurodynamic model has two distinct pathways, a surface and an edge pathway, corresponding to the left half and the right half of Fig. 5.0.1, respectively. The surface pathway processes flat object surfaces for localization with respect to the object orientation, while the edge pathway detects and maps edges for localization with respect to the position. In the edge pathway, features of each finger are partly processed separately, i.e. some processing steps exist twice in the edge pathway. In the surface pathway the features of both fingers are processed jointly. In the following, both pathways are described in parallel, as the principals in each step are similar.

In the first stage of the neurodynamic model the features are encoded in population coding. In particular the position of tactile contacts in object space (two dimensional) is transformed into polar coordinates and serves as input to the detection DNFs in the edge pathway. The angle of the position together with the contact normal orientation in object space serve as feature dimensions to the detection DNFs in the surface pathway.

Figure Fig. 5.1.1 gives an illustration of the edge- and surface detection fields, where the activation for each of the feature dimensions of the detection DNFs is shown. For a tactile contact between a finger and the object the features contact position, sensor orientation and the EvR are computed. While contact position (i.e. position radius and angle) and sensor orientation serve as feature dimensions for the DNFs in the model, the EvR serves as feature amplitude. Hence, the input to the DNFs at a given point in time is a circular Gaussian “blob” with the amplitude depending on the EvR at the according feature values. The width σ of the Gaussian is set to $n/50$ with n being the field size (in the according dimension). In the proposed model all fields have the size 100 in each feature dimension, i.e. the dynamics is evaluated at 100 equidistant

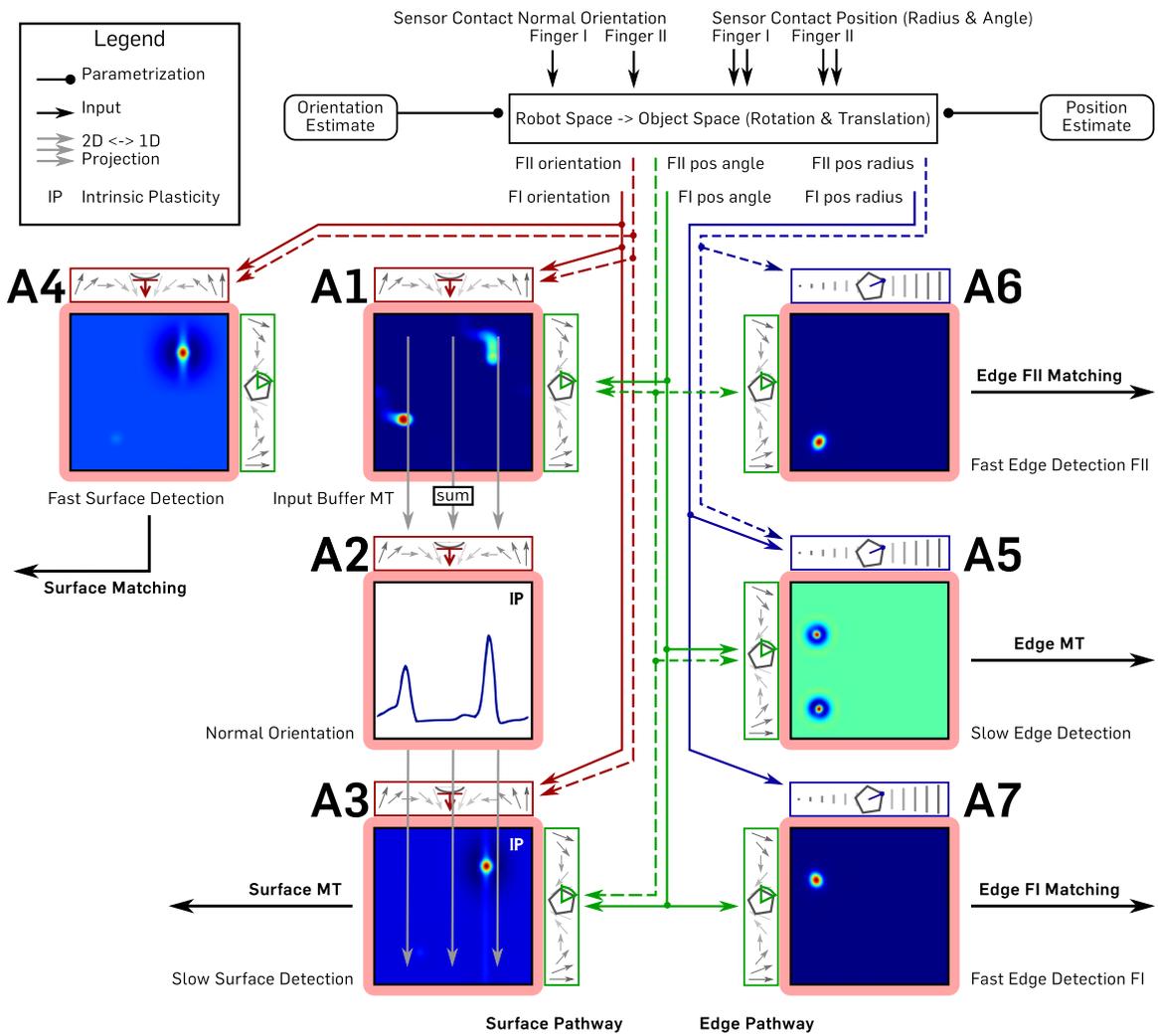


Figure 5.1.1: Overview of the surface and edge detection in the neurodynamic model, see text for description.

locations along each dimension of the DNFs.

Furthermore, the detection DNFs can be categorized into fast and slow fields, depending on how fast a detection is performed, i.e. how fast a peak is generated for according input patterns. In general, the slow detection fields need more time for the detection (i.e. classification) but have a lower false positive detection rate compared to the fast detection fields. These two properties makes them especially suited for mapping, i.e. storing the detections into the according object maps B1 and B2 in Fig. 5.0.1. The fast detection fields are used for the localization, i.e. matching detected features with the map. In the following subsections the detection of edges and surfaces in the neurodynamic model will be described, concluding with a tabular overview of the parameters of the DNFs and MTs.

5.1.1 Edge Detection

In the case of the edge pathway, the current tactile contact position in object space is transformed into polar coordinates and the angle and radius are encoded in the x and y axis of two dimensional edge detection DNFs (A5-A7 in Fig. 5.1.1). The amplitude is given by the negative eigenvalue ratio ($1 - \text{EvR}$) of the according tactile contact area, i.e. high input amplitudes for sharp and narrow contact areas. The edge detection DNFs have standard Mexican hat kernels and no global inhibition to generate activity peaks for high object curvature at stable contact positions. Only stable inputs with respect to the position over time allow the field dynamics to reach the detection instability due to the inertial dynamics. The DNF is parametrized such that no self sustaining activity in the field is possible, the parameters are given in Tab. 5.1.3. In particular, there are one “slow edge detection” DNF (A5) for the mapping and two “fast edge detection” DNFs (A6 + A7) for localization of detected edges in the edge pathway. In slow edge detection DNF information from both fingers is processed, while the fast edge detection DNFs only process information of one finger, respectively. The slow edge detection DNF primarily has a higher time constant and a narrower kernel width, i.e. the lateral interaction is more locally, compared to the fast edge detection DNF. Both of these differences contribute to the slower detection rate, as the smaller kernel width and stronger inhibition impose a stronger requirement on the input sequence to be tightly clustered with respect to the location.

Hence, the slow detection fields generate peaks with precise localizations of detected edges, although more time is needed and the detection rates are more conservative compared to the fast detection fields.

5.1.2 Surface Detection

In the surface pathway, the contact normal orientation is encoded together with the contact position angle (in object space) as a two dimensional input for the DNFs. The amplitude is now given by the eigenvalue ratio (EvR), i.e. high input amplitudes for broad contact areas. As in the case of edge detection described in the previous subsection, there is a slow- (A3) and a fast surface detection DNF (A4 in Fig. 5.1.1). However, the slow surface detection has an additional input.

Slow surface detection

The slow detection of surfaces is achieved by combining information from temporal integration with the current EvR . The temporal integration is linked to the distributional approach to curvature estimation, described in Sec. 3.4. First, a memory trace (A1) serves as short time input buffer which stores the recent inputs to the surface pathway of the model (see Tab. 5.1.3 for the parametrization). This input buffer MT is then projected to a one dimensional normal orientation DNF (A2) by summing over the position angle dimension. This results in high input amplitude to the normal orientation DNF when the past inputs have similar normal orientations, i.e. several contact measurements on a planar object surface have been made. The normal orientation DNF thus detects peaks in the current distribution of contact orientation normals, based on an exponential fading time window. This idea of temporal

integration of normal orientations for surface detection was published in [STRUB et al. 2014b]. However, in the current revision the normal orientation DNF is continuously adapted by IP and thus autonomously tunes its sensitivity to the input amplitude. This is in particular relevant, as the amplitude of the projected input buffer MT (A1) is dependent on the spatial extension of the object surfaces. Large surfaces lead to an increased amount of tactile measurements with the same normal orientation and thus to higher sums of inputs. Hence, applying IP in the normal orientation DNF is crucial for the ability of generalizing over different object geometries. Note, that this online adaptation will lead to sporadic, highly input sensitive activations in the absence of variations in the normal orientation distribution which would e.g. be the case when rotating cylindrical objects. The normal orientation DNF (A2) is then projected as an additional input along the normal orientation dimension of the slow detection DNF (A3) in order to boost the EvR inputs for detected peaks in the normal orientation distribution.

In contrast to the slow edge detection field, the slow surface detection field (A3) uses a non-uniform Mexican hat kernel with a large excitatory σ in Eq. (5.0.4) along the contact position angle dimension and a small excitatory σ along the contact normal orientation dimension. The Inhibition of the slow surface detection kernel is approximately circular. The underlying principle here is that for subsequent contacts along a surface, the contact normal orientation should remain stable as the contact position angle varies. Thus, the kernel needs to be sensitive to high input amplitudes that vary along the contact position angle and are roughly constant along the contact normal orientation dimension. The slow surface detection field then generates a peak for detected surfaces which is sharp in the contact normal orientation dimension and broad in the contact position angle. Hence, the non-uniform kernel imposes an assumption on the input distribution in feature space of the DNF over temporal sequences. While this assumption on the input pattern holds for all objects, the particular spatiotemporal extent and amplitude is dependent on the spatial extent of the object surfaces. This requires the slow surface detection DNF (A3) to be autonomously adapted to the input characteristics by IP, like in the normal distribution DNF (A2).

For further processing only the contact normal orientation is relevant. The contact position angle dimension is only necessary to link the surface detection field to the edge detection field, which share this feature dimension.

Fast surface detection

In the surface pathway there is only one fast surface detection DNF (A4), as the tactile features of both fingers are processed jointly in one DNF. The fast surface detection DNF has standard circular Mexican hat kernels, thus the DNF is sensitive primarily to the input amplitude (EvR) and not to the input movement direction / spatial distribution. This enables a fast surface detection, although with higher rates of false positives.

Altogether, there are five detection fields in total used for further processing in the neurodynamic model, two (slow and fast) for detecting surfaces in sensory data of both fingers, one (slow) for detecting edges in sensory data from both fingers and two (fast) for detecting edges in sensory information from one of the fingers, respectively. The slow detection fields serve as input to the object mapping process described in

the next section, while the fast detection fields are utilized in the localization process described in Sec. 5.3.

5.1.3 Detection Parameters

In this subsection the parameter values of the DNFs and MTs related to the surface and edge detection in the neurodynamic model are given in Tab. 5.1.3 and Tab. 5.1.3, respectively.

DNF Name	I_i	w_i	τ	b	a	σ_{exc}	π_{exc}	σ_{inh}	π_{inh}	IP_μ	IP_η
A2: (1D) normal distribution	A1	1.0	50	-5.0*	1*	2	5	5	-5	0.2	0.001
A3: (2D) surf slow detection	EvR A2	4.4 1.25	300	-1.9*	1*	$\begin{pmatrix} 1.5 \\ 8 \end{pmatrix}$	6	$\begin{pmatrix} 10 \\ 12 \end{pmatrix}$	-28	0.4	0.0015
A4: (2D) surf fast detection	EvR	2.2	50	-1.0	6	$\begin{pmatrix} 2 \\ 8 \end{pmatrix}$	5	$\begin{pmatrix} 10 \\ 10 \end{pmatrix}$	-15	–	–
A5: (2D) edge slow detection	EvR EvR	0.3 0.3	80	-0.8	3	$\begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$	2	$\begin{pmatrix} 4 \\ 4 \end{pmatrix}$	-2	–	–
A6 & A7: (2D) edge fast detection	EvR	0.3	30	-1.0	3	$\begin{pmatrix} 7 \\ 7 \end{pmatrix}$	1	$\begin{pmatrix} 11 \\ 11 \end{pmatrix}$	-1	–	–

Table 5.1.1: Parameter values of the detection fields. The surface detection fields have the contact normal orientation as the first dimension and contact position angle as the second. The edge detection fields have the dimensions position radius and position angle. All 2D fields have a size of 100×100 , the 1D field has a size of 1×100 . The resting level and gain of the “normal distribution” and “surf slow detection” DNFs (denoted with *) are adapted with IP and the according values in the table only denote the initial values.

MT Name	τ_{build}	τ_{decay}	a	b
A1: buffer MT	5k	50k	1	0

Table 5.1.2: Parameter values of the input buffer memory trace. The MT has as size of 100×100 .

5.2 Object Mapping

Mapping of the object shape within the neurodynamic model occurs separately in the surface and the edge path, respectively. In particular, one map of object surfaces (B1) and one map of object edges (B2) is built, both fusing detections from both fingers. In

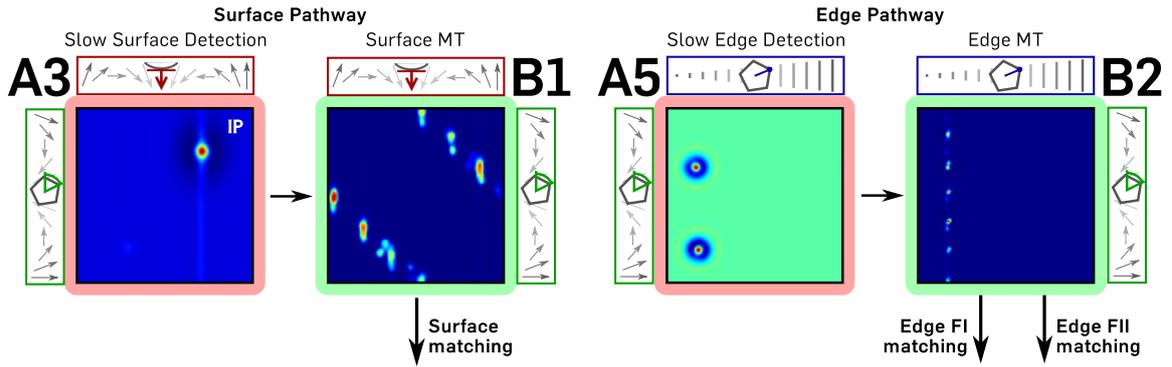


Figure 5.2.1: Overview of the surface and edge mapping in the neurodynamic model, see text for description.

both occasions, the according slow detection fields (A3, A5) serve as input to the map, which is represented in the activation of a MT (B1, B2) implementing an exponential fading memory (see Sec. 2.4). This implies, that information can not be corrected once it has been added to the object map, although all information fades out during time if not reinforced. This fading of memory is therefore critical as, information fused into the object map may be error prone – especially during the initial build up of the map. Assuming a localization mechanism at work, subsequent re-sensing of the same object area leads to a reinforcement of the fading out memory activities of the according map entries. Hence, the map is maintained only through memory fading and merging new information to reinforce the old. Therefore it is clear, that an effective localization mechanism is a necessary prerequisite for building a consistent map in the MTs. The higher the localization precision, the higher the spatial precision and activation level of the map, as activity of subsequent map entries accumulates in one position instead of being scattered in the MT. This increase of activation amplitudes in the MT for subsequently reinforced map entries increases the impact of the map on the localization and will therefore support a convergence of the SLAM process.

The surface map (B1) has the surface normal orientation on the x-axis and the angle of the surface position in object space on the y-axis. The representation of an object surface will therefore have an activity pattern with a small spatial extension on the x-axis and an extension on the y-axis according to the length of the surface, as visible in Fig. 5.2.1, on the left.

The edge map (B2) has the radius of the edge position in object space on the x-axis and the angle of the edge position in object space on the y-axis. The activation pattern representing an edge in the MT should be a circular peak of activity, like in the right half of Fig. 5.2.1.

MT Name	τ_{build}	τ_{decay}	a	b
B1: surf MT	6k	1M	10	0.5
B2: edge MT	5k	750k	10	0.3

Table 5.2.1: Parameter values of the surface and edge memory traces. Both MTs have as size of 100×100 .

The MT holding the edge map is parametrized with a faster decay rate than the MT for the surface map, as shown in Tab. 5.2, because of differences in the matching

procedure, described in the next section. Additionally, for the MTs to hold the object maps it is necessary to choose the time constants τ in Eq. (5.0.5) of the MTs with respect to the experimental setup. The time constants of the MTs have to be tuned according to the object rotation speed and the frequency of tactile feature readouts. If the decay τ is too small, the object map will fade out before features are revisited, while for a too large decay τ wrong information fades out very slowly. If the build τ is chosen too large the object map will hardly be build up at all. Hence, choosing the time scales for building and decaying of activity in the MT requires some manual tuning with respect to the operating regime of the robot hardware. The parameters used in the model are shown in Tab. 5.2.

5.3 Object Localization

The localization of the object pose requires to match the detected features with the object map and solve the correspondence problem (see Sec. 2.3 for further details). In the proposed model the only detected features are edges and surfaces. Edges and surfaces are detected by the slow detection fields (A3 + A5) and are stored into the MTs (B1 + B2). Additional to the slow detection fields, the matching process requires fast detection fields (A4 + A6 + A7) in order to match the current tactile features with the according object maps in the MTs (B1 + B2) *before* the current information is fused from the slow detection fields into the MTs. Matching A4 with B1 in the surface pathway enables the localization and correction of the object orientation, independent of its position. In the edge pathway, the matching A6 and A7 with B2 enables the localization of the object position, dependent on its orientation. The description of the matching is described separately in the following for the edge and the surface processing pathways.

5.3.1 Orientation from Surfaces

Localizing the object with respect to its orientation can be achieved as soon as a surface of the object is re-touched (i.e. re-visited). In the chosen experimental setup this occurs as soon as Finger II detects a surface formerly detected and mapped by Finger I and vice versa, i.e. after the object has been rotated ≈ 180 deg.

Matching features with the map

For localizing the object with respect to its orientation, it is necessary to associate a current detected surface (A4) with a previously detected one in the MT (B1). Then the deviance in the surface orientation corresponds to the error in the object orientation estimate. This association (matching) of a current detected surface to a previously mapped one is done by only utilizing information of the surface orientation, the surface position angle is neglected. This assumes that the surface in the map with the most similar orientation corresponds to the currently detected one. Thus a nearest neighbour matching with respect to contact normal orientation in object space is performed, if matched at all. This neglecting of surface position is done for two reasons: First, the manipulated objects are assumed to be convex – preventing the possibility

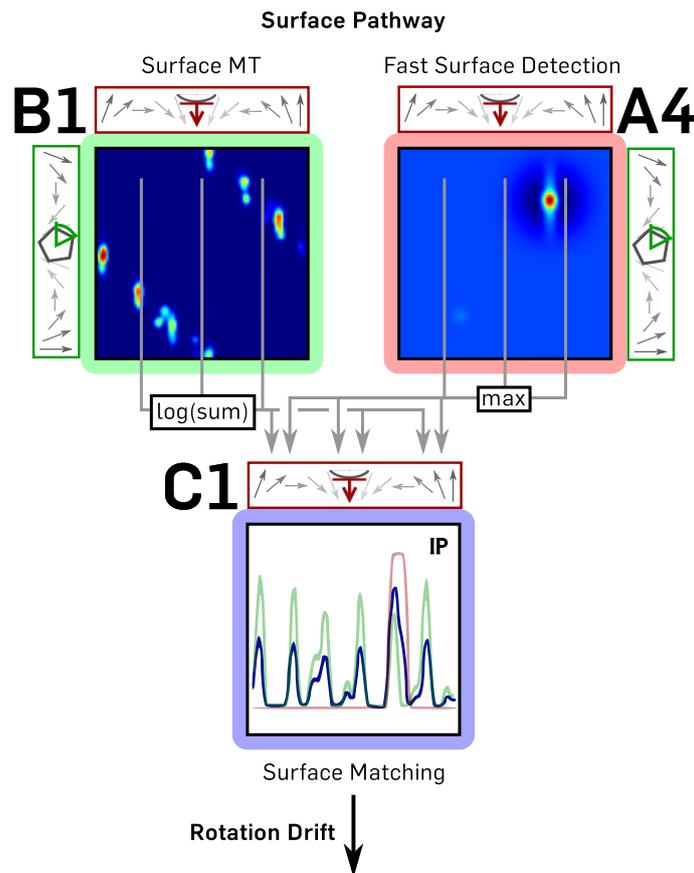


Figure 5.3.1: Overview of the surface and surface matching in the neurodynamic model, see text for description.

of multiple surfaces with the same orientation, which could only be distinguished by their position. Furthermore, in convex objects the position of a surface is strongly restricted by its orientation.

Therefore the fast surface detection field (A4) and the surface MT (B1) are both projected onto the surface normal orientation dimension. For projecting the surface MT, the logarithm of the summed activity along the contact position angle dimension is computed. This has the effect that surfaces with a large spatial extent have a larger impact to the matching (higher saliency). Hence, the spatial structure in the surface MT is better captured when using the summed projection. In the case of the fast surface detection DNF the maximum of the activity along the contact position angle dimension is taken, as the activation pattern is largely determined by the kernel parametrization and thus does not convey any additional information. These two projections are provided as input into the one dimensional surface matching DNF (C1), illustrated in Fig. 5.3.1.

The input from the surface MT projection (B1) primes positions of previously detected surfaces in the matching DNF (C1), but for itself is not sufficient to trigger a peak in the output activation of the matching DNF. If the fast surface detection field (A4) generates a peak at a novel location (i.e. a novel surface normal orientation is detected), its contribution to the surface matching field alone is also not sufficient for

generating a peak. It is only if the fast detection field generates a peak near (with respect to the normal orientation) to a previously stored detection in the surface MT, that the combination of both suffices to generate an output peak in the matching field (C1). The lateral interaction in the matching field leads to an output peak that lies in between the old and the current detected locations. Hence, the matched output peak lies closer to the previously detected surface orientation than the currently detected one. This difference in location of the surface matching DNF (C1) output peak with respect to the fast detection field (A4) output peak serves as a correction direction for the object orientation estimate, as will be described in Sec. 5.4. Since the system runs in a closed loop, the corrected object orientation will lead to a shift of the current detected surface normal orientation (see Fig. 5.0.1), thus the current detected surface orientation converges towards the past orientation stored in the MT.

The matching of the fast surface detection (A4) with previously mapped ones in the surface MT (B1) and the according correction of the object orientation estimate should be done before the slow surface detection (A3) fuses the current surface into the object map, i.e. the surface MT (B1). Therefore the slow and a fast surface detection DNFs are necessary. While the matching (A4 with B1 in C1) and correction (A4 with C1 in D1) take place, the slow surface detection field (A3) will not detect a surface, as the continuous correction of the object orientation imposes a drift in the surface normal orientation in object space. Therefore, the normal orientation detection field (A2) does not generate a peak as the normal distribution is broad during the correction, described in detail in Sec. 5.1. Additionally, the sharp kernel shape of the slow detection field (A3) along the surface normal orientation prevents the generation of peaks for inputs moving along the normal orientation dimension. As soon as the object orientation correction has converged, surface detection in the slow detection field may occur. This would then trigger a new entry into the map or the reinforcement of a previous detection.

In order for the matching system to operate, the proportion of input contributions from the fast detection DNF (A4) and the surface MT (B1) has to be tuned, as well as the build and decay rates of the surface MT. The proportion of inputs determines how conservative the matching is, i.e. how much past “evidence” in the MT is needed to trigger the matching process.

Adaptation of the matching DNF via IP

In order to keep the input from the projected MT (B1) in the regime just below the detection instability, the resting level of the matching DNF (C1) needs to be tuned accordingly, such that the input originating from the MT is always just sub threshold. This is particularly difficult, as the MT accumulates and leaks information over time and thus is sensitive to the temporal structure (i.e. frequency) and location (i.e. spatial density) of the detected surfaces. Objects with more (flat) surfaces on average also have surfaces with shorter spatial extension, for a fixed diameter. Thus, the input duration of each detected surface to the MT is on average shorter for objects with a higher number of surfaces and therefore the average activation amplitude in the MT is lower compared to objects with a fewer number of surfaces. Additionally, larger surfaces have a larger distribution of activity along the object position angle dimension and therefore the logarithmic summed projection onto the normal orientation dimension

scales with the surface length. Hence, an online parameter adaptation is necessary for a robust tuning of the matching DNF such that the input of the MT alone is always in a sub-threshold but input sensitive regime.

The adaptation of the matching DNF (C1) is done with IP, as described in Chp. 4. The DNF input is scaled by an adaptive gain and the DNF resting level is controlled by an adaptive bias. Both parameters are tuned by the IP algorithm in order to minimize the Kullback-Leibler divergence between the output distribution of the DNF and the target exponential distribution. This adaptation of the gain by IP ensures, that if, and only if, inputs from the fast detection DNF and the MT coincide, the matching DNF will generate a peak in the output activity. Introducing the IP adaptation of the matching DNF enhances the robustness of the neurodynamic model with respect to the different objects.

5.3.2 Position from Edges

While detected object surfaces are used to locate the orientation of the object, detected object edges serve for the localization of the object position. The localization of the object position can be split up into two subtasks: localizing along a distal-proximal axis (DPA) and the localization with respect to a lateral axis, as described in Sec 3.3. The localization along the lateral axis has little relevance for object localization in the chosen experimental setup, as the lateral object position during manipulation is mechanically restricted to the centre between the two fingers. Therefore the object localization along this axis is not implemented in the model, although a brief outline of how this could be done will be given in a subsequent subsection.

For object localization from an detected edge of finger one, there is an edge matching DNF (C2) with input from the edge MT (B2) and the according fast edge detection DNF (A6). The matching DNF generates an intermediate peak if the two input peaks are sufficiently near in their location, i.e. in the edge position radius. However, there are several differences with respect to the rotational case. First the relevant feature is now the position of a detected object edge in object space, in contrast to the surface normal orientation in the rotation pathway. Second, the features of both fingers are stored together into one MT (B2), but are then processed separately for each finger. Third, the edge matching of the edge MT with the fast edge detection DNF is performed in two dimensions, i.e. no projections are computed like in the surface case.

Localizing along the proximal-distal axis

In contrast to the localization with respect to the object orientation, the matching is not primarily performed with respect to edges that have been detected during previous object rotation cycles. Instead, the matching is done with respect to the same edge, detected in successive measurements during the object rotation with contact to this particular edge. Hence the highly ambiguous data association problem after re-grasping the object in the rotation pathway is now replaced with a much easier tracking problem of one edge during the manipulation with a stable grasp. When contact to the tracked edge is lost, the matching stops and is then re-initiated when the next edge is detected.

When touching an object edge, its position may be determined in each tactile sensor measurement. Successive measured positions of an edge during object manipulations while maintaining contact with the edge are:

- necessary identical in *sensor space* (i.e. the edge does not “slide” along the finger)
- may be different in *robot space* (as the object is manipulated)
- should be identical in *object space* (i.e. object deformations are not considered)

In object space this only holds for a correct object localization, any errors in the object position estimate induce systematic errors in the transformation of tactile features into object space, as described in Sec. 3.6 and Sec. 3.7. Hence, for localizing the object position with respect to the DPA the drift direction of the a detected edge position in object space during object manipulations with a stable grasp is used. For a drift of the edge position towards the centre of the object (in object space) the current position estimate is too proximal, for an outward drift it is too distal (see Sec. 3.6).

This drift of the edge position is indicated by differences in the peak locations (with respect to the radii) between the edge MT (B2) and the fast edge detection DNFs (A6 + A7), due to the differences in the temporal dynamics. In contrast to the case of rotation estimation in the surface path, the edge MT and the fast edge detection DNFs are not projected before matching. The edge position radii may be similar for multiple object edges and are indeed the same for all edges of the objects used in the experiments. Hence a projection of the MT would lead to a irreversible fusion of all edge radii information in the map and the matching would be performed to an arbitrary edge instead of matching to the memory of the according current edge position. Therefore, the projections are done after the matching, as the drift of the currently detected edge in its position radius between the fast edge detection DNFs and the edge MT is of particular interest. This projection will be described in Sec. 5.4. Accordingly, all DNFs in the edge pathway and the MT (B2) are two dimensional, featuring edge position angle and radius as feature dimensions.

In the neurodynamic model the localizing of the object position with respect to the DPA is implemented separately for each finger, except for the shared edge map. The separate processing of the fingers is necessary because the relevant information is on the object position radius dimension, thus information from both fingers in one DNF would interfere in the correction DNFs, discussed in the next section.

The two dimensional DPA position matching field of each finger (C2 + C3) have inputs from the fast edge detection DNF of the according finger (A6 + A7) and the joint edge MT of both fingers (B2), visible in Fig. 5.3.2.

If peaks in the fast edge detection DNFs (A6 + A7) and the edge MT (B2) are sufficiently close in their radii (depending on the kernel parametrization), an intermediate peak is generated in the edge matching fields (C2 + C3), analogous to the rotational case described previously. In Fig. 5.3.2 this procedure is illustrated. The peak position of the matching DNF is successively used for correcting the object position in order to minimize the drift of the edge position radius. Since the matching and correcting are based on drifts in tactile measurements transformed by the forward model (see Sec. 3.6), the position of previously mapped edges is only of minor interest

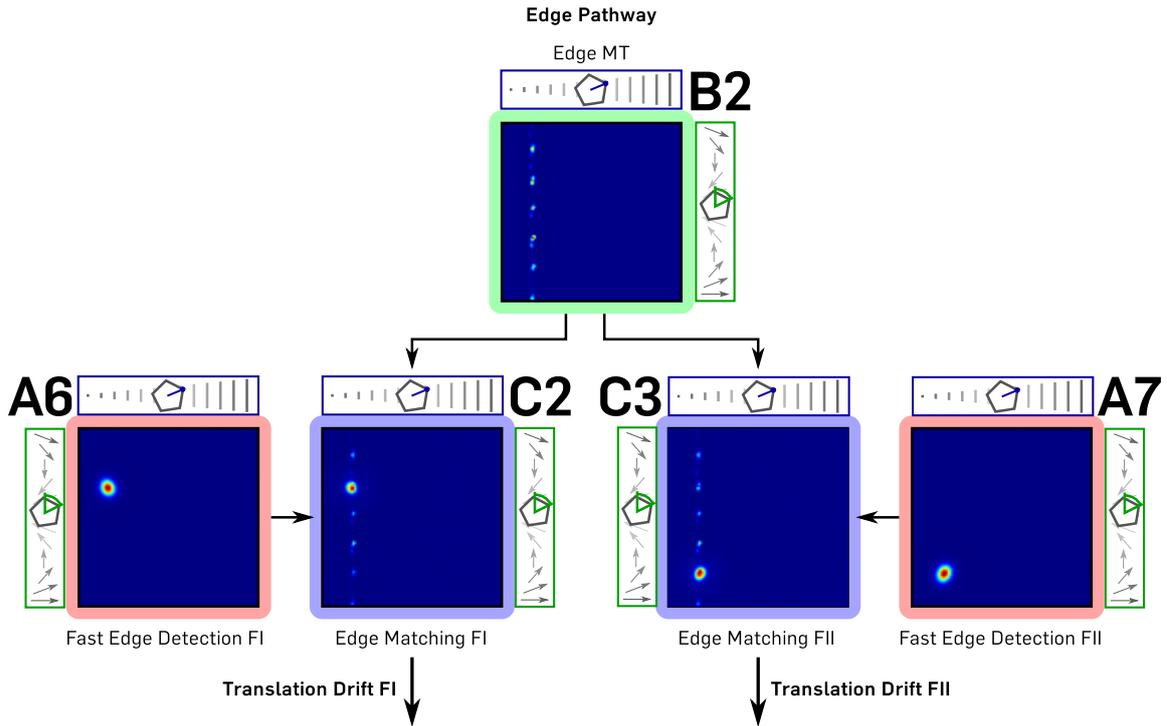


Figure 5.3.2: The matching of detected with mapped edges.

and therefore the MT (B2) holding the edge map has a faster decay rate than the MT of the surface map (B1).

The higher decay in the edge MT (B2) combined with the feature separation of the two fingers makes the matching more robust with respect to number and spatial distribution of object edges. Compared to the surface path, activity in the MT does not accumulate over successive object “full rotations” (360°) as the activity of previously visited edges has only a minor contribution. Additionally, the matching is done in two dimensions and projection to one dimension for comparison is only done after matching. Therefore the matching DNFs for the translation do not benefit from an IP adaptation.

Localizing along the lateral axis

The distortion pattern of the detected object edges in object space for localization errors in the distal-proximal axis is a drift of the edge position towards or away from the object centre during object rotation. In the case of localization errors with respect to the lateral axis, the distortion in object space is reflected in an asymmetry: the same object edge would be mapped with different radii by the two fingers. Therefore, correcting localization errors in the lateral axis requires a mechanism to detect systematic offsets between the edges detected by both fingers.

However, the gain of a localization mechanism with respect to the lateral axis during object manipulation is not clear, as natural grasps for object manipulation usually imply contacts at opposing locations [IBERALL et al. 1986]. In the experimental setup in this thesis this is also the case, thus the localization with respect to the lateral

DNF Name	I_i	w_i	τ	b	a	σ_{exc}	π_{exc}	σ_{inh}	π_{inh}	IP_μ	IP_η
C1: (1D) surf match	B1 A4	2.4 2	20	-4.0*	1*	3	4	5	-4	0.15	0.0005
C2 & C3: (2D) edge match	B2 A6/A7	2 1	30	-1.1	6	$\begin{pmatrix} 9 \\ 9 \end{pmatrix}$	3	$\begin{pmatrix} 15 \\ 15 \end{pmatrix}$	-2	-	-

Table 5.3.1: Parameter values of the **matching DNFs**. The surface matching DNF is one dimensional and of size 1×100 , while the two edge matching DNFs both have the size of 100×100 . The resting level and gain of the surface matching DNF (denoted with *) are adapted with IP and the according values in the table only denote the initial values.

axis is not implemented.

5.3.3 Localization Parameters

The parametrization of all three matching fields is shown in Tab. 5.3.3, where the “edge” DNF parametrization holds for both instantiations of the DNF (C2 + C3), one for each finger. There is no global inhibition in the surface matching field, as multiple peaks are intended due to the possibility of both fingers simultaneously touching surfaces. In contrast, the edge matching DNFs (C2 + C3) prevent multiple peak solutions as a single finger cannot simultaneously detect multiple edges as the objects are assumed to be convex. This is implemented by a global inhibition parameter π_{glob} which is set to 0.001, corresponding to a uniform distributed inhibitory kernel.

5.4 Pose Correction

In this section the final part of the neurodynamic model is described, the correction of the object pose estimate, based on the localization in the map. The dynamics of the localization result in a corrected feature pose, thus a mechanism to extract the according correction term for the object pose estimate is required. For this, the matched activation peaks in the edge matching DNFs (C2+C3), (which represent the corrected feature pose) are compared to the original activation peaks in the according fast edge detection DNFs (A6+A7). In the surface pathway the surface matching field (C1) is one-dimensional and the fast surface detection DNF (A4) is two-dimensional. In contrary, in the edge pathway the edge matching DNFs (C2+C3) and the fast edge detection DNFs (A6+A7) are all two dimensional. All these two-dimensional DNFs are projected by using the maximum of the sigmoided DNF output activity along the edge position angle dimension, thus only the position radius dimensions remains in the DNFs of the edge pathway and the normal orientation dimension in the surface pathway (see Fig. 5.4.1).

The comparison of two of the resulting one-dimensional DNFs is performed in a two dimensional DNF (D1-D3), where each input field projects along one of the two dimensions and both projections sum up in their activity. This procedure is shown in the two DNFs depicted as rhombus (D1 + D2) in Fig. 5.4.1. A diagonal

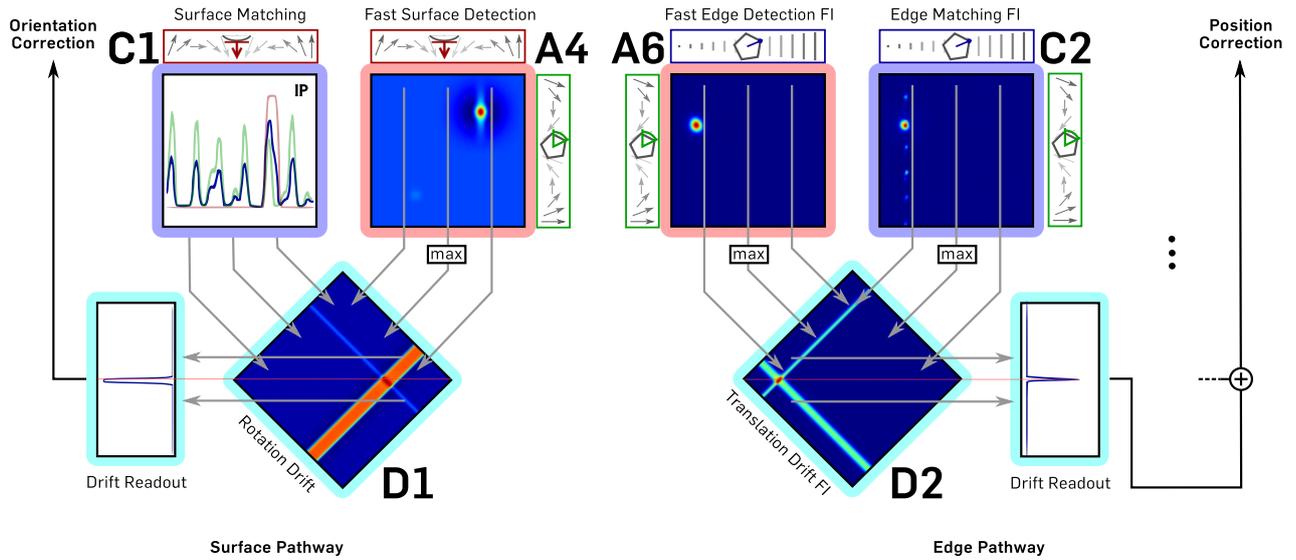


Figure 5.4.1: Correcting the pose estimate from the model localization. The translation correction is only shown for one finger (DNFs A6-C2-D2 on the right side of the figure). The correction for the second finger is analogous and therefore omitted (DNFs A7-C3-D3).

readout of this two-dimensional DNF, i.e. a diagonal projection to a one dimensional drift readout DNF (Fig. 5.4.1), creates a representation of the two input peaks in relation to each other. This mechanism is a neural implementation of reference frame transformations as described in [SCHNEEGANS & SCHÖNER 2012] and provides information of the peak shift due to the matching. The activation in the readout DNF is suppressed at locations far from the center to give close by matches a higher weighting and suppress matches with large deviations. This in particular necessary, as the projection of two matches of two detected surfaces results in four peaks in the rotation drift DNF. The suppression is implemented by multiplying the output of the readout DNF with a Gaussian with the mean located at the DNF centre and a standard deviation σ of 1.8° degree. The deviation of the peak position from the center of the drift readout DNF is the estimated error and is subsequently used to correct the current object pose estimate. Thus, for consecutive time steps, the perceived feature is mapped to the corrected location in object space. The object pose itself is encoded by a Gaussian distribution with the mean at the according position. Pose corrections generated by the model are continuously integrated into this representation, hence the location of the Gaussian mean is adapted by the pose correction mechanism. The object pose is additionally adapted by the forward model (Sec 3.6) for each tactile measurement. The tactile measurements are implemented as discrete events and therefore the pose update by the forward model is implemented by summing these up. Hence, the object pose representation is not implemented with neural dynamics but with a hard-coded integration of discrete and continuous updates. The parametrization of the drift DNFs and the according readout DNFs is given in Tab. 5.4.

DNF Name	I_i	w_i	τ	b	a	σ_{exc}	π_{exc}	σ_{inh}	π_{inh}	π_{glob}
D1: (2D) rot. drift	C1 A4	1 1	30	-1.5	4	$\begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$	1	$\begin{pmatrix} 4 \\ 4 \end{pmatrix}$	-2	–
D2 & D3: (2D) transl. drift	C2/C3 A6/A7	1 1	30	-1.5	6	$\begin{pmatrix} 3 \\ 3 \end{pmatrix}$	1	$\begin{pmatrix} 7 \\ 7 \end{pmatrix}$	-1	0.001
readout[3×]			20	-1.0	10.0	–	–	–	–	0.1

Table 5.4.1: Parameter values of the **drift DNFs**. The rotation drift DNF and the two translation drift DNFs all have a size of 100×100 . The three readout DNFs are one dimensional with a size of 1×100 and have no lateral interaction.

Chapter 6

Results

In the previous chapter the neurodynamic model was described, in this chapter the model is evaluated and the results are discussed. The discussion of the the model itself is postponed to the next chapter.

This chapter is structured in the following: first the results are discussed qualitatively with respect to the object mapping. For this, the obtained mappings are visualized and the surface MT holding the detected object surfaces is analysed. Then the object mapping is evaluated quantitatively, based on the number of detected surfaces and the matching to the ground truth shape. Last, the performance of the model with respect to the object localization is evaluated based on the pose corrections.

6.1 Object Mapping

The mapping of the object obtained by the neurodynamic model is evaluated in two steps, first qualitatively and then quantitatively. For this a combined representation of the surface and edge MTs is visualized, showing the build mapping of the object. Then the MT containing the surface information is analysed in detail. The results of the large six and eight sided objects are compared to the obtained object map from the model without any corrective feedback. This is done in order to distinguish effects of processing in the model from its localization performance. The complete set of results is provided in the appendix, Ap. B. In the second half of this section, a quantitative evaluation of all the recorded datasets is given.

6.1.1 Qualitative Evaluation of the Mapping

In the neurodynamic model the object mapping is distributed into two MTs which is qualitatively evaluated for six hand chosen examples in this subsection, three examples from the large six and eight sided objects and three examples from the asymmetric objects. The results of small six and eight sided objects are postponed to the quantitative analysis, as they do not add any qualitative insights.

Final object representation

For this, the information of the surface and edge MTs is combined and plotted in Cartesian object space, opposed to the polar coordinate system in the MTs. This procedure is straight forward for the information contained in the edge MT, as the dimensions are position angle and radius. In the surface MT the dimensions encode position angle and normal orientation of a surface. The activity distribution also encodes the spatial extent of a surface. As the surface MT does not have a representation of the position radius, a fixed radius is used for plotting. Alternatively, the location between two neighbouring (with respect to the position angle) edges in the edge MT would be a plausible estimate for the location of a surface. The marked edges in the plot on the left of Fig. 6.1.1 (A) are the exact positions contained in the edge MT, while the surfaces with their according normal orientations are plotted by adding the information of a fixed radius.

The mapped object in Fig. 6.1.1 is from the dataset #1, the large eight sided object. In the visualization of the object surfaces on the left plot (A) in Fig. 6.1.1, the fading information of a “ninth surface” is still visible at the bottom right corner (i.e. edge) of the object. The same information is also present in the top right plot (B) of the figure showing the surface MT, where activation for the normal orientation of approximately 290° is visible and accordingly results in a small peak in the summed activation (blue graph). The edge MT in the bottom right (C) of Fig. 6.1.1 reveals, that the object representation has undergone a change with respect to the object orientation. In principle, one would expect to see eight clearly separated activity peaks corresponding to the eight edges of the object. However, in (A,C) Fig. 6.1.1 a drift in the object orientation of the entire map is visible by the regular appearance of past detected (and thus weaker) edge representation in between the current (stronger) ones. As the object map is drifting consistently, this drift is only visible in the object position angle dimension but not in the surface normal orientation (B). The angular distance between the surfaces remains stable, yet the angular position of these drifts leading to a stronger surface overlap. This is visible in the tendency of the surfaces to have a pronounced counter-clockwise weighting in the left plot (A) of Fig. 6.1.1. Hence, while the model achieves a consistent (spatial) object representation it is not necessarily constant (temporal) in its orientation as the object is continuously rotated.

For two additional exemplary chosen datasets #4 and #8 the results are visualized in Fig. 6.1.2. In the set of plots at the top of the figure (#4), it is visible that several surfaces have been merged together, leaving only 2-3 clear surfaces in the object representation (A,B). This also leads to errors in the edge representation, with too many and inhomogeneous distributed edges due to the erroneous corrections in the object localization by the model (A,C). In the bottom set of plots in Fig. 6.1.2 (#8) the clear map of the six sided object is visible (A). The results here are representative for the datasets of the large six sided object (see Sec. B in the appendix for the object maps of the other datasets).

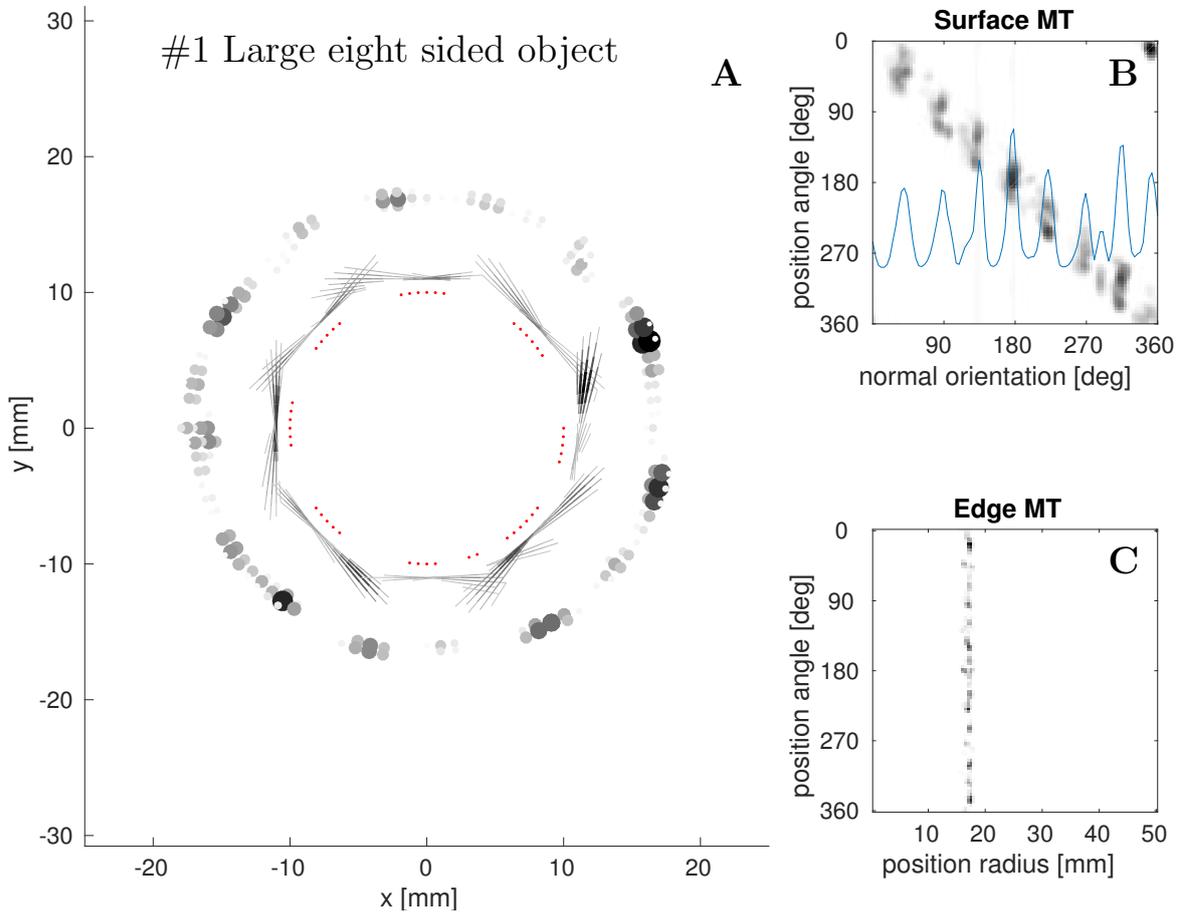


Figure 6.1.1: The object representation reconstructed from the surface and edge MTs at the end of the second iteration (i.e. 8th object rotation) through the dataset #1 (large eight sided object). On the right the activity in the MTs is visualized (A-B), darker shades correspond to higher activity. In the plot of the surface MT (B) there is an additional overlay showing the MT activity summed across the position angle dimension – i.e. projected to the normal orientation dimension (blue). On the left (A) the MT information is visualized in Cartesian object space: shaded dots indicate the positions of detected edges in the edge MT (C), darker and larger dots correspond to higher MT activity. The small red dots mark the surface normal orientations with activity of more than $1/4$ of the maximum peak activity of the projected surface MT (blue overlay). The shaded lines are located at the position angle of the according site in the surface MT with a fixed radius. The orientation of the lines is determined by the according normal orientation in the surface MT (B) and the intensity corresponds to the activity (i.e. amplitude) at the according position in the MT.

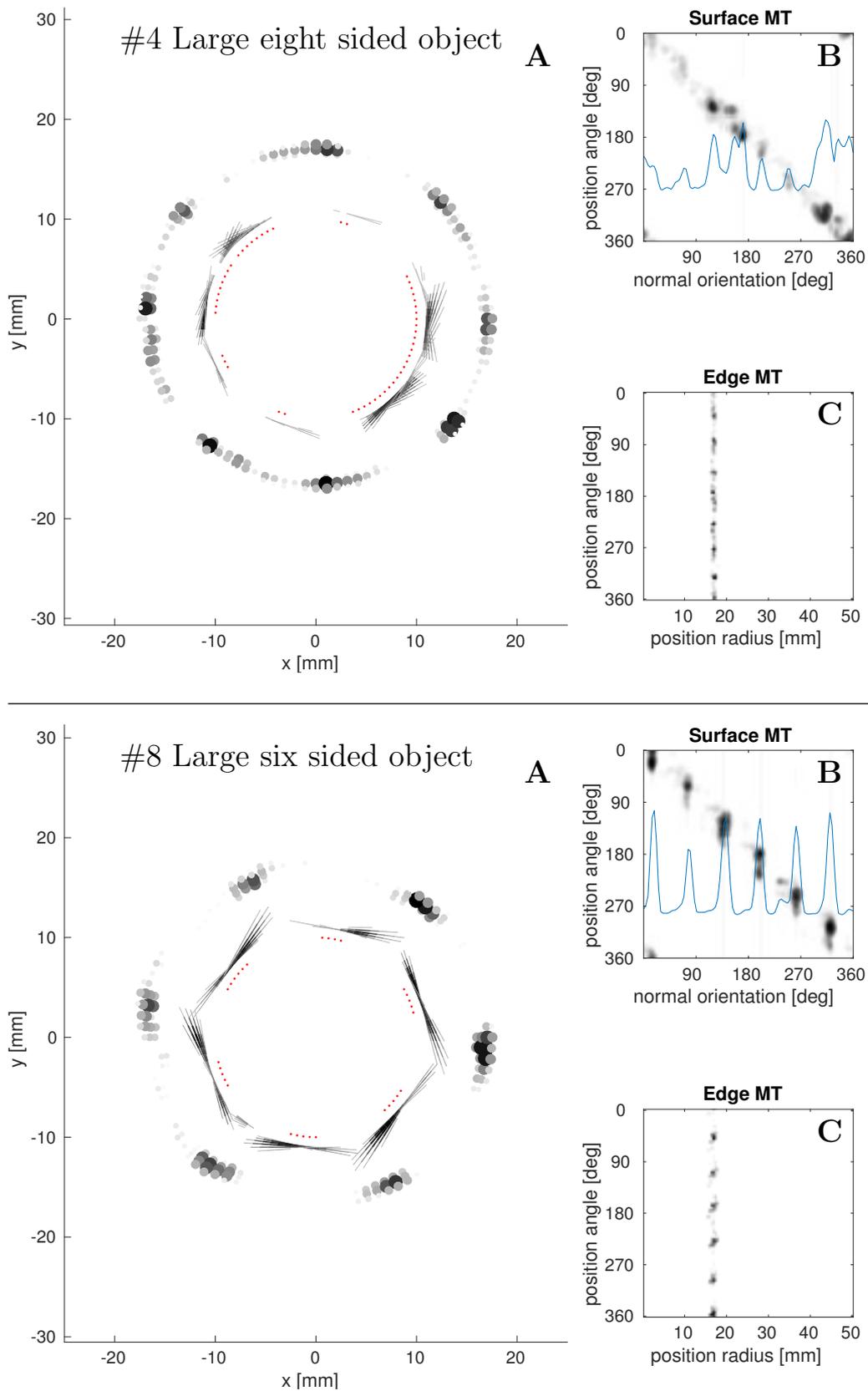


Figure 6.1.2: The object representations after the second iteration of datasets #4 (Top) and #8 (Bottom). See text and caption of Fig. 6.1.1 for further description.

#29 Asym. seven edged object

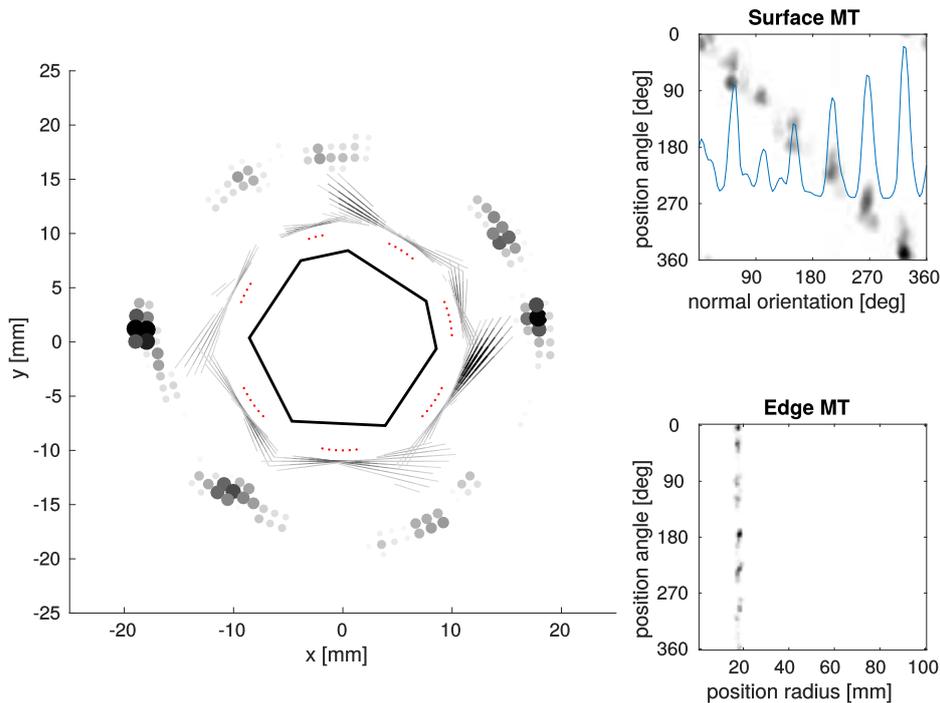


Figure 6.1.3: The object representations after the second iteration of datasets #29. The ground truth object shape was manually inserted into the centre of (A). See text and caption of Fig. 6.1.1 for further description.

A representative visualization of a mapping from the asymmetric object with seven edges is shown in Fig. 6.1.3. Here the ground truth object shape was inserted as a reference. Overall, the object maps obtained from the seven edged object are comparable to the results of the large eight sided object.

When inspecting the visualizations of the results for the five edged asymmetric object, one notices that it is not obvious to determine the quality of a mapping. Two exemplary object mappings are shown in Fig. 6.1.4, together with the ground truth shape. The ground truth shape was orientated in order to fit the generated map best. The mapping of the asymmetric five edged object is difficult to rate as it is not necessarily clear how the curved surfaces should be treated. The model classifies these partly as (weak) edges and / or (weak) surfaces as in #22 and partly does not map them at all as in #21. The parameters of the model remain fixed for all datasets, thus the variations in the feature mapping visible in Fig. 6.1.4 are purely due to trial variance in the tactile patterns of the recorded datasets.

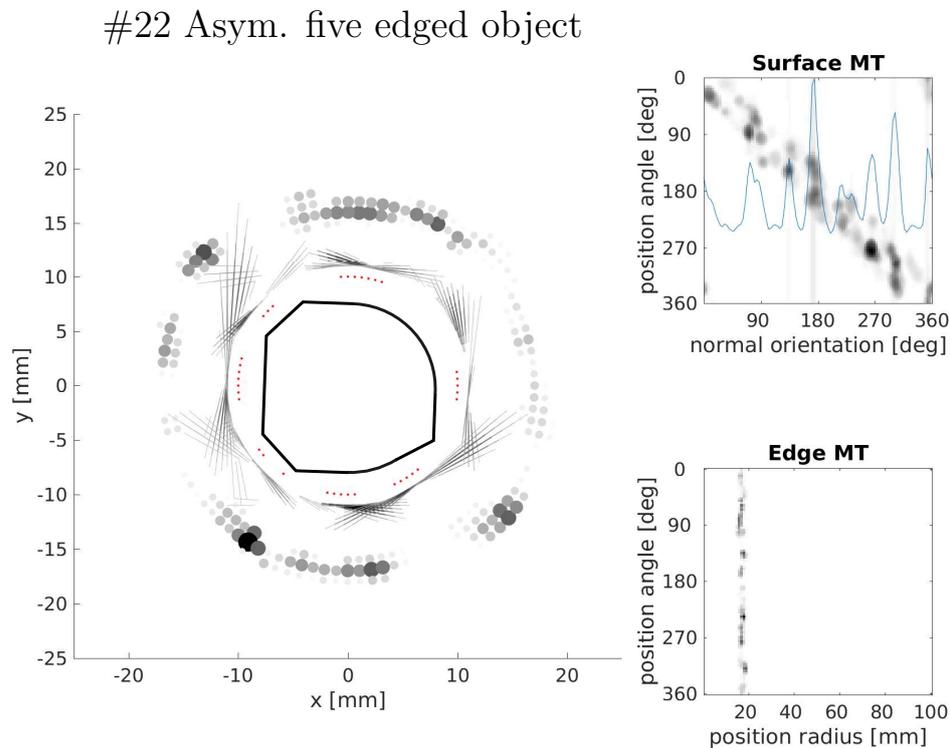
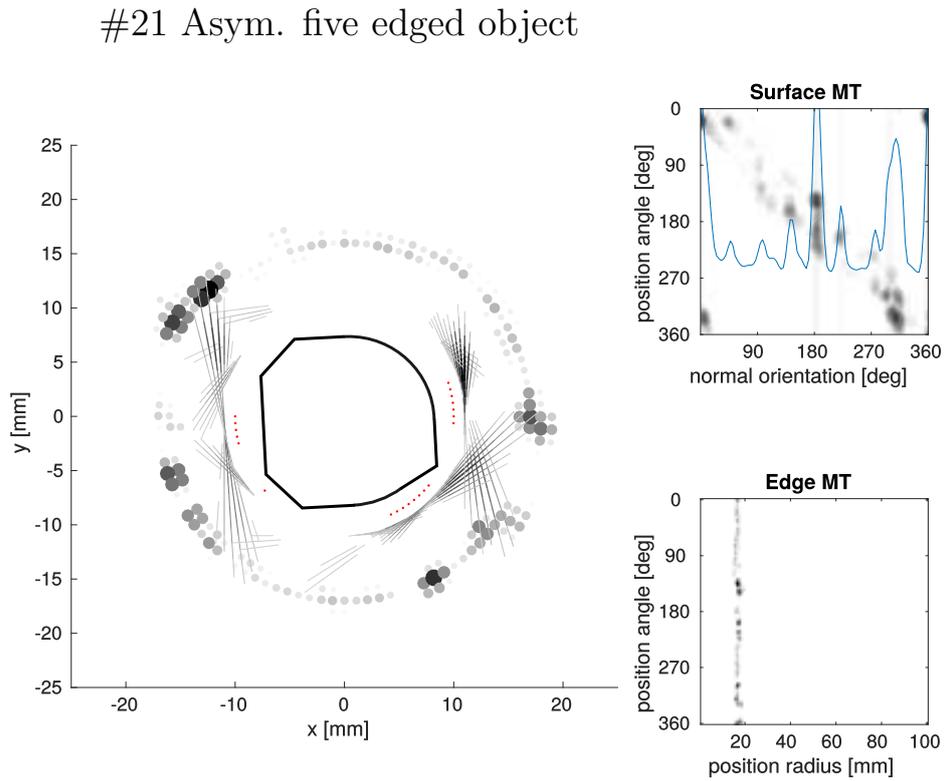


Figure 6.1.4: The object representations after the second iteration of datasets #21 (Top) and #22 (Bottom). The ground truth object shape was manually inserted into the centre of (A). See text and caption of Fig. 6.1.1 for further description.

Temporal development of the object surface representation

Furthermore, the temporal development of the object representation is qualitatively analysed, based on the activity in the surface MT. For this analysis the same exemplary datasets as in the above section are used. At each point in time the surface MT is projected to the surface normal orientation by summing over the position angel dimension. The temporal development of this projection is discussed in the following. The first example here is the dataset #1 from the large eight sided object for which the model performs well, the second example is the dataset #4 from the large eight sided object for which the model does not perform well and the third dataset is #8 from the large six sided object, which is representative for this object. The bottom row in Fig. 6.1.5 shows the MT activation for the same datasets, but with deactivated corrective object pose feedback in the model. For a visualization of all the datasets, see figures Fig. B.0.1 and the following in the appendix Sec. B. The datasets #1 and #8 in Fig. 6.1.5 are mapped in a precise and spatio-temporal consistent fashion, as can be seen in the according plots in the top row. The corresponding plots in the bottom row show, that the clear alignment is due to the matching and pose correction rather than an effect of representation in the model or intrinsic structure of the datasets. This clear detection and representation is not visible in the MT from the dataset #4 (Top row in Fig. 6.1.5). Here the surface detections are very broad in their normal orientation mapping, merging several surfaces into a single peak of activity. Thus, in the MT of dataset #4 the object is temporally represented with five surfaces, instead of eight. The reason is partly visible in the according plot in the bottom row of

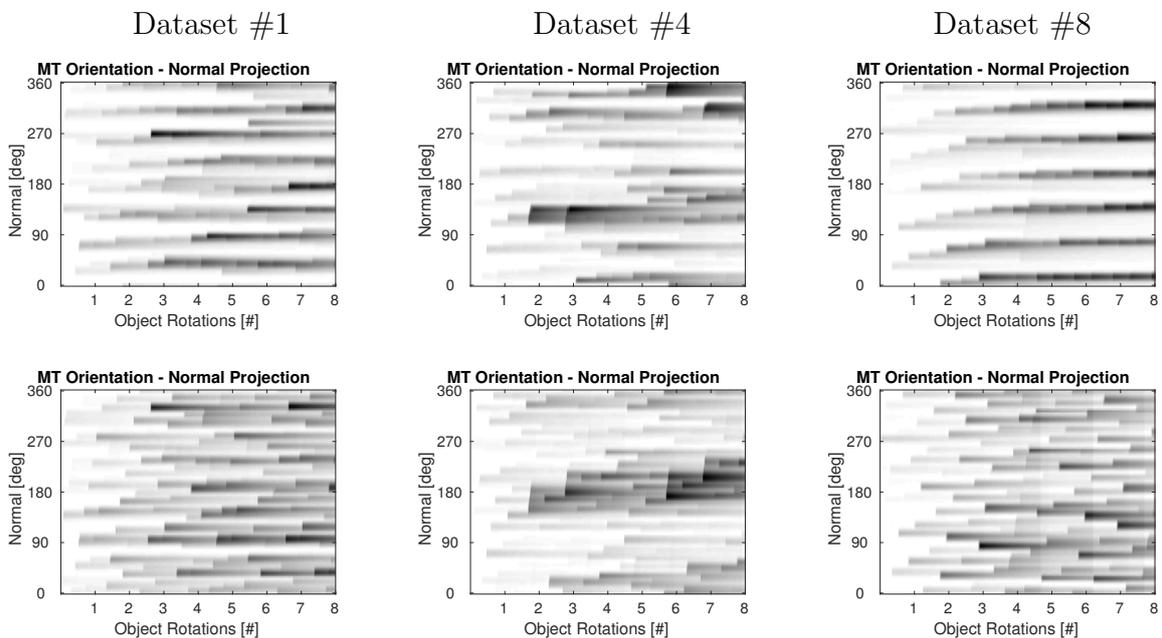


Figure 6.1.5: Exemplary qualitative view of the surface MT for three datasets: # 1 and # 4 (large eight sided object) and # 8 (large six sided object). The plots show the temporal development of the MT, projected to the normal orientation dimension, with the number of estimated full object rotations (i.e. time) on the x axis. The top row shows the results of the model, while the bottom row shows the results when the correction feedback of the model is deactivated, i.e. purely forward estimates of the object dynamics are used.

Fig. 6.1.5: several smeared (with respect to their surface normal orientation) surface detections with large amplitudes. These “smeared” surface detections have their origin in tangential shear forces along the tactile sensor surface during rotation. The shear forces lead to a higher conductivity of the rubber surface of the tactile sensors on the SDH2 which distorts the pressure pattern output of the sensor. This occasional distortion while rotating the object along edges produces broad, nearly round tactile pressure outputs. These then have a high EvR and the curvature feature has a high amplitude for low curvature, indicating an object surface to the neurodynamic model (see also Sec. 3.10). Hence, the reason of these “smeared” surfaces in the bottom, middle plot of Fig. 6.1.5 is a wrong input to the model from the curvature feature extraction. These smeared surfaces are partly stabilized in the model by corrections in the object pose estimate, i.e. the object orientation, thus leading to an error in the object localization which will be discussed in Sec. 6.2.

In Fig. 6.1.6 the temporal development of the surface MTs for the exemplary selected datasets from the asymmetric objects are shown.

In the plot of dataset #22 in Fig. 6.1.6 a change in the pattern of the surface MT activity is visible after the fourth object rotation. This is caused by the discontinuity in the model input, as the dataset is presented a second time. The datasets are recorded until the estimated fourth object rotation by the forward model. The true object rotation varies for every dataset within nearly \pm one object rotation. Hence, the repeated representation of the dataset corresponds to a random re-orientation of the object, which is not predictable by the forward model. In principle, the model will match detected edges or surfaces to the nearest ones in the map (if matched at all). Therefore the re-localization of the object orientation has a limited effect for the symmetric objects, as the model may match the next detected surface to an arbitrary mapped surface without any particular consequences to the object map. However, for the asymmetric objects the particular mapped surface to which the detected one is matched does have consequences. Subsequent object rotation leads to a change of the feature pattern with respect to the mapped (i.e. “expected”). The consequence is an overwrite of the existing map with the “new” feature pattern. This process is visible in the temporal structure of the surface MT plot of dataset #22 in Fig. 6.1.6. As the re-orientation of the object through the re-representation of the dataset is essentially

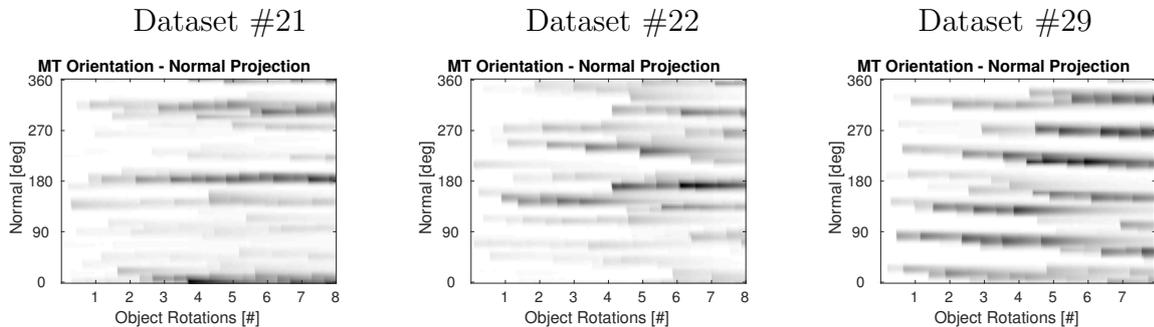


Figure 6.1.6: Exemplary results from the datasets #21 and #22 of the five edged object and #29 of the seven edged object. The plots show a qualitative view of the the temporal development of the surface MT, projected to the normal orientation dimension. The x-axis of the plots shows the number of estimated full object rotations (i.e. time).

random, the effect has a varying impact onto the object representation with respect to the particular dataset.

From the objects with a small diameter the model is only capable to map the six sided object. Out of the five datasets of the small eight sided object none leads to a correct mapping, but results in spatio-temporal inconsistent mappings. The results of the objects with the small diameter are shown in the appendix, see the Sec. B. Concerning the small six sided object, the model performed comparably to the large symmetric objects. The datasets #19 and #20 are erroneously mapped as an five sided object, the other datasets of this object are correctly mapped (see Fig. B.0.17 in the appendix).

6.1.2 Quantitative Evaluation of the Mapping

The qualitative evaluation from the previous subsection is now analysed based on quantitative measures for the quality of an object map in the edge and surface MT of the model. First, an analysis based on the number of surfaces in the surface MT is used as an indicator for rating the object mapping. Then, in a second approach the object mapping in the edge and surface MT is rated with respect to according templates encoding ground truth object shape. While the first measure (# of surfaces) has the advantage of an intuitive interpretation, the second captures the overall shape quality.

The number of mapped surfaces

For quantitative analysis of the model performance, the projected surface MTs from the previous section are analysed by first smoothing by a Gaussian filter along the surface normal orientation dimension, with $\mu = 0$ and $\sigma = 5.4^\circ$. Then all local maxima are counted for each time step via the Matlab function `findpeaks()` to determine the number of detected and mapped surfaces in the model representation. In Fig. 6.1.7 the quantitative results are shown for the three exemplary datasets. Here, the wrong mapping of dataset #4 is clearly reflected in the varying number of detected surfaces. To the contrary, the datasets #1 and #8 are mapped correctly.

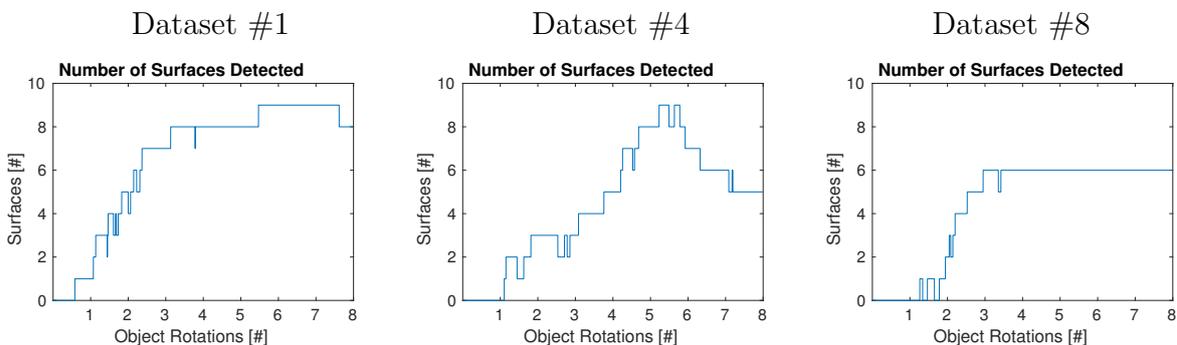


Figure 6.1.7: The number of detected Surfaces, i.e. the number of peaks in the activity of the surface MT for three exemplary datasets: #1 and #4 (large eight sided object) and #8 (large six sided object). The plots show the temporal development of the number of peaks in the MT, when projected to the normal orientation dimension, with the number of estimated full object rotations (i.e. time) on the x axis.

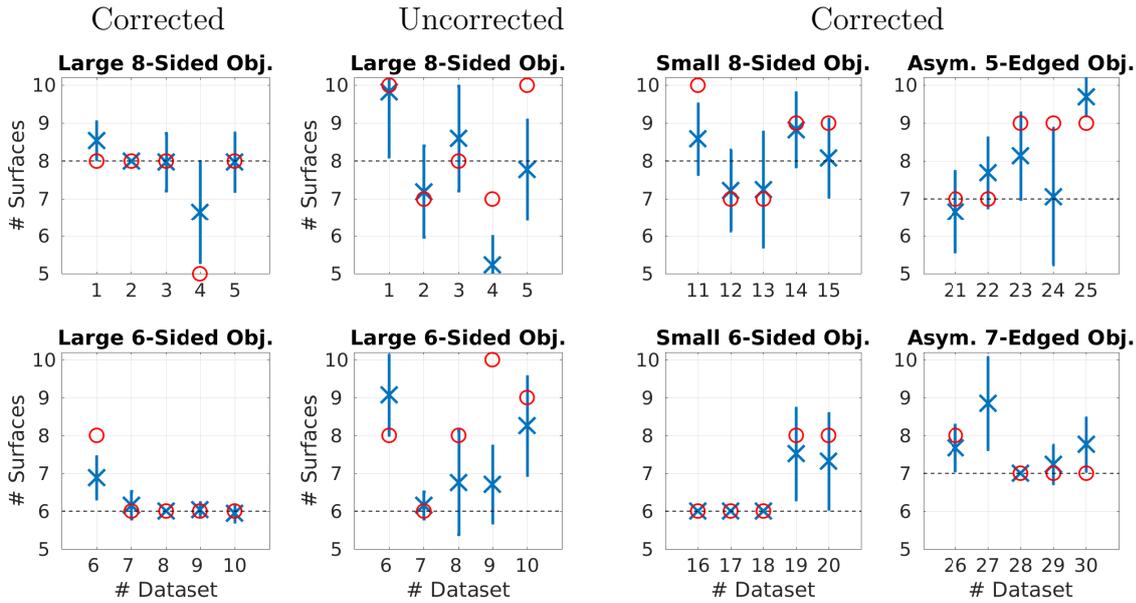


Figure 6.1.8: The mean (blue cross) number of detected surfaces and its standard deviation (error bars) of the model for the according datasets. The red circle is the number of detected surfaces at the end of the trial. Only the last iteration of the dataset is regarded. Left column: the model performance for the large symmetric objects with pose estimate correction. Second column: the same datasets are processed, but without pose correction. Third and fourth column: the datasets of the small symmetric and the asymmetric objects are shown, with pose correction from the model.

The plots in Fig. 6.1.7 visualize the results for the exemplary datasets, which are now quantified for all datasets. In Fig. 6.1.8 the mean result with median and standard deviation is shown across all datasets. Here only the number of detected surfaces for the last iteration through the dataset is included for the evaluation, i.e. the range from # object rotations four to eight in the previous figure Fig. 6.1.7. This ensures that the initial build up phase of the object representation is not included. In Fig. 6.1.8 the mean and standard deviation of detected surfaces are shown for the last half of the datasets, together with the final estimate (red circle). For comparison the same procedure is shown for the uncorrected object pose estimate in the case of the large six and eight sided objects. The uncorrected results for the other datasets are shown in Fig. B.0.7 in the appendix.

In theory, there is a principal tendency of overestimating the number of surfaces because changes in the object map are fast when fusing new information and slow for fading out information. Thus, when the tracking of the object localization is not yet established or is temporally lost (e.g. when the dataset is repeated without a smooth transition) a new feature might be included to the map instead of reinforcing the according old feature. This erroneous mapped feature will increase the number of features until it fades out, although the object might have been re-localized. This tendency can be observed in Fig. 6.1.8, where performance of the model on the individual datasets of these objects is shown.

Concerning the datasets of the large eight and six sided object, the errors in the number of object surfaces in the dataset #4 are clearly visible in the top plot of the

first column in Fig. 6.1.8. In the evaluation of dataset #6 the reason for the increased number of detected surfaces are two surfaces in the MT which have a double peak in their activity surface normal orientation distribution (visible in Fig. B.0.2). However, the resulting object map is clearly a six sided object, as can be verified in Fig. B.0.10 in the appendix. In the uncorrected datasets in the second column of Fig. 6.1.8 it is noteworthy, that dataset #7 has only a very small orientation drift of the true object pose. Therefore, the performance of the model with and without pose correction is comparable. However, this is an exceptional case in the recorded datasets.

The results of the small six and eight sided objects in the third column of Fig. 6.1.8 give the impression of rather small errors, when compared to the errors in the number of detected surfaces of other objects. However, this is illusive, as the number of surfaces is comparably with the the one if the model pose correction is deactivated (visible in the appendix, Fig. B.0.7). This will be discussed in the next section, together with the results of the asymmetric objects.

The objects with a small diameter, in particular the small eight sided object, are hard to map by the object, visible in the third column in Fig. 6.1.8. This may partly be due to the fact, that the temporal period where features are detected and matched is shorter for the objects with small diameter, in particular for the small eight sided object. When the temporal dynamics of the model correction is too slow to realize the orientation correction in the brief time of detecting and matching the feature, the object localization will lag behind.

Measuring quality of the object representation

In addition to the evaluation with respect to the number of detected surfaces in the surface MT, a measure for the overall quality of a mapping is defined.

For this, the activity patterns in the surface and edge MT are matched to templates of the ground truth object shape. The object surface and edges are encoded in templates with a Gaussian distribution with mean at the true position and a standard deviation of 5.4 degrees. The template is then normalized, such that the integrated volume is one. These templates capture the correct activity in the MTs and are shown in the top row of Fig. 6.1.9. Then an inverse template was implemented for measuring the false activity. For this the positive template was dilated by 16 degrees along both dimensions and was then inverted. The dilatation ensures that there is no penalty for activity near the correct position, tolerating small drifts in the object orientation. Activity further off from the correct positions is then captured by the negative templates, shown in the bottom row of Fig. 6.1.9.

Defining a meaningful metric for the quality of a shape representation is difficult, because there is typically a very non-linear relationship between the perceived mapped shape quality and the metric. For large ranges of the metric, only small qualitative differences can be observed and for a small regime in the metric, large differences in quality are perceived. Thus, a shape quality metric is useful only in the limited range near to the optimal reachable fit. In order to determine this regime, all shape fit values will be compared to a baseline of the shape fits of the mode without object pose correction.

The procedure for determining the quality of the shape representation in the surface and edge MT at a given point in time is the following:

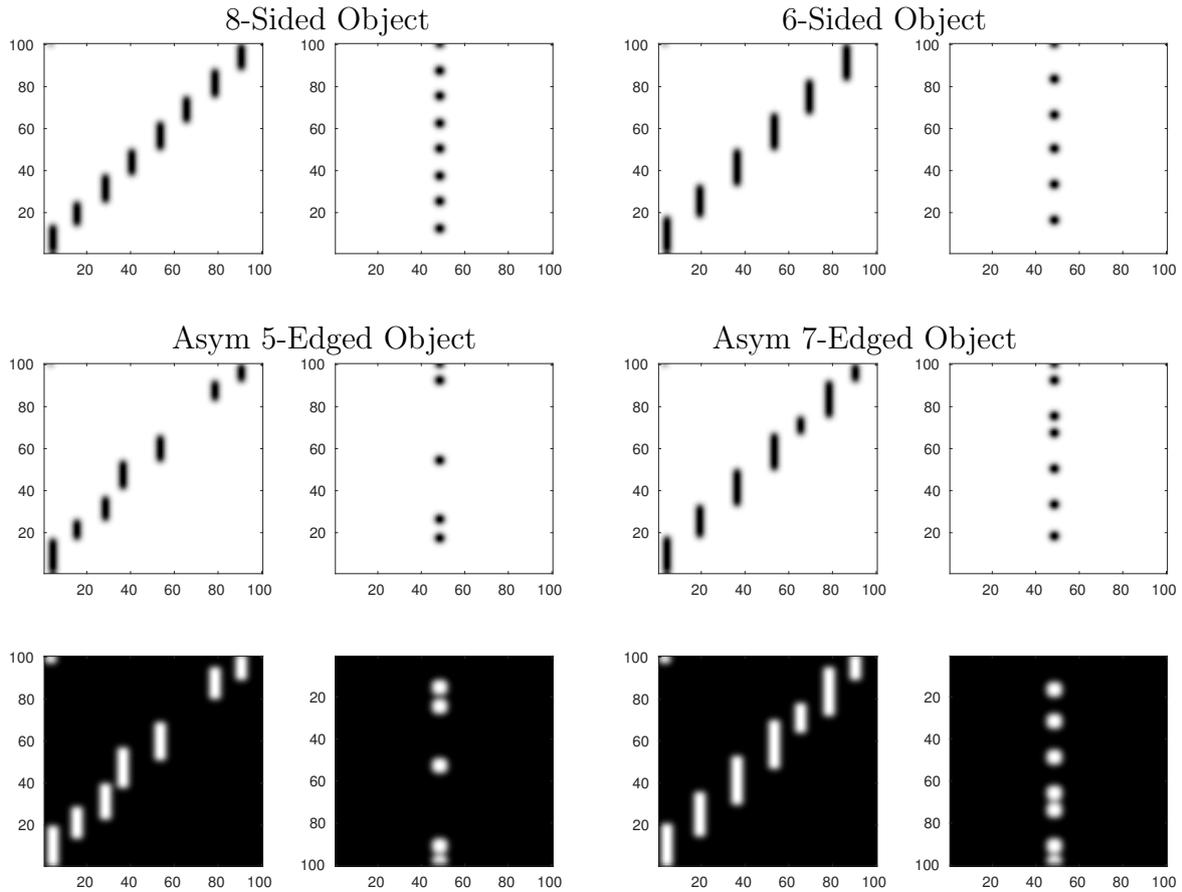


Figure 6.1.9: The shape templates. The Top row shows the positive templates for surfaces and edges for the eight and six sided objects. The left of the template pairs is the surface template and the according right the edge template. The middle row shows the positive templates for the asymmetric objects, the corresponding negative templates of the asymmetric objects are shown in the bottom row. Blue (dark) color denotes low amplitudes and yellow denotes maximum amplitude. Note the small increase of tolerance in the negative counterparts.

- (I) the activity in the edge and surface MT is normalized to $[0, 1]$.
- (II) each template is convolved with the according MT activity, hence four convolutions are done: the positive edge template with the edge MT (true edge), the negative edge template with the edge MT (false edge), the positive surface template with the surface MT (true surface) and the negative surface MT with the surface MT (false surface).
- (III) the positive template convolutions of the surface and edge templates (true positives) are divided by the corresponding negative template convolutions plus one (false positives). Thus, an edge fit ratio (edge fit) and an surface fit ratio (surface fit) are obtained.
- (IV) the maximum of each of these fit ratios is determined with respect to their unique feature dimension, i.e. the best fit with respect to translational and rotational

transformations, resulting in a best (w.r.t. position radius) edge fit and an best (w.r.t. normal orientation) surface fit value.

- (V) the final value of the shape fit is determined by adding the best surface fit and the best edge fit and maximizing with respect to the position angle.

The according equation is given in Eq. (6.1.1), where the value of the object shape template match is determined by

$$\text{Match} = \max_{\text{Pos. Angle}} \left[\max_{\text{Normal Ori.}} \left[\frac{T_{\text{SP}} * MT_{\text{S}}}{T_{\text{SN}} * MT_{\text{S}} + 1} \right] + \max_{\text{Pos. Radius}} \left[\frac{T_{\text{EP}} * MT_{\text{E}}}{T_{\text{EN}} * MT_{\text{E}} + 1} \right] \right] \quad (6.1.1)$$

Here T is the template for surface S and edge E in the positive P and negative N variants. MT denotes the according MT, the $*$ is a two-dimensional circular convolution and the division is a element wise division.

The theoretic range of this shape quality measure is $[0, 2]$, as the maximum of the surface fit and the edge fit can be one, respectively. However, this would require the activity in the MTs to be zero at all wrong locations and have a uniform amplitude at all correct locations. Since the activity in the MTs is exponentially fading out, the activity amplitudes are very heterogeneous – dependent on the temporal history of features fused into the MTs. The effective range of the shape fit measure was approximately $[0, 0.5]$.

In Fig. 6.1.10 the results of the proposed metric are shown for the dataset #1 (large eight sided object). On the left the performance of the proposed model can be seen, in the right plot the performance of the model with deactivated object pose correction

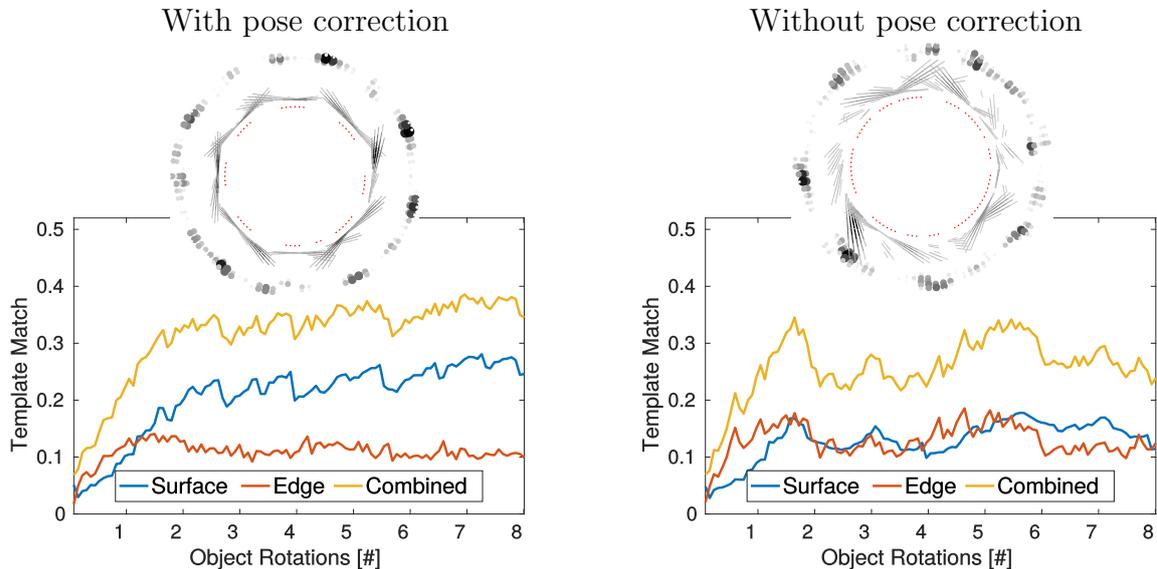


Figure 6.1.10: The shape fit of the surface and edge MTs with respect to ground truth templates for dataset #1 over time. The plot on the left is the fit of the shape template to the surface MT. The plot on the right shows the same, but with deactivated pose correction. Shown are the best fit value (yellow) and the separate components from the best surface fit (blue) and the best edge fit (red) for each step in time (i.e. object rotation on the x-axis). On top of each plot the object shape representation at the end of the trial is shown.

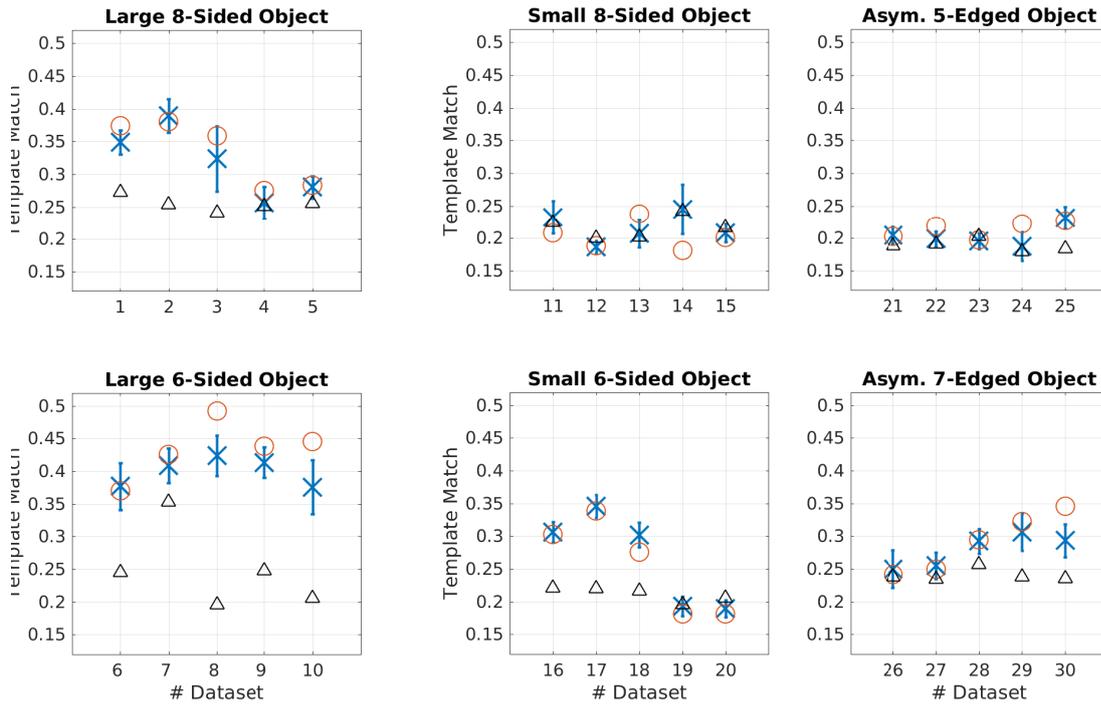


Figure 6.1.11: The shape fit of the surface and edge MTs with respect to ground truth templates for all datasets. The statistics are computed across time for each dataset. Shown are the mean (blue cross), standard deviation (blue error bars) and the fit at the end of the exploration (red circle). The black triangles show the mean template fit for deactivated pose correction, i.e. the baseline. Only the second of the two iterations of the dataset is regarded.

for comparison. Although the difference in the absolute template fit value seems very small, the perceived difference in the mapping quality is rather large. Furthermore, this difference is systematic, as can be seen in Fig. 6.1.11.

The template fit procedure is performed for all datasets, similar to the determination of the number of detected surfaces before. The results for all datasets are shown in Fig. 6.1.11, where the surface and edge MT are evaluated during the second iteration of each dataset. The corresponding development of the template fit measure over time is shown in Fig. 6.1.12.

Like in the previous sections of the mapping evaluation, the large six and eight sided objects are mapped well (left column in Fig. 6.1.12), except for the datasets #4 and #5 of the large eight sided object (left column in Fig. 6.1.11). This is partly also the case for the small six sided object, on which the model has a similar performance then on the large eight sided object (Fig. 6.1.12). This could partly be explained by the fact, that these two objects have the same size of surfaces.

In contrast, the model is incapable to map the small eight sided object. The explanation for this is the combination of a smaller object diameter (compared to the large symmetric and asymmetric objects) and the smaller size of the surfaces. Here the surface size of the small eight sided object corresponds to the surface size of the small surfaces in the asymmetric objects (see Sec. 3.1). The combination of these two facts

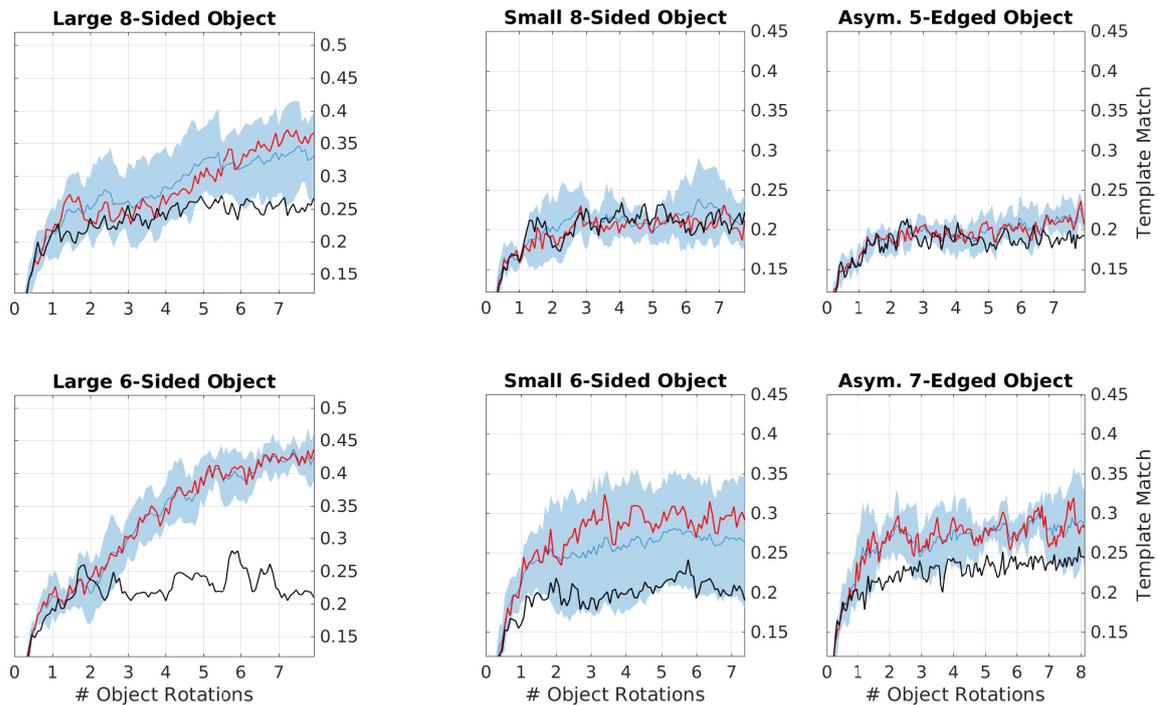


Figure 6.1.12: The shape fit of the surface and edge MTs with respect to ground truth templates for all objects. The statistics are computed across all datasets of an object for each step in time. Shown are the mean (blue cross), standard deviation (blue error bars) and the fit at the end of the exploration (red circle). The black triangles show the median of template fit for deactivated pose correction, i.e. the baseline.

results in too brief sensory inputs to the model which make the detection of surfaces and edges unreliable and degenerate the mapping and localization performance.

In Fig. 6.1.11 it is visible, that in several of the successful mappings of the datasets, the mean and standard deviation are worse than the final shape representation. This tendency is also reflected in the temporal development in Fig. 6.1.12, where the shape fit increases until the end of the trials, in particular in the large eight and six sided objects. Thus, the quality of the map might increase for further object exploration, which also might be the case in the five edged object in Fig. 6.1.12, although not tested.

It should be noted, that the second presentation of the dataset induces a large change in the true object orientation. The consequences of a wrong localization of the object orientation from matching the current detected surface to the nearest in the object map are in particular relevant in the asymmetric objects.

In the plot of the asymmetric seven sided object in Fig. 6.1.11 the results are comparable to the detected number of surfaces, shown previously. While the datasets #28, #29 and #30 show only minor improvements in the shape fit measure, the actual improvements in the object shape representations are rather large (see Fig. B.0.20 and the following in the appendix). Similar holds true concerning the asymmetric five sided object, where the results are once more difficult to interpret. The shape fit measure by

matching templates includes the problem of having to define the curved areas of the asymmetric five edged object. In the templates these were defined as neither surfaces nor edges, thus the model was required to not map them at all. However, the model frequently mapped these areas as edges or surfaces, as discussed previously.

Therefore, the results were additionally classified by a manual rating of the resulting maps which can be found in the appendix Fig. B.0.18 and following.

Overall mapping performance

An overview of the model performance on the datasets is given in Tab. 6.1.1.

For the classification into correct and incorrect mappings, the obtained object map in the model after a full run with the looped datasets is visually inspected and classified into the classes of a correct or incorrect mapping. All of the according maps are visualized in the appendix, see Fig. B.0.8 and the following. Note, that this manual classification is consistent with applying an object specific threshold on the shape fit measure.

In general, the good mapping (and localization) results of the large six sided object are expected. This is because the errors in the predicted object change of the forward model due to noise and unintended object contacts are essentially independent of the number of surfaces of the object. As the errors in the pose predictions are comparable across the datasets and objects, objects with fewer surfaces are in theory easier to map and localize. The correspondence problem becomes easier as there are less and therefore spatially more unique object “landmarks” (i.e. surfaces and edges) which decrease the chance of wrong associations for a fixed prediction error.

From the asymmetric objects the model performed well for three of the five recorded datasets of the seven edged object. The mapping of the asymmetric five edged object is difficult to rate as it is not necessarily clear how the curved surfaces should be treated. The model maps these partly as (weak) edges and partly as (weak) surfaces (see the previous Sec. 6.1.1). Whether this is correct or not is debatable, therefore the overall map from the five edged asymmetric object is rated with respect to its temporal consistency. The maps of datasets #21, #22 and #25 are consistent in the sense that the object orientation of the map does not or only slightly drift during the object exploration. Thus surfaces repeatedly measured are mapped to the same location (and orientation). This was determined by visual inspection of videos¹ showing the temporal development of the object map, the last frame is identical to the according one shown in Fig. 6.1.1. Notably, this procedure lead to classifications consistent with the shape fit measure, although the range of fit values for the asymmetric five edged object is very small.

Interestingly, with the recorded datasets the model either showed a robust convergence of the mapping or performed a permanent re-mapping. This is in particular visible in the figures showing the number of detected surfaces over time of the large symmetric objects and the seven edged asymmetric objects (see Fig. B.0.1 and the following in the appendix Sec. B). In all the experiments only one example was observed (#26), where the model lead to a wrong object map which was clear and spatiotem-

¹The videos are located on the supplementary CD

poral consistent. All other wrong object maps had a strong drift in orientation over time. However, the vice versa cannot be concluded: not all object mappings with a drift in the orientation over time are necessarily wrong.

In general, the results in this subsection show that the object shape representations built in the neurodynamic model are sufficient to map and discriminate several objects. The model failed on the small eight sided object, presumably as the surfaces are in particular small and combined with the small object diameter, features are only measured for a brief period in time. The ability of the model to recover from wrong localization estimates (and thus wrong map entries) by rebuilding the map is demonstrated.

# Dataset	Object	Shape fit	Shape fit +	Correct	Object	Class
1	8-sided large	0.3744	0.1222	✓	60%	80%
2		0.3802	0.1513	✓		
3		0.3584	0.1351	✓		
4		0.2754	-0.0188	✗		
5		0.2836	0.0029	✗		
6	6-sided large	0.3699	0.1263	✓	100%	
7		0.4256	0.0825	✓		
8		0.4922	0.2934	✓		
9		0.4381	0.2209	✓		
10		0.4452	0.2453	✓		
11	8-sided small	0.2079	-0.0111	✗	0%	30%
12		0.1884	-0.0036	✗		
13		0.2375	0.0036	✗		
14		0.1811	-0.0179	✗		
15		0.2003	-0.0229	✗		
16	6-sided small	0.3024	0.1233	✓	60%	
17		0.3387	0.1455	✓		
18		0.2753	0.0620	✓		
19		0.1817	-0.0003	✗		
20		0.1812	-0.0246	✗		
21	5-edged asym.	0.2022	0.0284	✓	80%	70%
22		0.2184	0.0169	✓		
23		0.1969	-0.0132	✗		
24		0.2221	0.0253	✓		
25		0.2266	0.0525	✓		
26	7-edged asym.	0.2411	0.0151	✗	60%	
27		0.2498	-0.0168	✗		
28		0.2948	0.0357	✓		
29		0.3228	0.0703	✓		
30		0.3455	0.1264	✓		

Table 6.1.1: Overview of the performance of the model with respect to the recorded datasets. The “shape fit” column shows the shape fit value at the end of the second dataset iteration, the “shape fit +” column shows the difference with respect to the performance without pose correction, i.e. a base line. The classification into correct (✓) and incorrect (✗) was done by manual visual inspection of the built object map after the 8th object rotation (i.e. at the end of each dataset) and its temporal consistency during the object manipulation. The according object maps are shown in the appendix, in Fig. B.0.8 and the following. The results are furthermore grouped into the object performance and the object class performance.

6.2 Object Localization

The aim of the model is to implement spatiotemporal integration of haptic sensory information into a consistent object shape representation. For this, the model needs to continuously localize the object, however the focus is not on the exact object localization. Therefore, the evaluation of the object localization in this section is less detailed than the previous analysis of the mapping performance. In particular, the object position estimate is quantitatively evaluated and the localization of the object is used to correct the original tactile data and qualitatively visualize it.

6.2.1 Object Position Estimate

The effect of the position correction was implicitly visible in the evaluation of the mapping in the previous section Sec. 6.1. Here, the localization of the object with respect to the DPA is analysed for the large six and eight sided objects by systematically manipulating the initial position estimate. Normally, the object position estimate is initialized with the arithmetic mean of the first object contact positions of both fingers. In order to evaluate the translation correction, the initial estimate of the object position was systematically varied from -1.0 cm to $+1.0$ cm for each of the 10 datasets. This corresponds to a translational error of up to $\pm 25\%$ of the object diameter (4.0 cm). The estimated position by the model is evaluated during four full rotations of the object, i.e. until the object has turned $8 \times \pi$. For each dataset of the large six and eight sided objects the mean position estimate across the five initializations was computed at every step in time. Then the standard deviation across all five initializations with respect to this computed mean position estimate is computed. This gives a mean position estimate and the according standard deviation for each dataset, based on five initializations with different initial object position estimates. Then the mean of the standard deviations is computed across all five datasets of an object, again with the according standard deviation. The mean standard deviation of the position estimate with the according standard deviation of the standard deviation of the position estimate are plotted for both objects in Fig. 6.2.1.

The results in Fig. 6.2.1 show that for all tested datasets the object position estimate converges. Indeed, the position estimates do not converge to the same object

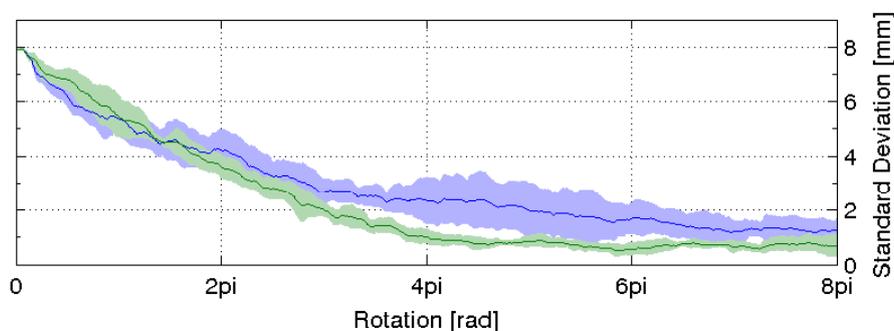


Figure 6.2.1: Temporal development of the mean and standard deviation (across all datasets of an object) of the position estimate standard deviation (across five different initial translation estimates). Blue: the large eight sided object; Green: the large six sided object.

position trajectory across datasets (not shown). This is due to the variability of the object position in the recorded datasets, where the object position was mechanically constrained within a range of approximately $\pm 1.0\text{cm}$ along the distal-proximal translation axis. The other axis was not constrained, as the grasp enforced the object position along the lateral axis. As no ground truth of the object position was tracked during the experiments, the evaluation of the localization is limited to the analysis of the standard deviations.

It should be noted, that the final mean standard deviation of circa 1mm roughly corresponds to the spatial resolution of the position MT with one sampling point per mm.

6.2.2 Pose Corrected Tactile Data

The results of the object localization by the neurodynamic model are used here to correct the recorded tactile features from the according dataset with respect to their location in object space. For this the last two object rotations from the exemplary datasets from the large symmetric objects used in Sec. 6.1.1 are visualized in Fig. 6.2.2.

The raw sensory data are plotted in three variations: first the raw data is plotted according to the predictions of the forward model. This refers to the tactile features transformed from sensory space to object space by relying only on the pose estimates from the forward model (object kinematics, see Sec. 3.6). These features are plotted in the first column of Fig. 6.2.2 for three exemplary datasets. Each feature is composed of two coloured lines: a blue line marks the estimated surface orientation with its length determined by the EvR. A color-coded orthogonal line marks the surface normal, where the color indicates the temporal order in which the features were recorded, starting from teal and ending at magenta. Thus, the features of the two object rotations are distinguishable by their color-code. The second column of the figure shows the tactile features when the object orientation estimate is corrected by the model. The third, right column shows the tactile features when the orientation and position corrections of the model are regarded.

In Fig. 6.2.3 the same procedure is done for the exemplary datasets of the asymmetric objects.

The visualization for all the datasets is given in the appendix, Sec. B. The corrected raw data from dataset #1 show that the orientation correction in principle works however, the slight drift in the orientation of the object representation is visible in the imprecision of the orientation alignment of consecutive mapped surfaces. As to be expected from the previous evaluations, the performance with dataset #4 is poor. In the plots on the bottom row (dataset #8) of Fig. 6.2.2 the effects of the orientation and the position correction are best visible. The localization with respect to the orientation is (from visual inspection) precise and the localization with respect to position accounts for the majority of the errors from the forward model. In most of the tested datasets from the six sided object the correction of the position was only achieved to a certain degree, as is apparent by the remaining small “bumps” at the object edges.

This limitation in position correction is visible even stronger in the data of the asymmetric objects in Fig. 6.2.3.

It is notable, that the errors in the object position estimate responsible for the

”bumps” in the transformed raw data (Fig. 6.2.2 and Fig. 6.2.3) are increasing with the amplitude of the original errors. An explanation could be, that the speed of the position correction in the model is implemented by a dynamical system and is thus limited by its parametrization (e.g. the choice of the time constants, the number of fields, etc.). Therefore, the correction for positional shifts may not be fast enough to fully compensate translations of the object position in time.

In general, the object orientation correction leads to good results when the surfaces are matched correctly. The position estimates have a robust convergence, however with a limited precision. Together with the previous section it is shown that the localization performance of the model is sufficient for a consistent haptic spatiotemporal integration into a map of the object shape.

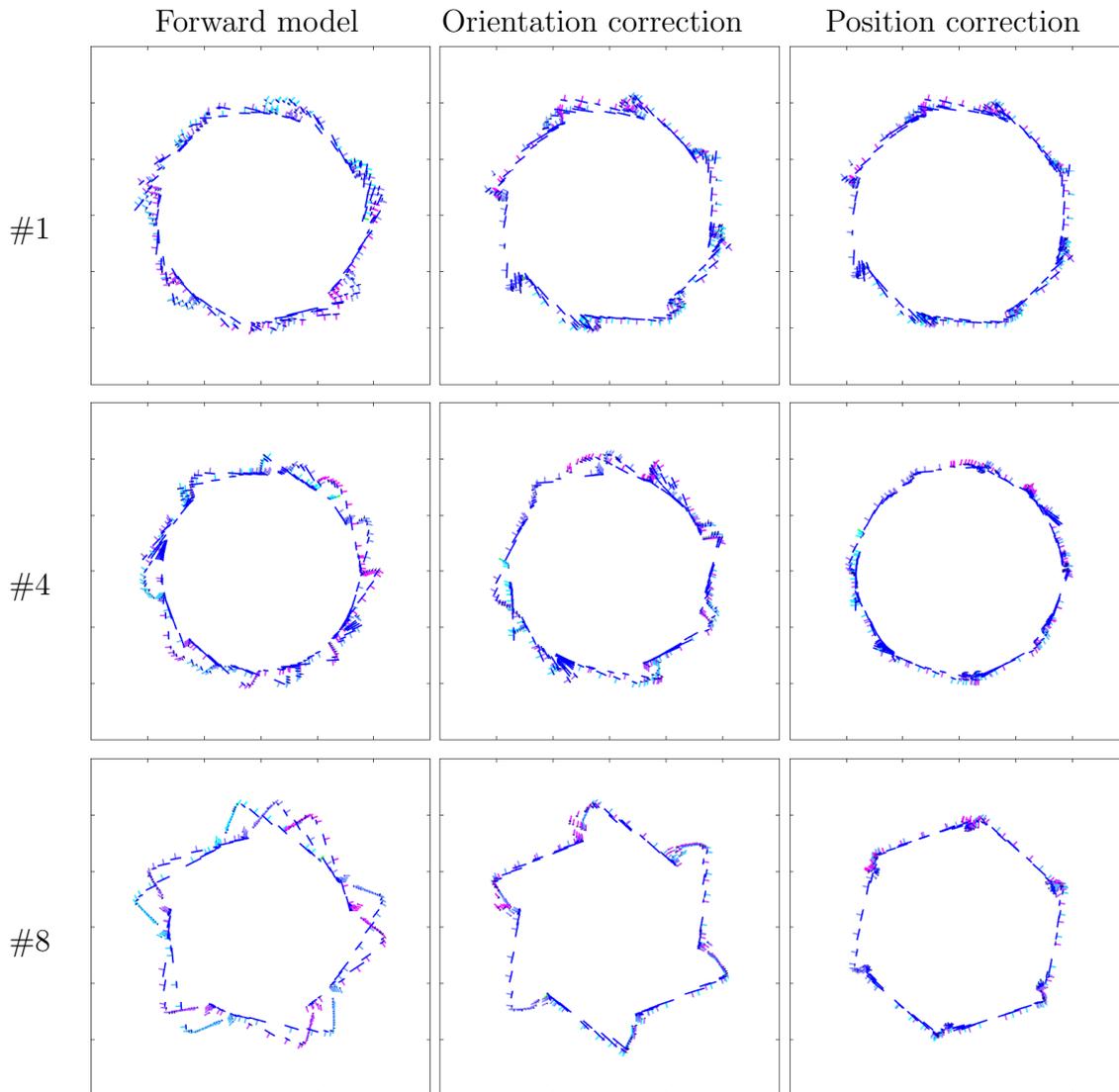


Figure 6.2.2: Tactile features in object space from exemplary datasets: #1, #4 (large eight sided object) and #8 (large six sided object). The left column shows the features based on the object pose estimate from the forward model (input to the neurodynamic model). The middle column includes the orientation correction from the model, the third column additionally includes the position correction. All plots are in Cartesian coordinates, with the x-axis at the bottom and the y-axis on the left. The ticks correspond to 1cm intervals.

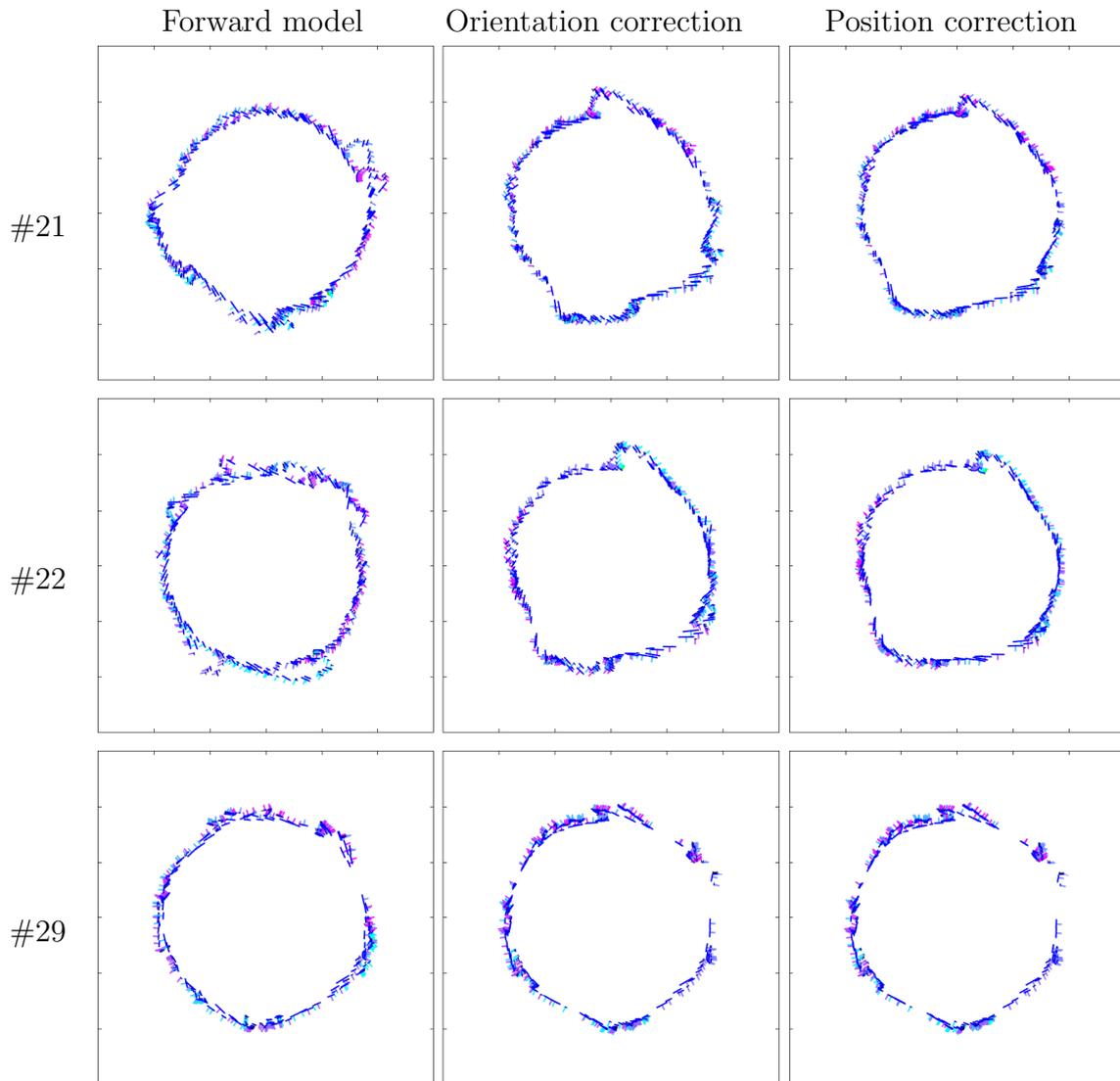


Figure 6.2.3: Tactile features in object space from exemplary datasets: #21, #22 (asymmetric five edged object) and #29 (asymmetric seven edged object). The left column shows the features based on the object pose estimate from the forward model (input to the neurodynamic model). The middle column includes the orientation correction from the model, the third column additionally includes the position correction. All plots are in Cartesian coordinates, with the x-axis at the bottom and the y-axis on the left. The ticks correspond to 1cm intervals.

Chapter 7

Discussion

In this chapter the proposed neurodynamic model will be discussed. Starting with a discussion on a conceptual level of the model in general, it will be set into the context of related work in the subsequent sections. Finally the scientific contribution will be highlighted, followed by a concluding section.

7.1 The Model in General

In this section the proposed model is discussed with respect to its general properties.

Concerning the object shape representation (i.e. the map), it is rather unconventional to use two maps. The motivation for a distributed object representation is partly biologically inspired, as different brain regions have different representations of haptic retrieved object properties (see Sec. 2.1). On the other hand, the distribution of the object representation onto two two-dimensional maps has a computational advantage, as a combined map would be three dimensional (in a space encoding the number of required neurons grows exponentially with the number of dimensions). Only detected surfaces allow for a unique localization with respect to the orientation of the object, assuming convex objects. It is sufficient to only use the surface normal orientation for this orientation localization, as the position along a previously mapped surface can not be determined with certainty. Thus, a one dimensional surface orientation map is sufficient for unique object localization with respect to the orientation. Detected edges are spatially distinct features of two dimensional object mappings (accordingly corners in three dimensional mappings). Therefore, only edges allow a positional localization with respect to a map, when assuming the orientation is known. As the object orientation localization is independent of the object position localization, this assumption is approximately true. Therefore the position localization of the object is dependent on the orientation localization and requires a two dimensional map of previously detected edge positions. The dependency on the orientation is released if two (ore more) features are detected simultaneously, which makes the localization unique.

These two localization mechanisms together span a three dimensional mapping space, each localization mechanism only requires a subspace of the map. Therefore the splitting of the map into a one dimensional and a two dimensional map is sufficient. Then each localization is a unique localization of the feature in the according map.

Although a one dimensional surface map is in principal sufficient, the proposed

model has a two dimensional surface map featuring the additional position angle dimension. This dimension is in particular relevant for an enhanced object mapping, linking surface orientations to a spatial extension in object space and therefore encoding the object boundaries. The additional position angle dimension also implements a direct link between the surface and the edge map which is of relevance for object shape reconstruction. Considering the generalization to three dimensional space, the hypothesis would be that three dimensional surface and edge maps could encode the object boundaries. Both maps would be extended with the additional position angle required for three dimensional polar coordinates.

When discussing the proposed model it should be noted that the object position is encoded in a Gaussian activity distribution, however it is not implemented as a dynamical system like the remaining model. This is because of the temporal discrete readout of tactile measurements and the algorithmic implementation of the forward model (see Sec. 3.6) which predicts a change in the object pose for every measurement. Integrating these predictions over discrete steps in time is implemented by summing these up and setting the mean of the Gaussian pose to the according value. Although this implementation is not done within neural dynamics, this is more a technical issue than a conceptual one.

Assumptions and Limitations

The processing by the model is based on several assumptions.

First, the model assumes convex objects. This is mainly reflected in the orientation localization based on surface orientation normals, independent of the contact position. In order to relax this assumption, the object orientation localization would additionally require the surface position, i.e. taking into account positional information. This would impose an bidirectional dependency, requiring the orientation for the position and the position for the orientation estimates. To determine the consequences from this conceptual change would require further research.

Second, the model utilizes temporal cues for estimating object curvature at contact locations, requiring a constant manipulation speed and a rotational movement of the fingers. In principle, this requirement could be relaxed if the tactile sensors would give more reliable feedback of the local object curvature. Still, an irregular exploration speed would influence the distribution of activity in the memory traces (i.e. higher activity amplitudes for slow exploration phases). Hence, the model performance would be dependent on the particular exploration strategy and its velocity profile. If primates process haptic information in a related fashion, the model would predict that they either try to keep the exploration speed constant or even execute strategies to facilitate the feature formation in the map(s).

The model has clear limitations with respect to the generalizability. The rotation speed of the object manipulation is restricted by the time constants τ in the model, in particular in the MTs. For small objects with smaller surfaces the rotation speed needs to be slower than for large objects with larger surfaces. This limitation is one reason why the model fails to map any of the datasets of the small eight sided object, although the rotation speed in joint space of the SDH2 is constant over all objects. The dynamics of the model are too slow, compared to the temporal dynamics of tactile

pattern formation. From a biological perspective an adaptation of the time constants is implausible as these model neurophysiological properties of the neural substrates. Therefore, the object rotation behaviour would need to be adapted accordingly. In addition, the object manipulation speed determining the overall input durations is also constrained by the correction rates of the object pose. For fast object manipulations the pose estimate cannot be sufficiently corrected as the attractor moves faster than the state of the dynamical system implemented by the model.

This constraint with respect to the object manipulation speed is in particular relevant concerning pauses between object grasps / contacts. For example, changing the arm configuration to grasp the object from a different direction requires time during which the dynamics in the model may fade out and information is lost. This is an intrinsic problem of the model, as the dynamics is not in a self-stabilizing regime and therefore is only input driven. The mapping of the object by the proposed model therefore may serve as a short term spatio-temporal working memory for higher level object representations.

Since the model does not utilize unique spatial features, the object localization will always be ambiguous. This prevents a robust mapping of the asymmetrical objects, as a localization with respect to the wrong feature leads to a re-mapping, i.e. an overwrite of the existing object map.

This ambiguity could be resolved by containing multiple localization (and map) hypothesis which would have to be resolved during the successive object exploration. Although this approach is common in mobile robot navigation, it seems implausible that the nervous system simultaneously tracks multiple of these localization and mapping hypotheses. The other possibility to resolve the ambiguity would be to compute features with a higher complexity or to assemble several features into higher-order features. These higher-order features could be inferred from the information in the maps, which in principle not only hold information on where an edge or surface has been detected. Evaluating the distances, e.g. the difference in surface normal orientations between neighbouring surfaces gives insight into the edge angle or the degree of curvature, depending if an edge has been detected in between the surfaces. The distance between two edges gives information about the surface length in between these. Exploiting these informations could enhance the localization and make it less ambiguous. These higher-order features would be of high relevance since the model as it is proposed in this thesis implements a greedy data association. Measured features are matched to their nearest neighbour, which is especially error prone when the object features are not sparsely distributed in space.

Although the feature complexity in the proposed model is very low and only a single location and map hypothesis is implemented, the model has a high recovery rate from false mapped features. This is apparent in particular in the experiments with the asymmetric objects, where the model recovers from an essentially random change in object orientation due to the representation of the dataset.

The following two subsections of the discussion relate the model to research in psychophysics and neurobiology, and to research in robotics.

7.2 Related Work on Primates

The model implements a prior of rigid objects and uses an estimate of contact curvature based on the shape of the tactile contact area, which is related to human haptic perception [HAYWARD 2008]. However, it is well known that humans and primates in general have access to a wide variety of haptic cues, such as texture, material properties, compliance, weight or temperature which influence perception [LEDERMAN & KLATZKY 2009]. As primates use all of these cues, it is difficult to differentiate on the role of the different kinds of information. The weighting of material properties as e.g. texture, temperature or compliance and geometric cues is known to be task dependent [LEDERMAN & KLATZKY 1993]. However, the proposed model has only access to local geometric cues and therefore a direct comparison to human object identification would require psychophysical experiments with constraints on the exploratory procedure as well as on the cutaneous perception.

In [PANDAY et al. 2012] the authors show that smooth surface and edge cues can interfere and edges seem to dominate, which in turn has a negative effect on the object orientation localization in their experiments. In the proposed model the object orientation localization is done purely based on the surface map, which might not be the case in primates.

Humans are capable to perform path integration of movements inferred from purely tactile cues as shown in [MOSCATELLI et al. 2014]. The subjects were able to temporally integrate the movements to geometric trajectories with high accuracy. Related to haptic object exploration this task would correspond to temporal integration of tactile cues while exploring an object fixed in its position. However, how the human performance changes when a simultaneous object localization has to be performed remains unaddressed.

Presumably somewhere in the posterior parietal cortex (PPC) spatio-temporal integration of haptic information into an object shape representation is performed 2.1. Several brain regions in the PPC have been shown to contribute in this task, as the anterior intraparietal area (AIP) in monkeys and the intraparietal sulcus (IPS) and supramarginal gyrus (ASM) in humans. The proposed neurodynamic model is not intended to model a specific neural substrate in a particular brain region but is rather a biological plausible model of the process of haptic spatiotemporal integration. Shape representation has been shown to be encoded in the population activity of neurons in the visual cortex (V4) [PASUPATHY & CONNOR 2002]. The encoding of shape in neurons of V4 is analogous to the shape coding of neurons in SII [YAU et al. 2009], which makes the findings in [PASUPATHY & CONNOR 2002] in particular interesting for the question of haptic shape encoding. The neuronal responses in V4 encode for boundary features of visually presented objects. In particular, the authors reconstructed two dimensional population codes of angular position over curvature. This object map in V4 is related to the object representation in the proposed model, which also uses two dimensional, object centred maps of population activity with angular position on one of the dimensions. The curvature feature in the model is discretized into two categories: surfaces (low curvature) and edges (high curvature). The reason for this is the poor curvature estimation capabilities with the tactile pressure sensors. However, this discretization is only an approximation as curvature is known to be detected and represented with a

high precision [LOUW et al. 2000]. Furthermore, in the current object representation curved surfaces (e.g. from cylinders) are not mapped which points to the necessity of a complete sampling of the curvature grades. Thus the proposed model only contains a minimal object representation for localizing and mapping an object. Still the shape representation of the surface map is similar to the shape representations identified in [PASUPATHY & CONNOR 2002], as both representations encode a scale and pose invariant object boundary. A slight difference in the proposed model is that the curvature dimension (i.e. second derivative) is replaced by normal orientation (i.e. first derivative). The authors did not use shapes without curved surfaces, i.e. flat surfaces as in the experiments in this thesis, which makes the direct comparison of the object representations difficult. Additionally, the object representation of the model uses the radial position, which is known to be encoded in neurons related to haptic shape processing but has not been explicitly verified as a population encoding jet.

7.3 Related Work in Robotics

In the following the proposed model is related to the probabilistic SLAM field of research.

The object map in the proposed model is a fading memory of Gaussian activations. While the Gaussian in the edge MT do not change in the variance, the variance of the Gaussian in the surface MT depends on the according surface detection. Clear and strong surface detections (i.e. circular tactile pressure patterns) result in sharp and long Gaussian activity distributions, which may move in their surface normal orientation mean as the finger explores large surfaces. In contrast, pressure patterns that are only weakly classified as surfaces will induce lower activity and thus lower amounts of lateral inhibitory interaction, resulting in nearly circular Gaussian distributions with lower amplitude. Hence, the activity in the MT does capture uncertainty with respect to detected features, although not in a strict probabilistic sense. In the subsequent matching of sensed features with the map, the strength of the activity amplitude has an impact on how large the range for matching is.

The object localization in the model is represented by a Gaussian distribution with uniform variance. This directly corresponds to the representation in classic probabilistic models, although the distribution in the model is not altered in its variance and thus the level of uncertainty remains fixed. Thus the representation of the object pose in the model might be better described as a fuzzy representation.

Concerning the data association (correspondence problem) the model implements a validation gate by constrained matching with an approximated mahalanobis distance. This is implemented by a Gaussian interaction kernel and higher DNF input amplitudes for clearer feature detection. The activity in the MT corresponds to the certainty of the according feature in the map. Although the kernel interaction width in the DNF is constant, the mahalanobis distance metric of the validation gate is influenced by the activity amplitudes in the MT. In general, the matching of sensed data with the map in the model is a greedy approach to data association with all the known drawbacks such as its unreliability in densely populated feature regions [BAILEY & DURRANT-WHYTE 2006]. The number of simultaneously measurable features is inherently very limited in the case of haptic object exploration (e.g. maximally two in the proposed setup) just

as the feature complexity computed from tactile pressure sensors. The feature complexity is mainly determined by the sensor type used, thus using deformable sensors – possibly optic tactile sensors [KAPPASSOV et al. 2015] – might increase the feature complexity. However, in the proposed setup the SDH2 tactile sensors were used and thus, the imposed limit on feature complexity.

The only remaining alternative to the greedy matching is then to contain multiple matching hypothesis, requiring a strategy for resolving these during further object exploration. However it is unlikely, that primates contain multiple hypothesis for the location of each contact sensation. Therefore the high error rate of the greedy matching is taken into account in the proposed model, implying strong limitations on the mapping capabilities of the objects manipulated. Whether the model compares to human capabilities in this respect remains to be shown.

Concerning related work in the mobile robotic navigation SLAM field of research, there is in particular one closely related approach to mapping, introduced in [LI et al. 2012]. There a shunting short term memory (STM) [GROSSBERG 1988] is used as map representation, which is closely related to the DNF and MT dynamics. The authors evaluate their approach with a database containing laser scan measurements from a mobile robot in an outdoor environment. The authors in [LI et al. 2012] claim, that the STM activity is representing a confidence in the existence of the according feature attributes and is thus a probabilistic representation of the environment. In principle, the same line of argument can be made for the representation in the MTs of the proposed neurodynamic model. However, in [LI et al. 2012] a scan matching algorithm is used for localization, finding the maximum likelihood match for sensory data with the map. In contrast, the proposed neurodynamic model implements a localization with DNFs, which is biologically inspired and in the same framework as the mapping (MT).

In the following, the state of the art approaches focusing on tactile SLAM shall be briefly discussed and related to the proposed model.

The particle filter approaches to the SLAM problem are in principal unsuitable for a purely haptics based setup, as sensory input is only available for a spatially very constrained action outcome. Only for direct contact to the object measurements can be made and thus noise in the particle location tends to degenerate the particle population over time. A modification for particle filters has recently been proposed in [KOVAL et al. 2013] to cope with the spatial sparsity of contact informations. However in [KOVAL et al. 2013] the objects remain fixed in their position during the haptic exploration. Nevertheless, particle filters remain a popular approach when considering haptic SLAM scenarios.

In [FOX et al. 2012] the tactile SLAM problem is approached by using artificial whiskers on a mobile robot with a particle filter and a occupancy grid map of a (small) environment. The main idea is the assumption that contacts are always with flat surfaces, hence the contact information is propagated according to its normal orientation to neighbouring cells of the grid map. It should be mentioned, that the tactile whiskers (14cm length) give contact information in a much larger range of surface distance than a pressure sensitive tactile sensor array of robotic grippers. The larger sensory measurement range softens the problem of particle degeneration mentioned earlier. The authors [FOX et al. 2012] demonstrate a proof of concept for a restricted

domain, however the obtained maps are not very precise. Using contact sensors in the proposed setup of this thesis gives the advantage of feedback with respect to the local properties of the contact, e.g. discriminating between edges and surfaces and in particular a precise contact location. The usage of the occupancy map in [Fox et al. 2012] is in principal comparable to the map contained in the surface MT of the proposed model. In general occupancy grids are a method for clustered storage of raw sensory data, in particular the contact positions. However, as [Fox et al. 2012] include the assumption of contacts with surfaces and fuse Gaussian “blobs” with large variance in the estimated surface orientation direction into the grid map, the map has a similar information content as the surface MT in the proposed neurodynamic model (despite different coordinate systems and the fading memory). However, the distinction of an edge and surface map gives the neurodynamic model an advantage. This is in particular true, as a surface map for itself is suboptimal for positional localization when only one surface can be detected at a time.

Recently there have been some publications that claim to solve haptic SLAM, which are reviewed in the following. In [ZHANG et al. 2013] the authors claim to solve the haptic SLAM problem with a dynamic Bayesian approach. However, in their evaluation they perform two distinct experiments: a mapping task with fixed object location and a localization task with a known map. In the mapping task the object is one dimensional and features two distinct roughness types which can be directly sensed. The localization task featured a two dimensional triangle object with an intrinsic random walk motion which was “grasped” by two planar tactile surfaces. Both experiments were carried out in simulations. Since no experiment with simultaneous localization and mapping is performed, the authors do not demonstrate a proof of solving a haptic SLAM problem.

In [YU et al. 2015] the authors claim to solve the haptic SLAM problem by optimizing a parametric polygon object model with respect to the action trajectory and the according sensory measurements. The simulated experiments consider the planar pushing of a two dimensional quadratic object by a probe. The probe implements a contour following exploratory behaviour whereby the object is moved. The movement kinematics are known and contact position and surface orientation are measured. One of the problems here is that the noise seems to be induced only in the observation and motion models (i.e. the motion prediction and sensory measurements) but not in the simulated state (i.e. the actual pose) of the object. As they include the physical model of the simulation into the optimization, the SLAM problem is degenerated. This is also visible in their results, where the errors of the estimated shape rise to chance level (i.e. a trivial solution) when the physical model is excluded. Furthermore no results for the simultaneous application of noise in the motion and observation models are presented, thus a proof of concept remains to be demonstrated as in [ZHANG et al. 2013].

In [BEHBAHANI et al. 2015] the authors claim to solve a haptic SLAM problem with a grid map and a standard particle filter approach. In their simulated experiment a cube is probed at random locations resulting in single contact position features. However the authors have a fixed object pose and only add a Gaussian noise term to the true contact position in every step. This does not lead to a SLAM problem, as there is no drift in the object pose estimate and the performance of their particle filter approach

could probably be matched by temporal averaging of the sensory information. Hence, the authors do not proof to solve a SLAM problem.

These publications show that the problem of haptic SLAM is still an open research question in robotics, as there are in particular very few publications featuring experiments with real robots. The pitfalls in simulations are the artificial induction of noise which is typically Gaussian and often times added to an insufficient number of system variables. Furthermore perfect predictions with superimposed Gaussian noise are not a good model for the complexity of error sources in interactions with real world objects.

The state of the art in haptic SLAM also indicates that the necessity of solutions to haptic SLAM seems not to be strong enough to motivate a large amount of research on this topic. This might be to the fact that in many robotic object manipulation setups a variety of sensors are available, including vision. As vision is frequently used for object localization, the haptic SLAM problem may be circumvented. However, if this will be the case as robotic systems go beyond controlled lab experiments remains to to be seen.

7.4 Scientific Contribution

The first scientific contribution of this thesis is on a conceptual level, understanding the problem that the nervous system is confronted with. Here the contribution is to qualitatively distinguish problem complexities in haptic object exploration into two classes. The first problem class is exploring object shape when the object can be directly localized with respect to the body, e.g. which are fixed in their location or are sustained with a stable grasp during exploration. This includes bi-manual exploration where one of the hands holds the object *without changing the grasp with respect to the object*, while and the other hand explores the object shape. In this case the location of the object is known at every point in time, i.e. can be *directly* measured from the joint angles. Although the measurements of the location may have superimposed measurement noise, the localization error in every point in time is independent of past measurements. Thus the first class of object shape exploration is a pure mapping problem. As soon as the object pose with respect to the hand holding the object changes the problem becomes a SLAM problem, i.e. falls into a second class. This may be the case when e.g. re-grasping the object for further exploration or object slippage due to an unstable grasp. The problem even becomes a SLAM problem, when the object is passed to the other hand while maintaining the object with a stable grasp at all times. Then the object location must be *inferred by temporal integration* of past localization estimates. The error of this integrated localization estimate will impose a systematic error in the mapping of all successive haptic object shape features, if not corrected. Hence, the second class of problems qualitatively differs from the first in that a simultaneous localization and mapping have to be performed. To my knowledge, this conceptual difference in the underlying haptic exploration problems not yet considered in psychophysical and neurobiological experimental setups. This distinction of haptic object shape exploration into a SLAM problem and a simpler mapping problem is relevant for future experimental research.

The second scientific contribution is the development of a neural, dynamic process

model of the haptic spatiotemporal integration. The model is based on principles of neural computation and takes into account the current state of the art of haptic shape processing by the primate nervous system. From a theoretical viewpoint the proposed model has clear limitations. It implements a greedy data association, tracks only a single location and map hypothesis and does not explicitly represent and compute with probabilities. Nevertheless, the model is capable to perform spatiotemporal integration of haptic information into consistent object maps in robotic experiments. Furthermore, the model is capable to perform this in an autonomous, unsupervised and incremental fashion with no distinction of phases like data acquisition, training and exploitation. The proposed model has implications which lead to further research questions.

From neurobiological perspective it would be interesting to analyse whether there are object boundary maps encoding positional radius and surface normal orientation in an population encoding similar to the ones found in the visual cortex [PASUPATHY & CONNOR 2002]. Furthermore, a question would be to what extent humans are capable to build an object shape representation from haptics when the object needs to be localized during object manipulation for exploration. The underlying question is the one of how robust the mapping process is with respect to errors in the localization, e.g. changes in the object location during the release of an object for re-grasping. A further question would be how complex and detailed the shape representation in humans is on the geometric level.

Altogether, this thesis contributes to a deeper understanding of the problem the nervous system solves and gives an impression of the capabilities of a neurodynamic approach to the haptic SLAM problem.

In a side-track of this thesis the autonomous adaptation of the gain and resting level of a dynamic neural field with intrinsic plasticity is introduced. This ensures that the DNF is always in an parameter regime where it is highly sensitive to input, i.e. regularly passes through the detection instability. This adaptation is in particular relevant when the input mean or variance may have a drift over time which could lead to a degeneration of the input encoding and processing in the DNF. Thus, IP prevents an over-saturation and input deprivation of the DNF in the presence of drifts in the first and second order statistical moments of the input.

7.5 Conclusion

This thesis focuses on the problem of haptic spatiotemporal integration and how in particular the primate nervous system could solve it. Although there is neurobiological and psychophysical research on haptic shape processing, the focus there is often on static processing of (possibly moving) stimuli. The first major contribution of this thesis is to analyse the underlying problem on a conceptual level and to make an explicit distinction between a pure mapping and a simultaneous localization and mapping problem (SLAM). This distinction is relevant for the transfer and generalization of experimental research results as the problems differ in their complexities. Nevertheless, past research has revealed several types of haptic features and their representations, as well as processing stages and multimodal interactions. Key brain areas haven been identified and theories on the processing pathways and hierarchies are elaborate. However, there is hardly any knowledge on how spatiotemporal integration of

haptic sensations during object manipulations is performed by the primate nervous system. To contribute to this research question, in this thesis a neurodynamic model is proposed as an exemplary approach to solving the problem of haptic spatiotemporal integration in a SLAM scenario.

The neurodynamic model considers the current knowledge of haptic shape processing in primates and is limited to computational principles from the nervous system. The process of haptic spatiotemporal integration is implemented as a dynamical system, which is in particular suited to model temporal information processing. The object shape representation is modelled within two feature maps encoding the object boundaries, a surface and an edge map. Similar maps have been identified in the visual cortex, which is known to have a shape encoding related to the haptic encoding in the secondary somatosensory cortex.

In general, the proposed model is confronted with a task of high intrinsic complexity, as discussed in the introduction Sec. 1.2. The capabilities and limits of the proposed model were explored and discussed in the context of the SLAM problem. In experiments with data from a robotic hand the model was able to build object maps fast (within \approx four object rotations) and with a robust recovery from erroneously mapped object areas. For the datasets where the model failed to build a consistent map, this often lead to a strong drift of the object map orientation as the object is continuously remapped. The major limitations on the mapping performance are due to the implementation of a greedy data association and a single hypothesis for the location and map. Thus the model can only perform a localization within a local parameter regime (i.e. rotation and translation), making it prone to local minima. In addition, there is the haptic specific problem of low numbers of simultaneous measurable features from different object locations and the sensor specific limitation on the complexity of tactile features. In combination these lead to a restricted localization capability, limiting the object shape mapping to regimes with moderate errors in the object pose predictions.

So what is this model good for? In a robotic scenario, the object shape representation in the proposed model could in principle be utilized for improving grasp precision an object manipulation, as it is encoded in a metric map. Furthermore, metric maps could also be used for predicting tactile features when combined with a body model. However, the main value of the model is the insight into the problem structure of haptic spatiotemporal integration and its possible solution. The model has implications which may be tested to gain further insights into the capabilities and mechanisms of haptic processing in the primate nervous system, discussed in the previous section Sec. 7.4.

Chapter 8

Bibliography

- [AMARI 1977] Amari, S. (1977). *Dynamics of pattern formation in lateral-inhibition type neural fields*. *Biological Cybernetics*, 27, 77–87.
- [AMARI & DOUGLAS 1998] Amari, S. & Douglas, S. (1998). *Why natural gradient?*. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, (pp. 1213–1216). IEEE.
- [AMARI 1998] Amari, S. (1998). *Natural gradient works efficiently in learning*. *Neural computation*, 10(2), 251–276.
- [AMEDI et al. 2002] Amedi, A., Jacobson, G., Hendler, T., Malach, R., & Zohary, E. (2002). *Convergence of visual and tactile shape processing in the human lateral occipital complex*. *Cerebral Cortex*, 12(11), 1202–1212.
- [ANTONELO & SCHRAUWEN 2012] Antonelo, E. & Schrauwen, B. (2012). *Learning slow features with reservoir computing for biologically-inspired robot localization*. *Neural Networks*, 25, 178–190.
- [ARGALL et al. 2009] Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). *A survey of robot learning from demonstration*. *Robotics and autonomous systems*, 57(5), 469–483.
- [AUGURELLE et al. 2003] Augurelle, A.-S., Smith, A. M., Lejeune, T., & Thonnard, J.-L. (2003). *Importance of cutaneous feedback in maintaining a secure grip during manipulation of hand-held objects*. *Journal of Neurophysiology*, 89(2), 665–671.
- [AZAÑÓN et al. 2010] Azañón, E., Longo, M. R., Soto-Faraco, S., & Haggard, P. (2010). *The posterior parietal cortex remaps touch into external space*. *Current Biology*, 20(14), 1304–1309.
- [BAILEY & DURRANT-WHYTE 2006] Bailey, T. & Durrant-Whyte, H. (2006). *Simultaneous localization and mapping (SLAM): Part II*. *Robotics & Automation Magazine, IEEE*, 13(3), 108–117.
- [BAILEY et al. 2006] Bailey, T., Uzcroft, B., & Durrant-Whyte, H. (2006). *Validation gating for non-linear non-Gaussian target tracking*. In *2006 9th International Conference on Information Fusion*, (pp. 1–6). IEEE.
- [BAR-SHALOM 1987] Bar-Shalom, Y. (1987). *Tracking and data association*: Academic Press Professional, Inc.

- [BEHBAHANI et al. 2015] Behbahani, F. M., Taunton, R., Thomik, A. A., & Faisal, A. A. (2015). *Haptic SLAM for context-aware robotic hand prosthetics-simultaneous inference of hand pose and object shape using particle filters*. In *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, (pp. 719–722). IEEE.
- [BIERBAUM et al. 2008] Bierbaum, A., Gubarev, I., & Dillmann, R. (2008). *Robust shape recovery for sparse contact location and normal data from haptic exploration*. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3200–3205). IEEE.
- [BODEGÅRD et al. 2001] Bodegård, A., Geyer, S., Grefkes, C., Zilles, K., & Roland, P. E. (2001). *Hierarchical processing of tactile shape in the human brain*. *Neuron*, 31(2), 317–328.
- [BOEDECKER et al. 2009a] Boedecker, J., Obst, O., Mayer, N., & Asada, M. (2009a). *Initialization and self-organized optimization of recurrent neural network connectivity*. *HFSP journal*, 3(5), 340–349.
- [BOEDECKER et al. 2009b] Boedecker, J., Obst, O., Mayer, N., & Asada, M. (2009b). *Studies on reservoir initialization and dynamics shaping in echo state networks*. In *Proceedings of the European Symposium on Neural Networks (ESANN)*.
- [BOLOGNINI et al. 2010] Bolognini, N., Papagno, C., Moroni, D., & Maravita, A. (2010). *Tactile temporal processing in the auditory cortex*. *Journal of cognitive neuroscience*, 22(6), 1201–1211.
- [BRANCO & HÄUSSER 2010] Branco, T. & Häusser, M. (2010). *The single dendritic branch as a fundamental functional unit in the nervous system*. *Current Opinion in Neurobiology*, 20(4), 494–502.
- [CARANDINI & HEEGER 2012] Carandini, M. & Heeger, D. J. (2012). *Normalization as a canonical neural computation*. *Nature Reviews Neuroscience*, 13(1), 51–62.
- [CHALON et al. 2013] Chalon, M., Reinecke, J., & Pfanne, M. (2013). *Online in-hand object localization*. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, (pp. 2977–2984). IEEE.
- [CICHY et al. 2011] Cichy, R. M., Chen, Y., & Haynes, J.-D. (2011). *Encoding the identity and location of objects in human LOC*. *Neuroimage*, 54(3), 2297–2307.
- [DAVIDSON et al. 1974] Davidson, P. W., Barnes, J. K., & Mullen, G. (1974). *Differential effects of task memory demand on haptic matching of shape by blind and sighted humans*. *Neuropsychologia*, 12(3), 395–397.
- [DESAI 2003] Desai, N. S. (2003). *Homeostatic plasticity in the CNS: synaptic and intrinsic forms*. *Journal of Physiology-Paris*, 97(4), 391–402.
- [DIJKERMAN & DE HAAN 2007] Dijkerman, H. C. & De Haan, E. H. (2007). *Somatosensory processes subserving perception and action*. *Behavioral and brain sciences*, 30(02), 189–201.
- [DRAGIEV et al. 2011] Dragiev, S., Toussaint, M., & Gienger, M. (2011). *Gaussian process implicit surfaces for shape estimation and grasping*. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, (pp. 2845–2850). IEEE.

-
- [DURRANT-WHYTE & BAILEY 2006] Durrant-Whyte, H. & Bailey, T. (2006). *Simultaneous localization and mapping: part I*. Robotics & Automation Magazine, IEEE, 13(2), 99–110.
- [DURRANT-WHYTE 1988] Durrant-Whyte, H. F. (1988). *Uncertain geometry in robotics*. IEEE Journal on Robotics and Automation, 4(1), 23–31.
- [EL SADDIK et al. 2011] El Saddik, A., Orozco, M., Eid, M., & Cha, J. (2011). *Human Haptic Perception*. In *Haptics Technologies*, (pp. 45–66).: Springer.
- [FAION et al. 2015] Faion, F., Zea, A., Steinbring, J., Baum, M., & Hanebeck, U. D. (2015). *Recursive Bayesian pose and shape estimation of 3D objects using transformed plane curves*. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2015*, (pp. 1–6). IEEE.
- [FAUBEL & SCHÖNER 2008] Faubel, C. & Schöner, G. (2008). *Learning to recognize objects on the fly: a neurally based dynamic field approach..* Neural networks : the official journal of the International Neural Network Society, 21(4), 562–76.
- [FOX et al. 2012] Fox, C., Evans, M., Pearson, M., & Prescott, T. (2012). *Tactile SLAM with a biomimetic whiskered robot*. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, (pp. 4925–4930). IEEE.
- [FRICK & JOHNSTON 2005] Frick, A. & Johnston, D. (2005). *Plasticity of dendritic excitability*. Journal of neurobiology, 64(1), 100–115.
- [GIRÃO et al. 2013] Girão, P. S., Ramos, P. M. P., Postolache, O., & Pereira, J. M. D. (2013). *Tactile sensors for robotic applications*. Measurement, 46(3), 1257–1271.
- [GOODWIN et al. 1997] Goodwin, A., Macefield, V., & Bisley, J. (1997). *Encoding of object curvature by tactile afferents from human fingers*. Journal of neurophysiology, 78(6), 2881–2888.
- [GRILL-SPECTOR et al. 2001] Grill-Spector, K., Kourtzi, Z., & Kanwisher, N. (2001). *The lateral occipital complex and its role in object recognition*. Vision research, 41(10), 1409–1422.
- [GROSSBERG 1988] Grossberg, S. (1988). *Nonlinear neural networks: Principles, mechanisms, and architectures*. Neural Networks, 1(1), 17–61.
- [HARRIS et al. 2002] Harris, J. A., Miniussi, C., Harris, I. M., & Diamond, M. E. (2002). *Transient storage of a tactile memory trace in primary somatosensory cortex*. The Journal of neuroscience, 22(19), 8720–8725.
- [HAYWARD 2008] Hayward, V. (2008). *Haptic shape cues, invariants, priors and interface design*. In *Human haptic perception: Basics and applications*, (pp. 381–392).: Springer.
- [HSIAO 2008] Hsiao, S. (2008). *Central mechanisms of tactile shape perception*. Current opinion in neurobiology, 18(4), 418.
- [HSIAO & GOMEZ-RAMIREZ 2011] Hsiao, S. & Gomez-Ramirez, M. (2011). *Neurobiology of Sensation and Reward - Chap. 7: Touch*. Frontiers in Neuroscience: CRC Press.
- [HUNG et al. 2012] Hung, C.-C., Carlson, E. T., & Connor, C. E. (2012). *Medial Axis Shape Coding in Macaque Inferotemporal Cortex*. Neuron, 74(6), 1099–1113.

- [IBERALL et al. 1986] Iberall, T., Bingham, G., & Arbib, M. (1986). *Opposition space as a structuring concept for the analysis of skilled hand movements*. Experimental brain research series, 15, 158–173.
- [JAMES et al. 2007] James, T. W., Kim, S., & Fisher, J. S. (2007). *The neural basis of haptic object processing*. Canadian Journal of Experimental Psychology, 61(3), 219–229.
- [JANSSEN et al. 2000] Janssen, P., Vogels, R., & Orban, G. (2000). *Three-dimensional shape coding in inferior temporal cortex*. Neuron, 27(2), 385–97.
- [JOHANSSON & FLANAGAN 2009] Johansson, R. S. & Flanagan, J. R. (2009). *Coding and use of tactile signals from the fingertips in object manipulation tasks*. Nature Reviews Neuroscience, 10(5), 345–359.
- [JOHNSON 2002] Johnson, K. (2002). *Neural basis of haptic perception*. Stevens’ handbook of experimental psychology.
- [KAPPASSOV et al. 2015] Kappassov, Z., Corrales, J.-A., & Perdereau, V. (2015). *Tactile sensing in dexterous robot hands* Review. Robotics and Autonomous Systems, 74, 195–220.
- [KITADA et al. 2003] Kitada, R., Kochiyama, T., Hashimoto, T., Naito, E., & Matsumura, M. (2003). *Moving tactile stimuli of fingers are integrated in the intraparietal and inferior parietal cortices*. Neuroreport, 14(5), 719–724.
- [KLATZKY et al. 1985] Klatzky, R. L., Lederman, S. J., & Metzger, V. A. (1985). *Identifying objects by touch: An expert system*. Perception & psychophysics, 37(4), 299–302.
- [KOHONEN 1982] Kohonen, T. (1982). *Self-organized formation of topologically correct feature maps*. Biological cybernetics, 43(1), 59–69.
- [KOPRINKOVA-HRISTOVA & PALM 2011] Koprinkova-Hristova, P. & Palm, G. (2011). *ESN intrinsic plasticity versus reservoir stability*. Artificial Neural Networks and Machine Learning–ICANN 2011, (pp. 69–76).
- [KOVAL et al. 2013] Koval, M. C., Dogar, M. R., Pollard, N., & Srinivasa, S. (2013). *Pose Estimation for Contact Manipulation with Manifold Particle Filters*. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.
- [KRAVITZ et al. 2011] Kravitz, D. J., Saleem, K. S., Baker, C. I., & Mishkin, M. (2011). *A new neural framework for visuospatial processing*. Nature Reviews Neuroscience, 12(4), 217–230.
- [KRAVITZ et al. 2013] Kravitz, D. J., Saleem, K. S., Baker, C. I., Ungerleider, L. G., & Mishkin, M. (2013). *The ventral visual pathway: an expanded neural framework for the processing of object quality*. Trends in cognitive sciences.
- [KULLBACK & LEIBLER 1951] Kullback, S. & Leibler, R. (1951). *On information and sufficiency*. The Annals of Mathematical Statistics, 22(1), 79–86.
- [LAZAR et al. 2007] Lazar, A., Pipa, G., & Triesch, J. (2007). *Fading memory and time series prediction in recurrent networks with different forms of plasticity*. Neural Networks, 20(3), 312–322.

-
- [LEDERMAN & KLATZKY 1993] Lederman, S. J. & Klatzky, R. L. (1993). *Extracting object properties through haptic exploration*. *Acta psychologica*, 84(1), 29–40.
- [LEDERMAN & KLATZKY 2009] Lederman, S. J. & Klatzky, R. L. (2009). *Haptic perception: A tutorial*. *Attention, Perception, & Psychophysics*, 71(7), 1439–1459.
- [LI et al. 2012] Li, Y., Li, S., & Ge, Y. (2012). *A biologically inspired solution to simultaneous localization and consistent mapping in dynamic environments*. *Neurocomputing*.
- [LIN et al. 2015] Lin, I.-C., Okun, M., Carandini, M., & Harris, K. D. (2015). *The nature of shared cortical variability*. *Neuron*, 87(3), 644–656.
- [LOMP et al. 2013] Lomp, O., Zibner, S. K. U., Richter, M., Ranó, I., & Schöner, G. (2013). *A software framework for cognition, embodiment, dynamics, and autonomy in robotics: cedar*. In *International Conference on Artificial Neural Networks*, (pp. 475–482). Springer.
- [LOSONCZY et al. 2008] Losonczy, A., Makara, J., & Magee, J. (2008). *Compartmentalized dendritic plasticity and input feature storage in neurons*. *Nature*, 452(7186), 436–441.
- [LOUW et al. 2000] Louw, S., Kappers, A. M., & Koenderink, J. J. (2000). *Haptic detection thresholds of Gaussian profiles over the whole range of spatial scales*. *Experimental brain research*, 132(3), 369–374.
- [LUO et al. 2015] Luo, S., Mou, W., Althoefer, K., & Liu, H. (2015). *Localizing the Object Contact through Matching Tactile Features with Visual Map*. ICRA.
- [MAKARA et al. 2009] Makara, J., Losonczy, A., Wen, Q., & Magee, J. (2009). *Experience-dependent compartmentalized dendritic plasticity in rat hippocampal CA1 pyramidal neurons*. *Nature neuroscience*, 12(12), 1485–1487.
- [MARKOVIĆ & GROS 2010] Marković, D. & Gros, C. (2010). *Self-Organized Chaos through Polyhomeostatic Optimization*. *Physical Review Letters*, 105(6), 068702.
- [MARKOVIĆ & GROS 2012] Marković, D. & Gros, C. (2012). *Intrinsic adaptation in autonomous recurrent neural networks*. *Neural Computation*, (pp. 1–18).
- [MASSON et al. 2015] Masson, H. L., Bulthé, J., de Beeck, H. P. O., & Wallraven, C. (2015). *Visual and Haptic Shape Processing in the Human Brain: Unisensory Processing, Multi-sensory Convergence, and Top-Down Influences*. *Cerebral Cortex*, (pp. bhv170).
- [MEIER et al. 2011] Meier, M., Schopfer, M., Haschke, R., & Ritter, H. (2011). *A probabilistic approach to tactile shape reconstruction*. *Robotics, IEEE Transactions on*, 27(3), 630–635.
- [MILFORD et al. 2004] Milford, M. J., Wyeth, G. F., & Prasser, D. (2004). *RatSLAM: a hippocampal model for simultaneous localization and mapping*. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, (pp. 403–408). IEEE.
- [MIQUÉE et al. 2008] Miquée, A., Xerri, C., Rainville, C., Anton, J.-L., Nazarian, B., Roth, M., & Zennou-Azogui, Y. (2008). *Neuronal substrates of haptic shape encoding and matching: a functional magnetic resonance imaging study*. *Neuroscience*, 152(1), 29–39.
- [MONTEMERLO & THRUN 2003] Montemerlo, M. & Thrun, S. (2003). *Simultaneous localization and mapping with unknown data association using FastSLAM*. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, (pp. 1985–1991). IEEE.

- [MONTEMERLO & THRUN 2007] Montemerlo, M. & Thrun, S. (2007). *The SLAM problem*. FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics, (pp. 13–26).
- [MOSCATELLI et al. 2014] Moscatelli, A., Naceri, A., & Ernst, M. O. (2014). *Path integration in tactile perception of shapes*. Behavioural Brain Research, 274, 355–364.
- [NAVARRO et al. 2012] Navarro, S. E., Gorges, N., Worn, H., Schill, J., Asfour, T., & Dillmann, R. (2012). *Haptic object recognition for multi-fingered robot hands*. In *Haptics Symposium (HAPTICS), 2012 IEEE*, (pp. 497–502). IEEE.
- [NEUMANN & STEIL 2011] Neumann, K. & Steil, J. (2011). *Batch intrinsic plasticity for extreme learning machines*. Artificial Neural Networks and Machine Learning–ICANN 2011, (pp. 339–346).
- [NEUMANN & STEIL 2012] Neumann, K. & Steil, J. J. (2012). *Intrinsic Plasticity via Natural Gradient Descent*. In *ESANN*, (pp. 555–560).
- [NEUMANN et al. 2013] Neumann, K., Strub, C., & Steil, J. J. (2013). *Intrinsic plasticity via natural gradient descent with application to drift compensation*. Neurocomputing, 112, 26–33.
- [NORMAN et al. 2008] Norman, J. F., Clayton, A. M., Norman, H. F., & Crabtree, C. E. (2008). *Learning to perceive differences in solid shape through vision and touch*. Perception, 37(2), 185–196.
- [OLIVER et al. 2009] Oliver, R. T., Geiger, E. J., Lewandowski, B. C., & Thompson-Schill, S. L. (2009). *Remembrance of things touched: How sensorimotor experience affects the neural instantiation of object form*. Neuropsychologia, 47(1), 239–247.
- [PANDAY et al. 2012] Panday, V., Tiest, W. M. B., & Kappers, A. M. (2012). *Influence of local properties on the perception of global object orientation*. IEEE transactions on haptics, 5(1), 58–65.
- [PARK et al. 2000] Park, H., Amari, S.-I., & Fukumizu, K. (2000). *Adaptive natural gradient learning algorithms for various stochastic models*. Neural Networks, 13(7), 755–764.
- [PASUPATHY & CONNOR 2002] Pasupathy, A. & Connor, C. E. (2002). *Population coding of shape in area V4*. Nature neuroscience, 5(12), 1332–1338.
- [PEZZEMENTI et al. 2011] Pezzementi, Z., Reyda, C., & Hager, G. D. (2011). *Object mapping, recognition, and localization from tactile geometry*. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, (pp. 5942–5948). IEEE.
- [PLATT et al. 2011] Platt, R., Kaelbling, L., Lozano-Perez, T., & Tedrake, R. (2011). *Simultaneous localization and grasping using belief space planning*. In *In Proceedings of the International Conference on Robotics and Automation (ICRA) 2011 Workshop on Manipulation Under Uncertainty*.
- [POPOVIĆ et al. 2010] Popović, M., Kraft, D., Bodenhagen, L., Bašeski, E., Pugeault, N., Kragic, D., Asfour, T., & Krüger, N. (2010). *A strategy for grasping unknown objects based on co-planarity and colour information*. Robotics and Autonomous Systems, 58(5), 551–565.

-
- [POZO & GODA 2010] Pozo, K. & Goda, Y. (2010). *Unraveling mechanisms of homeostatic synaptic plasticity*. *Neuron*, 66(3), 337–351.
- [PRUSZYNSKI & JOHANSSON 2014] Pruszynski, J. A. & Johansson, R. S. (2014). *Edge-orientation processing in first-order tactile neurons*. *Nature neuroscience*, 17(10), 1404–1409.
- [REED et al. 2004] Reed, C. L., Shoham, S., & Halgren, E. (2004). *Neural substrates of tactile object recognition: an fMRI study*. *Human brain mapping*, 21(4), 236–246.
- [REMY et al. 2010] Remy, S., Beck, H., & Yaari, Y. (2010). *Plasticity of voltage-gated ion channels in pyramidal cell dendrites*. *Current Opinion in Neurobiology*, 20(4), 503–509.
- [RICHTER et al. 2012] Richter, M., Sandamirskaya, Y., & Schöner, G. (2012). *A robotic architecture for action selection and behavioral organization inspired by human cognition*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*.
- [RICHTER et al. 2014] Richter, M., Lins, J., Schneegans, S., Sandamirskaya, Y., & Schöner, G. (2014). *Autonomous neural dynamics to test hypotheses in a model of spatial language*. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, (pp. 2847–2852).
- [ROMERO et al. 2014] Romero, M. C., Pani, P., & Janssen, P. (2014). *Coding of shape features in the macaque anterior intraparietal area*. *The Journal of Neuroscience*, 34(11), 4006–4021.
- [ROMERO et al. 2013] Romero, M. C., Van Dromme, I. C., & Janssen, P. (2013). *The role of binocular disparity in stereoscopic images of objects in the macaque anterior intraparietal area*. *PloS one*, 8(2), e55340.
- [SAAL & BENSMAIA 2014] Saal, H. P. & Bensmaia, S. J. (2014). *Touch is a team effort: interplay of submodalities in cutaneous sensibility*. *Trends in neurosciences*, 37(12), 689–697.
- [SANDAMIRSKAYA 2013] Sandamirskaya, Y. (2013). *Dynamic Neural Fields as a Step Towards Cognitive Neuromorphic Architectures*. *Frontiers in Neuroscience*, 7, 276.
- [SANDAMIRSKAYA & STORCK 2015] Sandamirskaya, Y. & Storck, T. (2015). *Learning to Look and Looking to Remember: A Neural-Dynamic Embodied Model for Generation of Saccadic Gaze Shifts and Memory Formation*. In *Artificial Neural Networks*, (pp. 175–200).: Springer.
- [SCHNEEGANS & SCHÖNER 2012] Schneegans, S. & Schöner, G. (2012). *A neural mechanism for coordinate transformation predicts pre-saccadic remapping..* *Biological cybernetics*, 106(2), 89–109.
- [SCHÖNER 2008] Schöner, G. (2008). *Dynamical Systems Approaches to Cognition*. In Sun, R. (Ed.): *Cambridge Handbook of Computational Cognitive Modeling*, (pp. 101–126)., Cambridge, UK: Cambridge University Press.
- [SCHRAUWEN et al. 2008] Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J., & Stroobandt, D. (2008). *Improving reservoirs using intrinsic plasticity*. *Neurocomputing*, 71(7), 1159–1171.

- [SCHULZ 2006] Schulz, D. (2006). *Plasticity and stability in neuronal output via changes in intrinsic excitability: it's what's inside that counts*. Journal of experimental biology, 209(24), 4821–4827.
- [SCHÜRSMANN et al. 2012] Schürsmann, C., Schöpfer, M., Haschke, R., & Ritter, H. (2012). *A high-speed tactile sensor for slip detection*. In *Towards service robots for everyday environments*, (pp. 403–415): Springer.
- [SLOMOWITZ et al. 2015] Slomowitz, E., Styr, B., Vertkin, I., Milshtein-Parush, H., Nelken, I., Slutsky, M., & Slutsky, I. (2015). *Interplay between population firing stability and single neuron dynamics in hippocampal networks*. Elife, 4, e04378.
- [SMITH 2009] Smith, L. B. (2009). *From Fragments to Geometric Shape Changes in Visual Object Recognition Between 18 and 24 Months*. Current Directions in Psychological Science, 18(5), 290–294.
- [SMITH & CHEESEMAN 1986] Smith, R. C. & Cheeseman, P. (1986). *On the representation and estimation of spatial uncertainty*. The international journal of Robotics Research, 5(4), 56–68.
- [SNOW et al. 2015] Snow, J. C., Goodale, M. A., & Culham, J. C. (2015). *Preserved haptic shape processing after bilateral LOC lesions*. The Journal of Neuroscience, 35(40), 13745–13760.
- [SNOW et al. 2013] Snow, J. C., Strother, L., & Humphreys, G. W. (2013). *Haptic Shape Processing in Visual Cortex*. Journal of Cognitive Neuroscience.
- [SPELKE et al. 2010] Spelke, E., Lee, S. A., & Izard, V. (2010). *Beyond core knowledge: Natural geometry*. Cognitive Science, 34(5), 863–884.
- [SRIVASTAVA et al. 2009] Srivastava, S., Orban, G. A., De Mazière, P. A., & Janssen, P. (2009). *A distinct representation of three-dimensional shape in macaque anterior intraparietal area: fast, metric, and coarse*. The Journal of Neuroscience, 29(34), 10613–10626.
- [STEIL 2007a] Steil, J. (2007a). *Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning*. Neural Networks, 20(3), 353–364.
- [STEIL 2007b] Steil, J. J. (2007b). *Several ways to solve the MSO problem..* In *ESANN*, (pp. 489–494).
- [STEMMLER & KOCH 1999] Stemmler, M. & Koch, C. (1999). *How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate*. nature neuroscience, 2, 521–527.
- [STRUB 2012] Strub, C. (2012). *Permanent intrinsic plasticity of dendritic and neuronal excitability*. Master's thesis, Universitaet Bielefeld, Germany.
- [STRUB et al. 2014a] Strub, C., Woörgötter, F., Ritter, H., & Sandamirskaya, Y. (2014a). *Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study*. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, (pp. 26–33). IEEE.
- [STRUB et al. 2014b] Strub, C., Wörgötter, F., Ritter, H., & Sandamirskaya, Y. (2014b). *Using haptics to extract object shape from rotational manipulations*. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, (pp. 2179–2186). IEEE.

-
- [THEYS et al. 2013] Theys, T., Pani, P., van Loon, J., Goffin, J., & Janssen, P. (2013). *Three-dimensional shape coding in grasping circuits: A comparison between the anterior intraparietal area and ventral premotor area f5a*. *Journal of cognitive neuroscience*, 25(3), 352–364.
- [THEYS et al. 2015] Theys, T., Romero, M. C., van Loon, J., & Janssen, P. (2015). *Shape representations in the primate dorsal visual stream*. *Frontiers in computational neuroscience*, 9, 43.
- [THRUN et al. 2002] Thrun, S. et al. (2002). *Robotic mapping: A survey*. *Exploring artificial intelligence in the new millennium*, (pp. 1–35).
- [TIWANA et al. 2012] Tiwana, M. I., Redmond, S. J., & Lovell, N. H. (2012). *A review of tactile sensing technologies with applications in biomedical engineering*. *Sensors and Actuators A: physical*, 179, 17–31.
- [TRIESCH 2005] Triesch, J. (2005). *A gradient rule for the plasticity of a neurons intrinsic excitability*. *Artificial Neural Networks: Biological Inspirations–ICANN 2005*, (pp. 65–70).
- [TRIESCH 2007] Triesch, J. (2007). *Synergies between intrinsic and synaptic plasticity mechanisms*. *Neural computation*, 19(4), 885–909.
- [TURRIGIANO 2011] Turrigiano, G. (2011). *Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement*. *Annual review of neuroscience*, 34, 89–103.
- [TURRIGIANO 2012] Turrigiano, G. (2012). *Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function*. *Cold Spring Harbor perspectives in biology*, 4(1), a005736.
- [VAN ERP et al. 2010] Van Erp, J. B., Kyung, K.-U., Kassner, S., Carter, J., Brewster, S., Weber, G., & Andrew, I. (2010). *Setting the standards for haptic and tactile interactions: ISOs work*. In *Haptics: Generating and Perceiving Tangible Sensations*, (pp. 353–358).: Springer.
- [VERSTRAETEN et al. 2007] Verstraeten, D., Schrauwen, B., & Stroobandt, D. (2007). *Adapting reservoirs to get Gaussian distributions*. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, (pp. 495–500).
- [WALLRAVEN et al. 2014] Wallraven, C., Bühlhoff, H. H., Waterkamp, S., van Dam, L., & Gaißert, N. (2014). *The eyes grasp, the hands see: Metric category knowledge transfers between vision and touch*. *Psychonomic bulletin & review*, 21(4), 976–985.
- [WARDERMANN & STEIL 2007] Wardermann, M. & Steil, J. (2007). *Intrinsic plasticity for reservoir learning algorithms*. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, (pp. 513–518).
- [WILSON & COWAN 1973] Wilson, H. R. & Cowan, J. D. (1973). *A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue*. *Kybernetik*, 13, 55–80.
- [YAMANE et al. 2008] Yamane, Y., Carlson, E. T., Bowman, K. C., Wang, Z., & Connor, C. E. (2008). *A neural code for three-dimensional object shape in macaque inferotemporal cortex*. *Nature neuroscience*, 11(11), 1352–1360.

- [YAU et al. 2013] Yau, J. M., Connor, C. E., & Hsiao, S. S. (2013). *Representation of tactile curvature in macaque somatosensory area 2*. *Journal of neurophysiology*, 109(12), 2999–3012.
- [YAU et al. 2009] Yau, J. M., Pasupathy, A., Fitzgerald, P. J., Hsiao, S. S., & Connor, C. E. (2009). *Analogous intermediate shape coding in vision and touch*. *Proceedings of the National Academy of Sciences*, 106(38), 16457–16462.
- [YU et al. 2015] Yu, K.-T., Leonard, J., & Rodriguez, A. (2015). *Shape and pose recovery from planar pushing*. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, (pp. 1208–1215). IEEE.
- [ZHANG et al. 2013] Zhang, L., Lyu, S., & Trinkle, J. (2013). *A dynamic bayesian approach to real-time estimation and filtering in grasp acquisition*. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, (pp. 85–92). IEEE.
- [ZIBNER et al. 2011] Zibner, S. K. U., Faubel, C., Iossifidis, I., & Schöner, G. (2011). *Dynamic Neural Fields as Building Blocks for a Cortex-Inspired Architecture of Robotic Scene Representation*. *IEEE Transactions on Autonomous Mental Development*, 3(1), 74–91.

Appendix A

Appendix: DNF with IP

the plots from the DNF IP with high initial bias

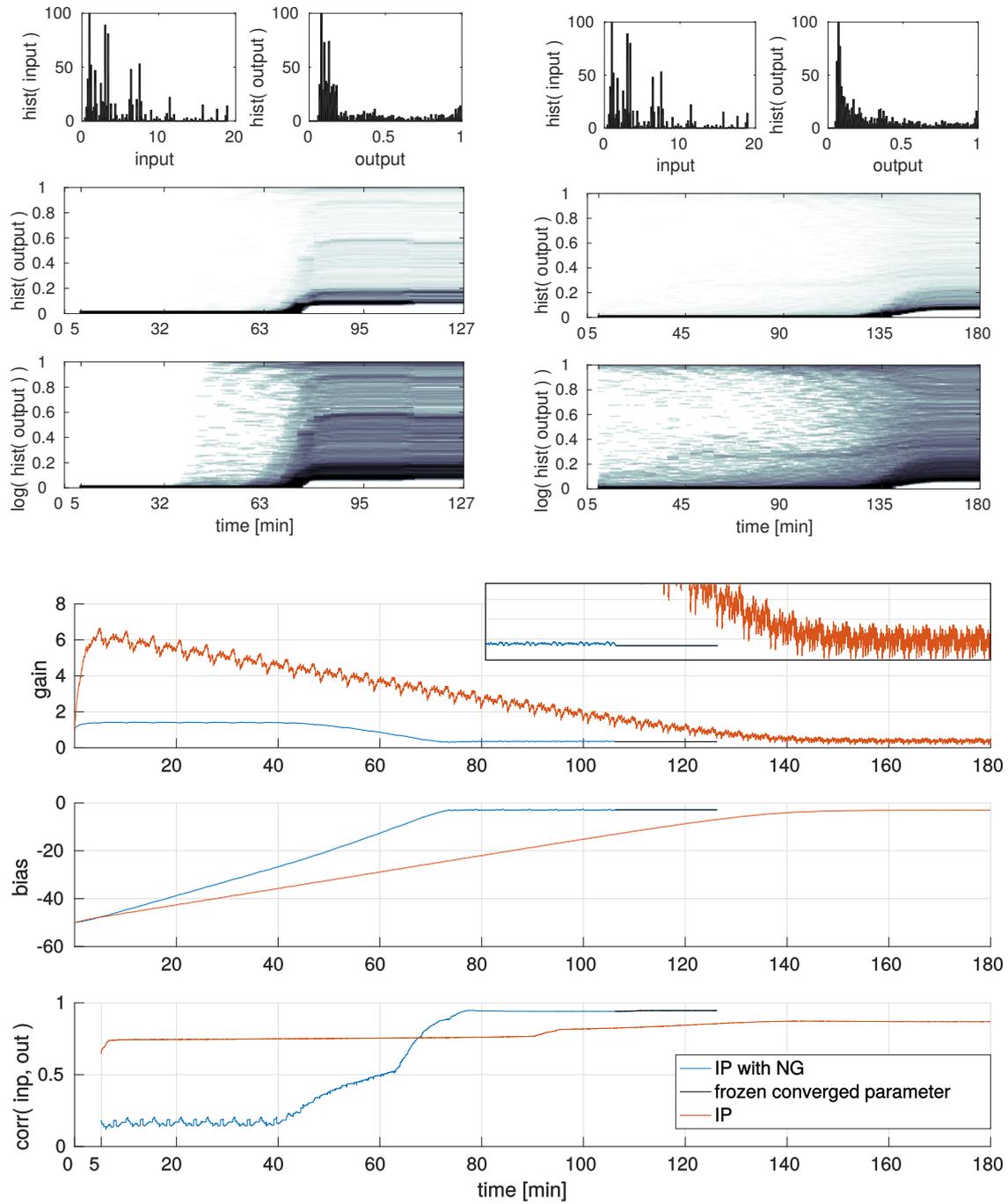


Figure A.0.1: IP in a DNF with initial **high bias**. Top, left three rows show input and output distribution for IP with NG. The top three rows on the right show the same plot for IP without NG. The bottom three rows show the parameter adaptation and input-output correlation for both procedures. From the 107th minute on, the converged parameter adaptation for IP with NG is frozen. See text for further description.

Appendix B

Appendix: Neurodynamic Model

In this section of the appendix all the results of the model with respect to the recorded datasets are shown. This is done for the sake of completeness and because quantitative measures in this thesis only capture aspects of the mapped object shape. In this section the individual performance is qualitatively shown. For the large six and eight sided objects the results are compared to the performance without pose corrections from the model. This comparison is omitted for all other results as it is not in particular relevant for rating the mapping performance.

Object Mapping

Surface MT Development Over Time

The following figures show the MT development during the twice iteration of the datasets, for all recorded datasets. The activity in the two dimensional MT is projected to the surface normal orientation dimension by summing up the activity along the position angle dimension. This projected activity is plotted on the y-axis over time on the x-axis.

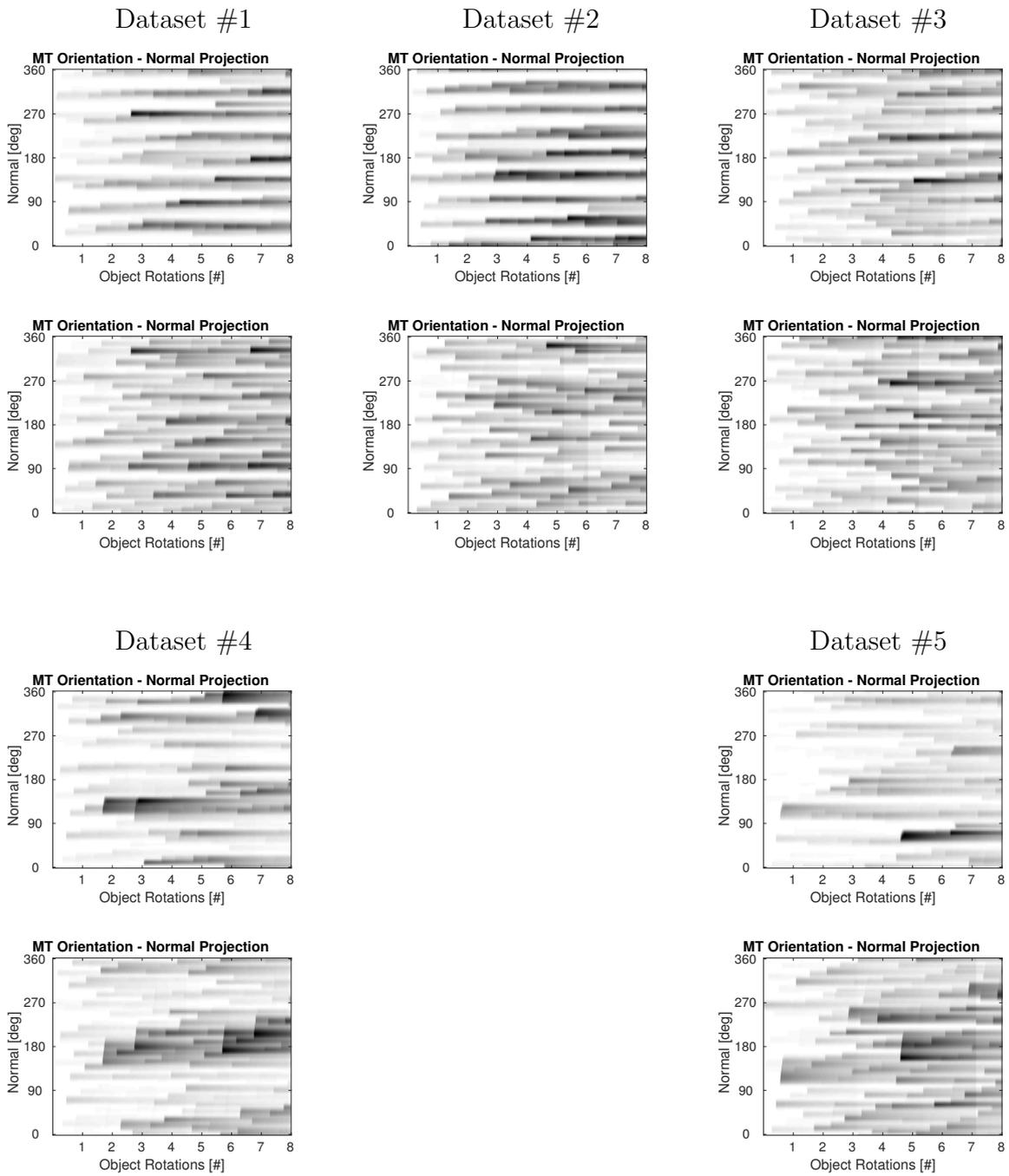


Figure B.0.1: The orientation MT for all datasets of the **large eight sided object**. The plots show the temporal development of the MT, projected to the normal orientation dimension, with the number of full object rotations (i.e. time) on the x axis. The top row shows the results of the model, while the bottom row shows the results when the correction feedback of the model is deactivated, i.e. purely forward estimates of the object dynamics are used.

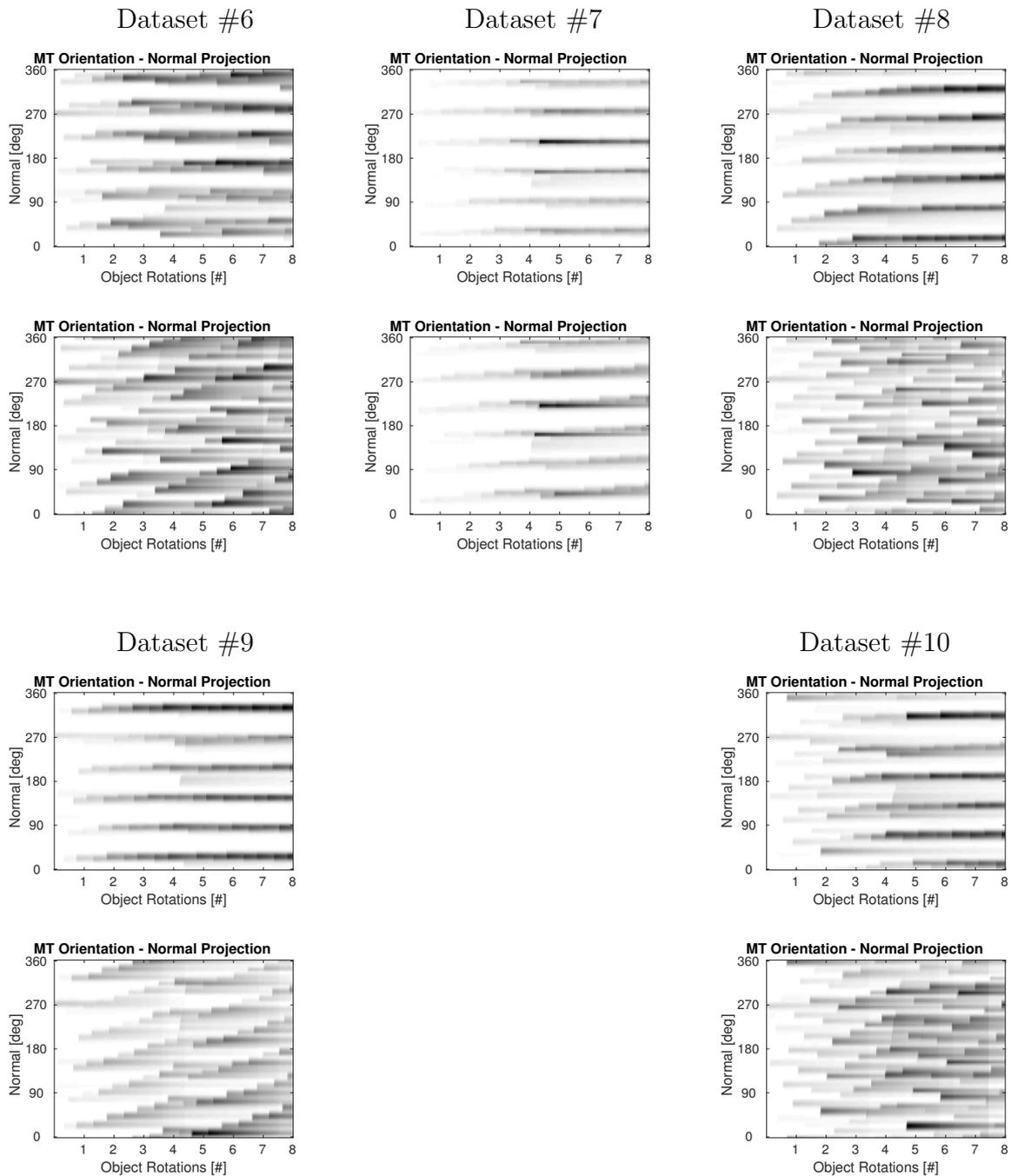


Figure B.0.2: The orientation MT for all datasets of the **large six sided object**. The plots show the temporal development of the MT, projected to the normal orientation dimension, with the number of full object rotations (i.e. time) on the x axis. The top row shows the results of the model, while the bottom row shows the results when the correction feedback of the model is deactivated, i.e. purely forward estimates of the object dynamics are used.

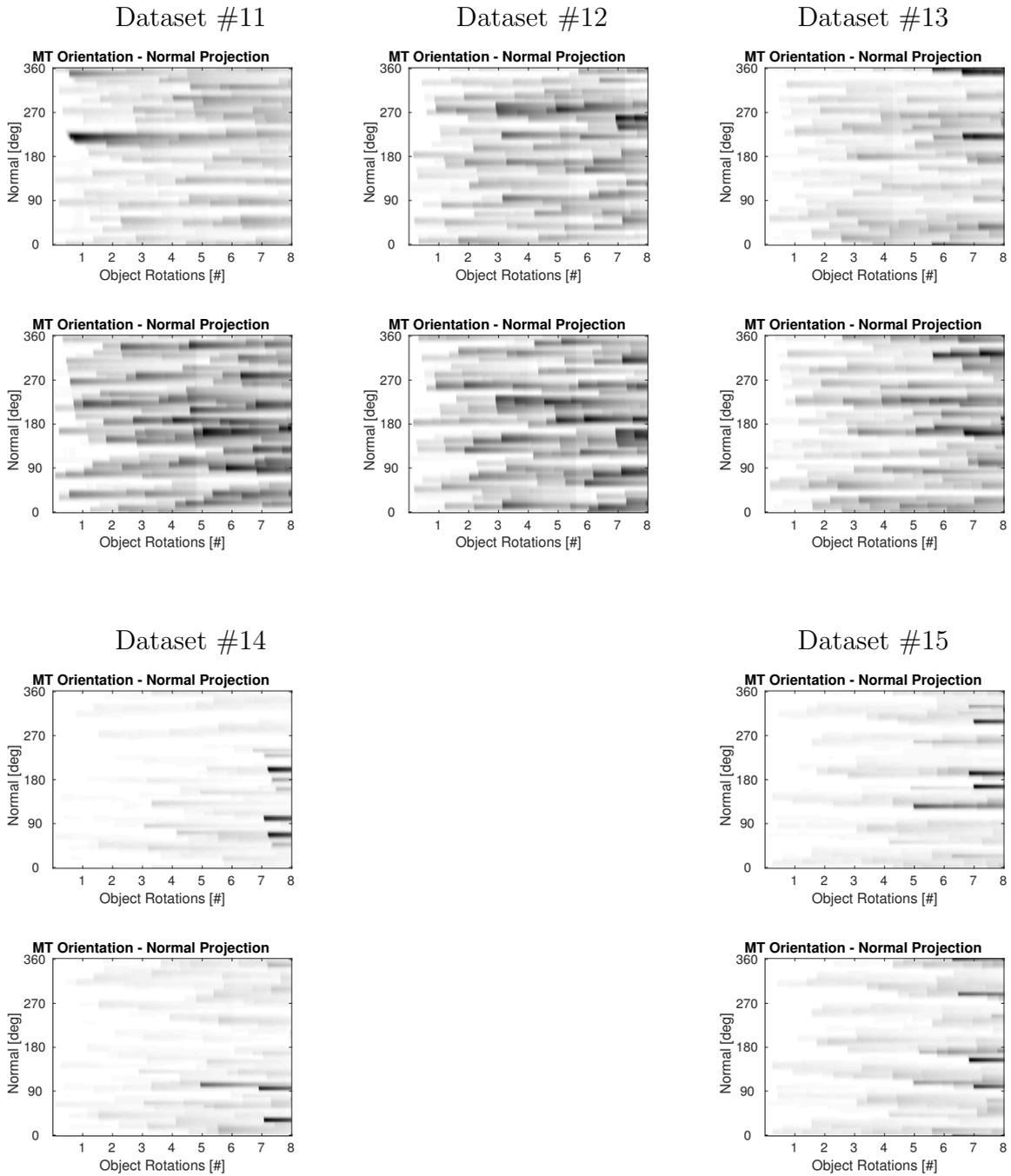


Figure B.0.3: The orientation MT for all datasets of the **small eight sided object**. For description see caption of Fig. B.0.1.

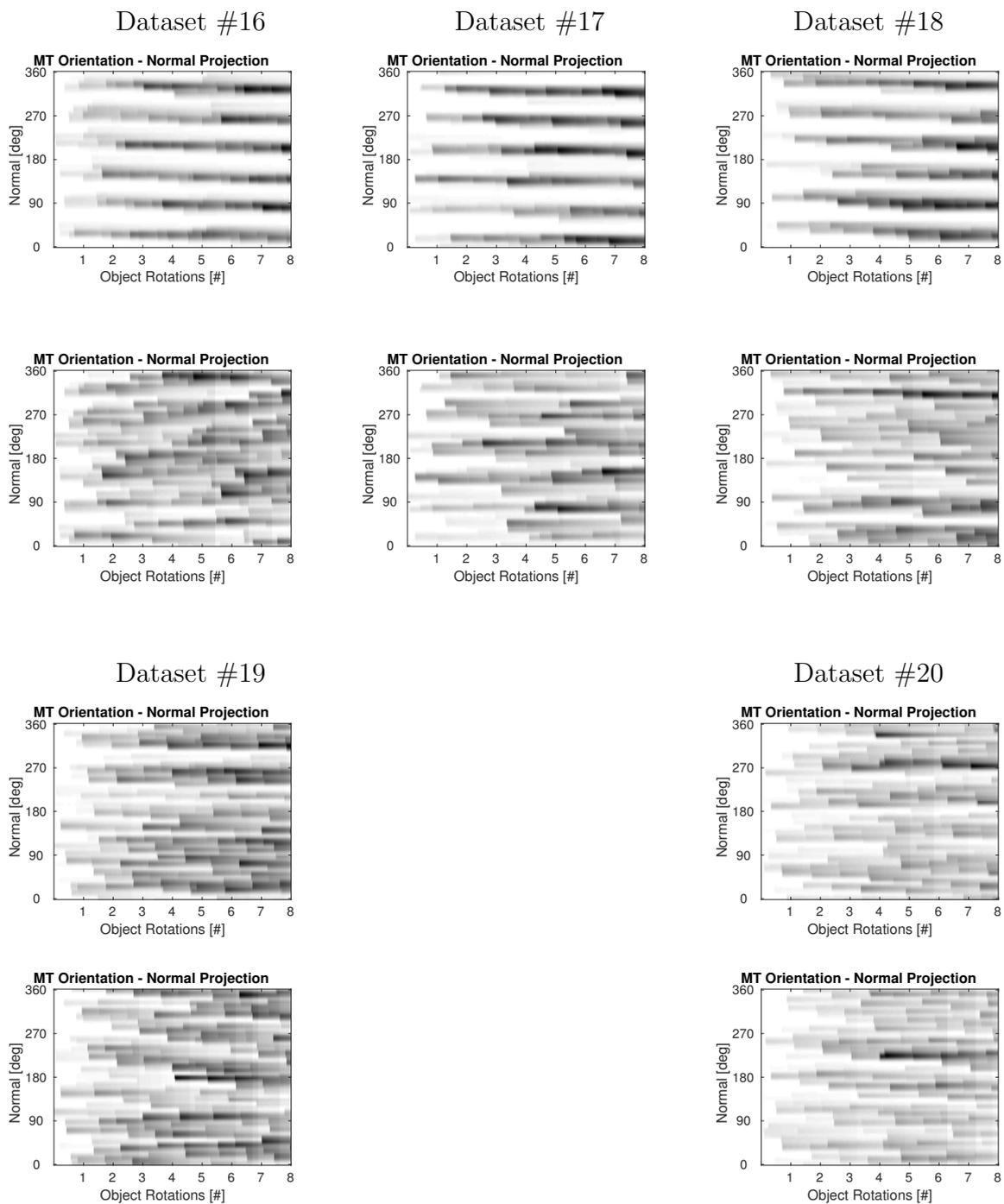


Figure B.0.4: The orientation MT for all datasets of the **small six sided object**. For description see caption of Fig. B.0.1.

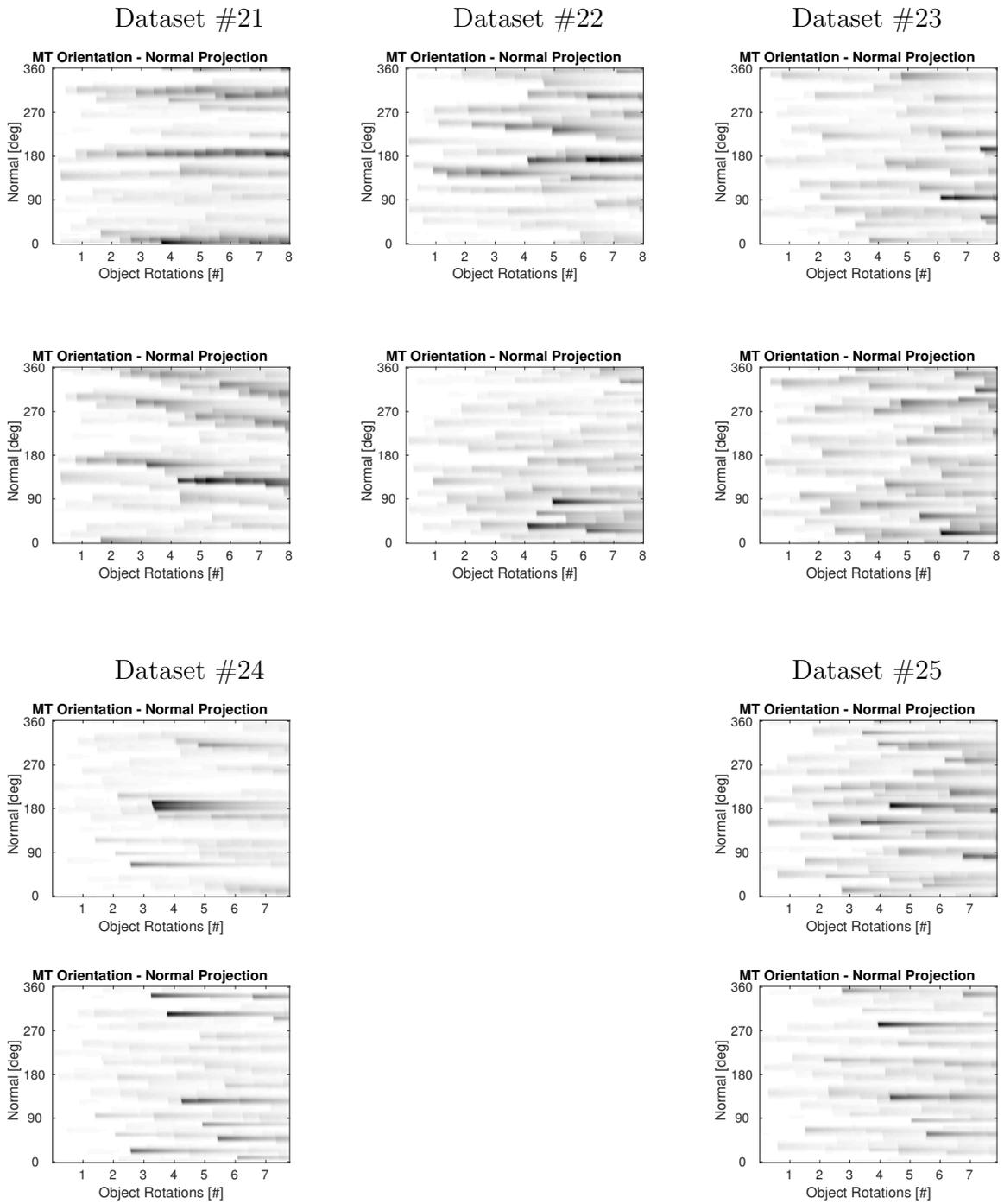


Figure B.0.5: The orientation MT for all datasets of the asymmetric five edged object. For description see caption of Fig. B.0.1.

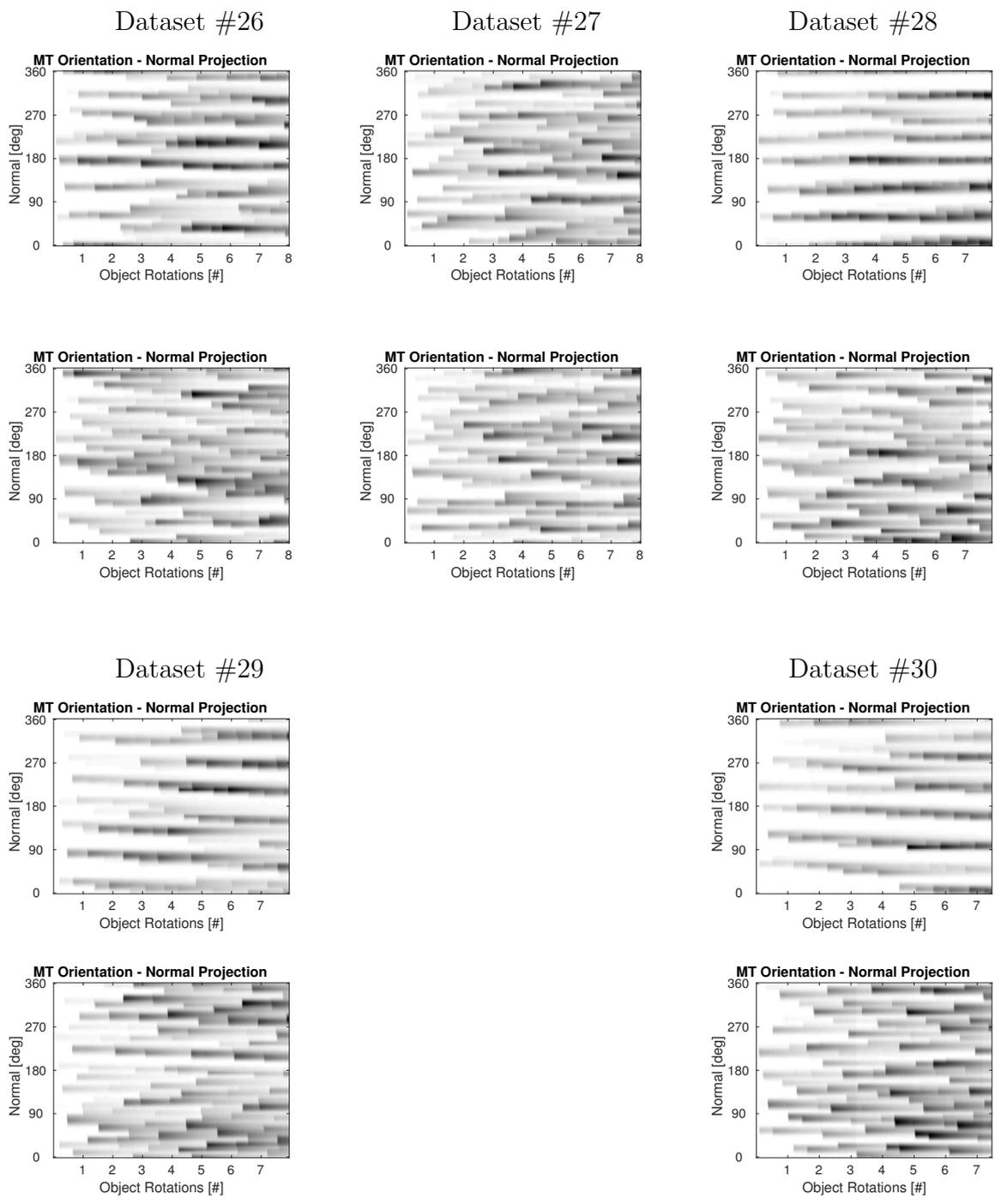


Figure B.0.6: The orientation MT for all datasets of the **asymmetric seven edged object**. For description see caption of Fig. B.0.1.

Number of detected surfaces without pose correction

Here the quantitative evaluation of the number of detected surfaces without any pose correction for the datasets of the small symmetric and the large asymmetric objects is shown.

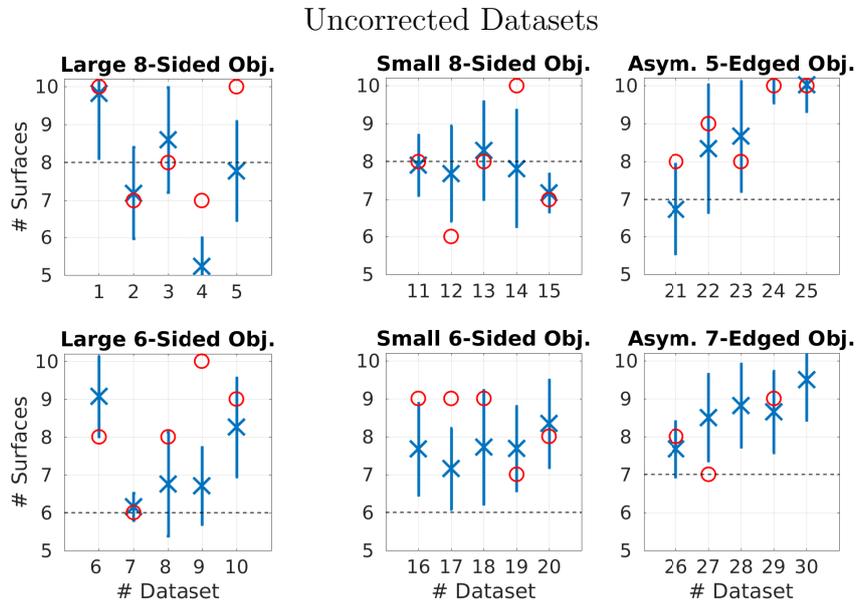


Figure B.0.7: The mean (blue cross) number of detected surfaces and its standard deviation (error bars) of the model without pose correction for the according datasets. The red circle is the number of detected surfaces at the end of the trial. Only the last iteration of the dataset is regarded.

Object Maps

In the following figures the object maps are shown at the end of the processing of a dataset by the model. For this the activation in the surface MT and edge MT are transformed into Cartesian coordinates and plotted. Note that the position radius is missing in the surface MT, it is set to a fixed value for this visualization. On the left of each figure the obtained object map is shown for the according dataset. Activity from the edge MT is indicated by dots with increasing size and intensity for higher activity in the MT. Activity of the surface MT is indicated by lines with increasing line width and intensity for higher activity in the MT. Additionally the surface normal orientation of activity peaks with more than $\frac{1}{4}$ of the maximum amplitude in the surface MT are indicated by red dots. On the right the according MT activity is shown from the surface MT (top) and the edge MT (bottom). The surface MT has an additional blue overlay showing the summed activity of the projected surface MT.

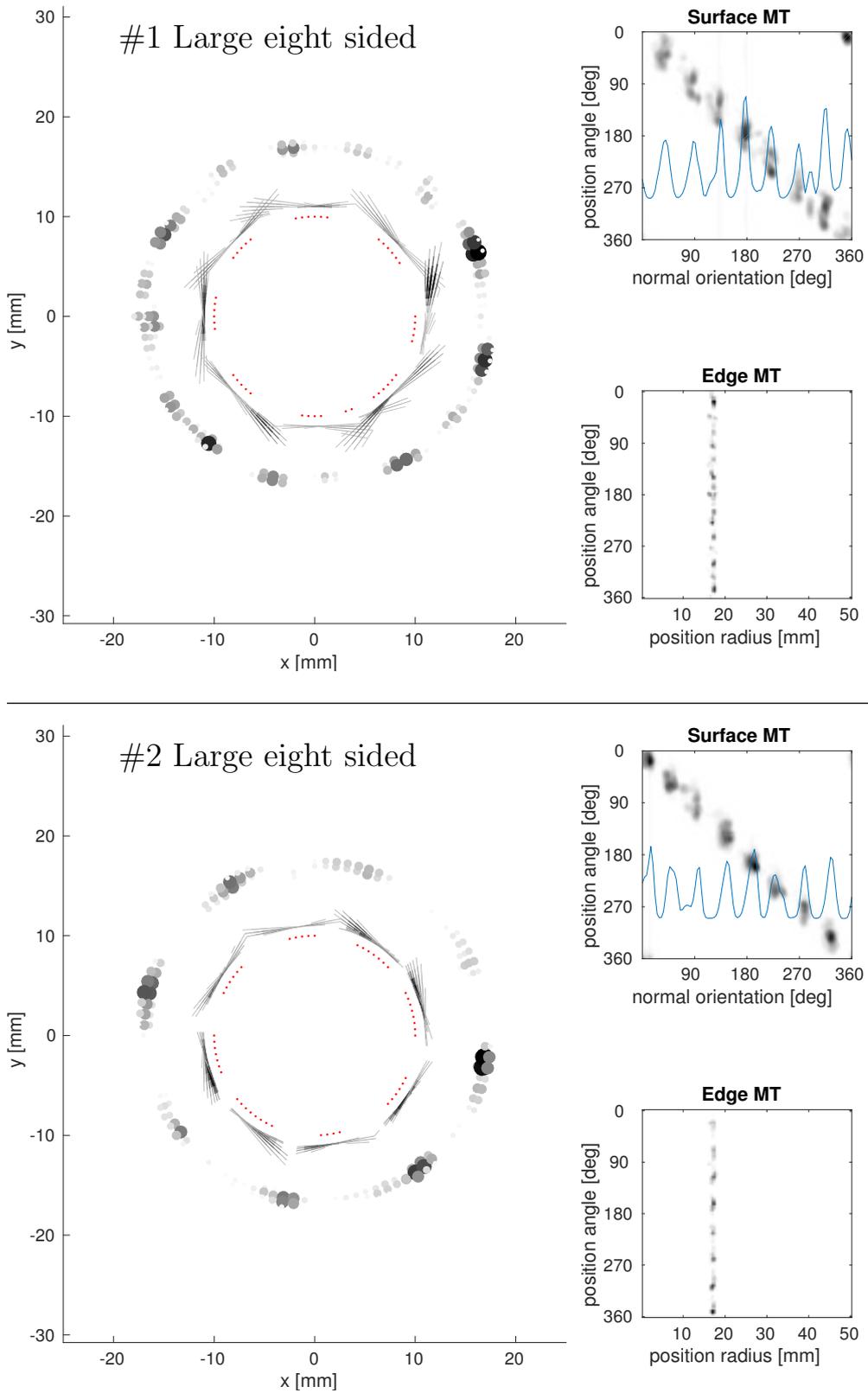


Figure B.0.8: The object representations after the second iteration of datasets #1 (Top) and #2 (Bottom). See Sec. 6.1 for further details.

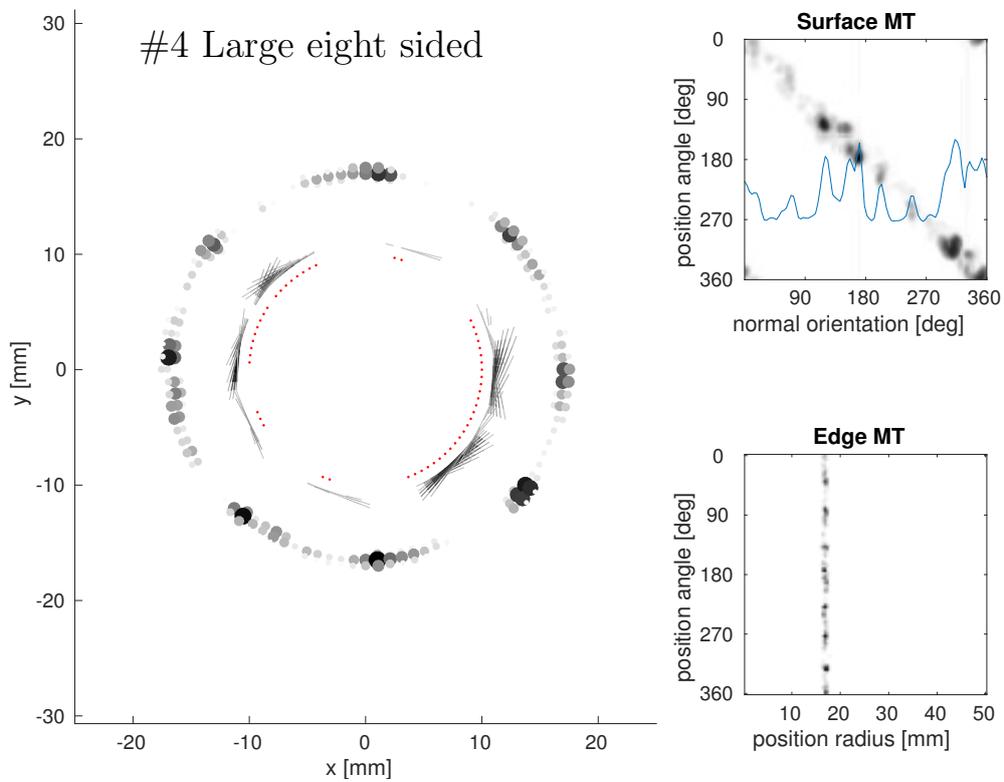
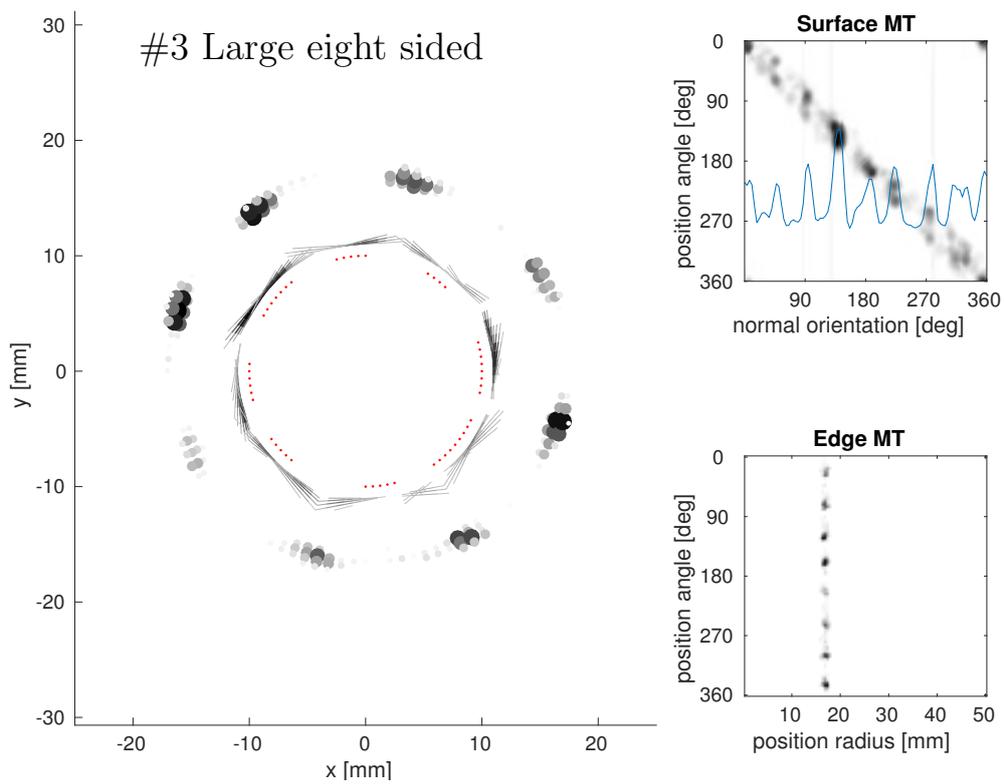


Figure B.0.9: The object representations after the second iteration of datasets #3 (Top) and #4 (Bottom). See Sec. 6.1 for further details.

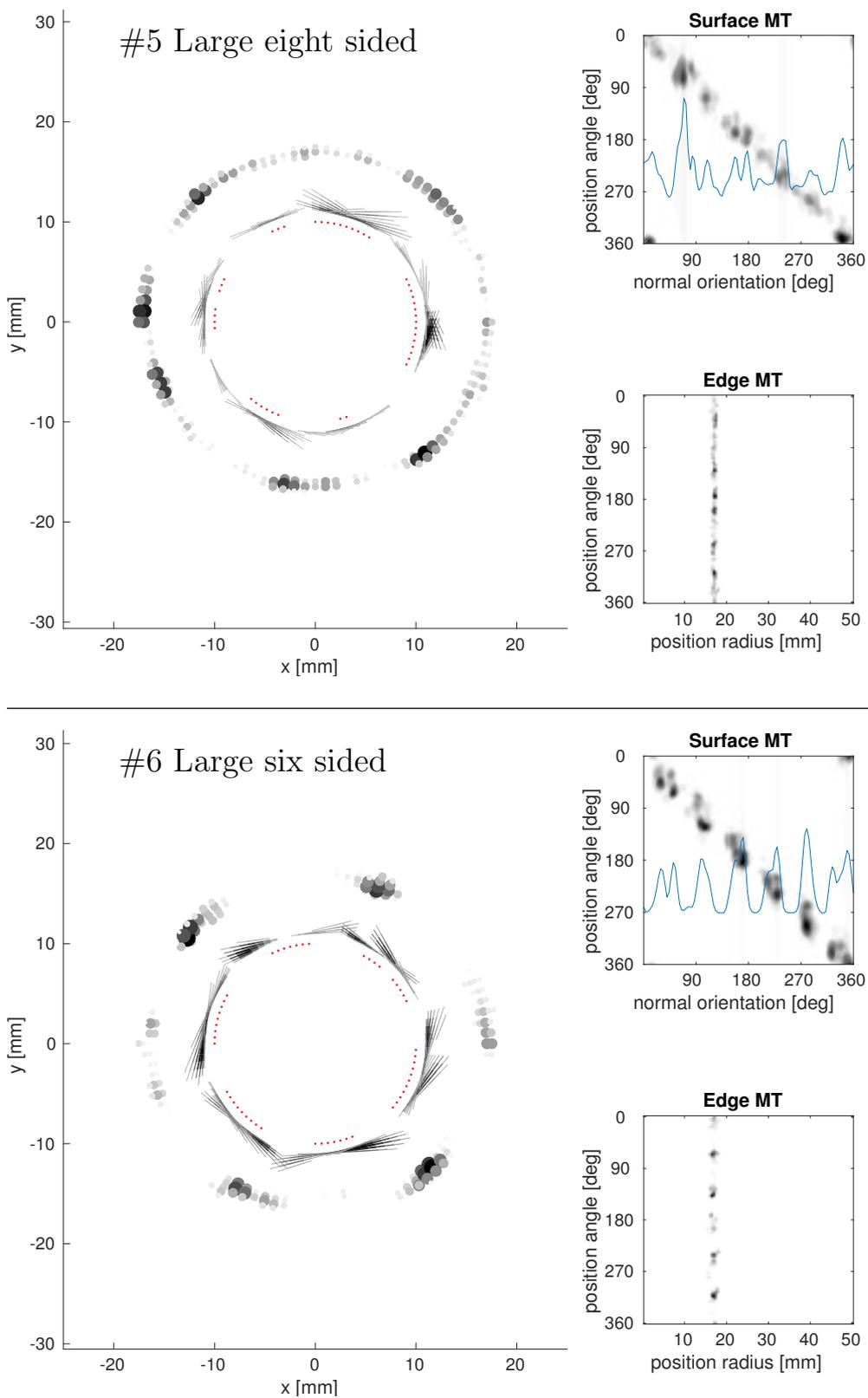


Figure B.0.10: The object representations after the second iteration of datasets #5 of the eight sided object (Top) and #6 from the six sided object (Bottom). See Sec. 6.1 for further details.

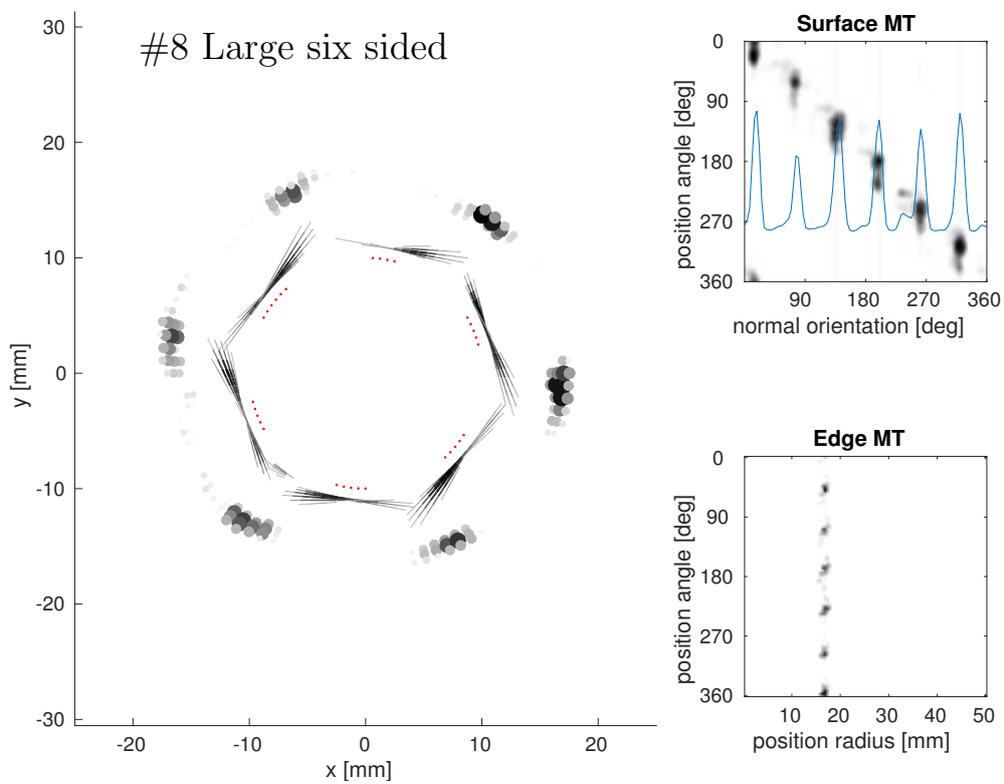
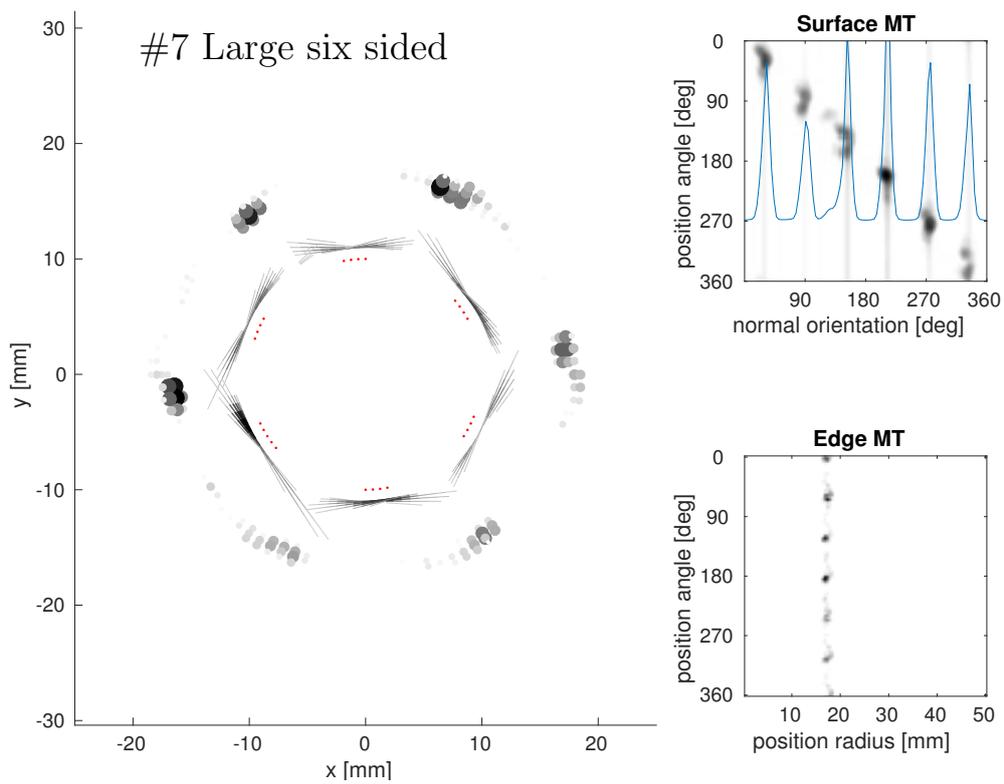


Figure B.0.11: The object representations after the second iteration of datasets #7 (Top) and #8 (Bottom). See Sec. 6.1 for further details.

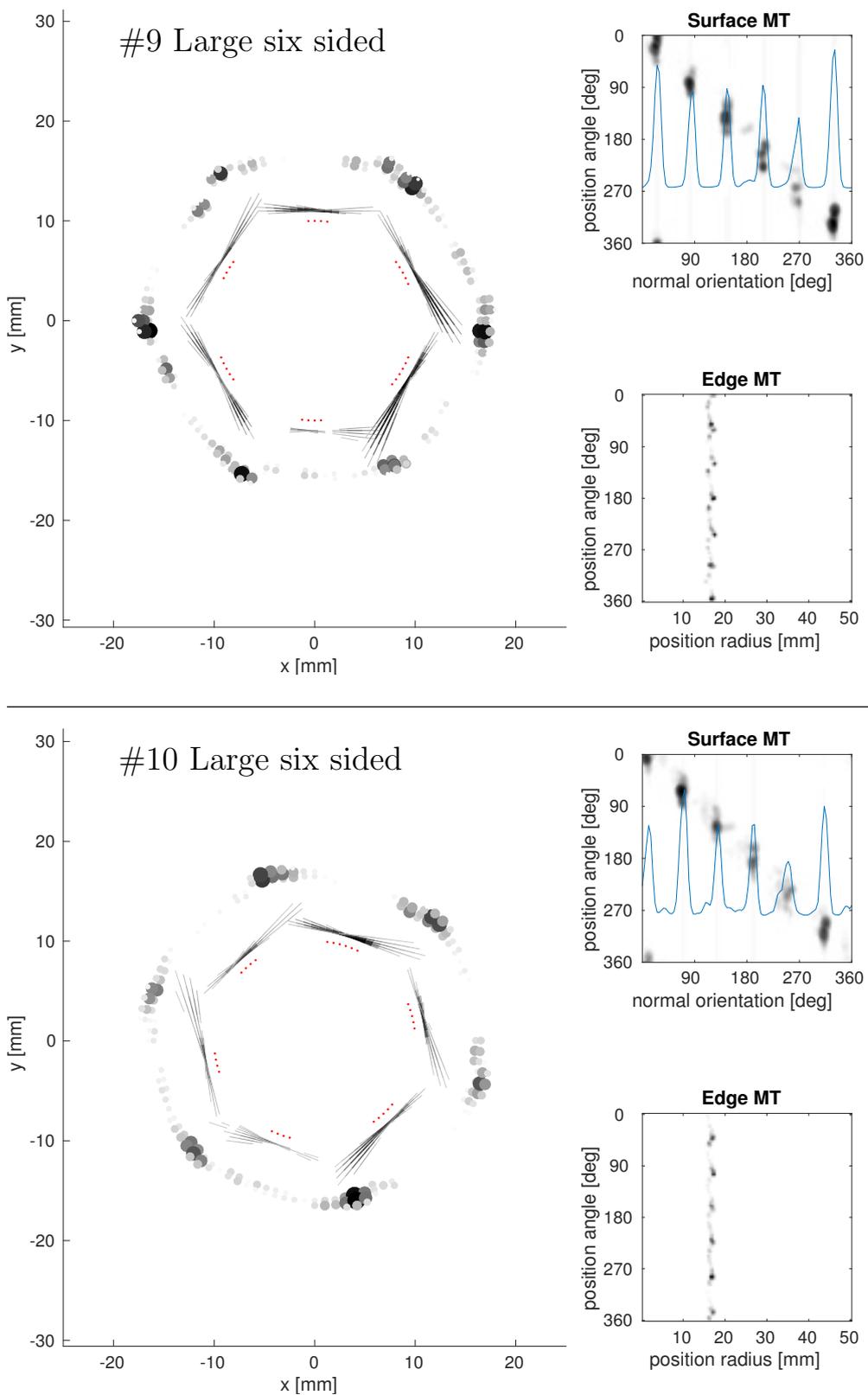


Figure B.0.12: The object representations after the second iteration of datasets #9 (Top) and #10 (Bottom). See Sec. 6.1 for further details.

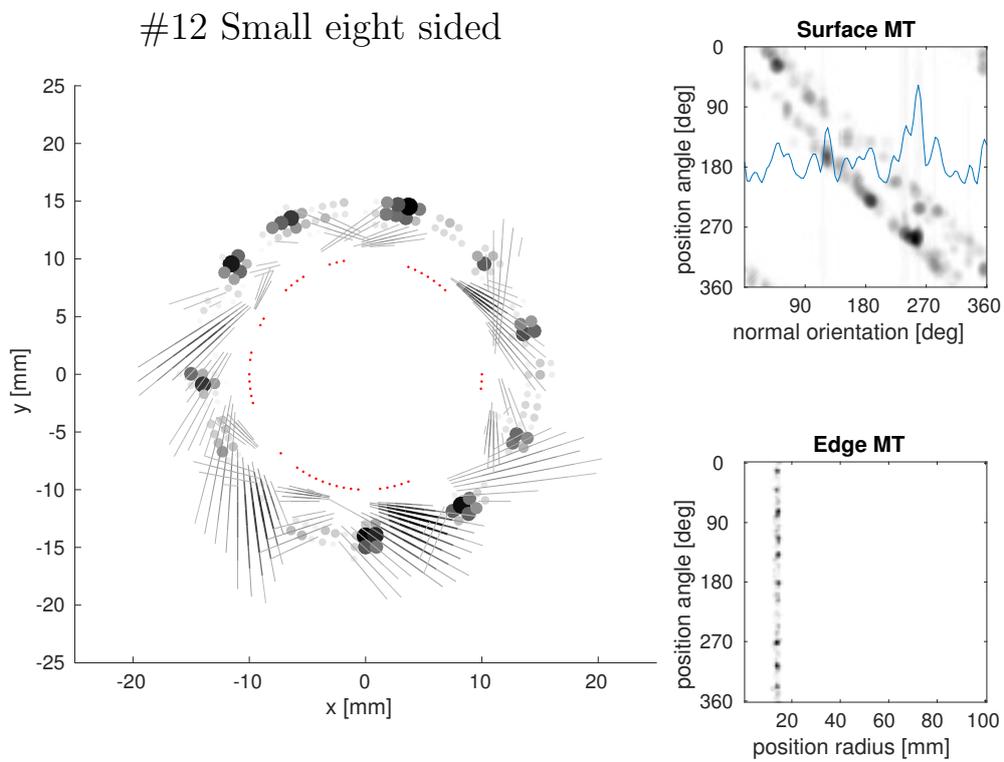
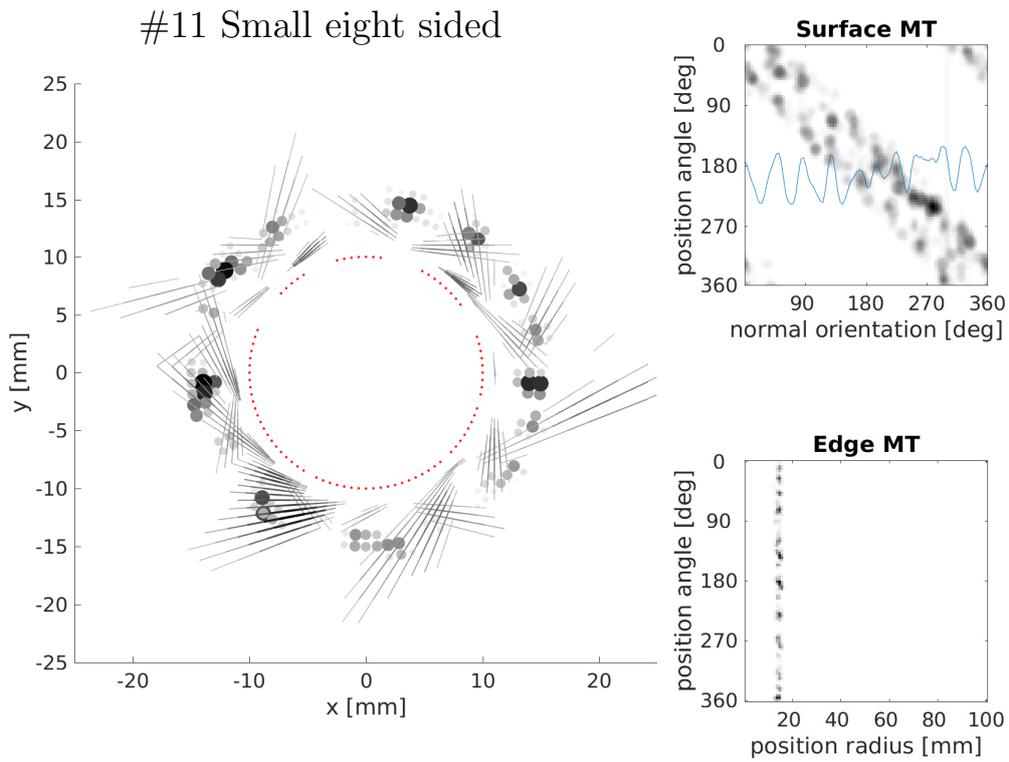


Figure B.0.13: The object representations after the second iteration of datasets #11 (Top) and #12 (Bottom). See Sec. 6.1 for further details.

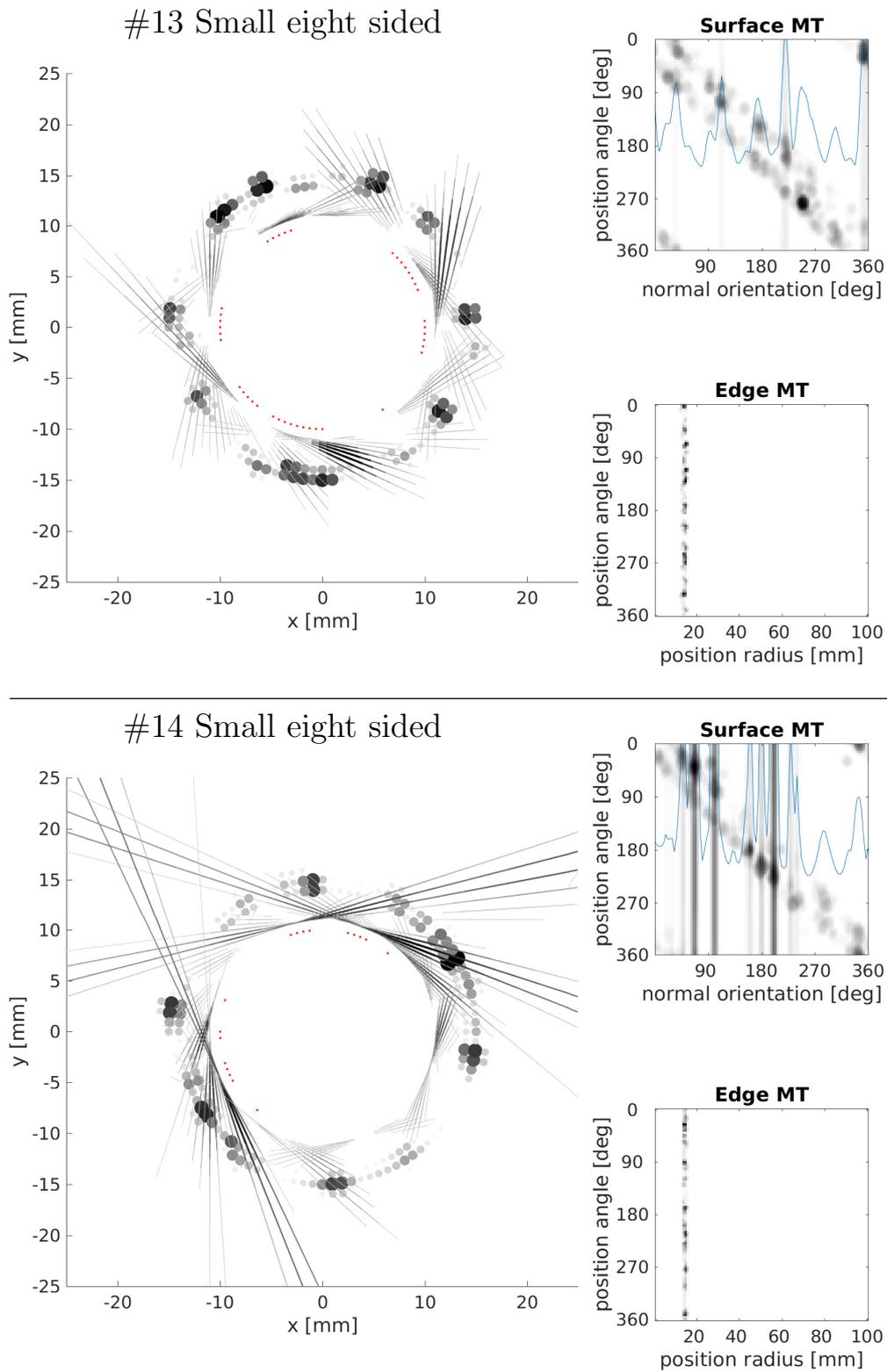


Figure B.0.14: The object representations after the second iteration of datasets #13 (Top) and #14 (Bottom). See Sec. 6.1 for further details.

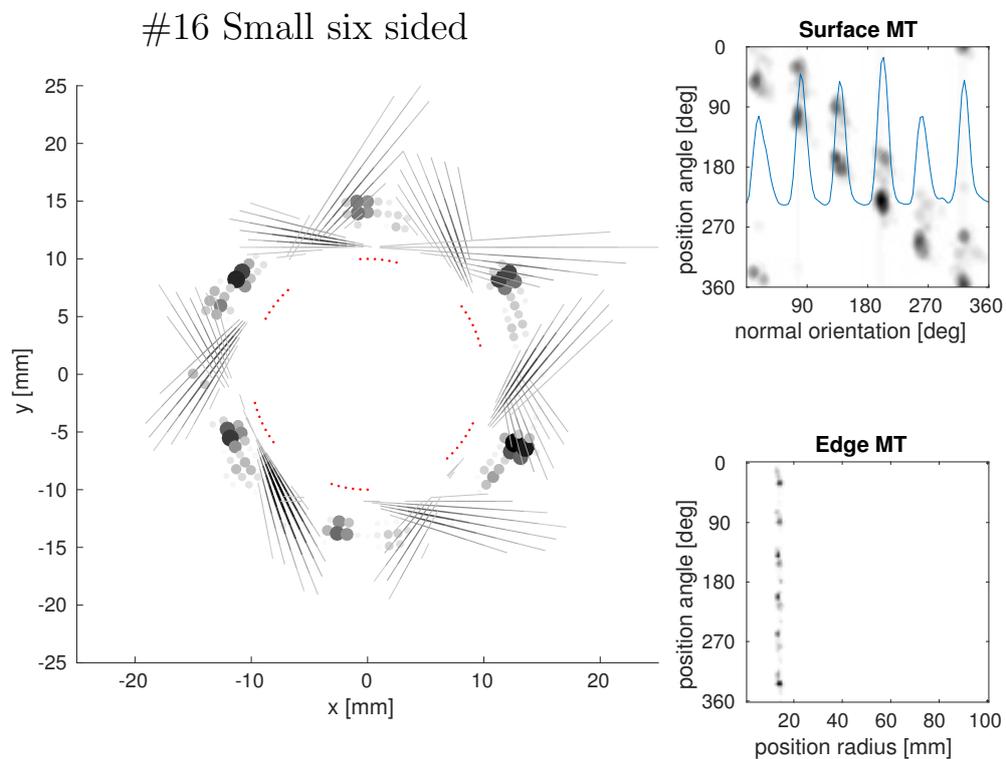
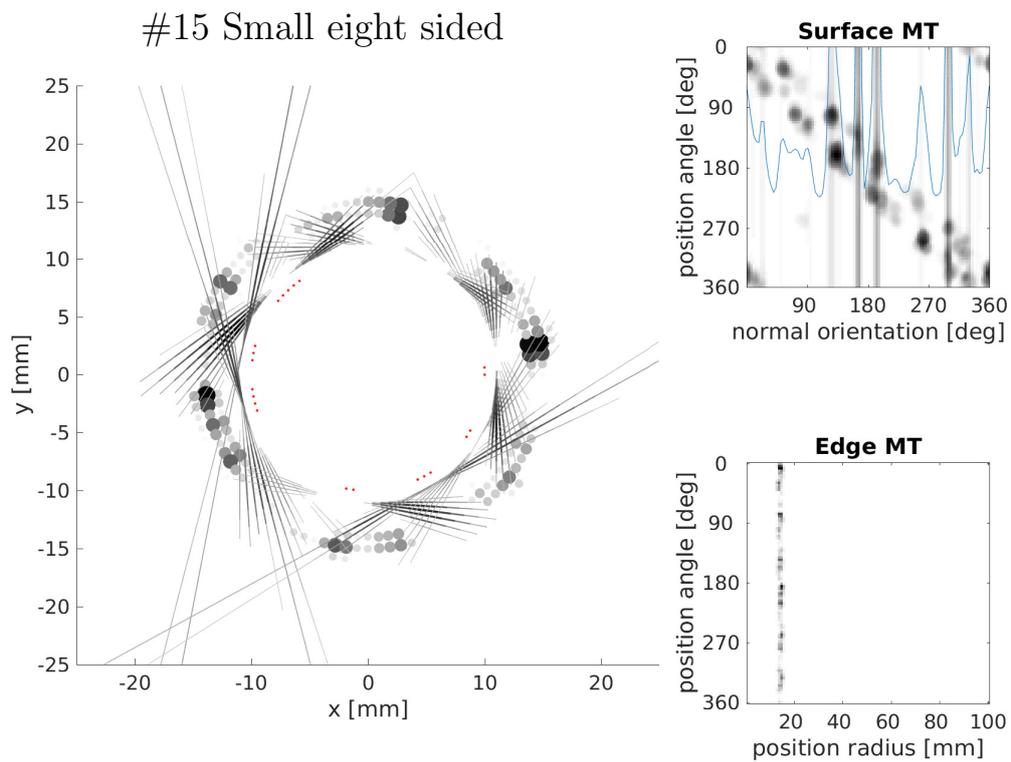


Figure B.0.15: The object representations after the second iteration of datasets #15 of the small eight sided object (Top) and #16 from the small six sided object (Bottom). See Sec. 6.1 for further details.

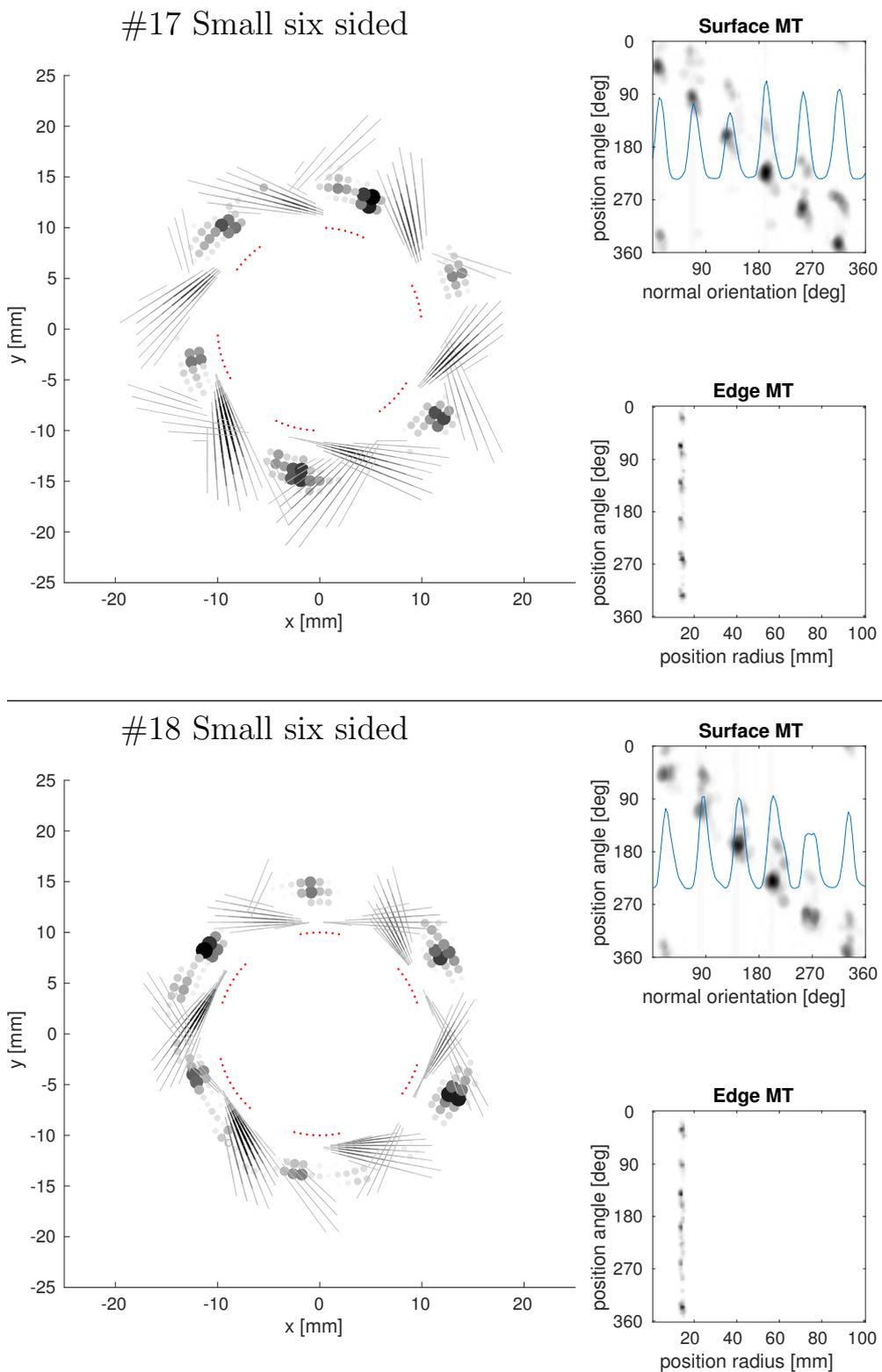


Figure B.0.16: The object representations after the second iteration of datasets #17 (Top) and #18 (Bottom). See Sec. 6.1 for further details.

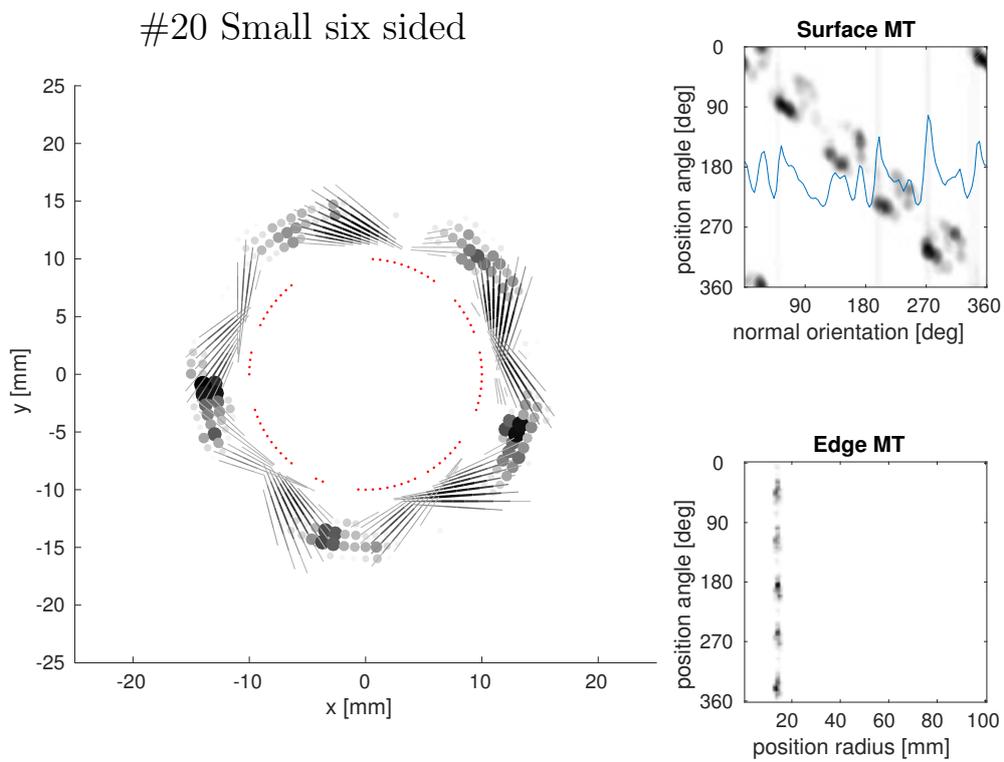
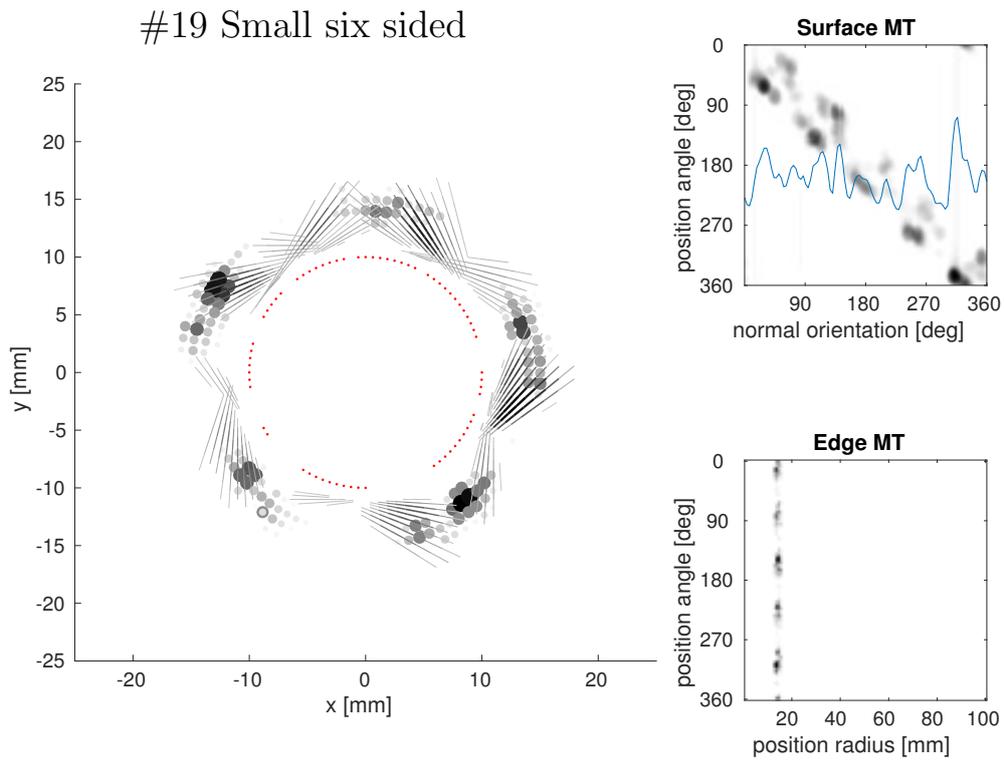
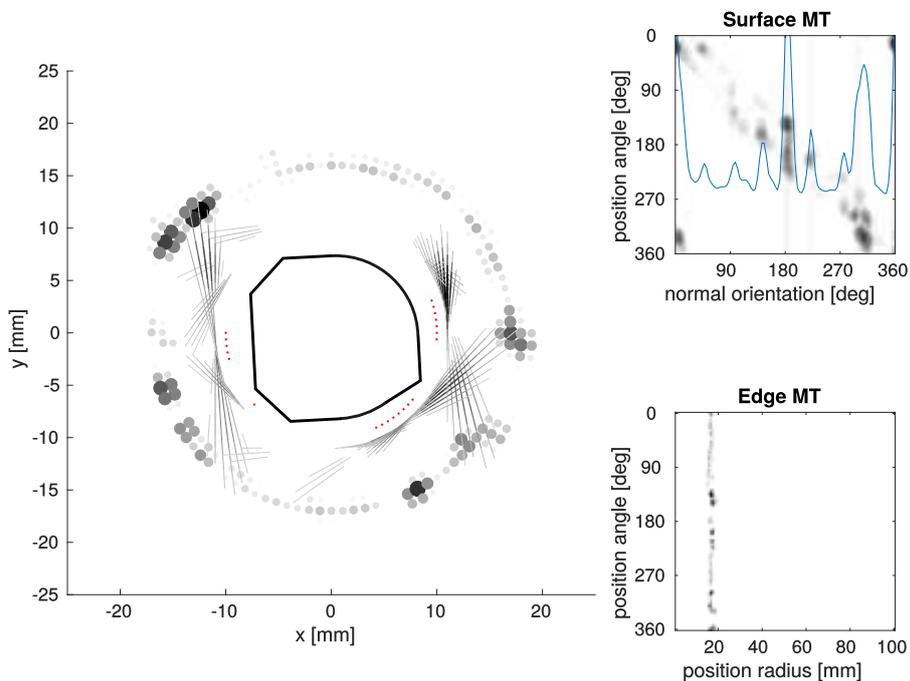


Figure B.0.17: The object representations after the second iteration of datasets #19 (Top) and #20 (Bottom). See Sec. 6.1 for further details.

#21 Asymmetric five edged



#22 Asymmetric five edged

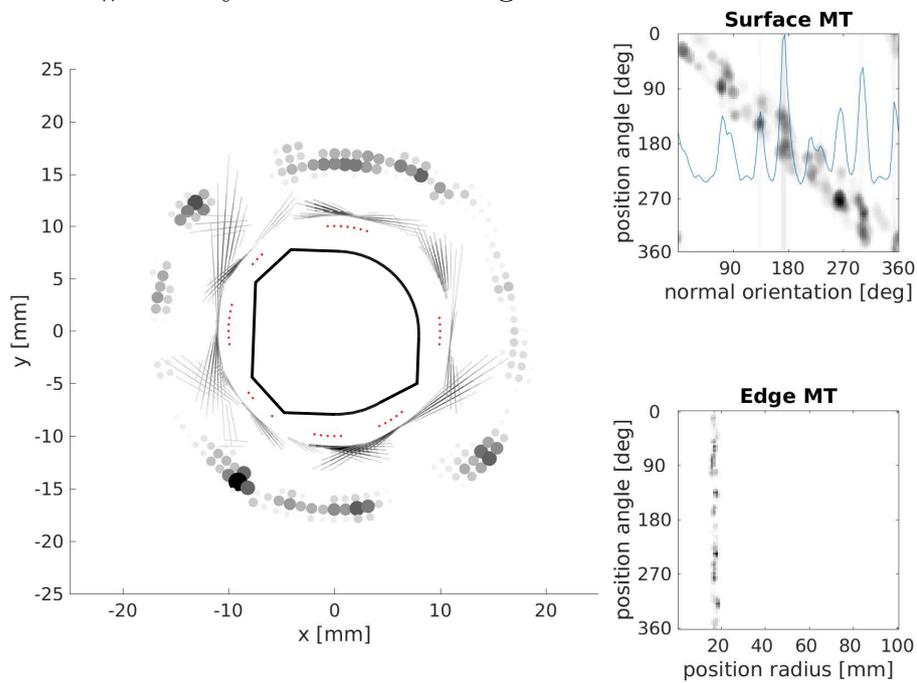


Figure B.0.18: The object representations after the second iteration of datasets #21 (Top) and #22 (Bottom). See Sec. 6.1 for further details.

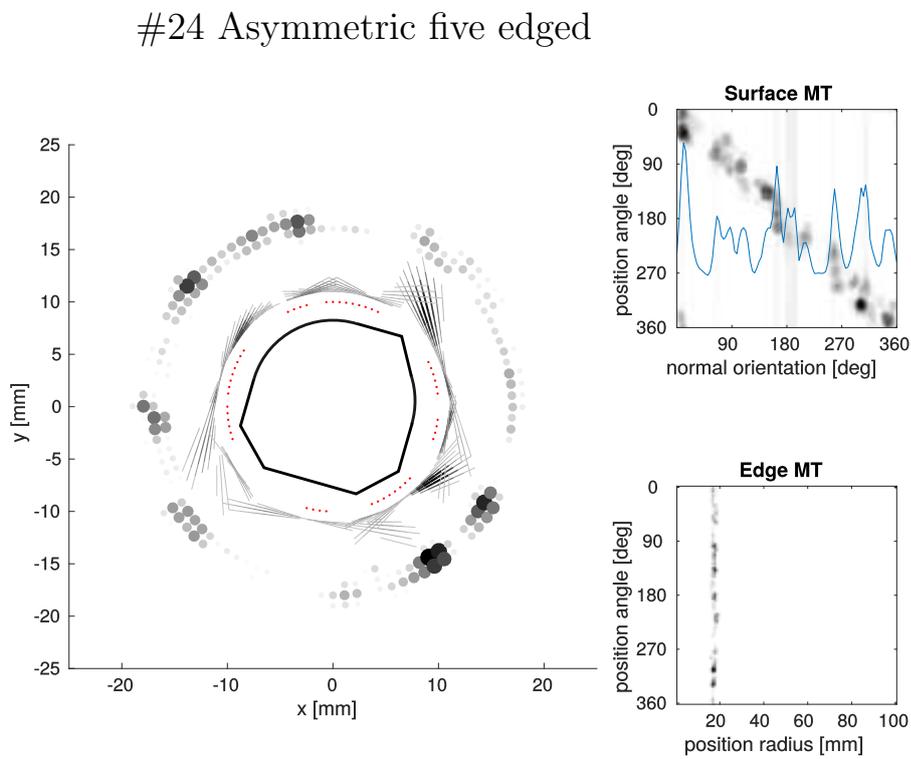
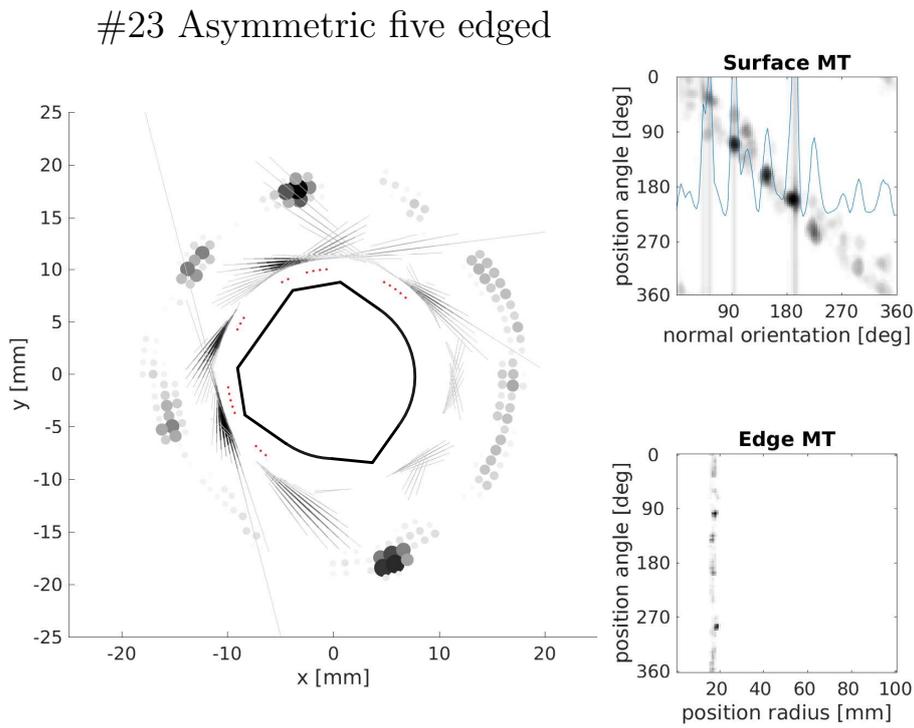
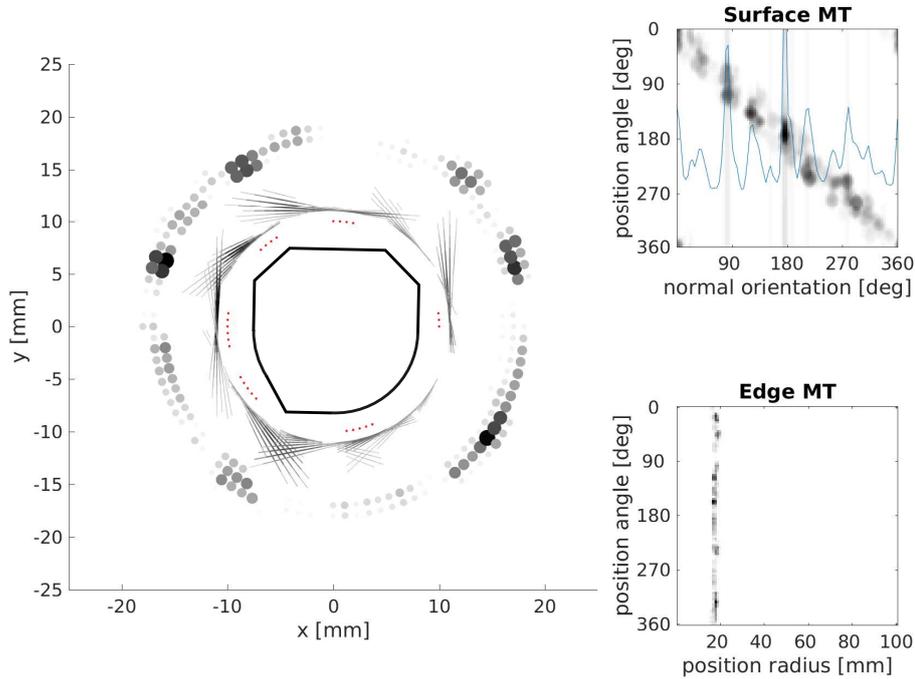


Figure B.0.19: The object representations after the second iteration of datasets #23 (Top) and #24 (Bottom). See Sec. 6.1 for further details.

#25 Asymmetric five edged



#26 Asymmetric seven edged

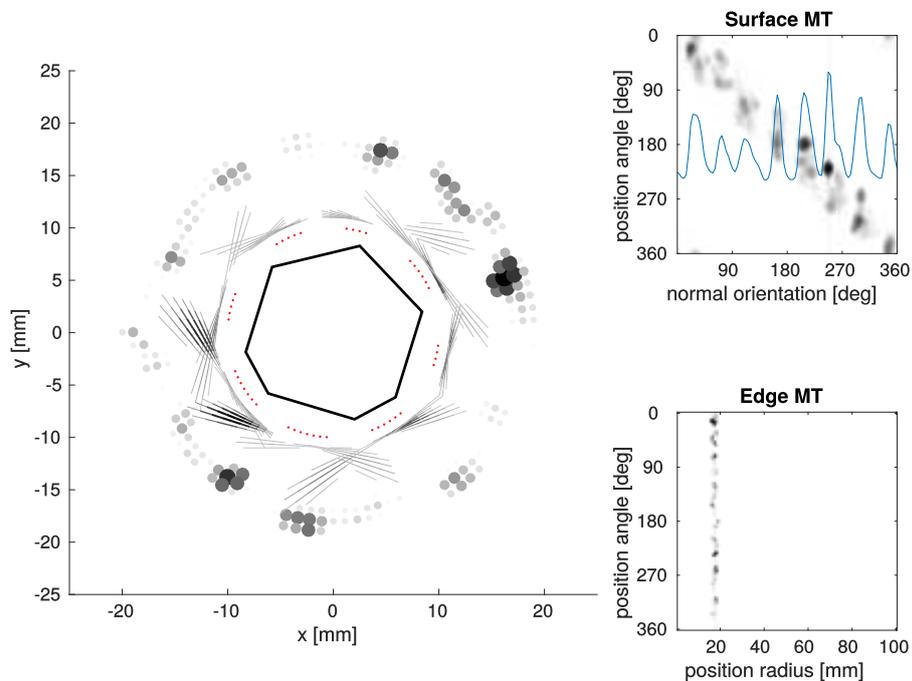


Figure B.0.20: The object representations after the second iteration of datasets #25 of the eight sided object (Top) and #26 from the six sided object (Bottom). See Sec. 6.1 for further details.

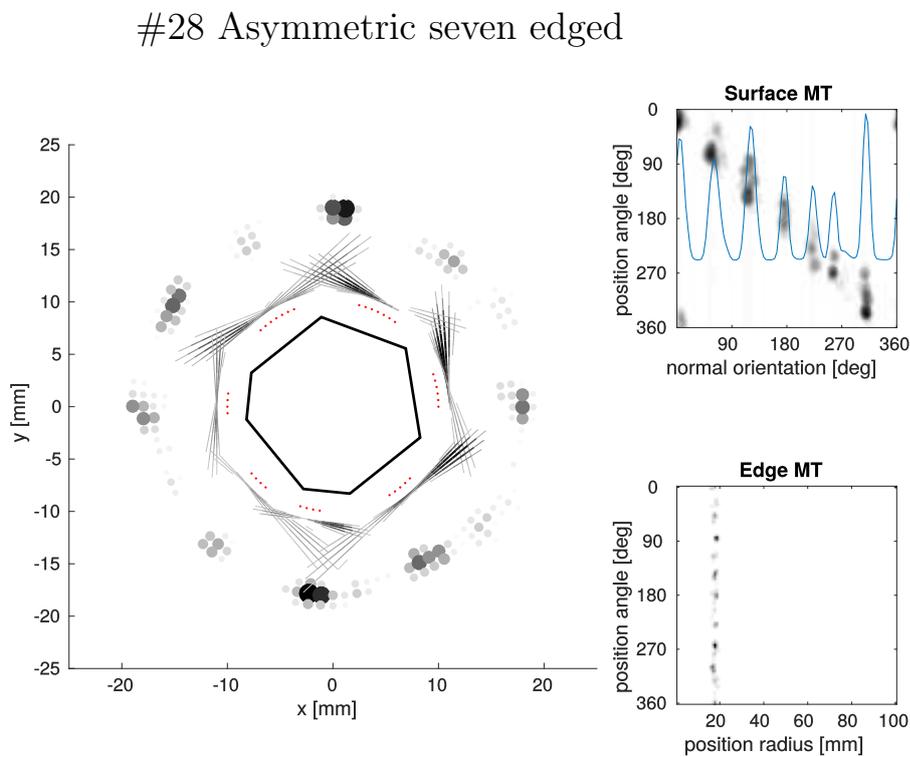
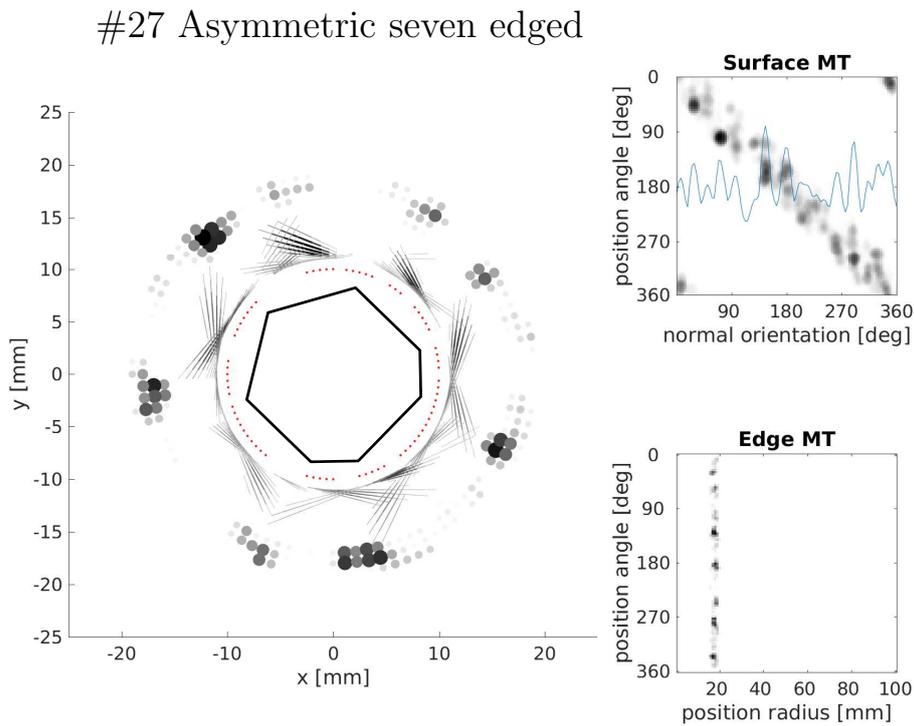
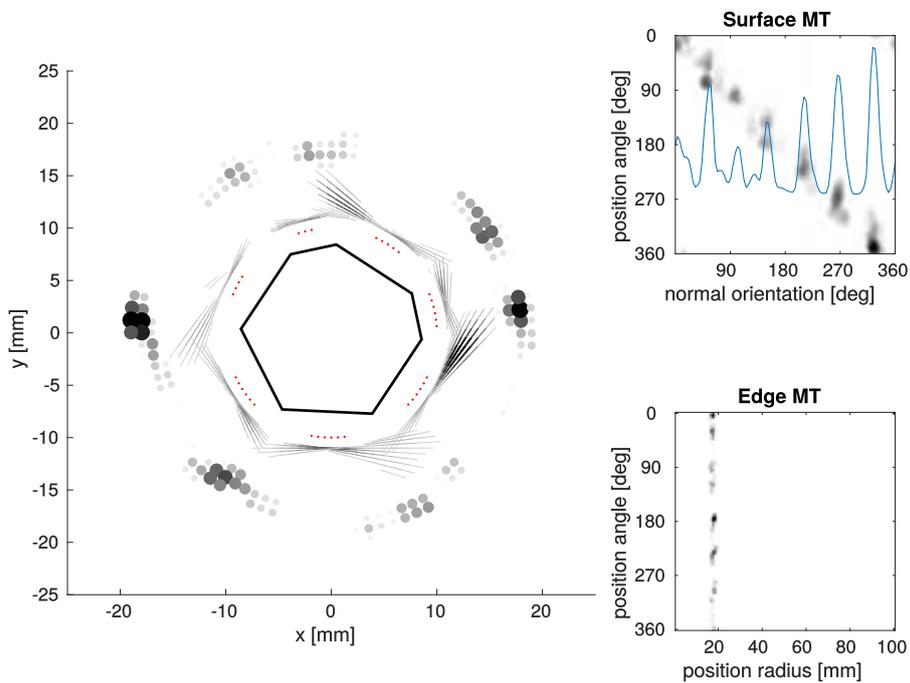


Figure B.0.21: The object representations after the second iteration of datasets #27 (Top) and #28 (Bottom). See Sec. 6.1 for further details.

#29 Asymmetric seven edged



#30 Asymmetric seven edged

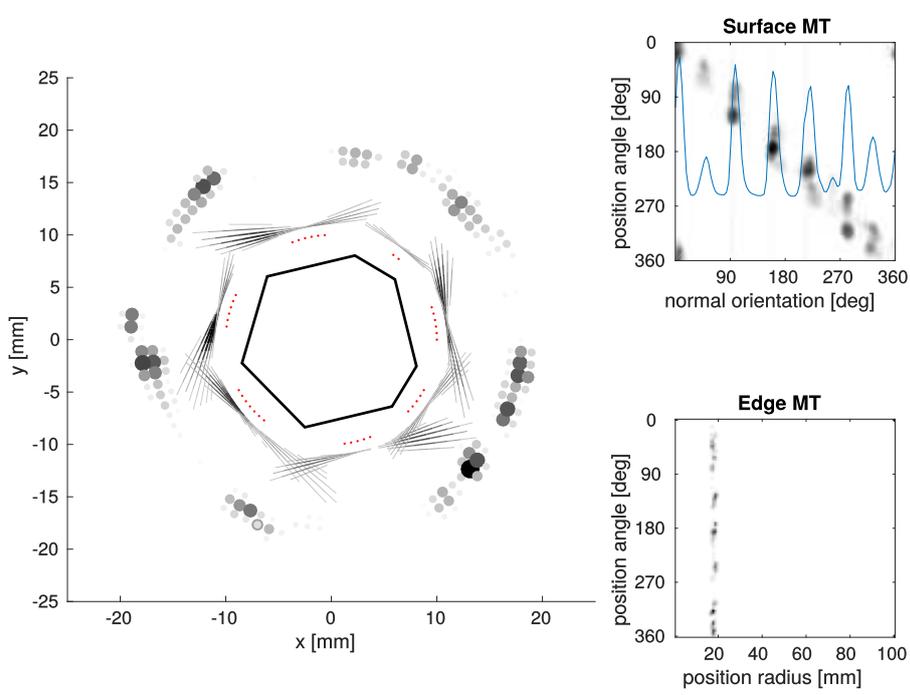


Figure B.0.22: The object representations after the second iteration of datasets #29 (Top) and #30 (Bottom). See Sec. 6.1 for further details.

Object Localization

In the following plots the raw haptic data recorded from the object manipulations are shown for every dataset. On the right column the raw data based on the predictions from the forward model are plotted. In the middle column the same data are plotted with a corrected object orientation estimate by the proposed model. In the third column the object position is additionally corrected in its DPA by the model.

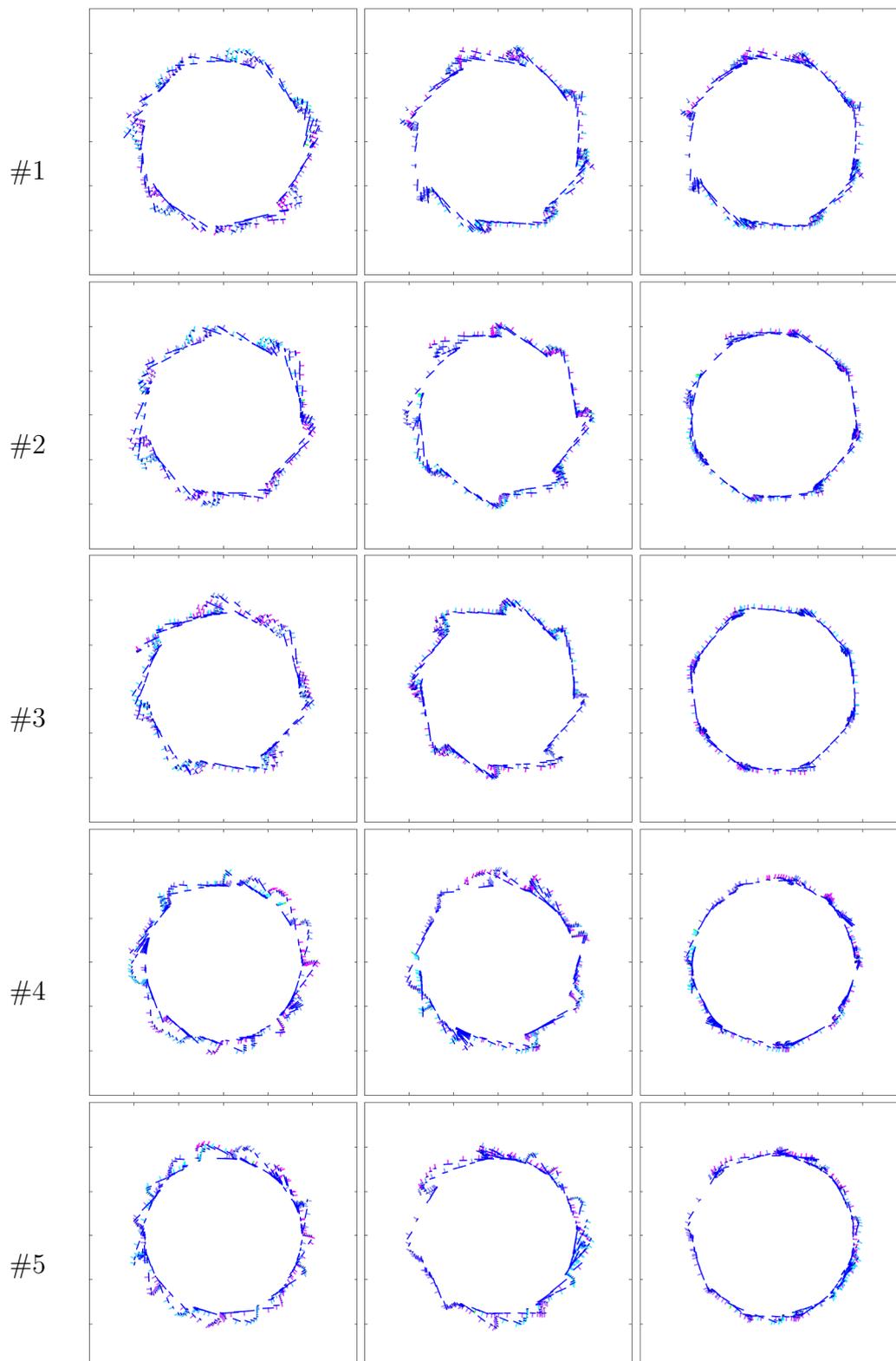


Figure B.0.23: Tactile features in object space from the datasets of the **large eight sided object** (row-wise). The first column shows the features based on the object pose estimate from the forward model. The middle column includes the orientation correction from the model, the third column additionally includes the position correction.

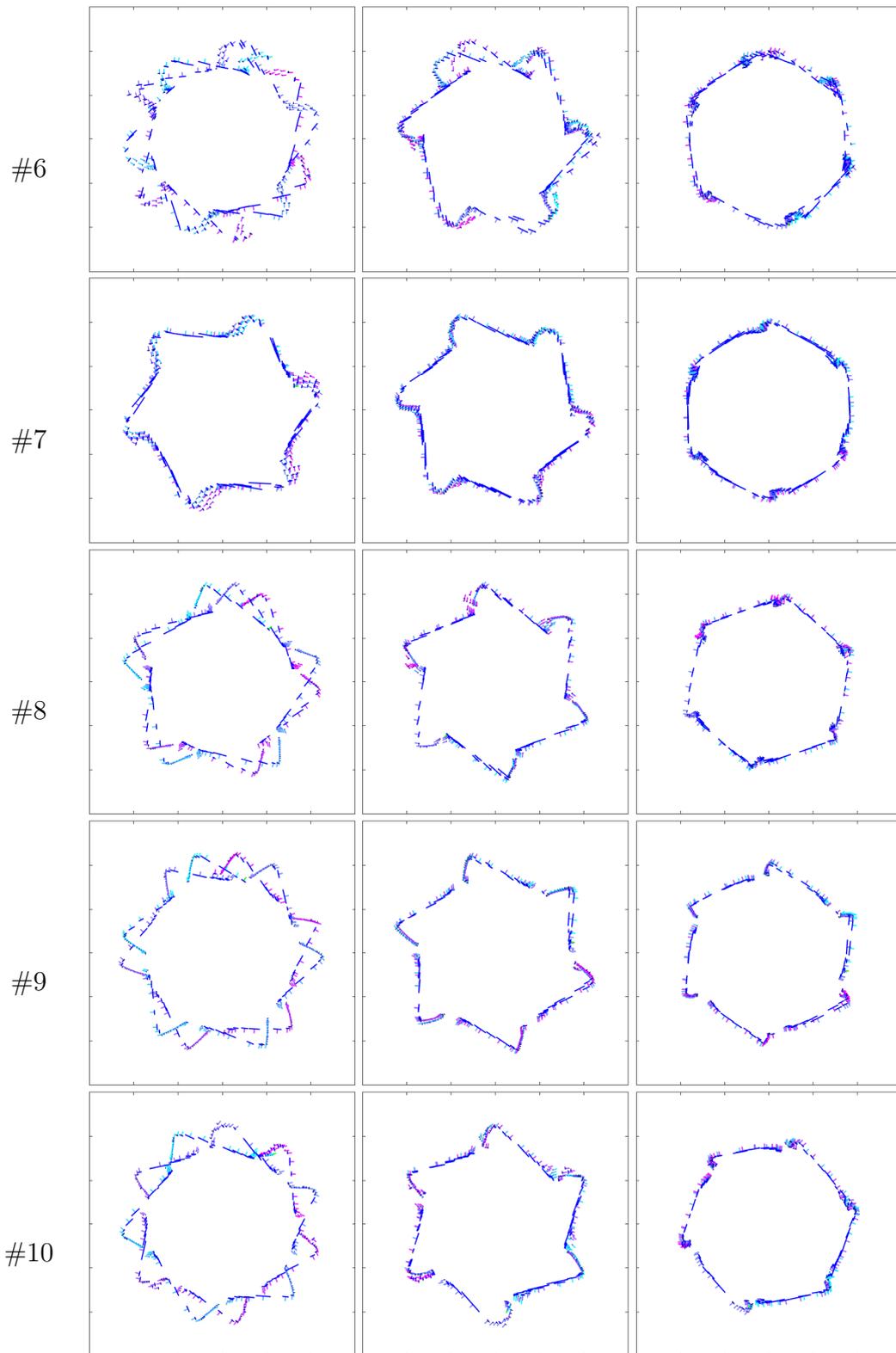


Figure B.0.24: Tactile features in object space from the datasets of the **large six sided object** (row-wise). See Fig. B.0.23 for more information (above).

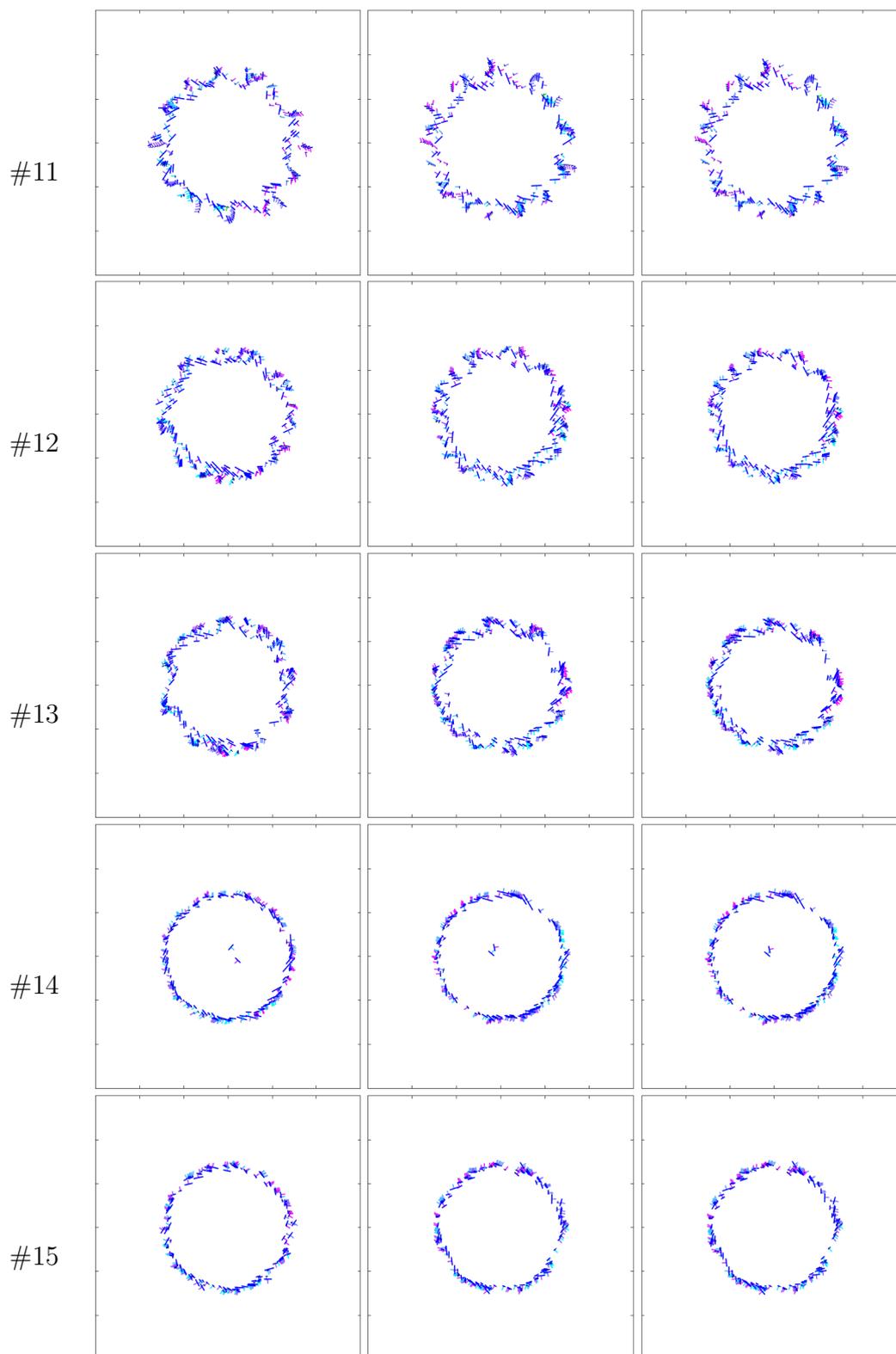


Figure B.0.25: Tactile features in object space from the datasets of the **small eight sided object** (row-wise). See Fig. B.0.23 for more information.

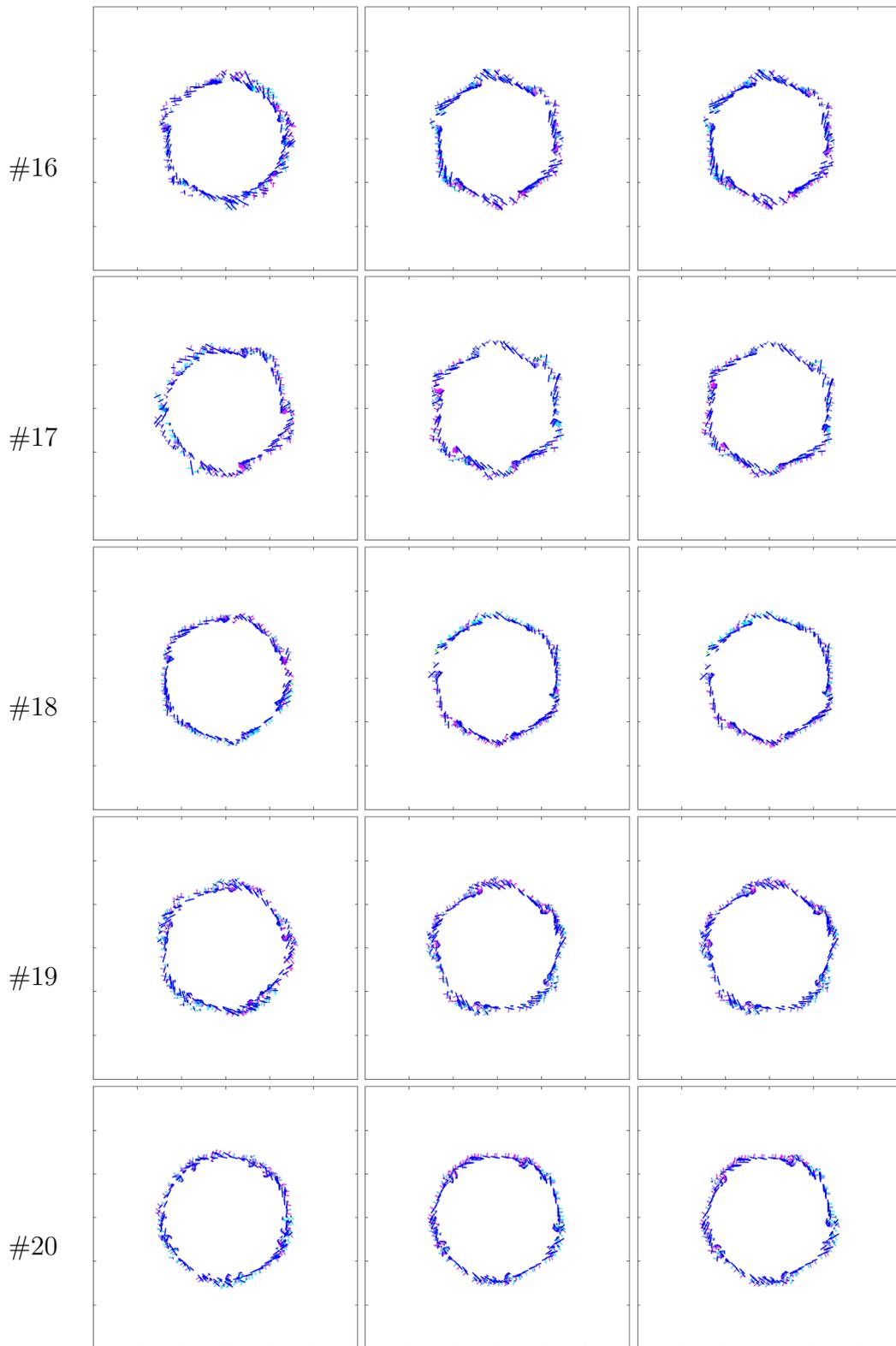


Figure B.0.26: Tactile features in object space from the datasets of the **small six sided object** (row-wise). See Fig. B.0.23 for more information (above).

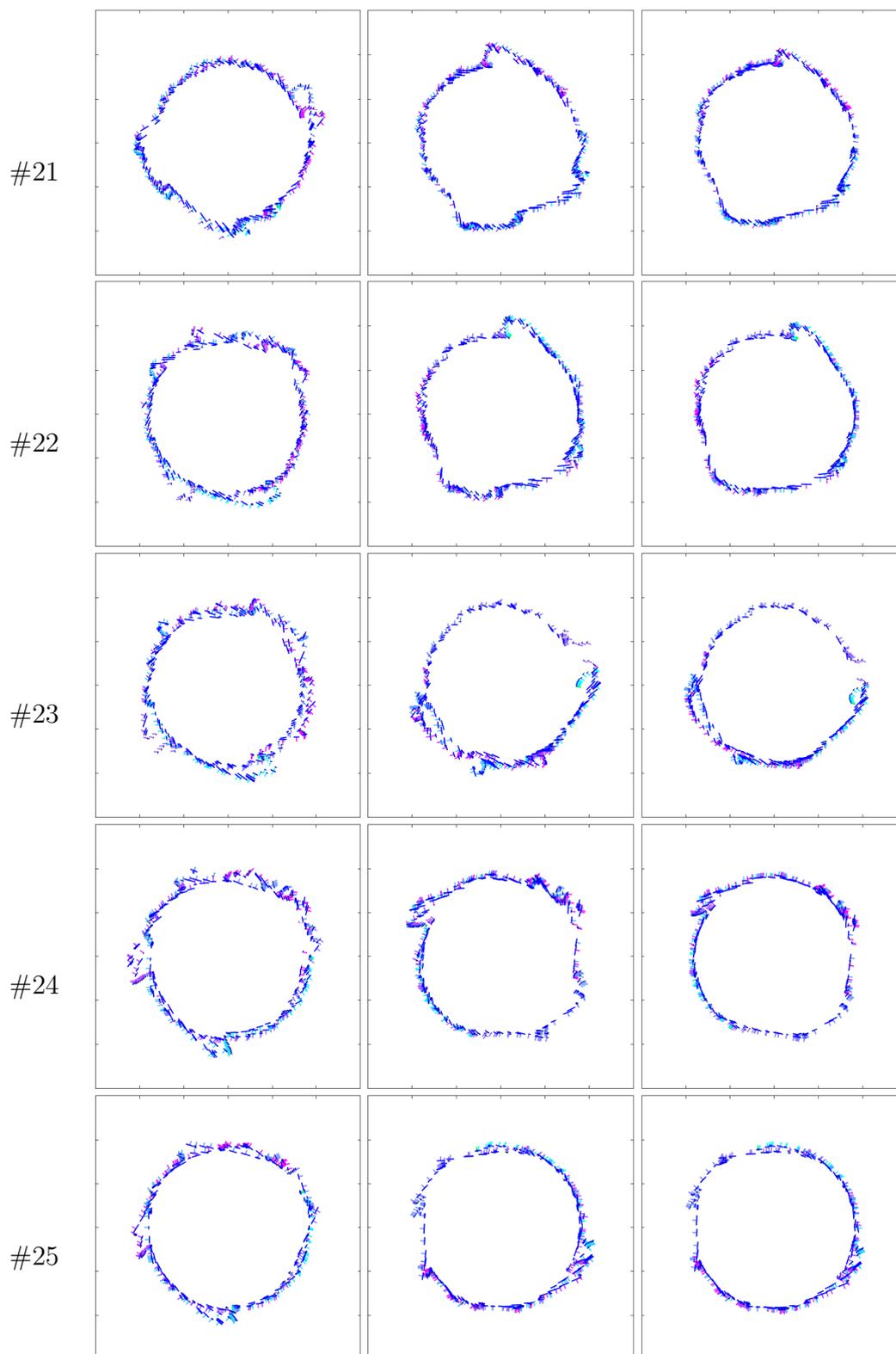


Figure B.0.27: Tactile features in object space from the datasets of the **asymmetric five edged object** (row-wise). See Fig. B.0.23 for more information.

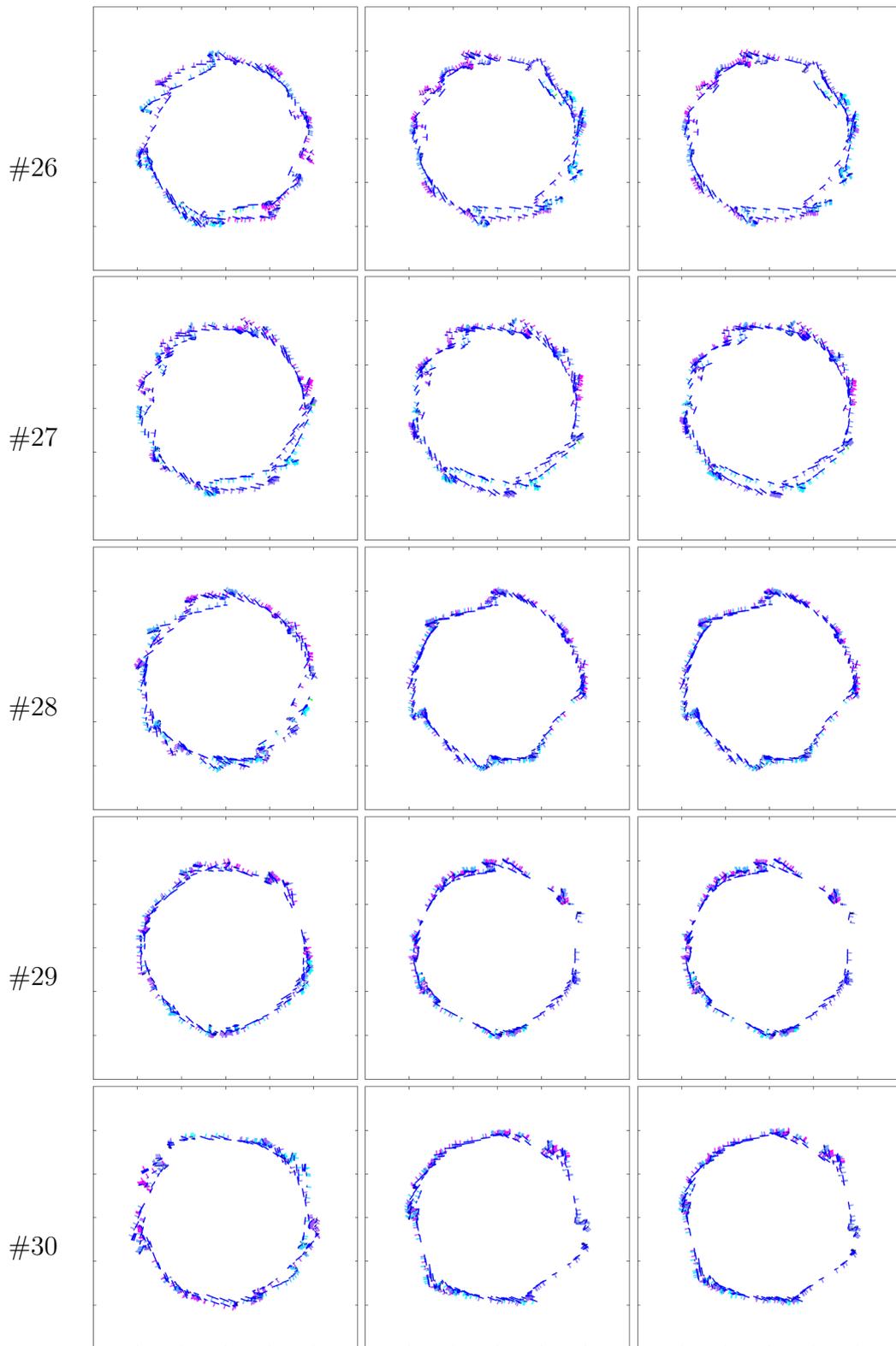


Figure B.0.28: Tactile features in object space from the datasets of the **asymmetric seven edged object** (row-wise). See Fig. B.0.23 for more information.

