



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Visual Articulated Tracking in Cluttered Environments



Christian Rauch

School of Informatics
University of Edinburgh

This dissertation is submitted for the degree of
Doctor of Philosophy

College of Science & Engineering

January 2020

Lay Summary

Robots that interact with the environment have to estimate their own internal state in addition to the external state of the world. Traditionally, this problem is approached by dedicated internal proprioceptive sensors. These sensors are limited in sensing the external world and are not available at all in some scenarios.

This thesis is therefore concerned with the estimation of the internal state of a robotic manipulator during grasping tasks, and focuses on the external perception of a robot's state using only visual sensors. Estimating the full state of an articulated robot arm and its fingers from images is difficult since an observed image of a robot manipulator will contain many additional distractions caused by the environment, the objects that are manipulated and especially by occlusions.

The visual tracking methods that are developed throughout this thesis enable the state estimation of an articulated robot manipulator in the presence of visual distractions and when this manipulator is partially occluded. These methods use machine learning techniques to detect parts and keypoints on the robot to support the discrimination between the robot and irrelevant visual distractions.

The robustness to visual distractions and the training of the discriminative methods is achieved without explicit prior knowledge about the environment, or about the manipulated and occluding objects.

Abstract

This thesis is concerned with the state estimation of an articulated robotic manipulator during interaction with its environment. Traditionally, robot state estimation has relied on proprioceptive sensors as the single source of information about the internal state. In this thesis, we are motivated to shift the focus from proprioceptive to exteroceptive sensing, which is capable to represent a holistic interpretation of the entire manipulation scene.

When visually observing grasping tasks, the tracked manipulator is subject to visual distractions caused by the background, the manipulated object and by occlusions from other objects present in the environment.

The aim of this thesis is to investigate and develop methods for the robust visual state estimation of articulated kinematic chains in cluttered environments which suffer from partial occlusions. To make these methods widely applicable to a variety of kinematic setups and unseen environments, we intentionally refrain from using prior information about the internal state of the articulated kinematic chain, and we do not explicitly model visual distractions such as the background and manipulated objects in the environment.

We approach this problem with model-fitting methods, in which an articulated model is associated to the observed data using discriminative information. We explore model-fitting objectives that are robust to occlusions and unseen environments, methods to generate synthetic training data for data-driven discriminative methods, and robust optimisers to minimise the tracking objective.

This thesis contributes (1) an automatic colour and depth image synthesis pipeline for data-driven learning without depending on a real articulated robot; (2) a training strategy for discriminative model-fitting objectives with an implicit representation of objects; (3) a tracking objective that is able to track occluded parts of a kinematic chain; and finally (4) a robust multi-hypotheses optimiser.

These contributions are evaluated on two robotic platforms in different environments and with different manipulated and occluding objects. We demonstrate that our image synthesis pipeline generalises well to colour and depth observations of the real robot without requiring real ground truth labelled images. While this synthesis approach introduces a visual simulation-to-reality gap, the combination of our robust tracking objective and optimiser enables stable tracking of an occluded end-effector during manipulation tasks.

Acknowledgements

It's done. This very thesis has taken a central place of the past years of my life. It would not have been possible without the people that supported me during this time and to which I owe my gratitude.

Foremost, I want to express my deep gratitude to my supervisors Dr. Maurice Fallon, Dr. Timothy Hospedales, and Dr. Jamie Shotton for their guidance and professional advice. I also want to thank Microsoft Research for funding this work through its PhD Scholarship Programme, and the Edinburgh Centre for Robotics for creating this inspiring research environment.

I wish to thank the MIG group for their inspiration and feedback on my ideas, and for after-hour activities. I am grateful to the SLMC group for access to the KUKA LWR arm, a robot that has seen many generations of PhD students, to collect evaluation sequences. A special thank you goes to Vladimir Ivan and Wolfgang Merkt for help with the KUKA setup and for advice on using EXOTica.

A big thank you goes to the DRS group at ORI for hosting me during my visit, and to Sudhanshu Kasewa, Chia-Man Hung and Kevin Li Sun for help with the Jaco setup.

I also want to mention Raluca Scona, Jan Stankiewicz and Radim Tyleček for fruitful lunch and coffee-break discussions.

At last, my deepest gratitude goes to my family and Julia Feist for their support and patience on this way:

Ich danke euch von ganzem Herzen für eure Unterstützung und Geduld in dieser Zeit. Auch wenn ich oft zu beschäftigt war – Ich habe immer an euch gedacht. Ohne euch wäre das alles nicht möglich gewesen.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Christian Rauch
January 2020

Wahrlich es ist nicht das Wissen, sondern das Lernen,
nicht das Besitzen, sondern das Erwerben,
nicht das Da-Seyn, sondern das Hinkommen,
was den grössten Genuss gewährt.

— Johann Carl Friedrich Gauß

“Truly it is not the knowledge but the learning,
not the owning but the acquiring,
not the being there but the getting there,
which grants the greatest pleasures.”

— From a letter from Carl Friedrich Gauss to Wolfgang Bolyai
(Göttingen, 2. September 1808)

Table of Contents

Nomenclature	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Formulation	2
1.3 Requirements and Assumptions	4
1.4 Research Questions	5
1.5 Outline of Contributions	5
2 Background	7
2.1 Preliminaries	7
2.1.1 Image Generation	8
2.1.2 State Recovery	14
2.2 Distance Metric Representation	20
2.2.1 Direct Methods	21
2.2.2 Generative Metrics	22
2.2.3 Data-Driven Metrics	26
2.3 Optimisation Approaches	29
2.3.1 Gradient-Based	29
2.3.2 Particle-Based	30
2.4 Conclusion	31
2.4.1 Model-Based Tracking	31
2.4.2 Objective	32
2.4.3 Optimisation	33
3 Visual Articulated Tracking in the Presence of Occlusions	35
3.1 Introduction	36
3.2 Related Work	36
3.2.1 Generative Model-Fitting	36
3.2.2 Discriminative Tracking	37
3.2.3 Hybrid Objectives	38

3.3	Proposed Method	38
3.3.1	The Signed Distance Function	38
3.3.2	Data Association	39
3.3.3	Generating Training Data for Pixel Classification	41
3.3.4	Training	41
3.4	Evaluation	44
3.4.1	Platform	44
3.4.2	Data Collection	45
3.4.3	Tracking Error Metrics	46
3.4.4	Experiment 1: Discriminative Tracking	46
3.4.5	Experiment 2: Grasping	48
3.4.6	Experiment 3: Tracking in the Presence of Occlusions	48
3.5	Conclusion	58
4	Learning-driven Coarse-to-Fine Articulated Robot Tracking	59
4.1	Introduction	60
4.2	Related Work	62
4.2.1	Joint Position Distribution Prediction	62
4.2.2	Visual Features	62
4.2.3	Kinematic Optimisation	63
4.3	Method	63
4.3.1	Overview	63
4.3.2	Multi-Task Prediction	64
4.3.3	Sampling of Initial Configuration	66
4.3.4	Training	68
4.3.5	Tracking Objective	69
4.3.6	Optimisation	71
4.3.7	Tracking Pipeline	73
4.4	Evaluation	74
4.4.1	Sampling Robot States	74
4.4.2	Tracking	75
4.5	Conclusion	80
5	Multi-Hypotheses Tracking of Robotic Manipulators in Cluttered Scenes	83
5.1	Introduction	83
5.2	Related Work	84
5.2.1	Synthetic Training Sets	84
5.2.2	Optimisation with Multiple Hypotheses	85

5.3	Method	86
5.3.1	Tracking Objective	86
5.3.2	Training Data Generation	90
5.3.3	Optimiser	95
5.4	Evaluation	102
5.4.1	Platform	102
5.4.2	Tracking Sequences	102
5.4.3	Background Image Synthesis	104
5.4.4	Generalisability by Input Modality	107
5.4.5	Generalisability by Background	112
5.4.6	Reduced Prediction Complexity	115
5.4.7	Optimiser	117
5.5	Conclusion	125
6	Conclusion	127
6.1	Summary	127
6.2	Contributions	128
6.3	Discussion and Future Work	129
6.3.1	Objective	129
6.3.2	Image Synthesis	130
6.3.3	Models	131
6.3.4	Optimisation	131

Nomenclature

Roman Symbols

e	model-fitting objective
\mathbf{f}	feature vector
H_K	keypoint heatmap
\mathbf{I}	Image
J	Jacobian matrix
K	camera projection matrix
R	Rotation matrix
T	isometric Transformation
t	translation vector
\mathbf{x}	3D coordinate
x	2D coordinate

Greek Symbols

α	scalar weight
θ	articulated state

Other Symbols

$\ \cdot\ $	L2 norm
-------------	---------

Acronyms / Abbreviations

CNN	Convolutional Neural Network
DoF	Degrees of Freedom

FK	Forward Kinematics
FoV	Field of View
ICP	Iterative Closest Point
IK	Inverse Kinematics
KUKA	Keller und Knappich Augsburg
LWR	lightweight robot
PF	Particle Filter
PSO	Particle Swarm Optimisation
RF	Random Forest
RGB-D	Image channel order: red, green, blue, depth
ROI	Region Of Interest
SDF	Signed Distance Function
w.r.t.	with respect to

Chapter 1

Introduction

1.1 Motivation

Estimating one's own state is the most crucial capability when interacting with the environment. Humans have evolved a set of proprioceptive senses to perceive the internal state of our bodies and exteroceptive sensors to perceive the environment in which we act. Of these sensors, vision is arguably the most complex but also most versatile as it bridges the gap between the internal and the external state. This becomes evident when trying to reach for and grasping an object with closed eyes. Visual feedback is crucial for estimating our own state for manipulation tasks in our environment.

In robot manipulation tasks, the ability to estimate the internal state is as crucial as for humans. Traditionally, robotic manipulators rely on joint position encoder sensing to report the position of each joint in the kinematic chain. This reported joint state, together with forward kinematics of the kinematic chain, provides the pose of robot links with respect to the base frame.

Joint position encoders have the inherent issue that if their single dimensional perceived value is affected by a perturbation, they have no means to detect this unreliability through redundancy and even a small perturbation close to the root of the kinematic chain will have a large impact on the reported state of the end-effector. Visual sensors on the other hand provide a high-dimensional signal that contains sufficient redundancy to cope with low-dimensional perturbations. Moreover, joint position encoders are incapable of sensing certain effects like linkage bending and are difficult to employ in small links, like fingers for dexterous manipulation, or soft-robots.

This approach to robotic manipulation has long been used for industrial setups with sub-millimetre repeatability and precise calibration of the workspace. As manipulators become smaller and more compliant and have to operate in



Figure 1.1: BROKK BM3000 robot manipulator at Sellafield nuclear decommissioning site [59]. Since the manipulator is devoid of joint encoder sensing, a human operator (top left window) has to control each joint independently. Visual state estimation could serve as a safety system in this case and support the operator.

dynamic environments, purely proprioceptive manipulation approaches become less applicable to underactuated manipulators, such as the Robotiq 3-Finger Adaptive Robot Gripper and Kinova Jaco hand, and scenarios like nuclear decommissioning [59] that are entirely devoid of joint encoder sensing and rely on exteroceptive sensing (Figure 1.1).

This thesis therefore motivates the use of visual perception for articulated state estimation and investigates methods to enable visual state estimation in cluttered environments during manipulation tasks.

1.2 Problem Formulation

The observation of an articulated object is a function

$$f : \mathbb{R}^N \mapsto \mathbb{R}^{W \times H \times D} \quad (1.1)$$

that projects the articulated state $\theta \in \mathbb{R}^N$ to an image $\mathbf{I} \in \mathbb{R}^{W \times H \times D}$. In the real world, this projection is a physical process with many unknowns such as unmodelled objects, material properties and lighting conditions, and can only be approximated by a rendering process in the image domain that synthesises images.

The aim of articulated tracking is to recover the model's articulated state θ from an observation \mathbf{I} in a sequence of consecutive observations. That is, we need to find the inverse function

$$g = f^{-1} : \mathbb{R}^{W \times H \times D} \mapsto \mathbb{R}^N . \quad (1.2)$$

Since the original observation function f is unknown and therefore not invertible, the problem is framed as finding the optimal state θ^* that most likely explains the observation \mathbf{I} :

$$\theta^* = \arg \max_{\theta} p(\theta \mid \mathbf{I}) . \quad (1.3)$$

The likelihood of a state θ_{est} being observed in \mathbf{I}_{obs} can also be expressed by an inversely related distance metric $e(\theta, \mathbf{I})$. Since the articulated state and the image are represented in different domains, we have to find a common intermediate representation $h \in \mathbb{R}^O$ in which a distance between the estimated state and observed image can be established. Both domains are mapped to this intermediate representation by functions $h_{est} : \mathbb{R}^N \mapsto \mathbb{R}^O$ and $h_{obs} : \mathbb{R}^{W \times H \times D} \mapsto \mathbb{R}^O$.

Given a proper distance metric $e(\theta, \mathbf{I})$ and the mappings h_{est} and h_{obs} to this intermediate representation, the problem then becomes the optimisation problem:

$$\theta_{est}^* = \arg \min_{\theta} e(\theta_{est}, \mathbf{I}_{obs}) , \quad (1.4)$$

with the objective to minimise the distance:

$$e(\theta_{est}, \mathbf{I}_{obs}) = |h_{est}(\theta) - h_{obs}(\mathbf{I})| . \quad (1.5)$$

In the special case $h = \mathbf{I}$, $h_{est}(\theta)$ becomes a *generative* forward mapping process that will entirely rely on image synthesis, and in the case $h = \theta$, $h_{obs}(\mathbf{I})$ becomes a *discriminative* backward mapping process that relies on data-driven methods.

This thesis focuses on the general case, where h is a representation between θ and \mathbf{I} , and will explore different representations of this distance metric space and optimisation methods to find the optimal estimated state. For the mapping of an observation to the metric space via h_{obs} , this thesis will further explore different data-driven methods.

While motivated by state estimation for robotic manipulation, the problem formulation applies to any articulated object and the methods developed throughout this thesis are theoretically applicable to any articulated tracking problem where the state is observed by a sequence of images.

1.3 Requirements and Assumptions

This thesis aims to provide methods for the state estimation of an articulated robot manipulator with as few assumptions as possible to enable a wide range of applications. The requirements on such methods are formulated as follows:

Plausibility

The estimated state has to be kinematically and visually plausible and needs to obey physical properties. The estimator must not provide states that are impossible to reach by the physical robot.

Applicability

To be generally applicable to scenarios with poor or no proprioception, an articulated robot manipulator tracking approach needs to reduce the dependency on joint position encoder sensing and ideally can be used entirely without proprioception. The estimator should also generalise to different manipulator configurations.

Accuracy

A manipulator estimation method must yield states that are as close to the true observed state to allow manipulation of the objects. The end-effector pose error must therefore be within the bounds of the manipulated object (spatial accuracy). It is further important to maintain this accuracy over time to prevent jumps in the estimated state (temporal accuracy).

Robustness

Manipulation scenes naturally contain additional untracked objects such as the manipulated object and occlusions, as well as arbitrary backgrounds. Tracking must be robust to visual distractions from unrelated objects and specifically continue to operate if the tracked manipulator is partially occluded. This requires a robust visual feature extractor and distance metric, and an optimiser that robustly converges based on this distance metric.

To enforce plausibility, we will assume that a kinematic and geometric model of the tracked object is available. This is readily the case in robotic scenarios as this model is needed for planning and control.

In a sequence of observations of a manipulation task, we assume temporal coherency with relative small changes of the observed state between frames. Instead of estimating the state independently for every observed image frame, the optimisation of the current frame will be initialised at the estimated state of the chronological previous frame.

1.4 Research Questions

Researching methods for visual articulated tracking with the aforementioned requirements needs to answer the following questions:

Representation

Which feature representation minimises ambiguity in the scene, is robust to visual distractions and enables estimation of the state of a partially occluded articulated manipulator?

Training

In the absence of the ground truth robot state in observations, how can we make use of data-driven approaches to extract features as intermediate representation?

Optimisation

How can we efficiently explore the state space to reliably find the optimum without prior knowledge about the optimisation problem?

This thesis investigates different data-driven approaches for extracting an intermediate representation from images, training of these approaches on synthetic colour and depth images, and optimisation methods to minimise distances in this intermediate space. To solve the problem of purely visual articulated tracking without prior information, we have to find ways to train a data-driven method without real labelled data, find a feature representation that is abstract enough to be used without real data, and we have to find an optimisation method that can handle these abstract and possibly ambiguous features.

To this end, this research work contributes methods for articulated tracking and insight into the general optimisation problem on data-driven methods without real labelled data. Throughout this thesis, we will develop and evaluate methods to solve these problems with increasing complexity and while reducing limitations.

1.5 Outline of Contributions

The following chapter (Chapter 2) will introduce the concept of model-based tracking that is used throughout this thesis, and also presents related work on commonly used intermediate representations, data-driven methods and optimisation approaches for articulated tracking.

Our first contribution in Chapter 3 analyses the behaviour of state-of-the-art ICP-like methods in the presence of visual distractions and under the assumption of poor proprioception. With this insight into the problem of model-fitting

with ambiguous correspondences from raw features, we propose a discriminative model-fitting approach to unambiguously relate the estimated model to a depth observation. To differentiate between observed robot parts, a random forest (RF) is trained on synthesised depth images using abstract features that generalise well to the real depth images in our test sequences. Without prior knowledge about additional objects in the scene, we further propose a training strategy to make extracted features robust to distractions from previously unseen objects.

This approach is further expanded on in Chapter 4 by extending the input space to features from colour images to get a wider range of correspondences. This chapter introduces a multi-task convolutional neural network (CNN) for the simultaneous learning of low-level and semantic high-level depth image features as basis for the tracking objective. We reuse the previous depth image synthesis pipeline, but add a general object class to represent arbitrary manipulanda and occluders in the scene. While the initial optimisation approach relied on joint position readings for the very first image in a sequence, we explore in this chapter the initialisation of such an optimiser using a predicted distribution of states. While still relying on a classic gradient-based optimisation approach, we thereby become independent of any proprioception during training or tracking.

The final contribution in Chapter 5 builds on the findings of Chapter 4, primarily the beneficial use of colour and depth as the representation and the optimisation using a distribution of states. Instead of independently extracting features from colour and depth images as before, we combine both modalities as single input to a data-driven feature extractor. We further make no assumptions about the workspace and process the raw colour and depth images without background filtering. This reduces the complexity of the processing pipeline and has the advantage that no ad-hoc heuristics for the background model have to be optimised. To process the raw images, we extend the image synthesis pipeline to colour and provide insight into the generalisability of synthetic and real images to real colour and depth sequences under varying synthetic and real properties. Inspired by the initialisation of the optimiser from a distribution and related work on particle-based optimisers, we propose a novel optimiser that uses multiple hypotheses and resampling to robustly discover and avoid local minima.

In summary, this thesis provides (1) a fully automated image synthesis pipeline to generate colour and depth training data for data-driven approaches that does not rely on real labelled data; (2) insight into different representations as tracking objective that are robust to visual distractions; (3) the ability to track a partially occluded articulated model, without prior knowledge about specific objects in the scene; and finally (4) an optimiser that can efficiently explore the state space and robustly converge to the global optimum without using prior knowledge.

Chapter 2

Background

This chapter gives an overview of methods that are used for model-based and data-driven articulated tracking. Related to the problem formulation in Section 1.2, these methods are presented with a focus on how a distance metric between observed and estimated state is established, and how this distance is minimised. Since the minimisation of the *distance metric* is the *objective* of tracking, this chapter will use both terms interchangeably.

Preliminary methods for mapping between the image and model state space, and their notation, are introduced in Section 2.1. At this point, we will also relate the imaging and kinematic processes to the robotic platforms and imaging sensors that were used to evaluate our contributions.

Sections 2.2 and 2.3 gives an overview of different categories of tracking objectives and optimiser approaches and how these are used in relevant state-of-the-art literature. Additional more specific related work is provided in the dedicated sections in Chapters 3 to 5.

Finally, Section 2.4 motivates model-based tracking and discusses the related work and their choice of certain objectives and optimiser approaches, and thereby relates the contributions of this thesis with the state-of-the-art.

2.1 Preliminaries

This section presents some of the basic principles that are used to map between the underlying state that articulated tracking attempts to recover, and the image representation of this state, which is externally observed. Processes for the forward mapping from the state to the image are introduced in Section 2.1.1. The approximated inverse processes that are required for recovering the state from an image are introduced in Section 2.1.2.

Since the forward observation process itself is not entirely invertible, we have to find an intermediate representation h between the state θ and the image \mathbf{I} , and provide methods to map from those to this intermediate representation. In model-based tracking, the forward mapping $\theta \mapsto h$ and backward mapping $\mathbf{I} \mapsto h$ depend on the modelled kinematic and visual representation of the tracked object.

We will further discuss optimisation approaches to minimise the distances in this intermediate representation.

2.1.1 Image Generation

The observation process (eq. 1.1) projects a model state θ to an observation. Throughout this thesis, this observation is a 2D image $\mathbf{I} \in \mathbb{R}^{W \times H \times D}$, with width $W \in \mathbb{N}^+$, height $H \in \mathbb{N}^+$ and $D \in \mathbb{N}^+$ colour and depth channels.

The observation process involves two individual processes (Figure 2.1). Firstly, the state vector $\theta \in \mathbb{R}^N$ is mapped to a set of rigid 3D transformations $T \in \text{SE}(3)$ between a hierarchy of frames in a kinematic chain:

$$\theta \mapsto \{T_{i-1,i}(\theta_i) \mid i \in [1, N]\} . \quad (2.1)$$

Secondly, the visual representation of the robot is transformed to the observation frame via these chained transformations, and projected from their 3D representation $\mathbf{x} \in \mathbb{R}^3$ to 2D representations $x \in \mathbb{R}^2$ onto the image plane:

$$\{\mathbf{x}_j \mid j \in \mathbb{N}^+\} \mapsto \{x_j \mid j \in \mathbb{N}^+\} . \quad (2.2)$$

Forward Kinematics

A kinematic structure, such as a hand or arm, is represented by a tree structure, where vertices represent coordinate frames and edges represent the transformation between them. By selecting a root node, the tree becomes directed with parent and child nodes.

A rigid, isometric, transformation combines an orthogonal rotation $R \in \text{SO}(3)$ and a translation $t \in \mathbb{R}^3$. To apply rotation and translation as a single matrix multiplication, a transformation is represented as homogeneous matrix $T \in \mathbb{R}^{4 \times 4}$:

$$T = \left[\begin{array}{c|c} R & t \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (2.3)$$

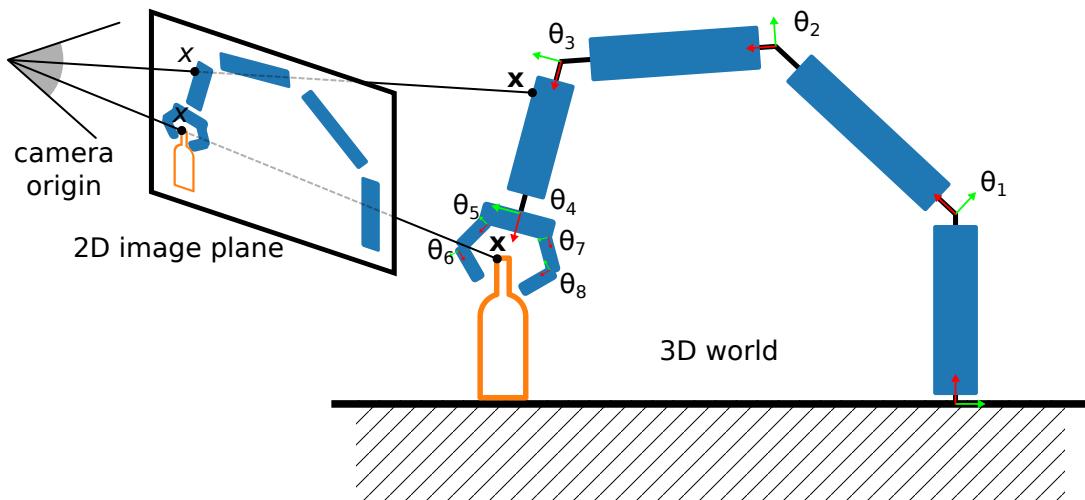


Figure 2.1: Image projection from model parameter θ . First, the tracked state θ transforms the visual parts of the robot (blue) into a common 3D space. Second, the 3D points \mathbf{x} are projected to 2D points x on the image plane. In colour images, these point coordinates on the image plane carry information about the amount of red, green and blue colour; in depth images, these points additionally hold the distance of \mathbf{x} to the image plane. Objects (orange) are not explicitly modelled or tracked.

that is applied to homogeneous coordinates $\mathbf{x}' = (x_x, x_y, x_z, 1)$ in place of the Cartesian coordinates $\mathbf{x} = (x_x, x_y, x_z)$. The transformation of a 3D point \mathbf{x}'_2 in frame 2 to a 3D point \mathbf{x}'_1 in frame 1 is denoted as

$$\mathbf{x}'_1 = T_{1,2}\mathbf{x}'_2 \quad . \quad (2.4)$$

For simplicity of notation, we drop the apostrophe and use Cartesian and homogeneous notation interchangeably, i.e. we will use \mathbf{x} in place of \mathbf{x}' .

A single joint in the kinematic tree joins two frames with a single degree of freedom. Different types of joints affect different parts of the transformation matrix. *Revolute* joints describe a rotation about the x-, y- or z-axis by ϑ :

$$R_x(\vartheta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\vartheta & -s_\vartheta \\ 0 & s_\vartheta & c_\vartheta \end{bmatrix}, \quad R_y(\vartheta) = \begin{bmatrix} c_\vartheta & 0 & s_\vartheta \\ 0 & 1 & 0 \\ -s_\vartheta & 0 & c_\vartheta \end{bmatrix}, \quad R_z(\vartheta) = \begin{bmatrix} c_\vartheta & -s_\vartheta & 0 \\ s_\vartheta & c_\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.5)$$

with $s_\vartheta = \sin(\vartheta)$ and $c_\vartheta = \cos(\vartheta)$. The translation t of a revolute joint is fixed and determined by its position in the parent frame. *Prismatic* joints describe a

translation along the x-, y- or z-axis by φ :

$$t_x(\varphi) = \begin{bmatrix} \varphi \\ 0 \\ 0 \end{bmatrix}, \quad t_y(\varphi) = \begin{bmatrix} 0 \\ \varphi \\ 0 \end{bmatrix}, \quad t_z(\varphi) = \begin{bmatrix} 0 \\ 0 \\ \varphi \end{bmatrix}. \quad (2.6)$$

The rotation R of a revolute joint is fixed and determined by its orientation in the parent frame. A full 6 DoF rigid pose transformation is represented by chaining 3 prismatic and 3 revolute joints.

Every individual single DoF in the state vector θ yields a single transformation

$$T_{i-1,i}(\theta_i) : \mathbb{R} \mapsto \text{SE}(3) \quad (2.7)$$

between the joint's parent frame $i - 1$ and its dependant child frame i . In such a chain, frame 0 is considered as the root frame. The transformation between arbitrary frames in the kinematic tree that are connected via multiple vertices is given by the chaining of the single transformations along the routed path between the frames, e.g. $T_{0,3} = T_{0,1}T_{1,2}T_{2,3}$.

Robotic manipulators may interact with a variety of rigid objects, whose state is not explicitly estimated. These objects are categorised into two classes, depending on how the robotic manipulator interacts with them. A *manipulandum* is an object that is actively manipulated and whose state changes are caused by the robot. An *occluder* is an object that is currently not manipulated and may occlude the tracked robot. An object might change between those two roles. The remaining observations are generally referred to as the *environment* or background, and are considered static and do not occlude the robot or the objects. A *scene* is the set of all tracked robot frames, manipulanda, occluders and the environment. The entire scene can be considered as a graph of transformations between any rigid visual entity, where articulated tracking is only concerned with the estimation of the transformations between the observation frame and the robot frames.

Image Projection

A 2D imaging sensor perceives the 3D world by the projection of properties of 3D points $\mathbf{x} = (x_c, y_c, z_c)$ in the camera frame to their corresponding 2D points $x = (x_i, y_i)$ on the 2D image plane. In the pinhole camera model, the physical camera imaging plane is x-y-axes-aligned and located at distance $z = f$ from the camera origin and intersects the line from this origin to the 3D point \mathbf{x} . Any point $\lambda\mathbf{x}$, with $\lambda \in \mathbb{R}^+$, on this line will result in the same 2D projection. Hence, we can arbitrarily set $\lambda = \frac{1}{z_c}$ and apply the projection on the point with homogeneous

coordinates $\mathbf{x}' = (\frac{x_c}{z_c}, \frac{y_c}{z_c}, 1)$ using the intrinsic camera matrix K :

$$x = K\mathbf{x}' \quad (2.8)$$

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \cdot \frac{1}{z_c} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad . \quad (2.9)$$

The focal lengths f_x and f_y , and the camera centre $c = (c_x, c_y)$ are expressed in pixels and are related to their physical representation by the size of a physical pixel on the camera sensor. These intrinsic parameters are obtained through calibration of the physical sensor and used for image synthesis and back-projection. For simplicity of notation, we will use the Cartesian instead of the homogeneous notation when referring to the projection, i.e. we will use $x = K\mathbf{x}$ to describe the projection of a Cartesian 3D point \mathbf{x} , via \mathbf{x}' , to its 2D image pixel location x .

Robotic Platforms

Two robotic manipulator systems (Figure 2.2) with similar kinematic structures are used for the experimental evaluation. The KUKA LWR4 arm (15kg) with the Schunk SDH2 hand (1.95kg) is targeted towards industrial applications. The more lightweight Kinova Jaco (5.2kg), on the other hand, targets assistive robotics with manual end-effector pose control.

The KUKA LWR and Kinova Jaco setup both have an arm which consists of a single kinematic path from the base frame to the palm frame, from where the kinematic structure branches into three fingers each with two joints. We refer to the base frame as the frame with which a robot is rigidly attached to the world frame, and refer to the palm frame as the vertex in the kinematic tree where the kinematic structure branches into paths for the finger frames. The camera frame is connected to this transformation graph by the visually estimated 6D pose of AprilTags [58], which are rigidly attached to the base frame of the kinematic tree.

In both setups, the fingers consist of two physical links each, which can also be referred to by their medical name *phalanges*. These phalanges are further separated into proximal phalanges, which have a small geodesic distance to the palm frame, and distal phalanges, which have a larger geodesic distance to the palm frame and at the same time define the leaf-vertices of the kinematic tree.

While the phalanges of the Schunk SDH2 are fully actuated by 7 DoF (2 degrees per finger and one additional degree for longitudinal paired rotation about 2 of the 3 finger axes), the Jaco fingers are each only actuated by a single degree. Each of the 3 Jaco fingers is tendon-driven. The proximal and distal phalanges adapt



Figure 2.2: Robotic manipulators. (a) The 7 DoF KUKA arm is rigidly attached to a table with AprilTag markers for camera pose estimation. (b) The Schunk SDH2 is a fully actuated 7 DoF hand that is mounted on the tip of the KUKA arm. (c) The Kinova Jaco combines a 6 DoF arm with an underactuated 6 DoF hand.

dynamically to the shape of a grasped manipulandum and cannot be actuated individually. Further, since there is no unique relation between the state of the tendon and the state of the phalanges, it is not possible to proprioceptively sense the full state of the finger joints.

Imaging Sensors

The tracking approaches developed in this thesis are generally applicable to visual sensors that provide a 2D image $\mathbf{I} \in \mathbb{R}^{W \times H}$ with intensity and depth information per pixel. A variety of visual sensors exist that are able to sense depth directly via time-of-flight or indirectly via triangulation. The Asus Xtion PRO LIVE sensor (Figure 2.3), that is used during the experiments, combines a traditional colour intensity sensor with a structured light sensor to provide an RGB-D observation $\mathbf{I} \in \mathbb{R}^{W \times H \times 4}$ of the manipulation scene.

The passive sensing of colour light intensity works according to the casting of light rays as described in the pinhole camera's projection equation (eq. 2.8). The depth of a projected pixel is additionally actively sensed by the triangulation of known patterns that are projected and sensed by an infrared (IR) projector and sensor, respectively. Both visual sensors operate according to the same pinhole camera principles, but maintain independent camera intrinsics and frames. That is,



Figure 2.3: Asus Xtion PRO LIVE RGB-D sensor [34]. The sensor combines a colour sensor (middle), with depth estimation from structured infrared light (left, right).

the colour and depth information about a pixel is expressed in different observation frames. With the known transformation between both optical centres, the depth information is registered with respect to the colour sensor frame. After this depth-to-colour frame registration, the colour and depth information is expressed in a common observation frame, using the same projection model.

Image Synthesis

The *real* image generation process, i.e. the mapping from an unknown state θ to an observed image \mathbf{I} , is a complex physical process that involves many unknowns about the propagation of light through the environment. The research area of computer graphics attempts to simulate these processes to generate *synthetic* images, i.e. map a known state to a synthetic image, through a simulation process that yields real visual properties.

While the geometric propagation of single rays of light is better understood and easier to simulate, additional intensity and colour forming processes are more difficult to simulate. Additional processes involve for example diffuse and specular reflection, which affects the propagation of light intensities, and the filtering of different wavelengths in the light spectrum, which affects the perception of colour. These processes strongly depend on material properties and the light source, and are in general difficult to simulate entirely realistically.

The geometric synthesis of an image, that is, the transformation of polygon meshes from the local robot frames by the estimated state into a common observation frame (eq. 2.4), and the projection of 3D points on these meshes onto the image plane (eq. 2.8), provides the estimated depth image $\mathbf{I}_{D,est}$.

By choosing this depth as an intermediate representation, the function $h_{est}(\theta)$ becomes the synthesis process and we already arrive at a simple way to map from state to image. Such generative methods (Section 2.2.2), also called analysis-by-synthesis, evaluate estimated states by comparing the estimated depth image with

the observed depth image. However, since the estimated image is only generated by the tracked state, it neglects any untracked objects in the scene and does not benefit from colour information.

2.1.2 State Recovery

The previous section described image formation as a forward mapping from a lower dimensional state to the higher dimensional image, by means of a real physical or synthetic rendering process. The aim of articulated state estimation is to invert this mapping (eq. 1.2), to recover the lower dimensional state θ from an image \mathbf{I} .

Back-Projection

The forward image projection (eq. 2.8) is a linear relation that can be reversed. Since $K \in \mathbb{R}^{2 \times 3}$ itself is not invertible, we will use its homogeneous representation

$$K' = \begin{bmatrix} K \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

with the inverse being:

$$K'^{-1} = \begin{bmatrix} \frac{1}{f_x} & 0 & -\frac{c_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} K^\dagger \\ 0 & 0 & 1 \end{bmatrix} . \quad (2.11)$$

The back-projection of a 2D image coordinate x to the 3D space is then given by:

$$\mathbf{x}' = K'^{-1}x' . \quad (2.12)$$

With $x' = (x_i, y_i, z_i)$, that incorporates the 2D image coordinate and its unknown depth z_i , the back-projection equation resolves to:

$$\lambda \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} K^\dagger \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} , \quad (2.13)$$

and it becomes clear that there is no unique solution to this inverse process, since neither λ nor z_i are known. This is because multiple points on the line-of-sight are projected to the same image coordinates. Only with known depth z_i can $\lambda = \frac{z_i}{z_c}$ be solved to recover the 3D point $\mathbf{x} = \frac{1}{\lambda}\mathbf{x}'$. To simplify the notation in later chapters,

we will drop the apostrophe for homogeneous representations, and will denote the back-projection as $\mathbf{x} = K^{-1}\mathbf{x}$.

Inverse Kinematics

In contrast to the linear image projection, the forward kinematic mapping from the state to frame poses is non-linear w.r.t. θ (eq. 2.5). However, very small changes in the joint state $\delta\theta = \theta_{t+1} - \theta_t$ can produce small changes in the task space pose $\delta T = T(\theta_{t+1}) - T(\theta_t)$, and this relation becomes linear.

The linear relation will be expressed by the 6 DoF pose $\mathbf{p} \in \mathbb{R}^6$ (3 DoF position p , and 3 DoF orientation o), $\mathbf{p}(\theta) = (p_x(\theta), p_y(\theta), p_z(\theta), o_x(\theta), o_y(\theta), o_z(\theta))$, instead of $T \in \text{SE}(3)$. The partial derivatives of $\frac{\partial \mathbf{p}}{\partial \theta}$ are arranged in matrix form as kinematic Jacobian $J \in \mathbb{R}^{6 \times N}$:

$$J(\theta) = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \dots & \frac{\partial p_x}{\partial \theta_N} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \dots & \frac{\partial p_y}{\partial \theta_N} \\ \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} & \dots & \frac{\partial p_z}{\partial \theta_N} \\ \frac{\partial o_x}{\partial \theta_1} & \frac{\partial o_x}{\partial \theta_2} & \dots & \frac{\partial o_x}{\partial \theta_N} \\ \frac{\partial o_y}{\partial \theta_1} & \frac{\partial o_y}{\partial \theta_2} & \dots & \frac{\partial o_y}{\partial \theta_N} \\ \frac{\partial o_z}{\partial \theta_1} & \frac{\partial o_z}{\partial \theta_2} & \dots & \frac{\partial o_z}{\partial \theta_N} \end{bmatrix}, \quad (2.14)$$

and finally provide the linear relation between joint state and task space:

$$\delta \mathbf{p} = \begin{bmatrix} \delta p_x \\ \delta p_y \\ \delta p_z \\ \delta o_x \\ \delta o_y \\ \delta o_z \end{bmatrix} = J(\theta) \cdot \delta \theta, \quad (2.15)$$

for a single Cartesian frame in the kinematic chain. For multiple Cartesian frames F , these Jacobians are stacked along the task space dimension:

$$J(\theta) = \begin{bmatrix} J_1(\theta) \\ J_2(\theta) \\ \vdots \\ J_F(\theta) \end{bmatrix}. \quad (2.16)$$

In the general case J is not a square matrix and thus not invertible. We will thus either use its Moore–Penrose inverse $J^\dagger = (J^\top J)^{-1} J^\top$, if an inverse is directly required, or indirectly and more efficiently solve for $\delta\theta$ numerically. In both cases,

we will denote the inverse mapping as:

$$\delta\theta = J^\dagger(\theta) \cdot \delta\mathbf{p} \quad . \quad (2.17)$$

During visual tracking, the frame pose \mathbf{p} is usually not directly observed. Tracking objectives in later chapters will refer to local features in these frames. When deriving task-specific gradients, by chaining the frame pose gradients (eq. 2.14), a task-specific Jacobian is formed and used to linearly relate the estimated state and the observed features in the same manner.

Optimisation Methods

The aforementioned intermediate representation provides a distance metric space, to which we map from the observed image and the estimated state. The distance metric (eq. 1.5) is the discrepancy between these two mapped representations and the optimal estimated state is the one which minimises this distance (eq. 1.4). This state is referred to as the *global minimum* of the objective. If the distance metric is non-convex it will have multiple additional *local minima*, to which an optimiser must not converge. This thesis will primarily discuss *gradient-based* and *particle-based* approaches to minimise this distance $e(\theta, \mathbf{I})$.

gradient-based In its principal form, gradient approaches use the partial derivatives $\frac{\partial e}{\partial \theta}$ of the objective as a linearised search direction. These objective gradients are derived from the kinematic gradients of the frame poses (eq. 2.14), and are arranged similarly in matrix form as Jacobian J . To minimise the distance in the intermediate space by $\delta e(\theta, \mathbf{I})$, we would need to update the state by the gradient descent step

$$\delta\theta = J(\theta)^\top \cdot \delta e(\theta, \mathbf{I}) \quad . \quad (2.18)$$

Since J is locally linearised, taking large steps $\delta e(\theta, \mathbf{I})$ will neglect non-linearities and result in over- or under-estimated updates. It is therefore required to apply this process iteratively using small steps: $\theta_{i+1} = \theta_i - \lambda \cdot \delta\theta$. Further, since we are taking small steps, the initial state θ_0 has to be close to the minimum.

We will briefly derive the Gauss-Newton method, as a commonly applied gradient-based minimisation approach, using the textbook notation x for the parameter of a function f that is to be minimised.

The Gauss-Newton method is a generalisation of Newton's method to non-linear vector-valued functions [77]. Newton's method is used to find the minimum

of a scalar function, by the iterative process:

$$x_{i+1} = x_i - \frac{f^{(1)}(x_i)}{f^{(2)}(x_i)} \quad (2.19)$$

$$= x_i - (f^{(2)}(x_i))^{-1} \cdot f^{(1)}(x_i) \quad (2.20)$$

where the ratio of the first- and second-order derivative serves as the search direction for the minimum. The general formulation for vector-valued functions $f : \mathbb{R}^n \mapsto \mathbb{R}^m$,

$$x_{i+1} = x_i - (\nabla^2 f(x_i))^{-1} \cdot \nabla f(x_i) \quad , \quad (2.21)$$

uses the gradient $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^\top$ in place of the derivative.

For non-linear least-squares problems of the form

$$\arg \min_x \sum_j^m r_j(x)^2 = \arg \min_x r(x)^\top r(x) \quad (2.22)$$

with the residual vector $r(x) = (r_1(x), \dots, r_m(x))^\top$, the partial derivatives are arranged as a Jacobian matrix,

$$J(x) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \dots & \frac{\partial r_m}{\partial x_1} \\ \vdots & \ddots & \\ \frac{\partial r_1}{\partial x_n} & & \frac{\partial r_m}{\partial x_n} \end{bmatrix} \quad , \quad (2.23)$$

which gives the first order gradient $\nabla f(x) = J(x)^\top r(x)$. For very small residuals r , the Hessian matrix with the second order partial derivatives can be approximated as $H(x) = \nabla^2 f(x) \approx J(x)^\top J(x)$. With this, the non-linear least-squares problem is solved by the iterative Gauss-Newton formulation:

$$x_{i+1} = x_i - (J(x_i)^\top J(x_i))^{-1} \cdot (J(x_i)^\top r(x_i)) \quad (2.24)$$

$$= x_i - J^\dagger(x_i) r(x_i) \quad (2.25)$$

with the pseudo-inverse $J^\dagger = (J^\top J)^{-1} J^\top$.

particle-based Without an analytic search direction, optimiser approaches have to resort to a random sampling of the objective function at different states. These approaches have to maintain a distribution of possible state hypotheses, also called particles, to gather sufficient information about the objective. This is typically inefficient but has the advantage that these approaches can be used for

non-differentiable objectives, and since randomly initialised, are less likely to get stuck in local minima.

The general process is to initialise a set of samples from the state space,

$$S = \{\theta_i \mid \theta_i \sim \mathcal{U}\} \quad , \quad (2.26)$$

evaluate the objective at every sample,

$$E = \{\epsilon_i \mid \epsilon_i = e(\theta_i, \mathbf{I}) \forall \theta_i \in S\} \quad , \quad (2.27)$$

and update these states with local and global information about the performance of the distribution. How these states are updated differs between types of particle-based approaches. Most approaches share information between particles, either by biasing them with the current globally best solution, or by weighted resampling, to shape the distribution towards the global minimum.

Data-Driven Methods

While the intermediate representation h can be an arbitrarily chosen space, the forward and backward processes need to be capable to map into this space. Ideally, the distance in this space is differentiable. On one side, image synthesis with controlled properties allows to map from the state space to a higher dimensional geometric or visual space. On the other side, the inverse processes for back-projection and inverse kinematics only account for geometric properties.

Image processing methods enable the extraction of more or less semantic meaningful visual features from images. Data-driven methods allow to optimise the mapping from the image space to a user-chosen intermediate representation, on a distribution of data samples. Hence, the aim of data-driven methods is to find the optimal function $h_{obs}(\mathbf{I})$ that maps from the visual observation to the target intermediate space, such that the optimised function produces the same mapping as the given target mapping from training samples.

With an unknown hidden state θ , this mapping cannot be provided for a real image \mathbf{I} . However, the image synthesis pipeline can be used to approximate these images for a variety of states. In contrast to generative methods, where the estimated observation is synthesised *a posteriori* from the estimated state ($\theta_{est} \mapsto \mathbf{I}_{est,syn}$), data-driven methods synthesise observations *a priori* for a distribution of possible expected observed states ($\theta_{obs} \mapsto \mathbf{I}_{obs,syn}$). Since the synthesised images are just an approximation of the real imaging process, this creates a gap in the expected prediction performance.

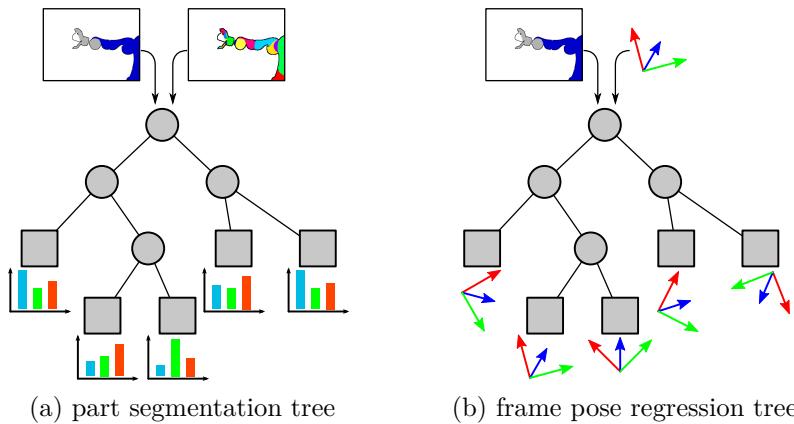


Figure 2.4: Illustrative examples for configurations of a decision trees. During training, the input samples are partitioned at split nodes (circles) and accumulated in leaf nodes (squares). During test, a single sample is routed along the split nodes to a leaf node to recover its distribution. (a) A classification tree assembles a distribution of discrete variables at its leaf nodes. This configuration can be used for a discrete classification of segments in an image. (b) A regression tree assembles a continuous distribution, here visualised by the mode. This can be used to predict continuous real-valued variables like a 6D pose vector.

In the following, we will give a high-level overview of two data-driven approaches, random forests (RF) and convolutional neural networks (CNN), to relate and contrast their properties for the application of mapping from observed images to an intermediate representation in later chapters. These methods are in general trained to minimise a loss that is defined within the targeted intermediate space. These targets can be discrete (classification) or continuous (regression) variables.

Random Forest A random forest [18] is an ensemble of individually trained decision trees (DT). Each DT consists of split nodes, that divide their input set into two partitions, and leaf nodes that accumulate a discrete or continuous probability distribution of the target property (Figure 2.4).

The input to these nodes is typically not the raw data such as pixel intensities of an image, but post-processed low-level features like Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HoG). These features are extracted for all training samples and randomly partitioned into subsets for the individual DTs. The split nodes contain very simple trainable functions that partition the feature space of the training sample subset, such that the two splits of this subset maximise the information gain. With the hierarchical alignment of these split nodes, the partitioning along a path in the DT becomes more specific and confident in the mapping of input features to the output target.

The outcome of the individual DTs in the RF is merged into a single discrete or continuous distribution. This prevents overfitting of a single DT and also yields smoother estimates. In the classic implementation of RF, each DT operates on a single batch of training samples and split nodes are optimised in a single instance. This has the disadvantage that the size of the training set is limited by the available memory and that the RF cannot be further optimised once trained on a training set. Further, the choice of input features is an input-data and output-task specific decision which requires domain knowledge and is not transferable to other modalities.

Convolutional Neural Network Convolutional neural networks [30] are a hierarchically layered ensemble of convolutional kernels, whose filter weights are optimised on the given training samples. These convolutions typically operate on the raw input data in the first layer, and consecutively on the filter responses in higher layers, with additional non-linear mappings of the filter responses by activation functions.

Compared to RFs, this convolutional and non-linear mapping on raw input data enables the simultaneous learning of the feature representation, without the need for domain knowledge. The arbitrary arrangement of these layers enables arbitrarily complex mappings between the image and the target intermediate representation h . This includes multi-input and multi-output configurations, where multiple image modalities, for example colour and depth, can simultaneously be processed and mapped to multiple targets, such as classification and regression tasks.

2.2 Distance Metric Representation

The distance metric is the space in which an estimated and observed state are projected into. This space can take an arbitrary form but has to represent a proper distance metric that can be minimised by the optimiser.

In this section we discuss three categories of intermediate representations: direct, generative and data-driven; and how they are extracted and applied in related work. Depending on the representation of the distance metric, related literature uses the term *joint position* to refer to different spaces. We will use the term *model joint position* to refer to the model parameters θ in the model state space, and *Cartesian joint position* to refer to the Cartesian 3D coordinates \mathbf{x} of a joint in the task space or observation frame. The Cartesian joint position is defined as the origin of the child frame that this 1 DoF joint directly affects.

2.2.1 Direct Methods

In direct methods, the distance metric is directly mapped to the state space of the tracked object without an intermediate representation. These methods do not involve any form of forward projection from the estimated state to the image during the tracking, but entirely rely on the backward mapping.

In the literature, two common formulations of this problem can be found. The problem is either framed as regressing from an image directly to the model joint positions (joint space), or it is framed as regressing to the Cartesian joint positions (task space). To some extent, both formulations are equivalent for rigid objects since a single frame pose can be expressed by joints. Recovering the model state from frame poses of articulated objects further requires an inverse kinematics (IK) stage.

Since the geometric methods for the inverse image projection and kinematics are not sufficient to map from a real observed image to the underlying state, finding $h_{obs}(\mathbf{I})$ becomes a regression problem that can be approached with data-driven methods.

Joint Space Regression

In joint space regression, the forward mapping h_{est} becomes the identity function $h_{est} : \theta \mapsto \theta$ and the backward mapping h_{obs} becomes the inverted projection function $h_{obs} = g : \mathbf{I} \mapsto \theta$. The regression from the image domain to the model's joint positions requires a data-driven method to approximate the relations that are otherwise given by the inverse image projection (eq. 2.12) and kinematic equations (eq. 2.17).

Such a direct approach has been demonstrated for the regression of robotic arm joint configurations, using a regression forest on a pre-segmented depth image [85], and using a CNN on stereo-image pairs [54]. In such an approach the accumulation of small joint space errors over the path within the kinematic chain might result in large task space errors, but no task space errors have been reported for these works.

Task Space Regression

Task space regression maps from the 2D image space to a 3D representation in the task space. While such an approach does not need to model kinematic properties, it still needs to infer the camera intrinsics.

A regression to 3D orientations, exemplarily for orientation estimation of rigid 3D objects from 2D images [47], using a similar architecture as [54], represents

orientations as a single real-valued vector. However, 3D orientations do not have a canonical form in which they can be expressed. Multiple representations like rotation matrices ($\mathbb{R}^{3 \times 3}$), quaternions (\mathbb{R}^4), axis-angle (\mathbb{R}^4) and Euler angle (\mathbb{R}^3) representations exist, and a regression approach has to select one of these representations. Further, these representations have a rather complex relation to the raw image data.

The regression to multiple 3D Cartesian joint positions in a kinematic tree, using the same approach of combining a CNN with a fully convolutional layer, was shown by [56] for hand pose estimation. For a hand model with $\theta \in \mathbb{R}^{N=14}$, this increases the regression target size to $\mathbb{R}^{3 \cdot 14}$. Through this model, the 48 Cartesian joint coordinates are kinematically related. An important finding of this work is that a lower dimensional embedding (8 dimensions) of this regression task provides better results than the direct regression to the 42 coordinates. This work is further extended with a regression to Cartesian position updates, from the distance of the synthesised estimated depth image to the observed depth image, to iteratively update the initially regressed Cartesian joint positions [55].

Kinematic Prior

The regression to task space Cartesian coordinates in a kinematic chain captures many redundancies due to the kinematic relation between these coordinates. While this issue can be approached by the dimensionality reduction in a lower-dimensional latent space [56], kinematic models, as commonly available in robotics, directly model this relation between Cartesian frames through forward kinematics (FK).

The integration of a kinematic model in the training of a regressor for model joint positions can improve the regression model by forward modelling of the model parameters to a loss based on the Cartesian position in the task space [90]. This is in line with findings from dimensionality reduction [56], but explicitly models this relation. However, this particular application with humanoid sized kinematic models reports average 3D position errors of above 10cm, which makes this particular approach unsuitable for manipulation tasks. It is further noted that the direct regression of model joint parameters did not work in their case.

2.2.2 Generative Metrics

As a generative metric, we categorise intermediate representations that are directly derived from the raw image data without using data-driven approaches. Specifically, the function h_{obs} that maps the input data to the intermediate space is not automatically optimised on mappings provided by training data. These

intermediate representations are usually less complex and therefore have less discriminative power. We refer to this as *generative*, since these intermediate representations are generated online during tracking by image synthesis in h_{est} and from the real observation in h_{obs} .

Points $p_h = (\mathbf{x}, \mathbf{f})$ in the space of this intermediate representation have a spatial property that is given by the coordinates in the 2D image ($x \in \mathbb{R}^2$) or 3D camera frame ($\mathbf{x} \in \mathbb{R}^3$), and a visual property that encodes features ($\mathbf{f} \in \mathbb{R}^{N_F}$) of this data point and its neighbourhood.

Generative metrics can further be categorised by how the points p_h in the intermediate space are associated to each other. This association problem is: Given a set of points $\{p_{h_{obs},i} \mid i \in N_{obs}\}$ from mapping from the observed image to the intermediate space, and a set of points $\{p_{h_{est},j} \mid j \in N_{est}\}$ from the synthesised estimated state, we have to find pairs $(p_{h_{obs},i}, p_{h_{est},j})$ to establish a distance between the observation and estimation.

Association by Identical Coordinates

Between two images $\mathbf{I} \in \mathbb{R}^{W \times H}$ with the same dimensionality (W, H) and discrete coordinates $x \in \mathbb{N}_{>0}^2$, points in both images can directly be associated by their identical 2D coordinates, i.e. where $x_{obs} = x_{est}$. That is, given an observed image \mathbf{I}_{obs} and a synthesised image of an estimated state $\mathbf{I}_{est,syn}$, a distance of the form $|\mathbf{I}_{obs}(x) - \mathbf{I}_{est,syn}(x)|$ is established over the entire domain of $W \times H$. Since such a metric does not consider alternative pairs of coordinates, it is entirely defined by the visual features \mathbf{f} .

Early approaches for 3D rigid pose estimation on 2D colour images used the overlap of observed and estimated edges as a distance metric [3]. In such an approach, the estimated edges are obtained by rendering the contour edges of an estimated model state.

With the rise of commodity 3D sensors, these approaches shifted to the comparison of raw depth readings with the synthesised depth image [86, 35, 27, 78]. This distance can take a simple form like $\sum_{x \in \mathbf{I}} |\mathbf{I}_{D,obs}(x) - \mathbf{I}_{D,est}(x)|$, with the observed image $\mathbf{I}_{D,obs}$ and the synthesised depth image $\mathbf{I}_{D,est}$.

Colour can be incorporated by weighting the scalar depth distance with a simple skin colour mask [48], or by representing colour differences as L2 distance in the RGB colour space [17].

Since such simple distance representations cover the entire image area, it is typically required that the observed image does not contain any other observations than those caused by the tracked object. Limiting the area for distance computation to a ROI [70] relaxes this constraint.

The distance metric by matching coordinates provides a computational efficient tracking objective that can be evaluated fast. Since the distance is established between properties of coordinates that are already associated to each other in the spatial space, this metric gives no information about potential alternative associations that would result in better matches in the feature space. That is, there is no information about how the estimated state has to be updated to minimise this distance. As a consequence of this, approaches that rely on correspondences by matching coordinates have to explore the model state space using particle-based approaches.

Template Matching

Similar to differences from properties at identical coordinates, template matching aims to establish this distance between a ROI in the reference image \mathbf{I}_{obs} and a synthesised template $\mathbf{I}_{est,syn}$ of the estimated state. In contrast, these templates are not synthesised from an estimated state online during tracking, but offline for a set of discrete model states $\{\theta_t \mid t \in \mathbb{N}\}$. The aim is then to find the template configuration θ_t and the 2D location of the template in \mathbf{I}_{obs} , or its matching ROI respectively, such that the distance between features \mathbf{f} of pixels with identical coordinates x , within the template and ROI, is minimal. A template distance is therefore defined in the feature space, but subject to its discrete configuration and 2D location in the image.

Early template methods for 3D pose estimation used gradient orientations [33] in colour images which can then be extended by normals in depth images within the same framework. While this was optimised for parallel computation, it still requires an exhaustive online search over the template configurations and locations. A more efficient method to find the matching template and its location in the observed image was presented by [12]. The coverage of the space of object orientations and scales yields a set of synthesised template patches, and all considered ROI locations in \mathbf{I}_{obs} yield a set of image patches. After transforming those patches by Laplacian of Gaussian (LoG), they are vectorised and concatenated within their set. The optimal template configuration and location is then given by the highest response in the cross-correlation matrix.

Compared to optimisation approaches where a continuous state estimate is maintained and the visual representation of an estimated state is synthesised online during the optimisation, template matching approaches involve an exhaustive search over discrete states. This is feasible for rigid object template configurations with a 3 DoF orientation and 1 DoF scale, but it becomes intractable for articulated

objects where each additional state variable increases the amount of templates exponentially by the number of discrete values in a state variable.

Association by Closest Coordinates

A distance by closest coordinates is primarily defined in the spatial space of 2D coordinates x in the image plane or 3D coordinates \mathbf{x} in the camera frame. Contrary to the association by identical coordinates or template matching, points in the intermediate space have to be associated during the optimisation. This association can additionally use features \mathbf{f} of these points as a secondary distance metric. If these features are not sufficiently discriminative, only the coordinates are considered and the distance is simply the L2 norm of the coordinates.

2D Distances Edges have long been used as simple binary feature for rigid pose estimation [57, 16]. Intensity edges are simple to compute from an observed image and contour edges are also simple to synthesise from an estimated state using the kinematic and geometric forward model. While the extraction of semantic meaningful edges works well on images with a low amount of structural information, such as uniformly coloured objects without texture, they are easily impaired by noise or texture. Further processed higher-level features such as SIFT [61] and HoG [52] are used to partially resolve ambiguity as found in low-level features. These higher-level representations encode a higher amount of information than edges and are therefore computationally more expensive to extract. Synthesising these features for an estimated model state requires structural visual models to yield similar feature responses. Keypoint coordinates are associated by their closest distance via a distance transform [57] or ICP [16], or by a robuster method like RANdom SAmples Consensus (RANSAC) [61].

3D Distances In 3D observations as point clouds from back-projected depth images, ICP-like relations between the point cloud and the model are commonly the basis for a distance metric. Originally applied to estimate rigid transforms between point clouds, this has since been extended to articulated objects.

With a known model configuration, excluding its 6 DoF rigid pose, an articulated model can be assumed rigid [60] and traditional ICP metrics can be used. Distance metrics for articulated objects that are represented as kinematically connected rigid objects can be locally defined per link [67, 21]. However, models that are represented as mesh [67], and not as rendered point cloud [21], can not directly rely on point distances and need to resort to a dedicated mesh distance function. This distance function can be defined for rigid meshes [67] or over

smooth blended meshes [79], and for a given mesh configuration and a point in the mesh frame provides the shortest Euclidean distance between them. In addition to coordinates, points can have 3D features such as normals [79] which can better discriminate them.

Distances in 3D can straightforwardly be related to the underlying kinematic model state via the inverse kinematic equations. This enables differentiation of this distance w.r.t. the state θ and provides analytic gradients for optimisation. As a result, many approaches in this area apply gradient-based optimisation approaches using an initial state close to the optimum.

2.2.3 Data-Driven Metrics

Encoding semantic meaningful features via h_{obs} with a higher amount of information requires more complex methods with tunable parameters. These parameters are optimised offline with a secondary objective independently from our primary tracking objective. Whereas in generative methods the extraction function h_{obs} is a design decision and the intermediate representation is a given, data-driven metrics allow to design this intermediate representation and automatically find h_{obs} . Although it can be an arbitrarily chosen space, this secondary objective needs to be a trainable mapping from the image and its representation must be able to be synthesised from the estimated state.

As a design decision, a data-driven representation should be chosen to maximise discrimination between points in the intermediate space, so as to minimise the amount of local minima in a thereon building primary tracking objective. Tracking objectives can use a combination of generative and data-driven metrics. While semantic features provide the advantage to better discriminate between feature points, they require training with a large amount of data.

Segmentation

In a segmentation task, regions of connected pixels are assigned with a segment or class ID. This is represented by an one-hot vector $\mathbf{f} \in \mathbb{B}_S^N$, with $\mathbb{B} = \{0, 1\}$, where the i -th segment is set to 1 ($f_i = 1$) and the remaining segments are set to 0. In practice a predictor, such as a RF or CNN, will provide a real-valued vector $\mathbf{f} \in \mathbb{R}^{N_S}$, with $f_i \in [0, 1]$, where the highest value, $i^* = \arg \max_i f_i$, determines the segment of a pixel. Ideally, all pixels belonging to the same segmented area will have high responses at their i^* -th segment, to confidently distinguish them from unrelated pixels.

An indirect application of segmentation is the pre-filtering of tracked objects from the unrelated background. Approaches relying on pre-segmentation may use

a mask of the tracked objects, such as hands [80, 91] or robotic arms [85], to remove unrelated data from the image, or to create a ROI bounding box [70] centred at the tracked object. These approaches apply tracking on the filtered image data, using a dedicated distance metric that is unrelated to the segmentation itself.

A distance metric that is applied on pre-segmented images does not have to handle visual distractions. But a data-driven distance metric further down the tracking pipeline will not be able to leverage low-level feature information about the segmented object, and it will not be able to resolve ambiguities within the masked area.

The pre-segmentation approach can be extended to segment parts within the tracked object. Applications of this approach range from the clustering of relative 3D [71, 6] or 2D offsets [88], to establishing direct point associations for model-fitting approaches [73, 74, 42]. The within-object part segmentation is especially useful for articulated objects to resolve ambiguities about shape similar parts, like fingers or repeating visuals of robot links, and associate regions of the image with individual parts of the tracked object.

Keypoints

The keypoint localisation task is concerned with recovering the 2D coordinates of distinguishable points in an image. In contrast to a regression task where coordinates are directly predicted as a real-valued coordinate vector, keypoints can also be represented as a heatmap. A keypoint heatmap H_k , which is defined for every keypoint ID k , represents how likely a pixel coordinate matches the keypoint coordinate. This score $s \in [0, 1]$ is therefore an indication about the distance of a pixel to the keypoint, with 1 being an exact match. The advantage of heatmaps over regression is that a heatmap can represent multimodal distributions of keypoints and does not need to regress to a single coordinate without any measure of confidence.

Per pixel, these N_k heatmaps result in a feature vector $\mathbf{f} \in \mathbb{R}^{N_K}$ that, in contrast to the segmentation task, is unique over the entire image. Segmentation can discriminate between regions of an image, but does not discriminate between pixels within this region which leaves ambiguity. Keypoints on the other hand provide a direct association between point coordinates in the observed image and the estimated model.

Heatmaps have been widely applied to hand and human joint position detection. Early use of heatmaps [80] extracted heatmaps from depth images and used the same depth to back-project the keypoint into the observation frame. A recursive approach to refine these heatmaps was presented for single-person 2D skeleton

estimation [83] on colour images. This approach was extended by an additional relation between keypoints [13] to enable multi-person 2D skeleton detection. A parallel detection of heatmap keypoints in stereo images [72] finally enables the back-projection of 2D keypoints to 3D and estimate the articulation of real-scale kinematic structures.

In combination with generative approaches, finger tips as keypoints [79, 81] can provide a discriminative feature to resolve otherwise ambiguous relations.

Dense Features

Keypoints provide unique point features for data association but are usually sparsely distributed. Dense features aim at providing unique discriminative features for every pixel in an observation. This combines the density of segments with the unique discrimination of keypoints. Dense features are represented by a real-valued feature vector $\mathbf{f} \in \mathbb{R}^{N_F}$ per pixel. In contrast to segments or heatmaps, this feature vector does not encode the identity of a pixel. Conceptually, a feature vector only relates to visual properties of a pixel and its neighbourhood and similar properties result in very similar feature vectors.

In 3D coordinate regression [7], every 2D image pixel is associated with a rigid object ID and the local 3D coordinate \mathbf{x}_l within that object's local frame ($N_F = 4$). The back-projection of the pixel's 2D coordinate x , together with the depth image, provides the corresponding 3D coordinate \mathbf{x}_o in the observation frame. The set of corresponding points in the local and observation frame ($\{(\mathbf{x}_l, \mathbf{x}_o)_i \mid i \in \mathbb{N}\}$) provides the distance metric. This type of orthogonal Procrustes problem is solved by the optimal transformation that minimises the least-squares distance when transforming coordinates from one frame to another. The dense nature of this metric increases the probability of outliers and requires robust methods, like RANSAC [7] or PF [43], to find the optimal transformation. An extension to this approach was presented by [8] which uses uncertainty in the coordinate prediction to improve the optimisation outcome. Commonly used for house-hold sized objects, it has also been applied to car-scale transformation estimation [5], albeit with reported average coordinate errors of 0.6m.

Whereas 3D coordinates on their own can only distinguish points within an object, multi-dimensional dense features descriptors [68, 24] provide a higher degree of discrimination between points of different objects and also carry some degree of semantics.

2.3 Optimisation Approaches

The distance metric defines the space in which a distance between the observed and estimated state is established. Minimising this distance with respect to the estimated state provides the state that best matches the observation. The distance, as a function of the state, is often neither linear nor convex, and may have many local minima for large articulated states. These local minima may be results of a proper similarity of the estimated and observed state (i.e. the global minimum), but can also be a result of ambiguous correspondences caused by indiscriminative features.

The aim of an optimiser is to minimise this distance in such a way that local minima are avoided and the global minimum is found. Often there is no closed-form solution to this problem and non-linear objectives have to be linearised and an optimiser has to take iterative steps towards a minimum. This linearised search direction distinguishes the two categories of optimisers that will be discussed in the following.

2.3.1 Gradient-Based

If the distance metric is differentiable with respect to the tracked state, then gradient-based approaches can be used with the analytic gradients as the linearised search direction. These kind of solvers fall into the broad category of Newtonian-like solvers [4]. The Gauss–Newton algorithm [29] provides an extension of this approach to multi-dimensional search spaces and problems, and the Levenberg–Marquardt algorithm [44] extends this for variable-specific damping.

The linearisation of the objective results in over- or under-estimation of classic gradient decent update steps, which can be mitigated by adaptive step lengths [73]. The Gauss–Newton approach, which uses the second order derivative to tune this step length, has been applied to 6D pose [84] and articulated state [67, 81] estimation. The use of the Levenberg–Marquardt for articulated tracking [79] provides additional damping on top of the tuned Gauss–Newton step lengths.

With the local linearisation of the search direction, these solvers have to be initialised close to the state of the global optimum. When estimating small relative rigid pose changes [84], this initial state can be initialised to identity or via velocity. The selection of this initial state is especially crucial for long kinematic chains with a large nullspace. Assuming that accurate proprioception is available [67], the optimiser can be initialised directly from joint encoder readings. For short kinematic chains, such as kinematically independent fingers, the optimisation is more robust to initialisation if distinct keypoints like finger tips [81] are available.

A data-driven method can also be used to directly predict an initial state [79], which is then updated by the actual tracking objective.

2.3.2 Particle-Based

If the distance metric is not differentiable, the search direction has to be approximated. Gradients can be numerically approximated by finite differences. This requires two evaluations of the objective per state variable but allows to reuse gradient-based methods. Alternatively, the state space can be explored by evaluating the tracking objective with multiple hypotheses, also called particles. The global knowledge about the performance of this distribution of particles is then used to update the local search direction of a hypothesis. Common approaches to update these particles either resample them via their performance score, as in Particle Filter (PF) [19], or update their state by a direction derived from the performance score, such as in Particle Swarm Optimisation (PSO) [39].

In PF approaches, as used for 6 DoF object tracking [43], particle states are updated by sampling from a distribution based on the motion of the previous state (motion model). These particles are then resampled according to the likelihood that their updated state matches the observation (observation model). This resampling stage is dependent on an observation model derived from the tracking objective. The motion model in a PF relates to the inertia term that is used in some implementations of gradient-based approaches. PF approaches are widely used for applications where the objective relies on raw depth comparisons [17, 86, 48, 27] or optical flow [61]. These raw distance metrics do not provide gradients but are computational very efficient.

A PSO uses the velocity of a particle as a motion model like the PF, but additionally combines this with the local best state of the particle and the global best state of the swarm. The classic approach has been applied to hand tracking [57, 80], and with resampling [70] to avoid collapsing to a local minimum.

All particle-based approaches have in common that the search direction has to be determined from exploring the search space, while gradient-based approaches directly rely on analytic gradients derived from the objective itself. As a result, particle-based approaches inherently have to use multiple particles to find a search direction and to find the optimal solution.

2.4 Conclusion

2.4.1 Model-Based Tracking

Ideally, a function g (eq. 1.2) for mapping an observed image \mathbf{I}_{obs} to the underlying state θ_{est} would be available through data-driven methods, by simply providing corresponding mappings from real training data.

However, the related work on direct model joint states and Cartesian joint positions regression suggests that it is a complex task to learn a direct mapping from the image space to the joint space, and to capture correlations of Cartesian positions in the task space. It is evident that a kinematic prior, either as direct kinematic model or as indirect dimensionality reduction, provides ways to reduce this complexity by mapping between the task and joint space and to reduce redundant dimensions.

A regression task to a real-valued vector of physical dimensions does not provide additional error bounds or context to interpret the reliability of this regressed state. In particular, learning the relation between image pixels and model joint positions is highly complex, but a small error in these predicted model joint positions can result in large errors in the task space. We further note that orientations, and thus rotational joint states, are periodic projections to the task space and therefore do not define a proper Euclidean distance metric, as often used as regression loss.

In contrast to pure data-driven regression approaches, which have to learn the relation between image and the state and thus implicitly model the imaging process and kinematics, model-based approaches explicitly use known backward models through the inverse imaging and kinematics (Section 2.1.2). A tracking method should therefore make use of these models. The tracking approaches presented in this thesis are therefore all model-based.

This thesis borrows ideas and methods from related work on articulated human body and hand state estimation. Because of the variety of human body and hand shapes, these works usually have to rely on an approximation of the tracked geometric model [80, 70, 79] or they resort to a 2D [83, 13] or 3D [91] kinematic skeleton representation. With the focus on keypoints, these works have established evaluation metrics, such as the PCK (probability of correct keypoint) [89] and the PCKh (PCK relative to the head length) [2], to quantify and compare the performance of keypoint detection within a 2D (image frame) or 3D (camera frame) threshold. As such, these metrics are only partially useful to quantify the task space accuracy for manipulation tasks, as motivated by this thesis.

Chapter 4 takes inspiration from work on model joint position regression and proposes an approach to initialise model-based tracking by a predicted distribution of model states.

2.4.2 Objective

Data Association

The tracking objective, that defines the distance between the observed and estimated state, takes a crucial part in model-based tracking. While simple comparisons between observed and synthesised images are obvious choices, they are not differentiable and require more inefficient optimisation approaches. Objectives that rely on the association between points in the observed and estimated space on the other hand are differentiable but ambiguous.

The less discriminative and the more ambiguous these points are, the more has the data association to rely on their position in the distance metric space. This has the inherent issue that the association and the optimisation are cyclically related which results in many local minima in the distance metric space. Especially for kinematic chains with large nullspaces this results in many local minima and becomes intractable without prior knowledge about the initial state.

Related work on articulated robot manipulator tracking that makes use of indirect associations by the closest distance between data points, such as DART [66] and SimTrack [61], avoid these local minima by modelling all observed objects and using proprioception to gain knowledge about the internal robot state.

This thesis builds upon these methods but focuses on robustness to unmodelled distractions and the applicability to scenarios with no proprioception. An explicit goal is to enable articulated tracking without prior knowledge about a robot's internal state and specific objects in the scene.

Chapter 3 demonstrates that implicit data association results in an objective function that is prone to initialisation errors and provides data-driven ways to mitigate this ambiguity with an explicit association of correspondences. DART [66] will be used here representative as a baseline for methods that require initialisation and observations without distractions. Later chapters will focus on robustness to visual distractions and improper initialisation and will not consider the constrained settings in which DART operates.

Semantic Association

Data-driven approaches can provide strong semantic information to resolve ambiguity in tracking objectives. However, such approaches rely on training data as

reference for the desired mapping from image to semantics, and all observations in the image have to be modelled. In robotic applications, modelling specific manipulanda and occluders is undesirable since a data-driven model has to be retrained for every new combination of robotic manipulator and objects.

To overcome this limitation, Chapter 3 proposes a training strategy for a segmentation-based objective that does not explicitly model manipulanda or occluders in the environment. This implicit object modelling enables tracking in the presence of visual distractions, without relying on prior knowledge about objects present in the scene.

Many works that use keypoint heatmaps are only concerned with recovering the 2D projection of a kinematic configuration. While sufficient for tasks like behaviour detection, this creates ambiguities in the back-projection to 3D, which have to be resolved by additional depth or stereo information. Even then, unmodelled occlusions prevent recovering the true 3D position of keypoints.

Chapter 4 therefore proposes a tracking objective that takes possible occluded keypoints into account, and thereby explicitly enables data association and tracking behind occlusions.

2.4.3 Optimisation

While data-driven augmented objectives reduce the amount of local minima, they often do not entirely remove them. Especially for kinematic structures with large nullspaces, a data-driven shaped objective may have less local minima but in return have those located far off in the kinematic configuration. In such settings, the initialisation of gradient-based optimisers is problematic.

This thesis explores two initialisation approaches to remove the dependency on the initial state for gradient-based optimisers. Chapter 4 proposes an initialisation from a predicted distribution of model states, to select an optimal starting state for a gradient-based optimiser. Chapter 5 borrows ideas from particle-based approaches and proposes an optimiser with multiple hypotheses that uses gradients as search directions and a resampling strategy to efficiently explore the search space with fewer hypotheses than required in classical PF or PSO implementations.

Chapter 3

Visual Articulated Tracking in the Presence of Occlusions

This chapter introduces the approach of applying discriminative information to a generative iterative model-fitting technique. The application focuses on visual articulated tracking of a robotic manipulator during manipulation tasks. In typical manipulation tasks, a robot manipulator interacts with a manipulandum and may also be occluded by other objects in the scene. Visual tracking is prone to failure in these scenarios due to the visual distraction created by a non-tracked object in the environment.

Current state-of-the-art approaches, which typically rely on model-fitting using Iterative Closest Point (ICP), fail in the presence of distracting data points and are unable to recover. Meanwhile, discriminative methods which are trained only to distinguish parts of the tracked object can also fail in these scenarios as data points from the occlusions are incorrectly classified as being from the manipulator. We instead propose to use the per-pixel data-to-model associations provided from a random forest to avoid local minima during model-fitting. By training the random forest with artificial occlusions we can achieve increased robustness to occlusion and clutter present in the scene. We do this without specific knowledge about the type or location of the manipulandum and occluders. Our approach is demonstrated by using dense depth data from an RGB-D camera to track a robotic manipulator during manipulation and in presence of occlusions.

The work presented in this chapter has been peer-reviewed and published as:

Visual Articulated Tracking in the Presence of Occlusions by Christian Rauch, Timothy Hospedales, Jamie Shotton and Maurice Fallon in 2018 IEEE International Conference on Robotics and Automation (ICRA)

<https://doi.org/10.1109/ICRA.2018.8462873>

3.1 Introduction

When estimating the state of a robot during manipulation, a common approach is to use joint sensing, forward kinematics (FK) and a complete description of the kinematic model of the robot to compute the position of the end effector. However, joint sensing can be affected by calibration inaccuracies, quantisation noise and non-linearities, or may not be available at all for underactuated and dexterous manipulators. This traditional industrial approach does not consider tactile information nor does it incorporate visual sensing, which is of course heavily used during human manipulation. Finally, it cannot track the state of the manipulated object.

We are motivated to explore jointly visual tracking of a manipulator and an object by the prior work of [57, 66]. A key challenge for visual tracking is the presence of distractor objects. These objects occlude the manipulator and add irrelevant visual information which can impair tracking.

Estimating the full and valid configuration of an articulated object directly from images is a challenging problem. In this work we propose an approach similar to [70, 79, 73, 42] to combine model-based tracking, which simplifies the kinematically plausible state estimation, with discriminative information to prevent failures due to the distracting visual information.

The core contribution of this chapter is the integration of pixel-wise predictions from a random forest into a model-fitting framework that is robust to incorrect initialisation and unmodelled occlusions, as illustrated in Figure 3.11.

3.2 Related Work

We categorise visual articulated tracking into **generative model-fitting** and **discriminative** approaches as well as **hybrid** methods which combine generative model-fitting with discriminative information. In the following we give a brief overview of the relevant literature for each approach.

3.2.1 Generative Model-Fitting

Given a model of the tracked object, generative model-fitting aims to synthesise a set of hypotheses of the model’s state and compare these hypotheses with the observed state. These methods rely on a good metric to quantify the similarity between the synthesised state and the real observation (the objective function), and an efficient method for exploring the large state space of the articulated model.

Early work by Oikonomidis et al. [57] used colour and edge cues as a similarity metric on 2D images for tracking a hand and an object in interaction. This objective was minimised using particle swarm optimisation (PSO) over the combined state space. This concept was later applied to data from depth sensors by Schmidt et al. [67] which used the signed distance function (SDF) as the similarity metric and gradient-based Gauss-Newton optimisation to minimise this objective.

Pauwels et al. [60] simplified articulated tracking as a 6D pose estimation problem given proprioceptive sensing and an initial camera pose. After articulating the manipulator according to the sensed joint positions, the manipulator is assumed rigid and fitted to the depth observation.

Generative model-fitting methods can be extended to track multiple objects in parallel and allow hypotheses rejection by applying kinematic and physical constraints [66]. However, these methods have similar properties and disadvantages as the iterative closest point (ICP) algorithm. Their similarity metric is typically dependent upon local visual features such as edges and gradients and hence can suffer from local minima, which is why tracking needs to be initialised close to the optimal solution.

3.2.2 Discriminative Tracking

Meanwhile, discriminative methods learn the visual representation of a model with respect to the true state or joint configuration. This requires an extensive amount of labelled training images which show the tracked object in many different states. In this problem domain, these states are synthesised using known articulated models *a priori*.

A popular approach for depth-based tracking of articulated objects is to use simple depth probe offset features in a random forest (RF) for segmentation and keypoint localisation. This was used for human pose estimation [71] and more recently was applied to robot manipulator configuration estimation [6]. In our work we also use this type of feature and classification method, but our approach uses the raw class probability for model-fitting instead of joint position prediction or mean-shift.

Direct regression of the full manipulator configuration has been demonstrated in [85], again using depth probe offset features. Tompson et al. [80] applied convolutional neural networks to depth data to detect the locations of hand keypoints on joints and to infer the joint configuration from inverse kinematics.

3.2.3 Hybrid Objectives

Hybrid tracking methods use a combination of generative and discriminative methods, so as to augment model-fitting with discriminative information.

The detection of fingertips was used by Tzionas et al. [81] and Taylor et al. [79] in their objective function to guide optimisation towards the optimal hand pose. In our work we propose to instead rely on a full segmentation of the image to prevent cases where these specific keypoints are occluded, e.g. when reaching behind an object.

Our work is related to Sridhar et al. [73] and Krejov et al. [42], where a RF segmentation of depth images was used to support model-fitting of a human hand. Compared to the Gaussian volumetric approximation in [73] we use the full pixel-wise data-to-model association and a more realistic mesh model of the robot. Compared to [42], our approach extends to cases when the tracked manipulator is occluded by unmodelled objects.

Finally, approaches which simultaneously track hands and objects typically rely on knowledge specific to the object of interest such as colour (Sridhar et al. [74]) or shape (Schmidt et al. [66]). In our proposed tracking approach we aim to track the manipulator generally without knowledge of the object of interest, relying only on the 3D model of the manipulator. Specifically we do not require a volumetric representation of the object nor any specific properties of the object to enable manipulator tracking near and behind distractions.

3.3 Proposed Method

3.3.1 The Signed Distance Function

Model-fitting approaches rely on the minimisation of an objective function $e(\cdot)$ which contains a term for the discrepancy between the estimated and the observed state, as well as other criteria which impose physical or kinematic constraints. For depth-based model-fitting, the truncated signed distance function (SDF) has been commonly used as the metric when minimising data-to-model discrepancy.

The SDF is the 3D distance transform, $\text{SDF}(\mathbf{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$, in a 3D spatial grid with binary voxel states and provides the Euclidean distance of a given data point \mathbf{x} to its closest active voxel [23]. For closed polygon model meshes, a voxel in this 3D grid is active if it is occupied by the model mesh and inactive otherwise. The SDF is positive outside the model, or when the query data point occupies an inactive voxel, and negative inside the model, or when the query data point occupies an active voxel, respectively.

For articulated models, the objective can be represented by a single articulated SDF that is defined by the model configuration θ :

$$e(\theta, \mathbf{I}) = \sum_{\mathbf{x} \in \mathbf{I}} \text{SDF}(\theta, \mathbf{x})^2 \quad , \quad (3.1)$$

or it can be piecewise locally defined as individual rigid SDF_{*i*}:

$$e(\theta, \mathbf{I}) = \sum_{\mathbf{x} \in \mathbf{I}} \text{SDF}_{i^*}(T_{i^*}^{-1}(\theta)\mathbf{x})^2 \quad , \quad (3.2)$$

for all M tracked model parts $P = \{i \in \mathbb{N} \mid 0 \leq i < M\}$, with $T_i(\theta)$ as the transformation from the observation frame to the local link frame i . The choice of the individual transformation T_i for every data point \mathbf{x} depends on the optimal association i^* of the point to the individual SDF_{*i*} (Section 3.3.2). In this model-fitting objective the signed distance is used as a squared residual in a least-squares problem formulation.

In both cases, the 2D depth readings $z \in \mathbf{I}$ are back-projected to the 3D points \mathbf{x} in the observation frame via the camera projection matrix K , $h_{obs}(\mathbf{I}) = \{\mathbf{x} \mid \mathbf{x} = K^{-1}zx \forall z \in \mathbf{I}\}$, which is expressed by the shortened notation $\mathbf{x} \in \mathbf{I}$. The SDF is truncated in a way that a data point \mathbf{x} is not considered for the objective if it exceeds a given distance threshold.

In the latter case, the 3D points are further transformed from the observation frame to the individual SDF_{*i*} frames by FK on the estimated state θ , $h_{est}(\theta) = \{T_i(\theta) \mid i \in P\}$. The locally rigid SDF_{*i*} are defined by the model visuals. After the optimal association of the data point \mathbf{x} to i^* , the corresponding transformation T_{i^*} maps the estimation and observation into the same local coordinate frames to establish a distance metric between the estimated state and the observed data.

3.3.2 Data Association

Without knowledge of the true identity of a data point, prior approaches, such as [67], have assigned \mathbf{x} to the closest SDF_{*i**} using

$$i^* = \arg \min_{i \in P} |\text{SDF}_i(T_i^{-1}(\theta)\mathbf{x})| \quad . \quad (3.3)$$

The optimal pose $\theta^* \in \mathbb{R}^{6+N}$ of an articulated object is then the θ which minimises the local data-to-model error when transforming each \mathbf{x} to the SDF_{*i**} frames, according to the kinematic chain articulated by the N joints and the 6D pose, and the optimal associations i^* per \mathbf{x} .

So as to minimise this objective over the huge state space of articulated models, it is common to use iterative gradient-based approaches such as the Gauss-Newton algorithm initialised close to the true solution. The gradient of the SDF with respect to θ is based on a temporary association between the data and model parts which is re-evaluated with each iteration. This data association criteria (minimal distance) is the same as the objective function which reinforces incorrect data associations and can lead to irreversible tracking failure.

We propose to instead replace the implicit data association (eq. 3.3) with an explicit association using a discriminative pixel-wise classifier to provide a class probability distribution $p(c | \mathbf{f})$ per class c , given a feature vector \mathbf{f} , computed per pixel in the depth image \mathbf{I} . A data point is then explicitly assigned to the part i with the highest class probability

$$i^* = \arg \max_{i \in P} p(c = i | \mathbf{f}) . \quad (3.4)$$

In what follows we refer to implicit data association using the shortest SDF distance (eq. 3.3) as DA-SDF (our baseline approach), with the objective:

$$e(\theta, \mathbf{I}) = \sum_{\mathbf{x} \in \mathbf{I}} \left[\text{SDF}_{i^*}(T_{i^*}^{-1}(\theta)\mathbf{x}) \Big|_{i^*=\arg \min_{i \in P} |\text{SDF}_i(T_i^{-1}(\theta)\mathbf{x})|} \right]^2 , \quad (3.5)$$

and refer to our proposed approach which uses explicit data association from pixel-wise classifications (eq. 3.4) as DA-RF, with the objective:

$$e(\theta, \mathbf{I}) = \sum_{\mathbf{x} \in \mathbf{I}} \left[\text{SDF}_{i^*}(T_{i^*}^{-1}(\theta)\mathbf{x}) \Big|_{i^*=\arg \max_{i \in P} p(c=i|\mathbf{f})} \right]^2 . \quad (3.6)$$

After carrying out data association, DA-SDF and DA-RF rely on the same Gauss-Newton optimisation shown in Figure 3.1, to minimise the data-to-model distance. After initialising the optimiser once at the reported robot state for the very first image of a sequence, the algorithm iteratively converges to a minimum (eq. 2.24). The image frames and their data association are updated continuously at 30Hz and the optimisation on an image frame is initialised from the solution of the optimisation on the previous image frame.

The objective is only differentiable within the association of a single image. In an image sequence, these associations change between consecutive images which can cause oscillation and jumps. In practice this is mitigated by damping. Compared to [73], we evaluate only one hypothesis at a time but we use the gradients for all pixel-wise associations for the optimisation.

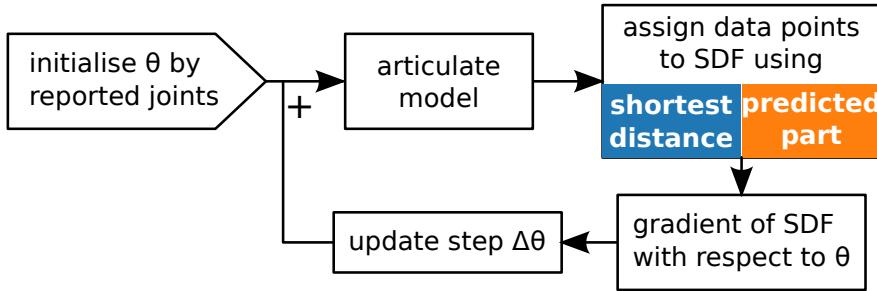


Figure 3.1: Flow chart of iterative pose optimisation using either DA-SDF (shortest distance) or DA-RF (predicted part).

3.3.3 Generating Training Data for Pixel Classification

To obtain a sufficiently large set of labelled training data, we synthesise depth images using the Z-buffer of an OpenGL renderer. Each part of the robot is associated to a dedicated class and its geometric appearance is represented by a mesh. We do not add any sensor-specific noise. Importantly, to train an occlusion robust segmentation we do not add a specific occlusion object class but instead sample pixel-wise occlusions during the feature generation phase.

To generate the training data, the robot model is articulated using a set of joint configurations which provide good coverage of the expected range of manipulation poses. We initially sample 20000 target palm poses with a position within the camera frustum in a distance range of [0.5, 1.5]m, and axes of the rotation matrix such that the palm-face is in the direction of typical grasping. The arm joint configurations for these palm poses are obtained via inverse kinematics (IK). Since many of these initial target poses are kinematically infeasible, for example if they are too far away or violate collision or joint limit constraints, we only accept valid arm joint configurations that result in palm poses within 1mm and 1deg of the initial target palm pose after convergence of the IK optimiser. This results in 4477 valid palm pose and arm joint configurations that are further combined with four discrete finger grasping states between fully opened and closed, resulting in a total of 17908 labelled training images (Figure 3.2).

3.3.4 Training

Pixel-Wise Segmentation of Robot Parts

To train the classification random forest (RF) for the task of pixel-wise labelling of depth images we use the depth probe offset features presented in [71]:

$$d_{\Theta}(\mathbf{I}, x) = d_{\mathbf{I}} \left(x + \frac{\mathbf{u} \cdot f}{d_{\mathbf{I}}(x)} \right) - d_{\mathbf{I}} \left(x + \frac{\mathbf{v} \cdot f}{d_{\mathbf{I}}(x)} \right) . \quad (3.7)$$

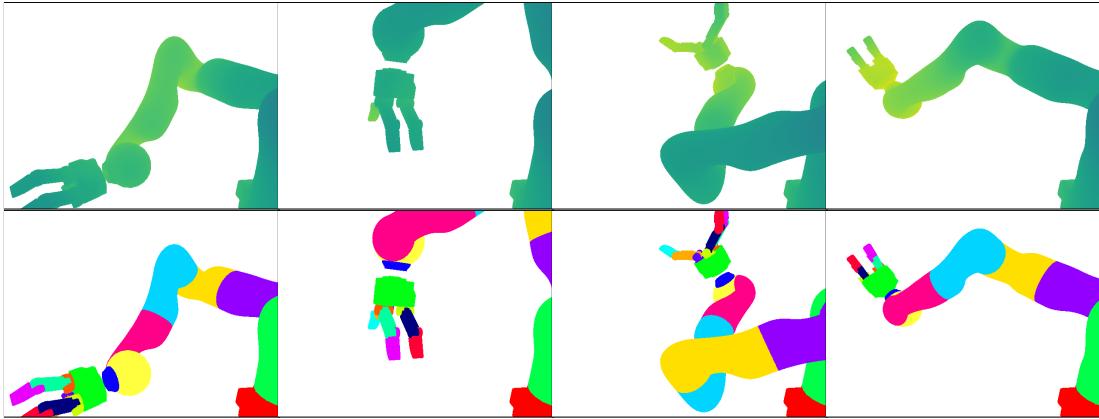


Figure 3.2: Examples of colour-mapped depth (top) and label (bottom) images for valid sampled joint configurations of the KUKA arm with Schunk SDH2 hand. Through the depth buffer, self occlusions, such as the partially occluded yellow ball link in columns two to four, are already part of the training set.

The function $d_{\mathbf{I}}(\cdot)$ gives the depth value at the queried pixel location x of a depth image \mathbf{I} . The relative offsets $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2$ are randomly sampled on a plane in world space and stored as individual feature configurations $\Theta = (\mathbf{u}, \mathbf{v})$. After projecting the offsets onto the image plane using the focal length f and depth at queried pixel x , the feature response $d_{\Theta}(\mathbf{I}, x)$ is computed as the difference of the depth at the two pixel-offset locations. Sampling offsets in world space makes the feature responses independent from the depth.

We apply the same procedure as in [71] to train 30 randomised decision trees to maximum depth. While deeper trees enable more complex decision functions, larger forests provide a smoother prediction of the pixel’s class. In independent experiments we observed an asymptotic convergence of the test accuracy towards the full depth of these trees and thus observed no overfitting effects.

Since the training data is processed as a single batch which is processed at once, the number of training samples and features is a trade-off between accuracy and memory requirements. We empirically chose 1000 random feature configurations Θ before the training. During training, a split node is then optimised over a randomised subset of 32 ($\approx \sqrt{1000}$) of these feature configurations. This smaller randomised subset reduces the correlation between trees of the forest which in turn increases the smoothness of the prediction.

For simplicity, we train and test without a background scene which we set to the constant value of 3m. It has been demonstrated [85] that an additional prepended RF stage can be used for foreground/background segmentation.

A qualitative example for the segmentation of the palm and finger links is shown in Figure 3.9d.

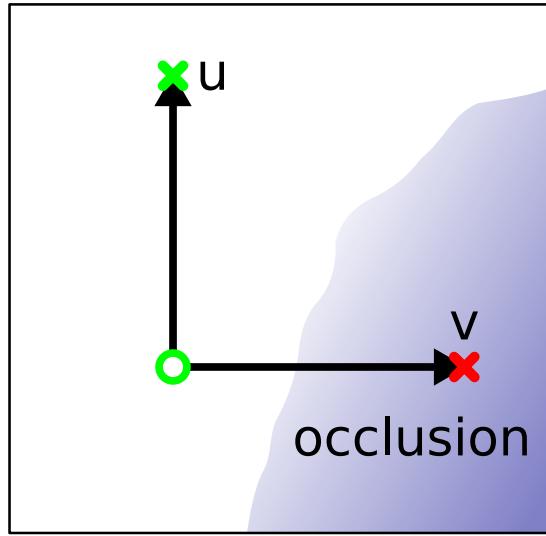


Figure 3.3: Probing near the borders of occlusions (blue area) with a feature configuration $\Theta = (\mathbf{u}, \mathbf{v})$. Offset \mathbf{v} is probing an occluded pixel. During training, a surrogate depth value will be simulated at this location in the image which generates a different feature response.

Occlusion Sampling

If a RF is only trained on the parts of a robot, this usually results in an over-confident classification of unseen data, such as occlusions or objects, as parts of the robot. Doing this would distract the data association of the model-fitting stage by assigning model parts with irrelevant data and drawing the SDF optimisation away from the true configuration. We address this problem by training the random forest with generic and randomised occlusions so as to reduce the confidence of predictions in the area of occlusions. The effect of this is that the RF becomes less confident when classifying occlusions as robot parts. These less confident classifications can then be rejected using a threshold on the class probability, with only the more confident data associations then used for model-fitting.

At training time, this confidence can be shaped by randomly sampling occlusion pixels when generating the feature responses. Each time a probe offset (eq. 3.7) \mathbf{u} or \mathbf{v} accesses a pixel of the original synthetic training image, it is marked as accessing an occluding pixel with a certain probability (Figure 3.3).

We temporarily replace the depth value at the probe offset that has been marked as occluded by a simulated occlusion depth value. The reference pixel x keeps its label but receives a different response from the same feature configuration. In this manner, the RF is forced to learn a certain variance of the feature response resulting from nearby occlusions.

The simulated occlusion depth value is drawn from a half-normal distribution whose mean is placed in front of the occluded part (Figure 3.4). The half-normal

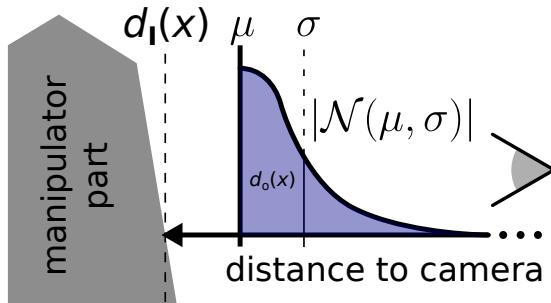


Figure 3.4: Adding robustness to occlusion: During training, the original depth value $d_I(x)$ of a robot part (grey) is replaced by a depth value $d_o(x)$ drawn from the shifted probability density function of a half-normal distribution (blue).

distribution has been chosen because it has no support for points behind the farthest occluded distance and allows the occluder to have a varying shape.

A depth value $d_I(x)$ of an occluded probe is replaced in eq. 3.7 by

$$d_o(x) \sim |\mathcal{N}(d_I(x) - \mu, \sigma)| \quad (3.8)$$

with

$$\begin{aligned} \mu &\sim \mathcal{U}(0, \mu_{max}) \\ \sigma &\sim \mathcal{U}(\sigma_{min}, \sigma_{max}) \end{aligned}$$

where μ is sampled per image and σ is sampled per probe. The hyperparameters μ_{max} , σ_{min} and σ_{max} are constant during training.

We sample these simulated occlusions with a probability of 0.15 from a half-normal distribution with $\mu = \mathcal{U}(0, 0.1)\text{m}$ and $\sigma = \mathcal{U}(0.05, 0.15)\text{m}$. These parameters reflect the expected object distances and dimensions. We only use pixels that are classified with a class probability of more than 0.55 for tracking and refer to the RF training with occlusions as DA-RF-OCCL. We will refer to the class probability, which is visualised in Figures 3.14 and 3.16, also as confidence.

3.4 Evaluation

3.4.1 Platform

We tested the proposed approach using a KUKA LWR4 7 DoF arm with a Schunk SDH2 7 DoF hand (Figure 3.5). The hand contains 3 fingers with 2 joints each and an additional joint that allows two of the fingers to rotate around their longitudinal axis. Depth images were collected using an Asus Xtion PRO LIVE structured



Figure 3.5: KUKA LWR4 (7 DoF) with Schunk SDH2 (7 DoF) mounted on a table with AprilTags for camera pose estimation.

light sensor. Since the depth sensor is not part of the kinematic chain, its pose in the robot frame is estimated using an AprilTag [58] mounted on the robot’s base.

During our experiments, we only track a subset of the robot’s links which contains the hand and the last 4 links of the arm. The tracked state therefore consists of the 6D pose and 10 joints. The camera pose is chosen such that the arm enters the scene from the right side of the image and the camera pose is held static during a sequence.

3.4.2 Data Collection

We collected three different sequences with different degrees of visual distractions. For all sequences, we manually define end-effector poses as task space waypoints and use MoveIt to obtain their corresponding joint space configuration and to interpolate between these joint space points. The selection and validation of these waypoints have to be done manually to prevent collisions with the environment,

such as the table and the object, and to prevent excessive movements of the arm close to joint limits. The finger states are manually chosen to firmly grasp an object.

Once a complete joint trajectory, including arm and finger states, has been found and stored, it is replayed on the robot without further interaction through MoveIt and recorded via the Asus Xtion.

Depth readings from the background are removed by manually defining planes in the workspace of the robot and removing all depth readings behind these planes as viewed from the camera. These planes are located above the table, in front of the wall and on both sides. This effectively filters all depth readings but the robot and the object in the workspace.

3.4.3 Tracking Error Metrics

The tracked state for the baseline algorithm (DA-SDF) and variants of our approach (DA-RF with and without occlusion sampling) is compared to the robot state as reported by joint position sensing. The reference pose of a frame is obtained by forward kinematics using the reported joint positions.

We define the pose tracking error T_{err} as the transformation that needs to be applied to the estimated pose T_{est} to obtain the reference pose T_{ref} in the camera frame. Decomposing $T_{err} = T_{est}^{-1}T_{ref}$ into its translation part t_{err} and rotation part R_{err} , the magnitude of the position error p_{err} and orientation error o_{err} are defined as

$$p_{err} = \|t_{err}\| \quad (3.9)$$

$$o_{err} = \left| \cos^{-1} \left(\frac{\text{Trace}(R_{err}) - 1}{2} \right) \right|. \quad (3.10)$$

3.4.4 Experiment 1: Discriminative Tracking

In this experiment, we articulated the palm and the fingers of the manipulator without any external occlusions. This is to show the general ability of our approach to track palm pose and finger motions.

Palm Pose Tracking

Figure 3.6 shows that the selected scenario, with correct initialisation and observations of manipulator parts only, provides the optimal conditions for the implicit data association (DA-SDF). Discriminative methods (DA-RF, DA-RF-OCCL) are affected by misclassified pixels. These idealised conditions are however unrealistic

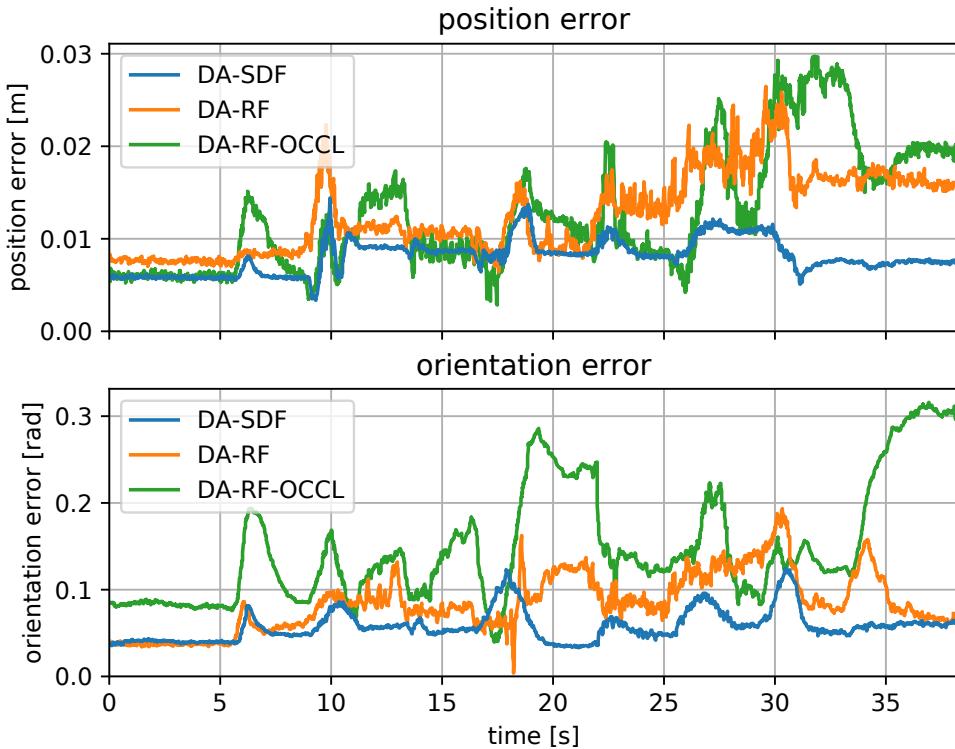


Figure 3.6: Experiment 1: Palm pose tracking error. Average: DA-SDF: $0.8 \pm 0.2\text{cm}$, $0.06 \pm 0.02\text{rad}$; DA-RF: $1.3 \pm 0.4\text{cm}$, $0.09 \pm 0.03\text{rad}$; DA-RF-OCCL: $1.3 \pm 0.6\text{cm}$, $0.15 \pm 0.07\text{rad}$.

in scenarios like grasping, which require more robust visual tracking approaches that can deal with distractions.

Convergence of Optimisation

To evaluate the convergence properties of Gauss-Newton optimisation using both data association approaches, we selected a static palm pose with all fingers visible (Figure 3.7). We initialised the optimisation with a perturbation applied to the true palm pose. 100 of these pose perturbations were randomly sampled within the range of $\pm 0.1\text{m}$ per coordinate and $\pm \frac{\pi}{2}\text{rad}$ per Euler angle.

The estimated palm pose error after converging with 500 iterations is reported in Figure 3.8 with cumulative histograms. Using DA-SDF as objective results in many local minima, which are located far away from the original reference pose. Only 25% of DA-SDF trials converge to palm poses with errors less than 1.5cm and 0.3rad. The DA-RF objective has less local minima and 75% of trials converge to poses within the same error bounds. The rejection of manipulator pixels in DA-RF-OCCL removes gradients from the optimisation, that would otherwise

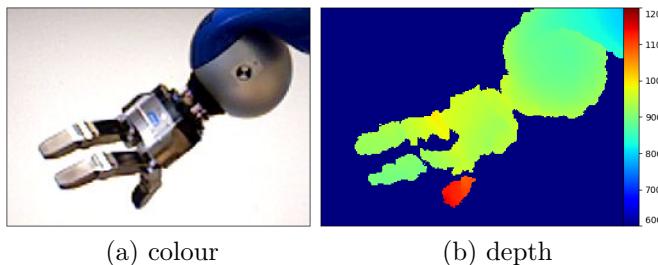


Figure 3.7: Experiment 1: A static pose showing all fingers. This posture is used to analyse the convergence of the gradient-based optimiser.

prevent local minima in the original objective (DA-SDF). This is particularly apparent through the high orientation errors of the local minima. In this scenario without distractions, the proposed DA-RF-OCCL is therefore a trade-off between the performance of the direct association in DA-RF and the robustness against unmodelled distractions by reducing the confidence in these direct associations.

The example failure case in Figure 3.9 demonstrates the need to explicitly associate data points to model parts (DA-RF) to avoid local minima caused by implicit data association (DA-SDF).

3.4.5 Experiment 2: Grasping

A more realistic scenario is presented by the grasping task shown in Figure 3.10. In this scenario, the manipulator: (1) approaches and grasps an object, (2) lifts and moves the object, (3) places it back on the table and (4) moves away from the object.

The baseline approach (DA-SDF) wrongly attaches the mis-tracked manipulator to the data corresponding to the object after the initial grasp (Figure 3.11c). This causes the palm pose estimate to be biased during subsequent tracking (Figure 3.12, $t > 4\text{s}$). When retracting the hand, the tracked manipulator remains associated to the object and tracking cannot recover ($t > 40\text{s}$).

By comparison, our approach (DA-RF) tracks the palm pose accurately throughout as parts of the manipulator are correctly classified during the grasping and therefore provides the correct data-to-model association.

3.4.6 Experiment 3: Tracking in the Presence of Occlusions

We evaluate our main contribution in an experiment where the manipulator is occluded by an object in the near table. This is different from the previous

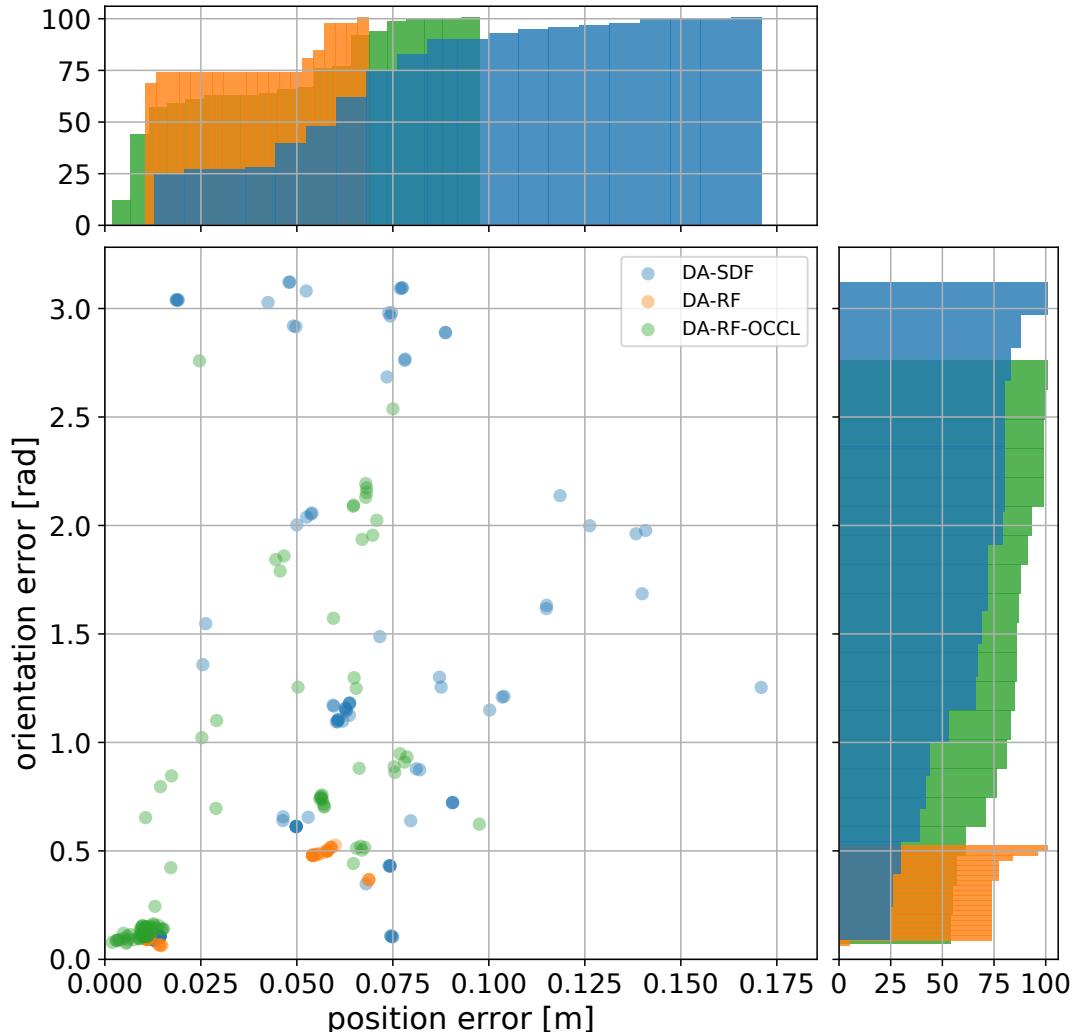


Figure 3.8: Experiment 1: Palm pose error after converging from a perturbed initial pose. DA-SDF has many local minima which causes the majority of trials to converge more than 1.5cm and 0.3rad away from the true reference pose, while the majority of DA-RF trials converge within these bounds.

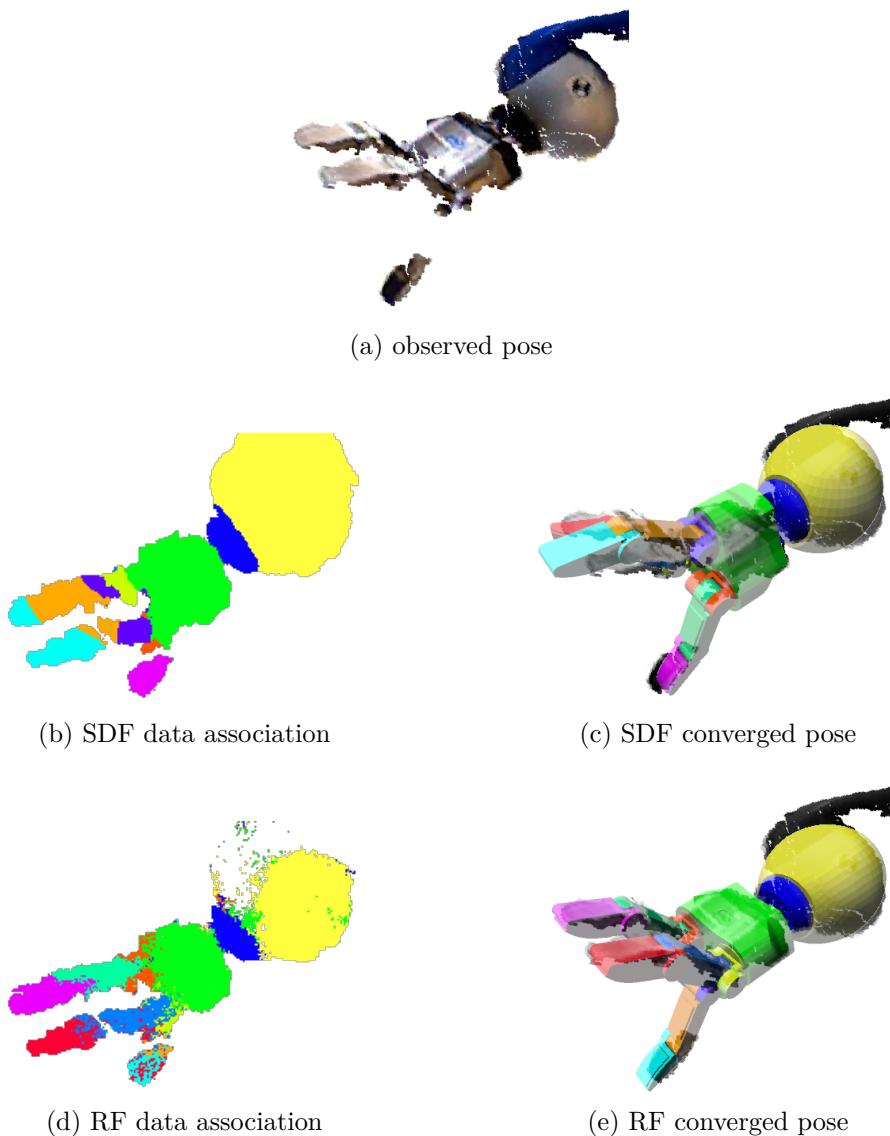


Figure 3.9: Experiment 1: Example poses and data association after convergence. (b) DA-SDF iteratively assigns the thumb (cyan) to both fingers, resulting in (c) convergence to a local minima. (d) The segmentation by the RF correctly distinguishes the fingers and the thumb and allows the optimisation to converge to the correct pose (e).

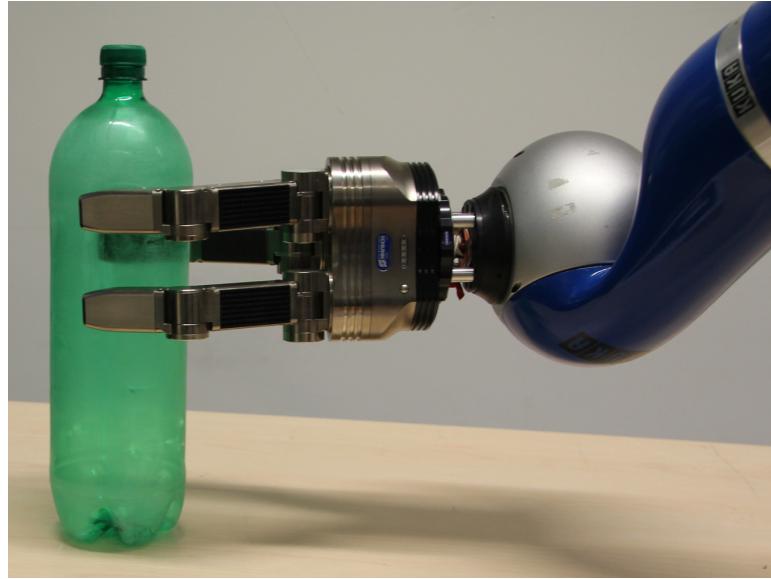


Figure 3.10: Experiment 2: Grasping and manipulating a bottle.

experiment as the occluding object is segmented into manipulator parts and the actual part is hidden and must not be associated to the occluder.

In this sequence, we initialise the robot in a state where none of its parts are occluded. The manipulator is then moved behind a green bottle such that it occludes the palm and fingers during movement so as to investigate the ability to fit the model to partial observations. The manipulator later moves back to a non-occluded configuration to demonstrate the ability to recover from tracking errors. Characteristic states of this sequence are shown in Figure 3.13.

Improved Data Association Through Occlusion Training

We wish to be implicitly robust to unknown objects and do not want to rely on object tracking. To overcome the distraction of the occluding green bottle, we train DA-RF-OCCL by adding a random sampling of occluding pixels as described in Section 3.3.4.

This random occlusion sampling reduces the probability of incorrect assignments of bottle pixels to palm parts (Figures 3.14a and 3.14b, left), while the visible finger tip keeps most of its confidence (Figures 3.14a and 3.14b, right). This improves model-fitting in that there are fewer gradients from bottle pixels that move the actual palm away from its original position.

Since we can treat the acceptance and rejection of data associations via thresholds as a binary classification problem, we can evaluate both RF data association variants (DA-RF, DA-RF-OCCL) given the true segmentation of the object. The Precision-Recall curve in Figure 3.15 shows that our proposed training

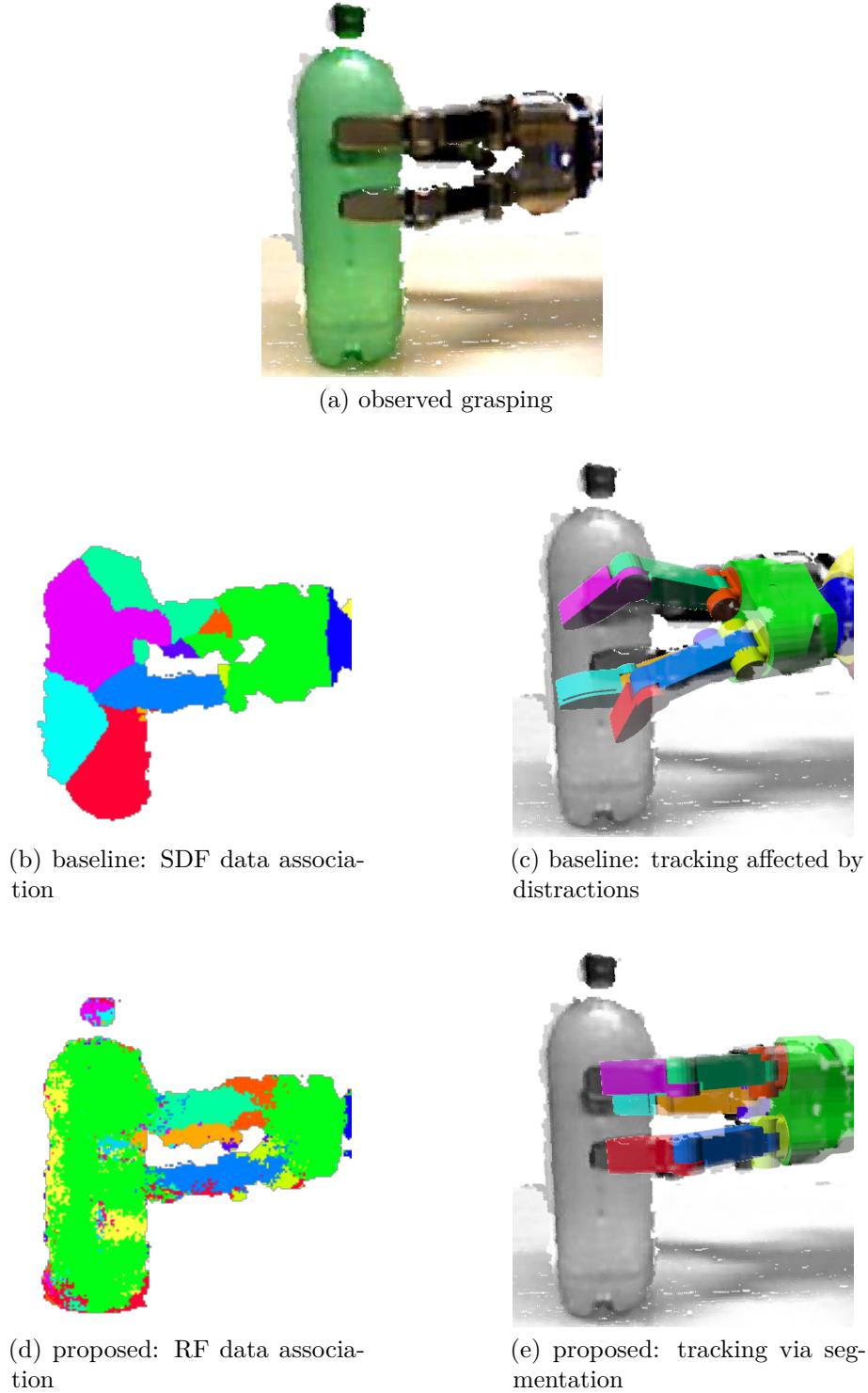


Figure 3.11: Data association with a generative and discriminative model-fitting approach. (a) Observation of a manipulator grasping a bottle. (b) The baseline generative approach assigns data points of the manipulated object to finger parts, which (c) results in a shift of the estimated pose towards the bottle. (d) Our discriminative approach correctly classifies palm and proximal finger pixels, and (e) keeps the palm pose estimate stabilised.

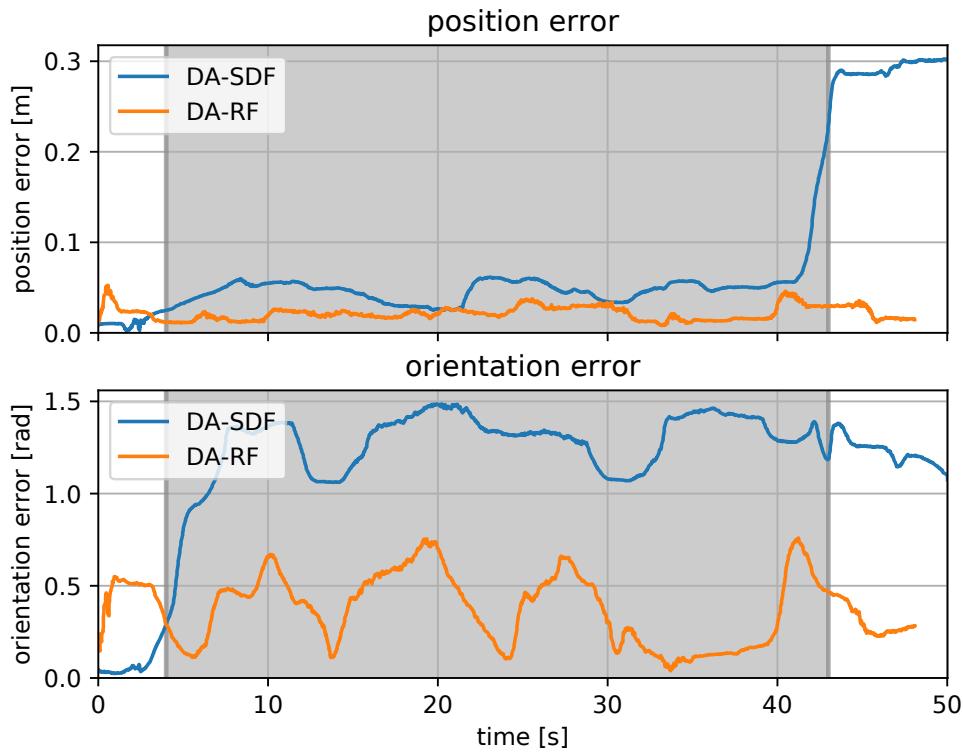


Figure 3.12: Experiment 2: Palm pose estimation when grasping and moving the bottle (grey shaded phase). DA-SDF tracker is biased as the manipulated object draws the palm away from its true position. Average tracking error: DA-SDF: $8.3 \pm 9\text{cm}$, $1.15 \pm 0.39\text{rad}$, DA-RF: $1.5 \pm 0.6\text{cm}$, $0.2 \pm 0.15\text{rad}$.

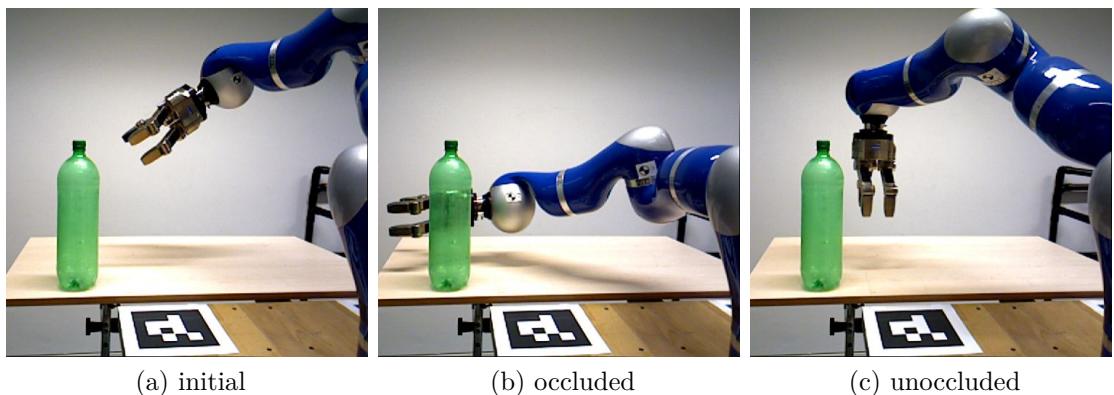


Figure 3.13: Sample images from Experiment 3. (a) Tracking is initialised at an unoccluded configuration, (b) the hand moves behind the green bottle and (c) returns to an unoccluded configuration.

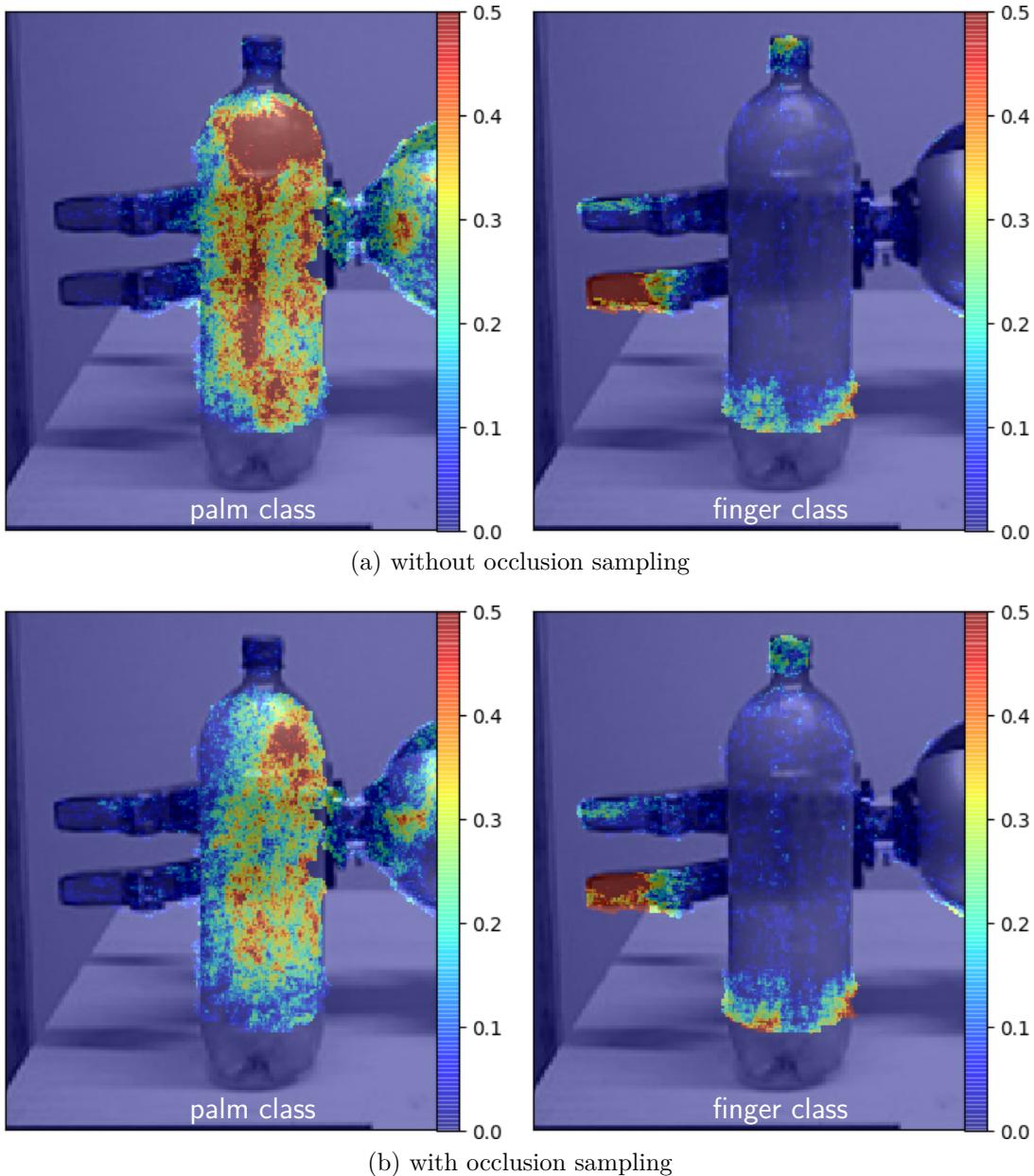


Figure 3.14: Class probabilities for the palm and a finger. (a) Training without occlusions results in a high classification confidence for wrongly assigning the occluded palm to the bottle (left) and assigning the finger tip to the robot’s actual finger tip (right). (b) After introducing random occlusions during training, we can reduce the confidence of assigning bottle pixels to the robot palm (left) but keep the high confidence of the finger tip classification (right). The region around the bottle has been manually selected for visualising the effect of occlusion sampling. See Figure 3.15 for a quantitative comparison on the complete sequence.

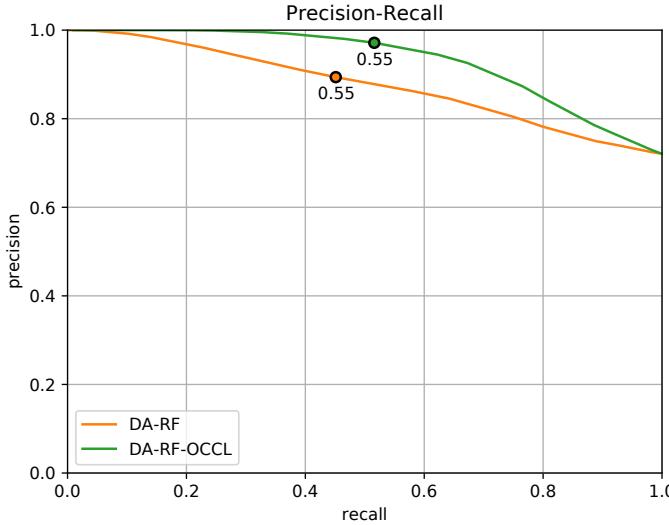


Figure 3.15: Precision-Recall curve for data association seen as binary classification, averaged over all images of Experiment 3. DA-RF: training without occlusions (AUC: 0.86), DA-RF-OCCL: training with occlusions (AUC: 0.89). The circled locations mark the selected rejection threshold (0.55) for tracking.

approach (DA-RF-OCCL) improves classification performance independently from the selected rejection threshold.

We found that the maximum depth probe distance is an important parameter that affects the model-fitting performance in presence of occlusions. For small manipulator parts like fingers, a short probe distance is important. This is visualised in Figure 3.16a where a large maximum probe distance ($\{\|\mathbf{u}\|, \|\mathbf{v}\|\} \leq 15\text{cm}$) results in similar finger tip probabilities for pixels on the bottle corner and the actual finger. By enforcing learning only from local information ($\{\|\mathbf{u}\|, \|\mathbf{v}\|\} \leq 5\text{cm}$) we can shift probability from the bottle to the actual finger tip (Figure 3.16b).

Baseline: Tracking with Known Object Pixels

As a baseline, we first use simple colour segmentation to remove the green bottle leaving only the pixels corresponding to the arm and hand (albeit with missing pixels). The idea being that this example can provide a baseline for what could be achieved when trying to be robust to a more complex unknown distractor object. The tracking error is reported for the DA-SDF, DA-RF and DA-RF-OCCL in Figure 3.17.

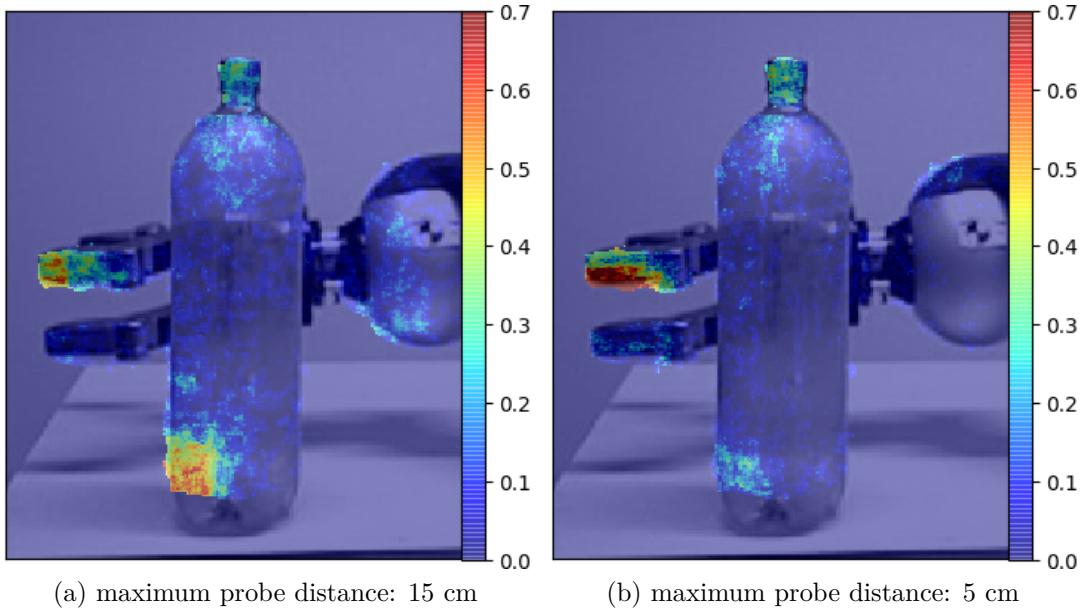


Figure 3.16: Class probability of the finger tip for different offset distances. By reducing the probe offset distance and providing more local information, we can move probability from the bottle corner (a) to the true finger tip location (b).

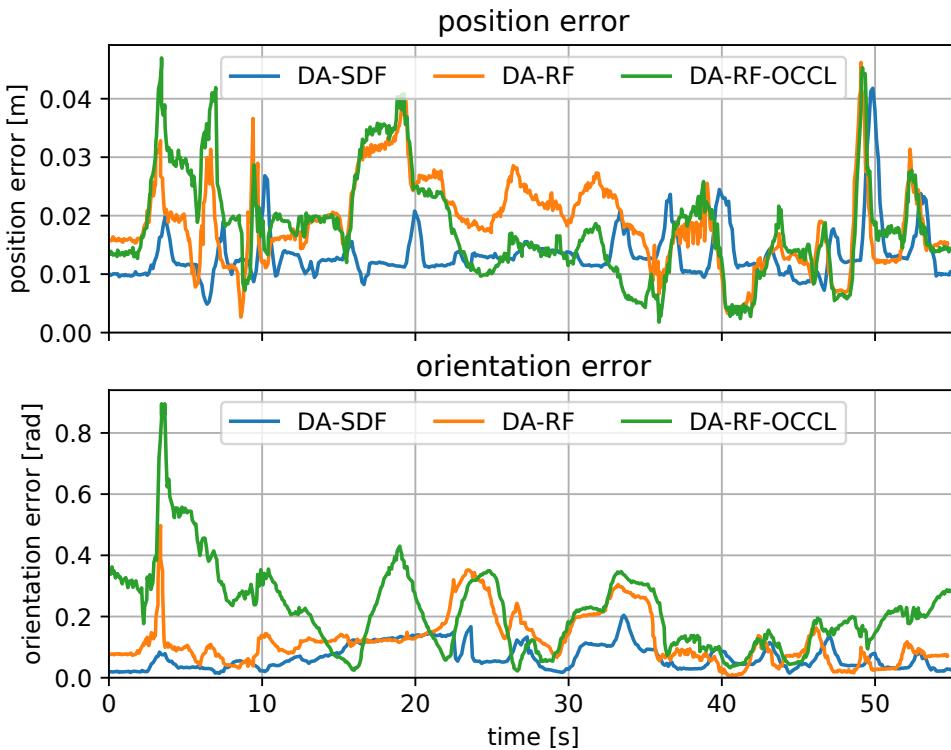


Figure 3.17: Experiment 3 *without occlusions*: Palm pose tracking error after object removal. Average: DA-SDF: 1.3 ± 0.4 cm, 0.07 ± 0.04 rad; DA-RF: 1.9 ± 0.7 cm, 0.12 ± 0.08 rad; DA-RF-OCCL: 1.8 ± 0.8 cm, 0.22 ± 0.14 rad.

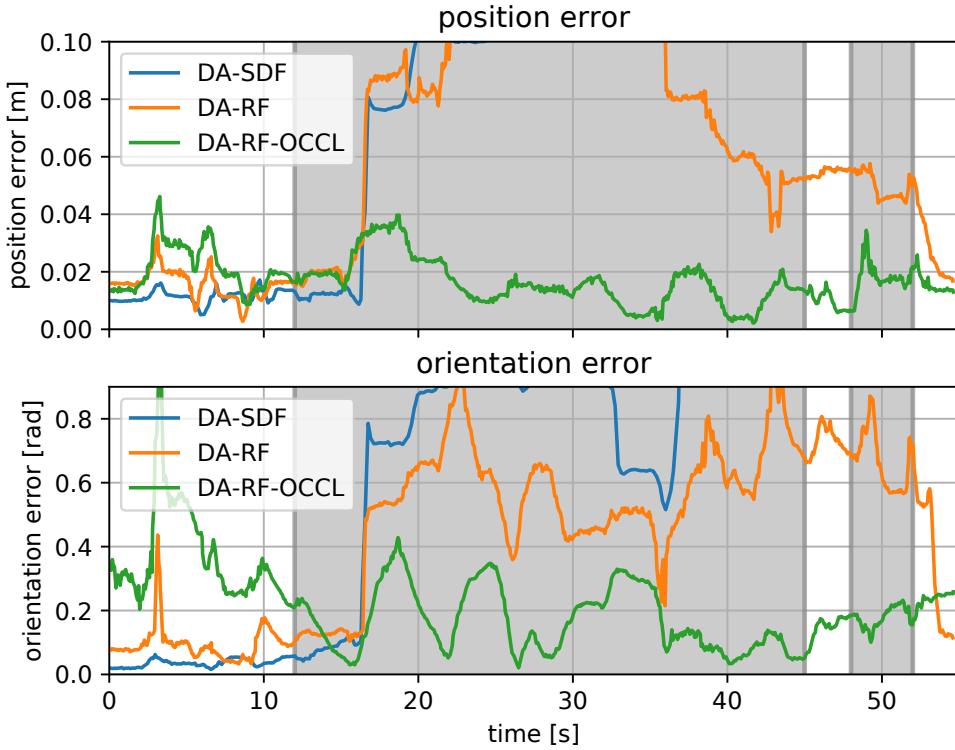


Figure 3.18: Experiment 3 *with occlusion*: Palm pose tracking error during movement close to occlusion (grey shaded phase). Average: DA-SDF: 10.4 ± 8.6 cm, 0.8 ± 0.63 rad; DA-RF: 6.8 ± 5.1 cm, 0.44 ± 0.26 rad; DA-RF-OCCL: 1.7 ± 0.8 cm, 0.22 ± 0.14 rad.

Occlusions: Pose Tracking Performance

Finally, Figure 3.18 shows the pose tracking error for DA-SDF (baseline), DA-RF and DA-RF-OCCL (proposed) when rejecting pixels with a probability of less than 0.55. There is a clear difference in performance after $t > 18$ s when the manipulator moves towards the bottle.

With DA-SDF, the model is fitted to the occluding object as it cannot distinguish between parts of the robot and the bottle. By assigning irrelevant points to the manipulator, DA-SDF finally diverges and cannot recover. A similar behaviour can be observed for DA-RF, since those pixels of the bottle that have been classified as palm (Figure 3.14a, left) distract the tracking.

Meanwhile, by rejecting these pixels, DA-RF-OCCL is able to track with a similar performance as without occlusions (Figure 3.17). Without data association of finger classes, tracking relies on the visible parts of the arm until the hand becomes fully visible again.

3.5 Conclusion

This chapter presented a discriminative model-fitting approach for depth-based tracking of articulated objects in the presence of distracting visual information. The approach is based on the explicit pixel-wise association of data points to model parts.

In our experimental analysis we were able to avoid local minima which often arise from distractions in currently existing tracking objectives. The proposed random sampling of unspecified occlusions during training enabled us to reject less confident data-to-model associations and provides a way of tracking partially occluded manipulators. An advantage over comparable discriminative approaches is that it does not rely on explicitly modelled manipulanda or occluders, and does not even require a dedicated class to represent those visual distractions.

At present, our approach does not explicitly provide estimates for occluding pixels, i.e. we can only indirectly infer occlusions from low class probabilities. In Chapter 4 we will explore multi-label classification so as to classify robot parts and occlusions so that we can independently access the occlusion probability of a pixel and also which part it is occluding.

The Gauss-Newton approach only tracks a single state hypothesis provided by the single gradient from the data-to-model association per pixel. This is feasible since the tracking is initialised at the beginning from the reported joint encoder values. To become further independent of proprioception at this stage, Chapter 4 proposes to initialise a gradient-based optimiser from a distribution of initial states and Chapter 5 proposes an optimiser approach that maintains a distribution of state hypotheses.

Finally, we note that many articulated tracking approaches only make use of depth information, presumably because it is easier to synthesise and more directly generalises to real sensor readings. We hypothesise that colour can provide much stronger cues, in particular for small parts such as fingers. Chapter 4 will therefore propose a tracking objective that combines generative colour and data-driven depth information. Chapter 5 will further expand on this idea and explore colour and depth image synthesis for data-driven methods.

Chapter 4

Learning-driven Coarse-to-Fine Articulated Robot Tracking

The initial discriminative tracking objective in Chapter 3 improved model-fitting by removing many local minima found in standard generative objectives and by making the data association robust to visual distractions. However, due to the gradient-based optimiser that only uses a single hypothesis, tracking had to be initialised from joint encoder readings for the very first observed image in a tracking sequence. As a result, the approach presented in Chapter 3 cannot be applied to systems that do not have proprioception at all.

This chapter presents work on articulated tracking that does not depend on proprioception at any stage of the tracking pipeline and only relies on visual cues from depth and colour images. We combine these cues in a novel objective, that makes use of direct data association from depth keypoints and indirect data association from colour edges, to associate the observations to the estimated model. In addition to extracting these cues from the observed depth image, we also predict a distribution of model states from which we initialise the optimiser.

As before, our application focuses on the articulated tracking of a robotic manipulator during manipulation tasks that naturally contain visual distractions caused by the manipulandum itself and other unrelated occluding objects. Unlike with our previous approach of occlusion sampling, we use an explicit object class that is modelled via general object shapes during training.

The work presented in this chapter has been peer-reviewed and published as:

Learning-driven Coarse-to-Fine Articulated Robot Tracking by Christian Rauch, Vladimir Ivan, Timothy Hospedales, Jamie Shotton and Maurice Fallon in 2019 IEEE International Conference on Robotics and Automation (ICRA)

<https://doi.org/10.1109/ICRA.2019.8794359>

4.1 Introduction

Traditional robot manipulation requires a precisely modelled articulated robot arm with accurate position and torque sensing to execute trajectories with high precision. This approach has been most successful in industrial automotive manufacturing but typically does not use any exteroceptive sensing. In this chapter we focus on visually-driven manipulation where the scene is understood through visual object detection and fitting. The articulated robot arm is tracked purely visually. Many compliant robot arms suffer from structural bending and are not precise while in some industrial scenarios manipulators are entirely devoid of sensing such as in nuclear decommissioning [59]. In these scenarios vision-only manipulator tracking would be useful.

We explore model-based articulated tracking based entirely on RGB-D cameras passively detecting the arm. The goal is to determine the configuration of the robot arm model which best matches the observed state. One particular challenge is that a variety of different joint configurations can lead to visually similar observations. We are motivated by the work of [70, 73, 42] (in the field of human body tracking) to develop model-fitting approaches which leverage learned discriminative information to fit kinematically plausible states to the observed data.

Approaches such as [83] and [53] use 2D keypoint estimation to predict the 2D pixel location of joints but do not leverage the kinematic and visual information provided by the manufacturer’s 3D model of the robot. This kinematic information is only learned indirectly from a large training set and therefore needs to be explicitly enforced using an additional model-fitting stage. Although keypoints can provide reliable constraints for articulated tracking, they are only sparsely distributed. We therefore propose to use additional denser edge correspondences as a secondary tracking objective. These two objectives are visualised in Figure 4.1 for a cluttered environment. While edges provide densely distributed pixel-accurate correspondences, they are impaired by textured objects and are therefore unreliable as a sole tracking objective, in which case keypoints provide more stable cues.

Local optimisation algorithms leverage gradient information which, for kinematic models, can be easily obtained by differentiating the forward kinematics of the articulated model. However, initialising such a local optimisation solely from visual observations is challenging due to the visual ambiguity of shape symmetric robot links and the large range of possible joint motions. We therefore consider many possible candidates when initialising the optimisation. These candidates are drawn from a coarse robot state distribution that is predicted from a single depth

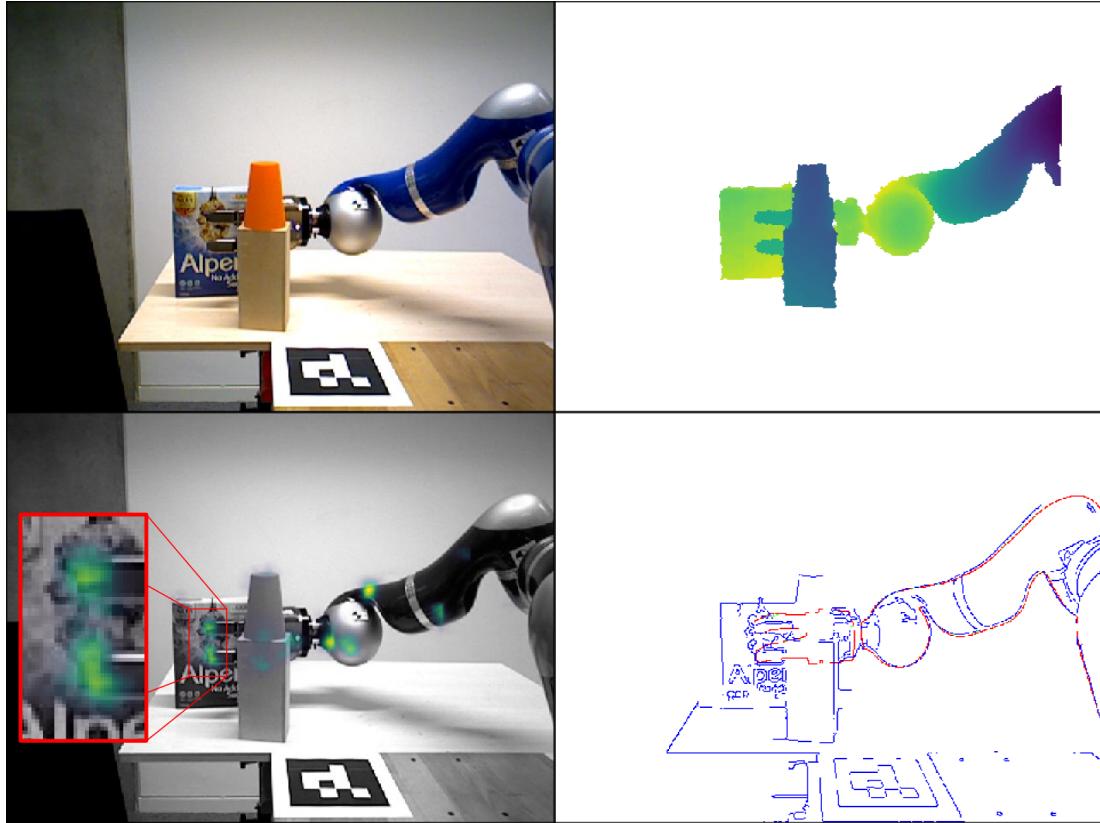


Figure 4.1: Grasping in a cluttered environment. **Top row:** *left*: colour image, *right*: colourised depth image of the manipulator and objects in the scene with the background removed. **Bottom row:** The combined tracking of keypoints (*left*, yellow/green) and edges (*right*, blue) enables precise tracking (red) during a grasping task even when parts of the manipulator are occluded. While keypoints provide sparse but stable visual cues for the fingers, edges provide pixel-accurate estimation of the upper arm. No joint encoder sensing was used here.

image. Sampling from this distribution allows us to consider many candidates and select the one that provides the best visual cues.

In summary, this chapter contributes:

1. a tracker initialisation strategy using a coarse joint position distribution predicted from a depth image,
2. a combined tracking objective that uses stable pixel-accurate cues from colour and depth images in a single unified framework.

The combination of these stages makes our proposed tracking approach independent of joint encoder sensing and consecutively refines the state from the initially sampled configuration via keypoint tracking until the basin of convergence for pixel-accurate edge correspondences is reached.

We show that, while keypoints already provide good performance for tracking a manipulator, the additional integration of edge information can reduce the end-effector tracking error to 2.5cm during grasping scenarios.

4.2 Related Work

A large corpus of work [60, 6, 66, 59] has investigated visual tracking for robotic manipulation in recent years. Visual tracking approaches ought to be able to mitigate effects such as linkage elasticity or joint encoder inaccuracies and could enable a more precise manipulation accuracy and a more holistic representation of the manipulation scene, including the manipulandum and obstacles in the scene.

4.2.1 Joint Position Distribution Prediction

Inspired by previous work on predicting the state of an articulated model from images in [70, 82], we propose to predict a distribution over the articulated state space of a robot manipulator. Similar to [70], we represent this distribution as discretised bins. Instead of training discrete state regressors for each of these bins, we propose to directly sample from the distribution that is represented by these bins. Compared to the discrete states provided by the retrieval forest in [82], our proposed sampling approach provides continuous interpolated samples from the state space and hence also includes samples that are not exactly part of the training set.

4.2.2 Visual Features

Different sparse and dense visual features have been used in the tracking literature to establish correspondences between the observed and estimated state of a 3D model. Early work in this area used dense features like colour image edges [57, 59, 52] and depth images [66]. These correspondences are based on the local appearance of the estimated state and change with each iteration of the optimisation. This results in many local minima which can be mitigated by introducing discriminative information [63].

Sparse keypoint features, learned from data, are commonly used for human pose estimation [83, 53] and used to estimate the skeleton configuration from 2D images. These approaches do not resolve 3D ambiguity, nor do they provide the exact visual representation that is required for robotic grasping tasks. An additional 3D pose estimation stage [91] can regress from these keypoint locations to 3D joint coordinates. As proposed in [10], we resolve the ambiguity when

mapping between a 2D keypoint to 3D pose by using a line-of-sight constraint and the camera intrinsics to constrain the optimisation state space.

Due to their stability, we propose to rely on keypoint tracking as the base objective. After initial optimisation, we then switch to dense but pixel-level accurate edge correspondence for accurate registration. Compared to Chapter 3, which used pixel-wise depth image segmentation as visual cues, the 2D keypoints proposed in this chapter can be located behind occlusions (see Figure 4.1) while intensity edges provide sharper contours than the imprecise edges in typical depth images.

4.2.3 Kinematic Optimisation

Model-fitting approaches, such as [66, 52, 42], rely on accurate models to find the optimal state that is kinematically and visually plausible. While global optimisation methods are less prone to local minima, they are also more difficult to tune and are computationally expensive. Local optimisation approaches on the other hand are well established and make use of gradient information to quickly converge to a minimum. We use the optimisation toolbox EXOTica [36], which provides a modular way to exchange solvers and objectives, to fit our robot model using the keypoint and edge objectives. Inspired by [78], we first optimise the kinematic chain from the base of the robot to the palm or wrist, before optimising the smaller finger links. This makes sure that the optimisation of fingers, which have less visual features and are more likely occluded, is initialised from a reasonable state.

4.3 Method

4.3.1 Overview

To find kinematically plausible robot configurations which match the observed depth image, we propose a coarse-to-fine inverse kinematic optimisation in three stages (Figure 4.2). First, we sample from a predicted distribution of joint values to propose a set of possible initial configurations (Section 4.3.3). These coarse samples are tested in the second stage to select the sample which minimises the keypoint tracking objective on the first image in a sequence. In the third stage, we minimise the combined keypoint and edge objective (Section 4.3.6) consecutively on a sequence of images (Section 4.3.7). All three stages use visual cues that we extract from the depth image using a multi-task convolutional neural network (Section 4.3.2).

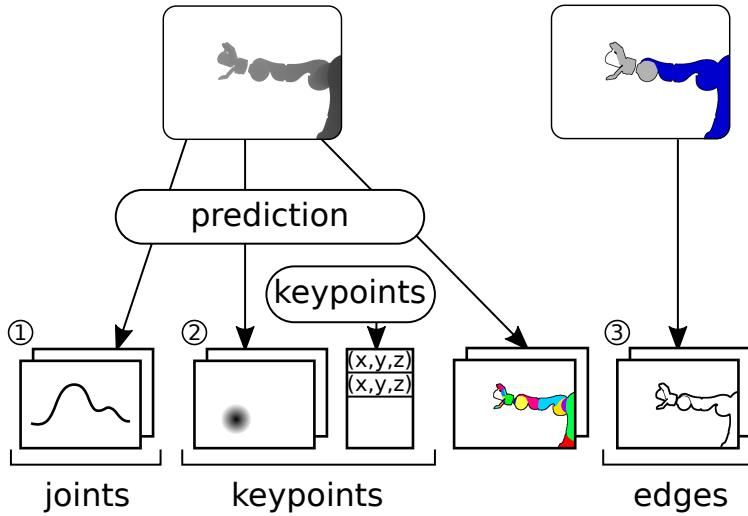


Figure 4.2: An overview of the sources of information extracted from an observed RGB-D image pair. From left to right, they provide increasingly detailed visual cues for the tracking system. (1) A broad joint position distribution provides samples to initialise an optimiser, which (2) optimises the keypoint objective until (3) we are close to the basin of convergence of the edge objective.

4.3.2 Multi-Task Prediction

Predicting the joint position distributions and keypoint heatmaps is done in parallel on a depth image \mathbf{I}_D . Since these tasks share depth image features, they are commonly trained in a multi-task setup (Figure 4.3). In our architecture, we use a ResNet-34 [32] to extract 256 feature maps that are used by the task specific branches. As the type of a keypoint relates to the link it belongs to, we train an additional segmentation task to support the training of the keypoint task. The segmentation provides a separation of the depth image into background and the individual robot links, but is not used for the tracking objective.

The segmentation and keypoint heatmaps provide information in the image space about pixels being occupied by a link or a 2D keypoint, respectively. The feature maps are therefore individually upscaled by 3×3 transposed convolutions to 128 task specific feature maps of the original depth image resolution. For the segmentation this is followed by a regular 2D 3×3 convolution and a softmax layer for providing the probabilities for the $N_L = 18$ robot links, the background and the object ($N_C = N_L + 2 = 20$ classes in total). To reuse information about the location of links for the keypoint localisation, we concatenate the upscaled heatmap features with the segmentation features and apply 3×3 2D convolution with a sigmoid activation function.

The $N_K = 22$ 3D keypoints are manually placed on the surface of the 18 links (Figure 4.4). During training, they are transformed via the true state into the

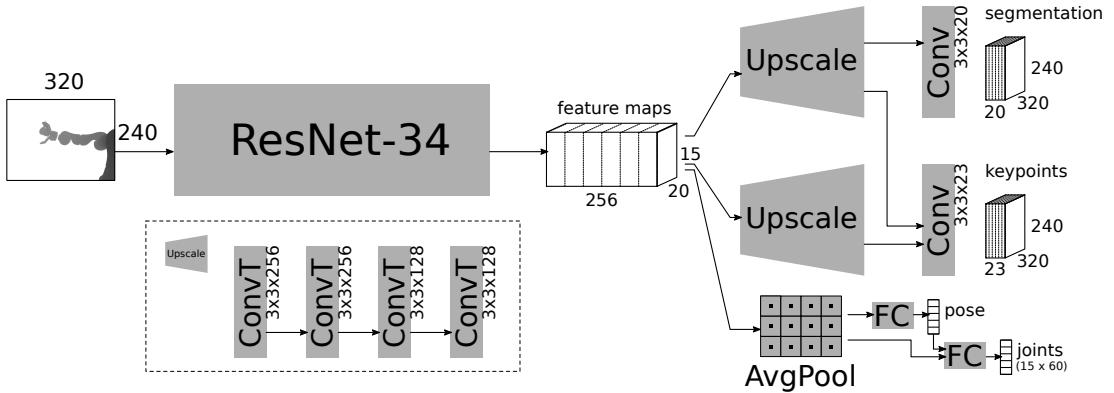


Figure 4.3: Our approach uses multi-task prediction to obtain segments, keypoints and joint estimates from a single depth image. All three tasks use common depth features extracted by a ResNet-34 [32].

camera frame and projected via the camera projection matrix K onto the 2D image plane. 2D Gaussians with $\sigma = 3\text{px}$ are centred on each of the 22 keypoint pixel locations to obtain the final heatmaps [83]. An additional background heatmap is created to represent the probability of a pixel not being assigned to any keypoint ($N_K + 1$). During prediction, we can only recover the line-of-sight from the camera origin to the 2D keypoint on the image plane. This ambiguity is resolved with the point-to-line constraint in Section 4.3.5. A qualitative example of the resulting keypoint scores above 0.5 is given in the green/yellow overlay in Figure 4.1 bottom left. The visualised keypoint scores are allocated over all keypoint heatmaps. Yellow indicates a high keypoint score, such as for the forearm and finger tips, and green indicates a lower score, such as for the occluded keypoints on the wooden box in front of the end-effector.

The third branch provides a joint state distribution to initialise the optimisation, and the 6D robot pose as support. Since a regular regression of the joint state only provides the state vector itself without a confidence measure, we train the network to predict a distribution of joint states. For each of the $N_J = 15$ joint positions, we place a 1D Gaussian with $\sigma = 0.2\pi\text{rad}$ on the true joint position. This Gaussian is then discretised in the value range $[-\pi, +\pi]\text{rad}$ into $N_B = 60$ histogram bins (Figure 4.5) which results in a resolution of $6\text{deg} \equiv 0.1\text{rad}$. Strictly speaking, this Gaussian is not a probability distribution since it is not normalised but we will use the term *distribution* since the histogram bars represent how likely a joint position will occur in this value range. All discretised joint positions are serialised into a single vector $\mathbb{R}^{(N_J N_B) \times 1}$ and then reshaped into a matrix $\mathbf{I}_\theta \in \mathbb{R}^{N_J \times N_B}$ containing the score values of discretised joint positions. After prediction we can treat the scores of each joint as an unnormalised probability distribution function (PDF) from which we can sample joint states. Since we are sampling from a continuous

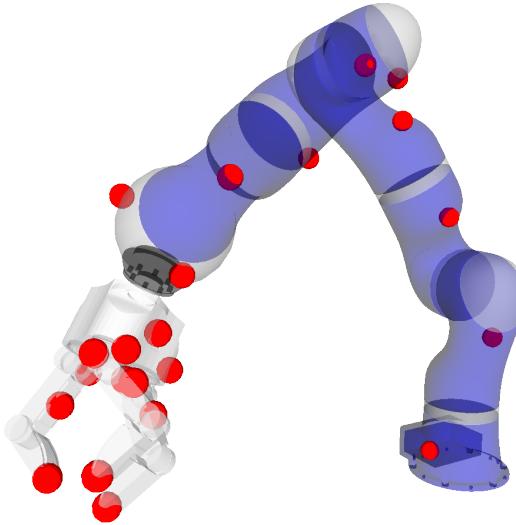


Figure 4.4: Manually selected 3D keypoints (red) on tracked robot links. These 22 keypoints are projected onto the image plane during training to obtain the true 2D keypoints and their heatmap.

distribution, the resolution of the discretised bins does not directly affect the sampled states, as long as it is small enough to represent multiple modes.

We will denote the extraction of segments $\mathbf{I}_S \in \mathbb{R}^{W \times H \times N_C}$, keypoint heatmaps $\mathbf{I}_K \in \mathbb{R}^{W \times H \times (N_K+1)}$, and joint position distribution $\mathbf{I}_\theta \in \mathbb{R}^{N_J \times N_B}$ from a depth image $\mathbf{I}_D \in \mathbb{R}^{W \times H \times 1}$ ($W = 320$, $H = 240$) by this network as:

$$\{\mathbf{I}_S, \mathbf{I}_K, \mathbf{I}_\theta\} = \text{extract}(\mathbf{I}_D) . \quad (4.1)$$

This inference function takes 0.07s per depth image.

4.3.3 Sampling of Initial Configuration

Robotic manipulators with long kinematic chains can have a large variety of possible joint position configurations that are far away in joint space but lead to the same end-effector pose with very similar local appearance. Trying to directly predict the joint position from a single depth image is therefore an ambiguous task. Although we provide a unimodal 1D Gaussian joint position distribution as a training target, it is likely that a multimodal distribution is predicted for visually similar appearing configurations of a link (Figure 4.6).

Since the mean or the strongest mode of such a distribution might not correspond to the observed link state, we propose to sample independently from each joint's distribution to initialise the optimisation several times. To obtain samples from the PDF represented by the score bins, we linearly interpolate the cumulative

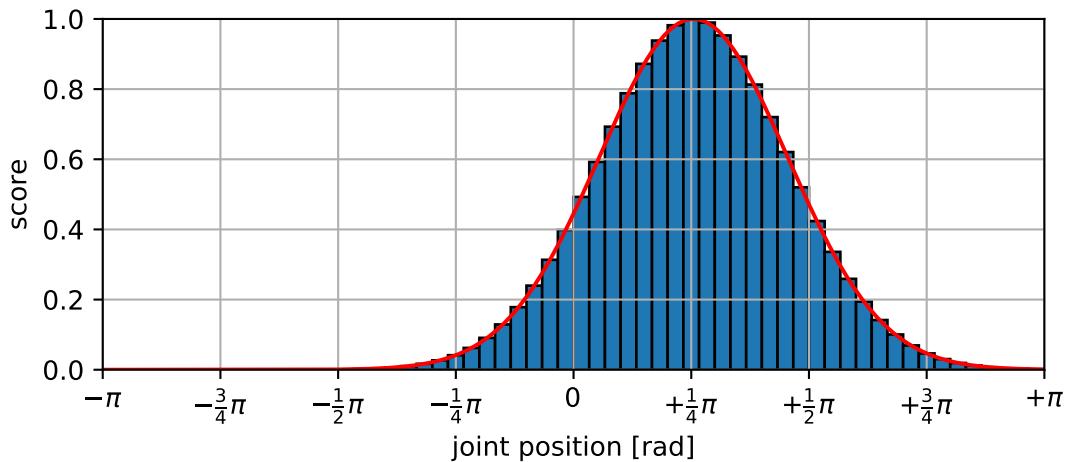


Figure 4.5: Example of a discretised Gaussian of a single joint position for training. The Gaussian (red) $e^{\frac{-(x-0.8\text{rad})^2}{2(0.2\pi\text{rad})^2}}$ has the highest score of 1 at the true joint position of $\mu = 0.8\text{rad}$. This function is discretised at 60 equidistant positions over the interval of $[-\pi, +\pi]\text{rad}$ to form the centres of the histogram bars (blue).

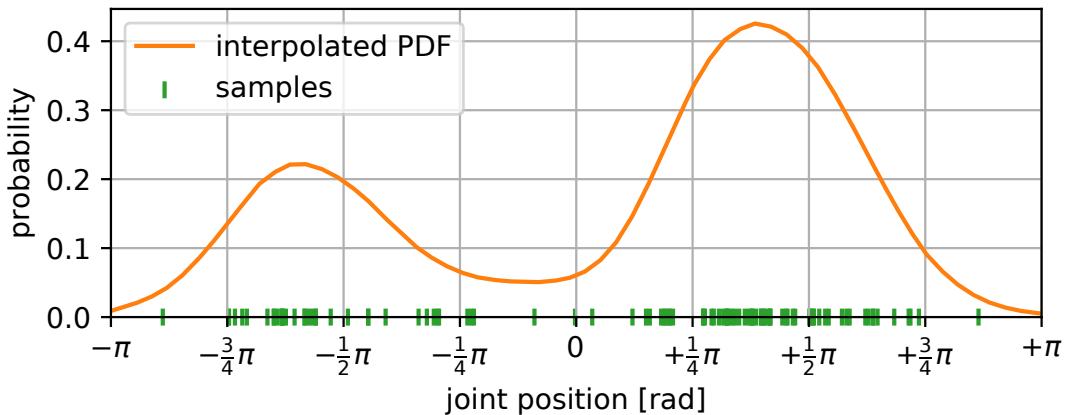


Figure 4.6: Example distribution (orange) and samples (green) of a lower arm joint position, predicted and sampled from an image of the occluded bottle sequence (Figure 4.13). The distribution shows two strong modes at $+1.2\text{rad}$ and -1.8rad since the link has a similar visual appearance for each half rotation.



Figure 4.7: Subset of object meshes. **Top:** synthetic meshes of typical IKEA objects, **bottom:** real 3D meshes acquired by an Artec Eva 3D scanner [1].

sum of the bin scores, which provides the cumulative distribution function (CDF), and sample uniformly in $\mathcal{U}(0, 1)$ from the inverse CDF.

4.3.4 Training

The segmentation, keypoint localisation and joint distribution prediction is trained using approximately 125000 synthetically rendered depth images in the manner described in Section 3.3.3. These synthetic images show the robot at different configurations sampled from a wide range of states where the palm is inside the camera frustum (Figure 3.2). As proposed in [51], we randomly select one of 30 objects (Figure 4.7) and place the object at a random pose inside the hand to simulate interaction with a manipulandum. We do not discriminate between these 30 objects, but treat them as a single object class.

During training we minimise a weighted cross entropy for the segmentation task, the mean absolute error on the keypoint heatmaps, and the mean squared error on the discretised joint scores. The pixel-wise cross entropy over N_C classes is defined as

$$H(y, \hat{y}) = - \sum_{c=1}^{N_C} y_c \log(\hat{y}_c) \quad (4.2)$$

between the true $y \in \{0, 1\}$ and predicted $\hat{y} \in [0, 1]$ outcome of the classification. This cross entropy is weighted by median frequency balancing [22]:

$$w_c = \frac{\text{median}(f_0, \dots, f_{N_C})}{f_c} \quad (4.3)$$

with the class frequency:

$$f_c = \frac{1}{N} \sum_x^N [x = c] \quad (4.4)$$

over all N pixels in the training set. This class balancing increases the classification accuracy for smaller links such as the proximal and distal finger phalanges.

4.3.5 Tracking Objective

The observed robot state is provided by the colour and depth images ($\mathbf{I}_C, \mathbf{I}_D$), the additionally predicted keypoints \mathbf{I}_K , and the joint position distribution \mathbf{I}_θ . The estimated robot state is initially provided by the sampled configurations and thereafter from the optimisation on consecutive image frames. The visual representation of the estimated state θ is obtained by rendering the link meshes at their estimated pose.

The objective for the optimisation is to minimise the distance between observed 2D keypoints and edges, and their corresponding estimated visual 3D representation. Since the depth of these observed keypoints and edges cannot be fully determined, e.g. a keypoint might be occluded, the objective is formulated using a line-of-sight constraint.

Keypoints

For each predicted heatmap,

$$H_k = (\mathbf{I}_K)_{i,j,k} \forall k \in [1, N_K] \quad , \quad (4.5)$$

we select the pixel coordinate x with the highest score s ,

$$x_k = \arg \max_s H_k \quad , \quad (4.6)$$

and its associated depth reading, $z = \mathbf{I}_D(x_k)$, to obtain the line-of-sight start \mathbf{l}_s via back-projection $\mathbf{l}_z = K^{-1} z x_k$. The corresponding 3D keypoint \mathbf{p} is transformed from its local coordinate frame to the camera frame via forward kinematics during the optimisation. The keypoint correspondences k are established per observed

image frame and stay constant during optimisation. Keypoints with a score of less than 0.5 are ignored for tracking.

Edge Pixels

Estimated edge pixels are related to their closest observed edge pixel by the distance transform of the Canny [11] edges, $\mathbf{I}_{E,obs} = \text{canny}(\mathbf{I}_C)$, of the observed colour image. The estimated edge pixels $\mathbf{I}_{E,est}$ and their 3D coordinate are provided by transforming, $\mathbf{p} = FK(\theta)$, and projecting, $x = K\mathbf{p}$, the estimated state θ on the image plane. We iterate through these estimated edge pixels and assign them to the closest observed edge pixel x_e if the angle between their normals is smaller than 8 degrees, i.e. if they point roughly in the same direction, and if their point-line distance is closer than 5cm. This is similar to the orthogonal line search proposed in [52]. The edge-to-edge association provides multiple edge pixel correspondences per link, which are updated at each iteration by rendering the new estimated state.

Point-Line Distance

The lines-of-sight \mathbf{l} in the camera frame are extracted from an observed colour and depth image pair, $\mathbf{I} = (\mathbf{I}_C, \mathbf{I}_D)$, by extracting image coordinates x from predicted keypoints (eq. 4.1, 4.5 and 4.6),

$$X_k = \{x_k \mid x_k = \arg \max_s H_k, H_k = (\mathbf{I}_K)_{i,j,k} \forall k \in [1, N_K], \mathbf{I}_K = \text{extract}(\mathbf{I}_D)\} \quad , \quad (4.7)$$

and Canny edges,

$$X_e = \{x \mid \mathbf{I}_{E,obs}(x) = 1, \mathbf{I}_{E,obs} = \text{canny}(\mathbf{I}_C)\} \quad , \quad (4.8)$$

and back-projecting them using the camera projection matrix K ,

$$h_{obs}(\mathbf{I}) = \{\mathbf{l}_z \mid \mathbf{l}_z = K^{-1}zx, x \in X_e \uplus X_k\} \quad , \quad (4.9)$$

where \mathbf{l}_z is the point on the line $\mathbf{l} = \mathbf{l}_e - \mathbf{l}_s$ with corresponding depth z . The disjoint set union, denoted by \uplus , is used here since a single image coordinate x can be part of X_e and X_k and thus provide an edge and keypoint point-line objective. If valid depth readings ($z > 0$) are available, we can constrain the start of a line as $\mathbf{l}_s = \mathbf{l}_z$, otherwise $\mathbf{l}_s = \mathbf{0}$. We cannot constrain the end of the lines-of-sight, \mathbf{l}_e , and thus can arbitrarily set $\mathbf{l}_e = a\mathbf{l}_z|_{z=1} \forall a \in (0, \infty)$.

The previously defined link keypoints and the link meshes are transformed from the link to the camera frame at each optimisation iteration and provide the corresponding estimated 3D visual representation,

$$h_{est}(\theta) = \{\mathbf{p} \mid \mathbf{p} = \text{FK}(\theta)\} . \quad (4.10)$$

The observed lines \mathbf{l} and estimated points \mathbf{p} are related to each other by their keypoint identity (k in eq. 4.5) or the edge distance transform, and the tracking objective is then to minimise the distance of \mathbf{p} to its projection onto \mathbf{l} ,

$$\text{proj}_{\mathbf{l}}\mathbf{p} = \mathbf{l}_s + \min \left(0, -\frac{(\mathbf{l}_s - \mathbf{p}) \cdot \mathbf{l}}{\|\mathbf{l}\|^2} \right) \mathbf{l} . \quad (4.11)$$

The magnitude and direction of the required updates are derived from the point-line vector

$$\mathbf{d} = \text{proj}_{\mathbf{l}}\mathbf{p} - \mathbf{p} \quad (4.12)$$

that points from the estimated point to its closest observed correspondence.

The final objective is then to minimise all N_d point-line distances:

$$e(\theta, \mathbf{I}) = \sum_i^{N_d} \|\mathbf{d}_i\|^2 . \quad (4.13)$$

4.3.6 Optimisation

We want to find a state θ that minimises the combined point-line distances over all keypoint and edge correspondences (eq. 4.13). The gradient of the point-line distance $\|\mathbf{d}\|$ w.r.t. θ is derived from the gradient of the frame position $\frac{\partial \text{FK}(\theta)}{\partial \theta}$ using the chain rule.

Point-Line Gradient

By differentiating the point-line distance $\|\mathbf{d}\|$ w.r.t. θ :

$$\frac{\partial \|\mathbf{d}\|}{\partial \theta} = \frac{\partial \sqrt{\mathbf{d}^\top \mathbf{d}}}{\partial \theta} = \frac{\partial \mathbf{d}}{\partial \theta} \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|} \quad (4.14)$$

with $\frac{\partial \mathbf{d}}{\partial \theta}$ derived from the differentiation of the point-line vector (eq. 4.11 and 4.12) w.r.t. θ :

$$\frac{\partial \mathbf{d}}{\partial \theta} = \left(\frac{\partial \mathbf{p}}{\partial \theta} \cdot \frac{\mathbf{l}}{\|\mathbf{l}\|^2} \right) \mathbf{l} - \frac{\partial \mathbf{p}}{\partial \theta} , \quad (4.15)$$

we arrive at the individual point-to-line objective derivatives:

$$\frac{\partial \|\mathbf{d}\|}{\partial \theta} = \left[\left(\frac{\partial \mathbf{p}}{\partial \theta} \cdot \frac{1}{\|\mathbf{l}\|^2} \right) \mathbf{l} - \frac{\partial \mathbf{p}}{\partial \theta} \right] \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|} \quad . \quad (4.16)$$

The partial derivatives of the estimated keypoints and edge points, $\frac{\partial \mathbf{p}(\theta)}{\partial \theta}$, in their respective frames, are derived from the differentiation of forward kinematics, $\frac{\partial \mathbf{FK}(\theta)}{\partial \theta}$.

Each of the $l \in [1, N_L]$ links yields a set of point-line correspondences from keypoints ($k_l \in [1, N_{K,l}]$) and edges ($e_l \in [1, N_{E,l}]$). $N_{K,l}$ is the specific number of detected keypoints per link l , hence $\sum_l N_{K,l} \leq N_K$. The relative contribution of keypoints and edges is weighted by link specific binary weights $\alpha_{K,l} \in \mathbb{B}$ and $\alpha_{E,l} \in \mathbb{B}$, with $\mathbb{B} = \{0, 1\}$. This binary weighting effectively switches between the two objectives. Alternatively, a continuous weighting in $[0, 1]$ can be used to superimpose both objectives.

The individual objectives ($\|\mathbf{d}_{k_l}\|$, $\|\mathbf{d}_{e_l}\|$) and their derivatives ($\frac{\partial \|\mathbf{d}_{k_l}\|}{\partial \theta}$, $\frac{\partial \|\mathbf{d}_{e_l}\|}{\partial \theta}$) are averaged and weighted by their contribution to form the objective and gradient per link:

$$\|\mathbf{d}_l\| = \alpha_{K,l} \frac{1}{N_{K,l}} \sum_{k_l}^{N_{K,l}} \|\mathbf{d}_{k_l}\| + \alpha_{E,l} \frac{1}{N_{E,l}} \sum_{e_l}^{N_{E,l}} \|\mathbf{d}_{e_l}\| \quad (4.17)$$

$$\frac{\partial \|\mathbf{d}_l\|}{\partial \theta} = \alpha_{K,l} \frac{1}{N_{K,l}} \sum_{k_l}^{N_{K,l}} \frac{\partial \|\mathbf{d}_{k_l}\|}{\partial \theta} + \alpha_{E,l} \frac{1}{N_{E,l}} \sum_{e_l}^{N_{E,l}} \frac{\partial \|\mathbf{d}_{e_l}\|}{\partial \theta} \quad . \quad (4.18)$$

The weights α are used to switch the tracking objective between keypoints ($\alpha_K = 1$, $\alpha_E = 0$) and edges ($\alpha_K = 0$, $\alpha_E = 1$) individually per link.

With the formulation of link-specific point-to-line objectives through the combination of individual keypoint and edge point-line correspondences (eq. 4.17), the final tracking objective (eq. 4.13) is reformulated as

$$e(\theta, \mathbf{I}) = \sum_l^{N_L} \|\mathbf{d}_l\|^2 \quad . \quad (4.19)$$

Solver

We now have single objectives and gradients per link which we jointly minimise through gradient-based optimisation by stacking the objectives $\|\mathbf{d}\| \in \mathbb{R}^{1 \times 1}$ and

gradients $\frac{\partial \|\mathbf{d}\|}{\partial \theta} \in \mathbb{R}^{1 \times N_J}$,

$$\phi = \begin{bmatrix} \|\mathbf{d}_1\| \\ \vdots \\ \|\mathbf{d}_{N_L}\| \end{bmatrix} \quad (4.20)$$

$$J = \begin{bmatrix} \frac{\partial \|\mathbf{d}_1\|}{\partial \theta} \\ \vdots \\ \frac{\partial \|\mathbf{d}_{N_L}\|}{\partial \theta} \end{bmatrix} \quad (4.21)$$

with gradient vector

$$\frac{\partial \|\mathbf{d}_l\|}{\partial \theta} = \left(\frac{\partial \|\mathbf{d}\|}{\partial \theta_1}, \frac{\partial \|\mathbf{d}\|}{\partial \theta_2}, \dots, \frac{\partial \|\mathbf{d}\|}{\partial \theta_{N_J}} \right) \quad (4.22)$$

to form the vector function $\phi(\theta) : \mathbb{R}^{N_J} \mapsto \mathbb{R}^{N_L}$ and the Jacobian $J(\theta) : \mathbb{R}^{N_J} \mapsto \mathbb{R}^{N_L \times N_J}$.

Given the pseudo-inverse Jacobian J^\dagger , the final objective (eq. 4.19) is iteratively minimised w.r.t. θ by gradient update steps:

$$\theta_{i+1} = \theta_i - J^\dagger \phi \quad . \quad (4.23)$$

Since the root link of the robot is rotational symmetric and often not observed in the depth image, we use the true 6D camera pose and do not optimise these state variables.

4.3.7 Tracking Pipeline

The tracking operates on a continuous sequence of paired colour and depth images $\mathbf{I}_t = (\mathbf{I}_{C,t}, \mathbf{I}_{D,t})$. The colour and depth images are timestamp synchronised by matching a colour image to the depth image with the lowest timestamp difference using the depth as the reference time t . This can cause a misalignment of the paired colour and depth image for up to 0.016s but can be neglected for low joint speeds. The tracking is initialised once at the beginning ($t = 0$) from the predicted distribution (Section 4.3.2), $\mathbf{I}_{\theta,0} = \text{extract}(\mathbf{I}_{D,0})$, from which we sample 50 configurations as described in Section 4.3.3. From these samples we select the initial configuration θ_0 with the smallest keypoint objective (Section 4.3.5), i.e. the forward kinematics state with the smallest average Euclidean distance between the 3D keypoints and their corresponding line-of-sight. The optimisation is then initialised at each new image pair using the previous solution and iterates for 10 iterations (0.37s).

The edge- and keypoint-weight configuration of the point-to-line objective is updated during tracking individually for each link. A link l switches from keypoint to edge tracking ($\alpha_{K,l} = 0$, $\alpha_{E,l} = 1$), if all of its keypoint distances are closer than 2cm ($\|\mathbf{d}\|_{k_l} \leq 0.02$) and vice versa. Finger links always use the keypoint objective.

We initially track only the arm and palm, and switch to full tracking when the keypoint error of the upper links are smaller than 2cm, and switch back to arm and palm tracking when this error becomes larger than 3cm. This hysteresis thresholding has been chosen to minimise oscillation, that would otherwise occur around a single threshold.

4.4 Evaluation

We evaluate our tracking approach on four sequences that show grasping of different objects and occlusions, using a KUKA LWR4 7 DoF arm with a Schunk SDH2 7 DoF end-effector, which is observed by a statically mounted Asus Xtion PRO LIVE RGB-D camera. The arrangement of robot and camera is the same as described in Section 3.4.1.

In the following, we report the tracking error as the distance between a reference frame T_{ref} , as reported by FK on the joint position encoders, and the estimated frame T_{est} as given by FK on the estimated state θ . If $T_{err} = T_{est}^{-1}T_{ref}$ is the transformation that has to be applied to the estimated pose to obtain the reference pose, and t_{err} is the translational part of this transformation, then the reported scalar tracking error is $p_{err} = \|t_{err}\|$.

4.4.1 Sampling Robot States

We will evaluate the first stage of our proposed pipeline, which provides initial robot joint configurations using the predicted distribution (Section 4.3.3). Figure 4.8 shows snapshots of two tracking sequences, each with three sampled configurations. For these visualisations, we sampled 50 configurations from the predicted distribution and automatically selected the three configurations with the smallest average edge-to-edge distance. These configurations coarsely align with the observed state and demonstrate that the predicted distribution provides reasonable robot states to initialise the local optimisation.

Figure 4.9 additionally compares samples from our predicted distribution with samples that are sampled uniformly between joint limits. This qualitative comparison shows that the end-effector position of predicted states is closer to the observed manipulandum, while states from the uniform distribution are much broader distributed within the workspace.

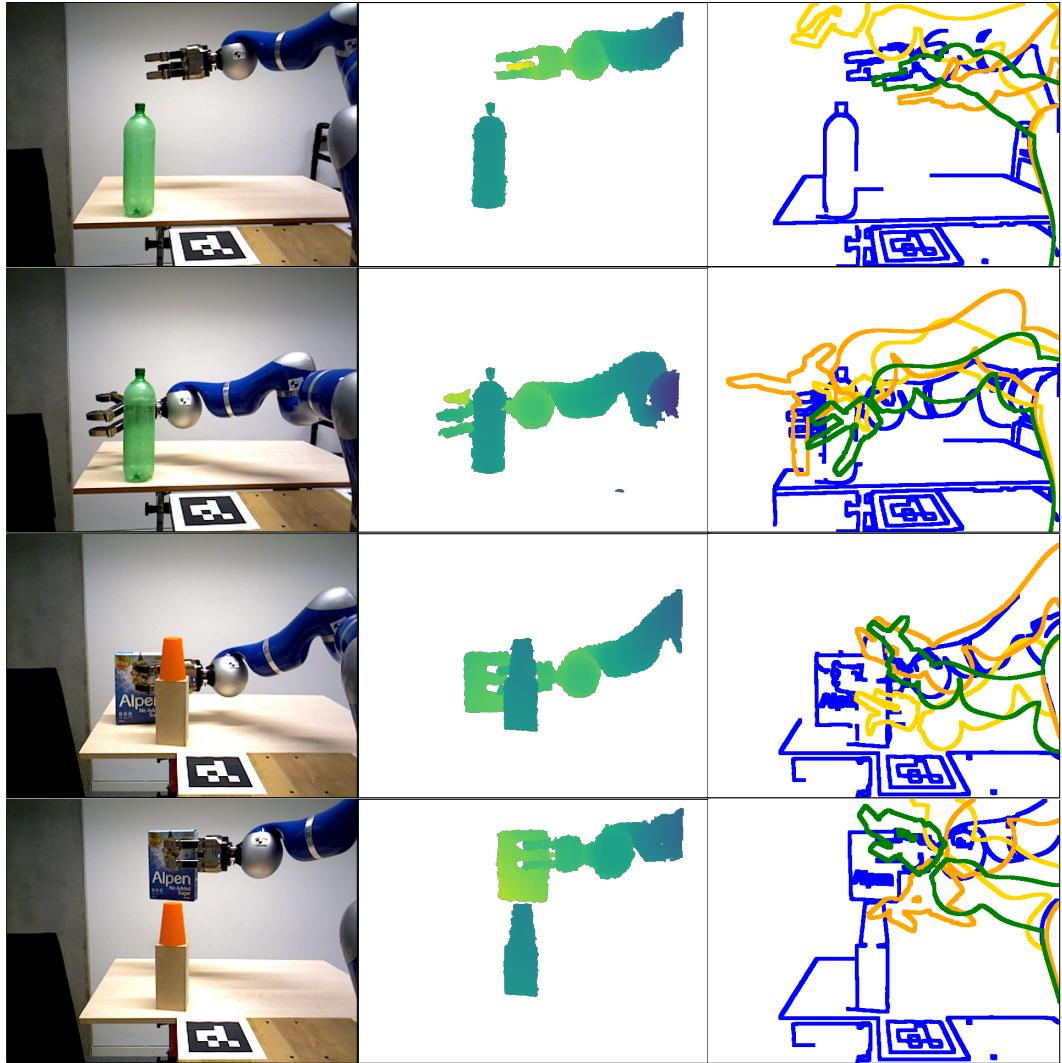


Figure 4.8: Sampling from the joint position distribution. **Left:** colour image of the observed scene, **Middle:** depth image from which we predict the joint position distribution, **Right:** observed edges (blue) overlaid with the contours of three sampled configurations (green, orange, yellow).

This qualitative observation is confirmed by the quantitative evaluation of the objective and end-effector position in task space (Figure 4.10). The convergence of the keypoint objective (Figure 4.10a) shows that samples from the predicted distribution start with a smaller initial error which eventually leads to a smaller end-effector position error (Figure 4.10b) after 50 iterations.

4.4.2 Tracking

We apply the proposed tracking approach on the four sequences as described in Section 4.3.7. To evaluate the contribution of edge tracking, we apply tracking once with the keypoint-only objective ($\alpha_{E,l} = 0$) and once with the combined

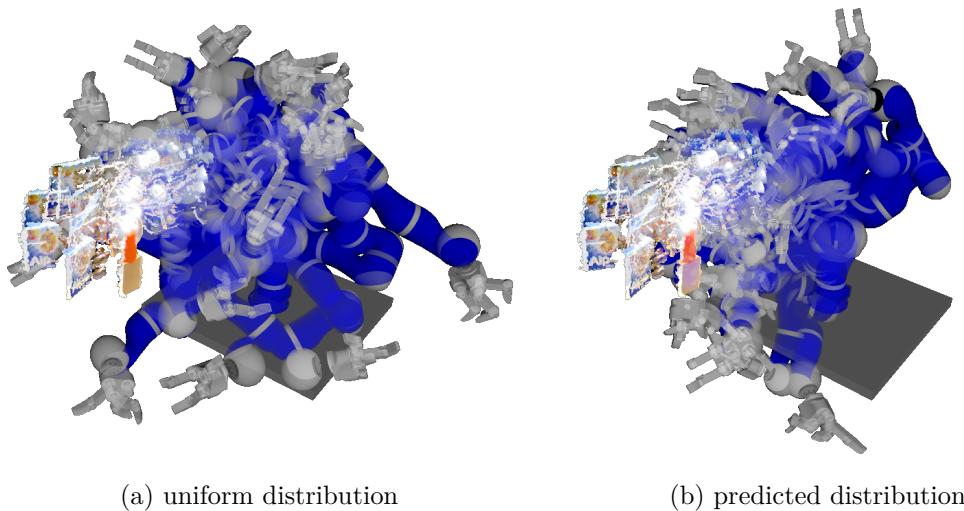


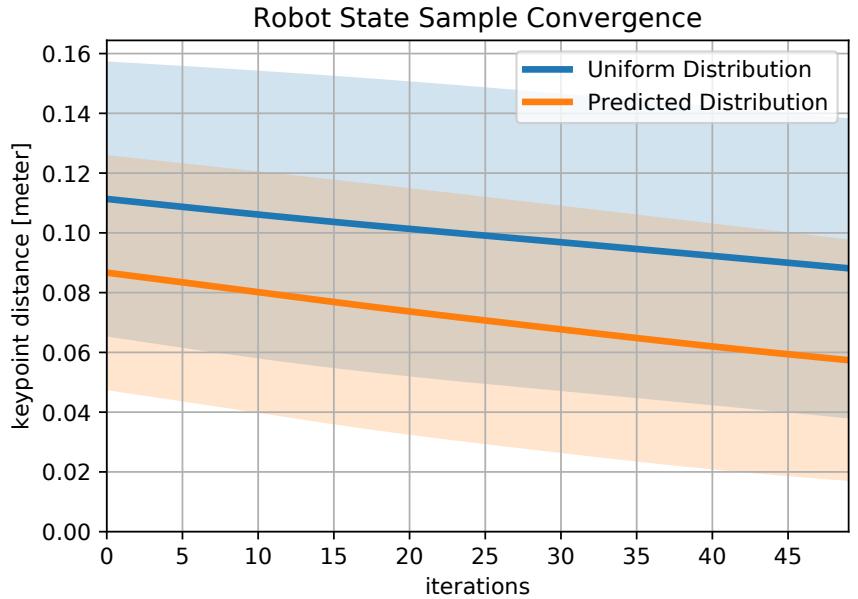
Figure 4.9: Sampled states for 40 equidistant still images of a grasping sequence (Figure 4.8, last two rows). 5 samples were drawn per still image from the uniform or predicted distribution and the sampled state that would converge closest to the predicted keypoints after 50 iterations is shown. Samples from the uniform distribution (a) are spread across the workspace, while samples from our predicted distribution (b) cluster in a banana-shaped area around the manipulandum.

keypoint and edge objective with the same sampled starting state. Apart from this objective setup, we use the same configuration for all sequences. Figures 4.11 to 4.14 report the position tracking error for a forearm link and the palm (fifth and ninth link in the kinematic chain) for the combined keypoint and edge objective (top), and show snapshots of the estimated state overlaid as contours on top of the observed colour images (bottom).

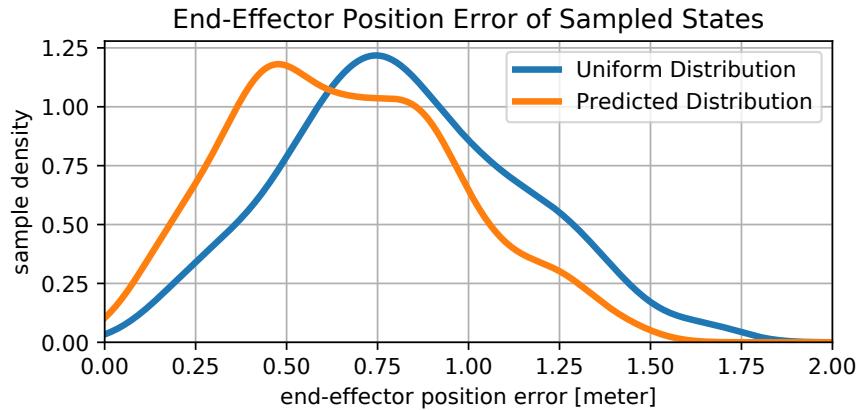
By using edges as an additional objective, the average palm position error in the non-occluded grasping sequences (Figures 4.11 and 4.12) has reduced from 3.7cm to 2.7cm and 3.1cm to 2.5cm, respectively. Although the *occluded bottle* sequence (Figure 4.13) shows improved tracking performance of the forearm link, this is not propagated to the palm.

The *grasping behind occlusions* sequence (Figure 4.14) is the most challenging of our sequences since it contains distractions of both types (manipulandum and occlusion) and has a textured occluder with many edge responses. In this adverse setting, we are still able to track the palm with an average position error of 4.5cm, which is less than half of the palm length (9.38cm). The keypoint-only baseline performs slightly better in this case.

Figure 4.15 provides a comparison of the point-line tracking with the SDF-based tracking approach (Section 3.3). While the SDF-based approach provides better palm pose estimates in the occlusion case, the performance of the point-line



(a) keypoint objective convergence of sampled state distribution



(b) distribution of end-effector positions error after 50 iterations

Figure 4.10: Performance of the keypoint objective with samples from a uniform (blue) and the predicted (orange) distribution. (a) Convergence of the keypoint error ($\frac{1}{N_K} \sqrt{\sum^{N_K} \|d_k\|^2}$) of 200 samples (40 still images à 5 samples) over 50 iterations, (b) final converged end-effector position error after 50 iterations. Samples from the predicted distribution start with a smaller initial keypoint error and also converge to end-effector positions closer to the true reference position.

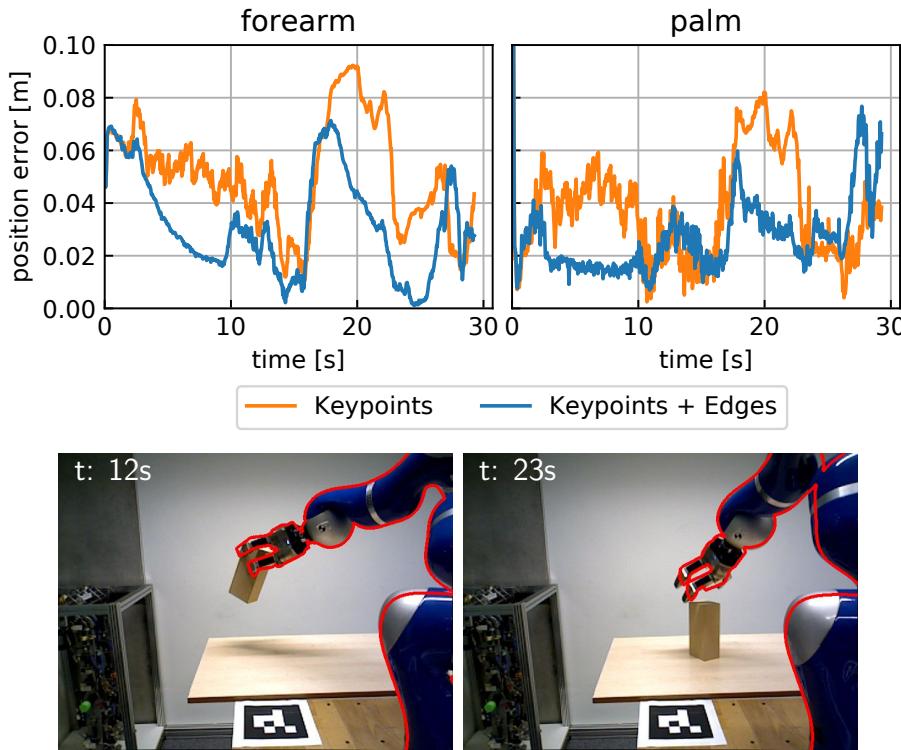


Figure 4.11: *Grasping box*. Using the additional edge objective reduces average position error from 5cm to 3.1cm (forearm) and 3.7cm to 2.7cm (palm).

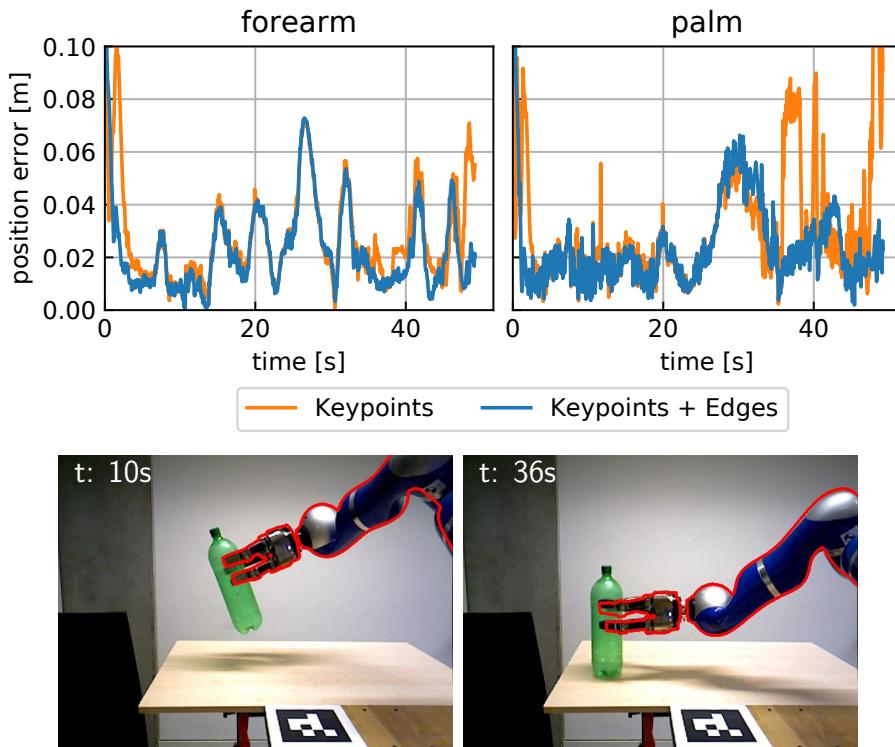


Figure 4.12: *Grasping bottle*. Using additional edge objective reduces average position error from 2.7cm to 2.3cm (forearm) and 3.1cm to 2.5cm (palm).

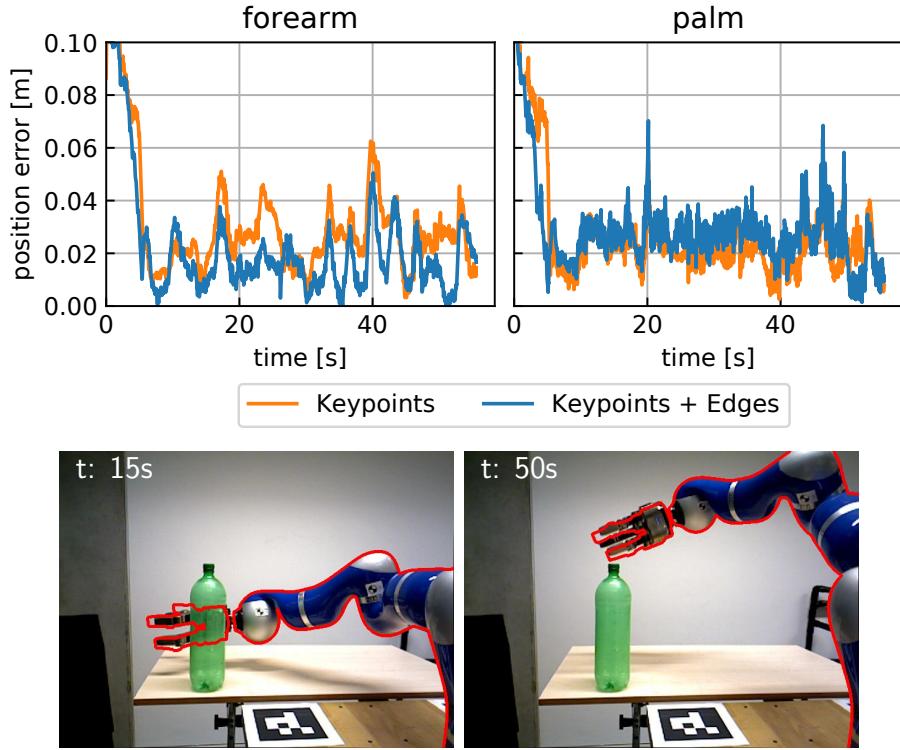


Figure 4.13: *Occluded bottle*. The additional edge objective improves the forearm error (3.1cm to 2.1cm), but marginally impairs the palm error (2.6cm to 2.8cm).

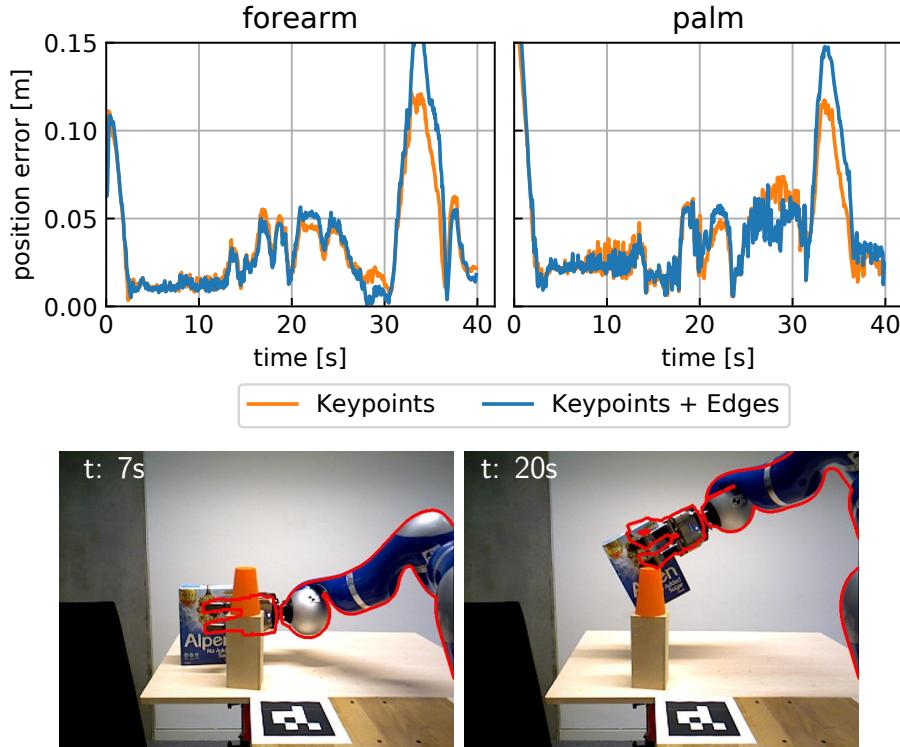


Figure 4.14: *Grasping Alpen box behind occlusion*. The additional edge objective impairs tracking (average palm position error increased from 4.3cm to 4.5cm), when strong visual distractions are present.

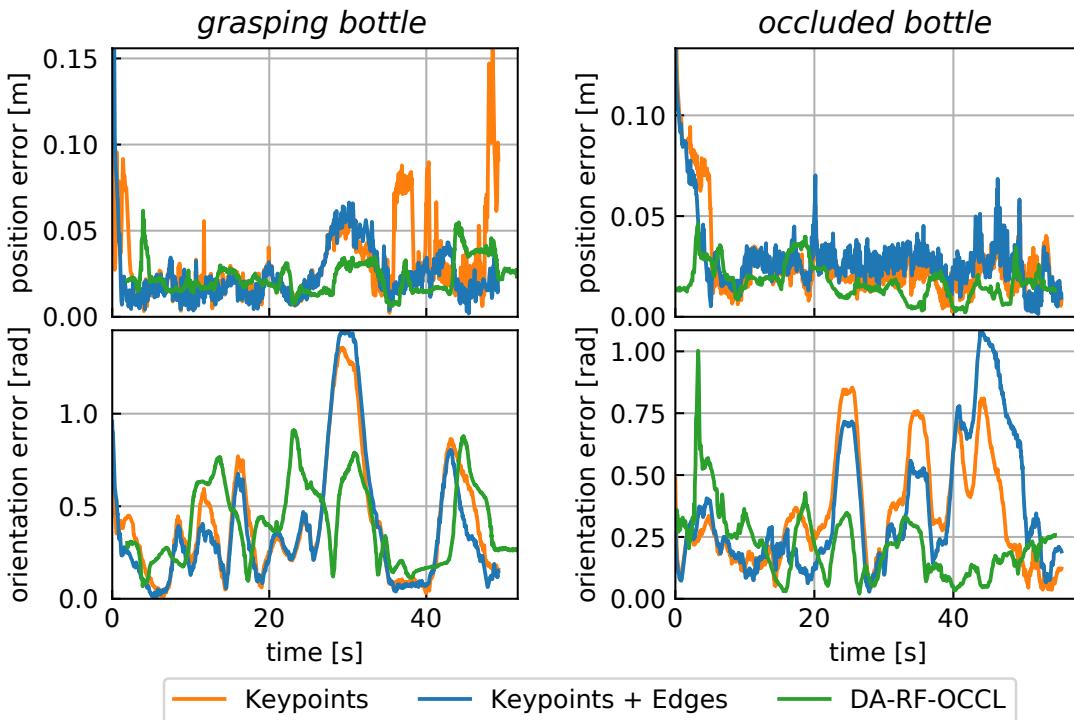


Figure 4.15: Comparison of the point-line objective with DA-RF-OCCL via the palm pose error on the bottle sequence.

approach is mostly similar or better when actually interacting with the object in the grasping case. We note that the SDF-based approach still requires an initialisation from proprioception in these cases, while the point-line tracking cannot rely on the accurate initialisation and only relies on the visual input.

In summary, our proposed tracking approach is able to reliably track an occluded manipulator when grasping, without making any assumptions on the presence of objects or the availability of joint encoder readings. This solves a common problem of articulated tracking approaches, which often need to be initialised from a known robot state. Our approach is therefore more generally applicable to scenarios where direct access to the robot is not available.

4.5 Conclusion

We presented a robotic manipulator tracking approach that relies solely upon visual cues to initialise tracking. At each iteration it consecutively updates the estimated state using a combination of colour edge and depth keypoint correspondences. The proposed deep multi-task network learns common depth image features that can be efficiently used in parallel for the coarse initialisation and the keypoint tracking objective. Dense colour image edges are then further used to refine the

estimated state. Our approach only requires an accurate kinematic and visual model to generate training data and to provide the estimated visual representation during tracking. No real robot data was required to train the network.

We evaluated our approach on four sequences showing the grasping of different objects and varying occlusions, and found that the additional edge tracking objective improves tracking of grasping scenes compared to only keypoint tracking. Even in cases with large occlusions and distractions, which the approach presented in Chapter 3 cannot handle, we are able to visually track the occluded palm with an average position error of less than half of the physical palm dimension. We note that we used the same tracking parameters for the sequences with and without occlusions and recommend tuning of the combination of the objectives depending on the expected amount of occlusions. In future work, we will investigate alternative dense features to provide more robust pixel-level correspondences.

The proposed prediction of a joint position distribution provides samples that are sufficient to initialise tracking. But the discrete bin scores and their interpolated PDF are predicted and sampled independently which makes this sampling approach inefficient. Chapter 5 proposes an optimisation approach that consistently maintains a state distribution and resamples complete states from a multivariate distribution.

Chapter 5

Multi-Hypotheses Tracking of Robotic Manipulators in Cluttered Scenes

5.1 Introduction

In previous chapters, we implicitly made the assumption that we know enough about the environment to be able to employ simple plane filtering for depth background removal (Section 3.4.2), and we assumed that once an initial configuration for the optimiser is found, it will converge from thereon to the optimal state. In this chapter, we are going to relax these assumptions and make the approach more robust and wider applicable to cluttered environments. This will be evaluated on a new dataset with a different robot model that, compared to the test sequences of previous chapters, has much stronger background clutter and a larger variety of visual distractions from objects.

The approach in Chapter 4 demonstrated the advantage of using multiple image modalities from colour and depth images and provided an initial approach for using multiple initial states to become independent of the proprioceptive initialisation of a gradient-based optimiser. Although the problem was simplified by not considering background as clutter, this increased the complexity of the method as it involved (1) a parallel pipeline for colour edge extraction (Section 4.3.1), (2) a dedicated prediction branch for the joint state distribution (Section 4.3.2), and (3) a dedicated sampling and selection stage to select the best initial sample from this distribution (Section 4.3.3).

To relax the assumption of a background model and the optimality of the initial optimiser state, we will assume in this chapter that no background model is available. The proposed tracking approach will therefore operate on the raw sensor

data without preprocessing and assume that a single initial optimiser state will not converge to the global optimum. To work under this more constrained scenario, the prediction and optimisation pipeline has to be more robust to cluttered background and non-optimal initialisation.

In the following, we propose to integrate colour and depth information into a data-driven objective to make use of this additional modality, and to move the tracking initialisation from the prediction network into the optimiser. While this reduces the complexity of the feature extraction and sample selection stage, this also increases the requirements on available training data to include colour images. In the absence of reliable proprioception for real RGB-D image ground truth, this chapter contributes an enhanced training data synthesis pipeline that only requires the tracked model, a set of generic background images and objects, and the bounds of the expected state space, to generate training data that simulates real properties and outperforms a manually labelled but smaller real training set. For tracker initialisation, this chapter further proposes an optimisation approach that maintains multiple state hypotheses and which is able to reliably and efficiently explore the search space to avoid converging at local minima.

5.2 Related Work

5.2.1 Synthetic Training Sets

Supervised learning and especially complex methods with a high capacity like deep learning, require large datasets of labelled data to generalise well to unseen test cases. Traditionally, large datasets like ImageNet [20], KITTI [28] and COCO [45] with a huge amount of labelled real images have served as the backbone of the machine learning research community. These standardised training sets are mainly used to compare the prediction performance of different networks and one has to gather task specific training data and labels to train a dedicated semantic feature extractor.

An automated data collection pipeline for robot configurations in the spirit of [24] would allow one to automatically cover a range of motions and gather RGB-D sequences, but would still require accurate proprioception and calibration. Additionally, it would still be difficult to cover dangerous states close to joint limits and collisions.

With the increasing quality of computer graphics, the collection of training data has shifted from manually labelled real data to synthetic rendered data that naturally contains perfect ground truth. Early work on this has focused on image properties that are easy to synthesise and that generalise well to real data

like depth sensors [71]. The large variety of other image properties caused by texture, lighting and the effect of material properties makes colour image synthesis much more difficult than depth image synthesis. Synthetic colour datasets have been used to train 3D hand pose estimation from RGB images [91] with 3D rendered hand models and randomised backgrounds, and for hand pose estimation in interaction with objects from RGB-D sequences [51] with additional randomised object textures.

It has been shown that training on real and synthetic data can achieve similar performance for segmentation of urban environments (cars, roads, pedestrians) similar to KITTI [25]. Furthermore, pre-training on synthetic data and fine-tuning on real data actually improves the performance over training on real data only. Pre-training on a large amount of synthetic images (5M images) can additionally outperform the same network trained on a smaller real dataset (ImageNet, 1M images) [50].

The requirements for synthetic datasets for the specific task of disparity and optical flow estimation are discussed in [49]. Their findings suggest that diversity, which can be achieved with synthetic datasets more easily, is important for generalisability and that realism is not important for performance improvements.

5.2.2 Optimisation with Multiple Hypotheses

Hybrid optimisation approaches that combine properties of particle-based and gradient-based approaches have been proposed as a meta-optimiser that alternates between converging from multiple initialisations and refining with gradients [62, 40], and as a variant that updates particles with gradient steps at every iteration [9, 31]. The combination of multi-hypotheses solvers with gradient update steps has not seen much application to the problem of articulated tracking in recent years. This is surprising, given their advantage to avoid local minima that are much more likely in articulated kinematic chains with large nullspaces.

Classic implementations of particle-based optimisation propagate particles based on their previous state (Particle Swarm Optimisation) or within the particle distribution (Particle Filter). For objectives with many local minima, this might lead to a multimodal distribution of particles. For robot manipulator tracking, we found that converged particles form a bimodal distribution and we therefore propose as in [70] to resample particles to shape the distribution towards the global optimum.

5.3 Method

5.3.1 Tracking Objective

RGB-D Feature Extraction

The tracking objectives in Chapter 4 – keypoints and edges – are directly derived from features in the 2D image plane. The line-of-sight in the 3D camera frame therefore provides the strongest cue in the x-y plane, while the z component is only constrained by the depth.

We reuse the keypoint based point-to-line objective (Section 4.3.5) because of its ability to directly associate the model with the observation and to enable tracking behind occlusions. In cluttered environments with many distracting edge responses in the background, the Canny edge objective (Section 4.3.5) is not applicable and we therefore only rely on the keypoint tracking to restrict the optimisation in the x-y space. To compensate for the missing colour-based objective, the colour image is used in addition to the depth image as input to the feature extraction network. To improve model-fitting in the z space of the camera frame, a secondary tracking objective based on the observed and estimated depth image is applied.

The network architecture (Figure 5.1) provides a common RGB-D feature extractor with two branches for the segmentation and keypoint heatmap targets. We use the second version of the MobileNet [65] as the feature extractor since it is much smaller (0.6M parameters) than the previously used ResNet (21.3M parameters) and thus provides a faster inference.

We built on an established network architecture as RGB-D feature extractor to benefit from the research in that area and to remove the need to independently evaluate a novel feature extractor. In the proposed multi-task architecture, the common RGB-D feature extractor can be exchanged by any trainable function $f(\mathbf{I}) : \mathbb{R}^{W \times H \times 4} \mapsto \mathbb{R}^{20 \times 15 \times N_D}$, without the need to adapt the task-specific hyperparameters.

The $N_D = 576$ feature maps extracted from the RGB-D input are individually upsampled by 4 transpose convolutions (0.9M parameters) to the original input resolution per target branch. The final convolutions use a target specific number of filters: The segmentation uses one filter for each of the three classes (background, robot and object), and the keypoint localisation uses one filter per keypoint and an additional filter for the background heatmap. In total the proposed network has 2.47M parameters and inference takes 17.7ms (57Hz).

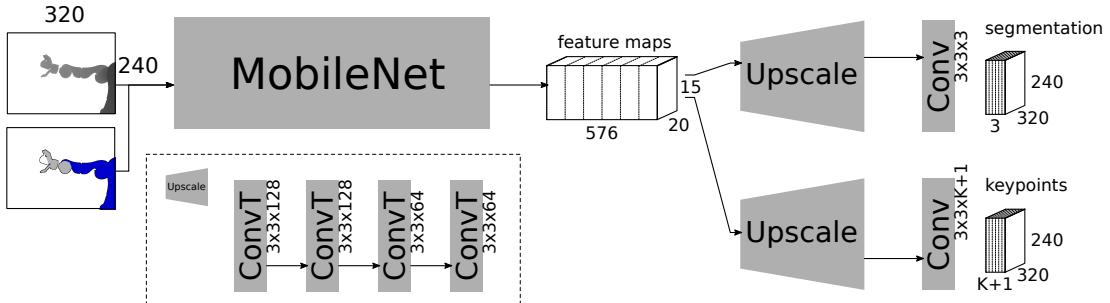


Figure 5.1: Network for segmentation and keypoint localisation. The segments and keypoints are used to provide correspondences between the tracked model and the observed image. The three predicted segments are background, robot and objects. Each of the K keypoints is represented by a 2D heatmap and absence of a keypoint is represented by an additional 2D background heatmap.

Similarly to the inference function in Chapter 4 (eq. 4.1), we formulate this two-branch predictor as

$$\{\mathbf{I}_S, \mathbf{I}_K\} = \text{extract}(\mathbf{I}_C, \mathbf{I}_D) \quad (5.1)$$

with 3-class segmentation $\mathbf{I}_S \in \mathbb{R}^{W \times H \times 3}$ and keypoint heatmaps $\mathbf{I}_K \in \mathbb{R}^{W \times H \times (N_K + 1)}$ predicted using a pair of colour and depth images $(\mathbf{I}_C, \mathbf{I}_D)$.

Keypoint Objective

This objective is identical to the previously used keypoint objective in Section 4.3.5. For consistency and completeness, we will elaborate further on this objective and provide details on how keypoint correspondences are established in the presence of occlusions.

Every 2D keypoint heatmap H_k provides a scalar score $s \in [0, 1]$ per pixel that represents the likelihood that a pixel is occupied by a keypoint. These heatmaps are created during training from a 2D Gaussian centred on the true 2D keypoint location, which is obtained by transforming the 3D keypoint \mathbf{p} from the mesh surface in the local frame to the camera frame and projecting it onto the image plane. A value of 1 corresponds to the true location of the keypoint and smaller values represent a measure of the distance of a pixel from the true keypoint location.

During inference, the pixel with the highest score s is assumed to be the 2D location of the keypoint, $x_k = \arg \max_s H_k$. The detected 2D location of a keypoint is back-projected to a 3D lines-of-sight \mathbf{l} in the camera frame and associated to 3D keypoints \mathbf{p} on the tracked model (Figure 5.2). Since the 3D keypoints are placed on the mesh surface, we can constrain the 3D line-of-sight

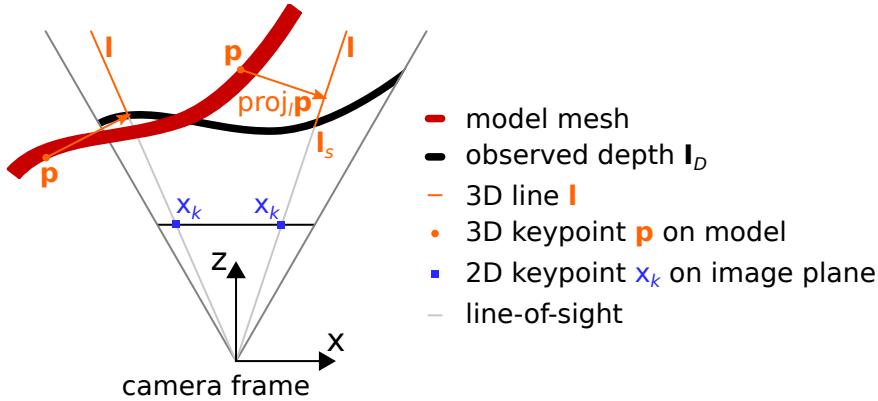


Figure 5.2: Keypoint objective. Predicted 2D keypoints (blue) are back-projected to line-of-sight rays (grey). The corresponding 3D keypoints (orange circle) are matched to the 3D lines beyond depth readings (orange line) and their minimal orthogonal distance (orange arrow) is minimised. This keypoint objective has the property that it can be established behind depth readings of occlusions.

to start at the depth reading. Still, a 3D keypoint might be occluded by an object in the scene (external occlusion) or it might be occluded by the robot itself (self occlusion) if the surface normal of the keypoint is facing away from the camera origin (see rightmost keypoint in Figure 5.2). An externally occluded keypoint is represented by a high heatmap score on occlusions, which could be recognised by the object class from the segmentation task, while a self occluded and a non-occluded keypoint will both be represented by a high score inside the area of the robot class. The keypoint objective covers both occlusion cases by associating a 3D keypoint p to its corresponding projection on line-of-sight l (eq. 4.11), instead of associating it directly to the back-projected point l_s . This enables us to track links behind occlusions, in which case the 2D keypoint will be located on an occlusion.

The keypoint objective is then to minimise the distances of 3D keypoints p_k to their corresponding projection on the lines-of-sight l_k ,

$$e_k(\theta, \mathbf{l}) = \sum_k^{N_K} \|\text{proj}_{l_k(\mathbf{l})} \mathbf{p}_k(\theta) - \mathbf{p}_k(\theta)\|^2 , \quad (5.2)$$

for all detected keypoints k . $\mathbf{p}_k(\theta)$ denotes the transformation of a 3D keypoint (compare Figure 4.4) from the link frame to the observation frame via the estimated state θ and $\text{proj}_{l_k(\mathbf{l})} \mathbf{p}_k(\theta)$ further denotes the projection of this transformed keypoint on the line-of-sight l_k , which is derived from the observed 2D keypoint location x_k .

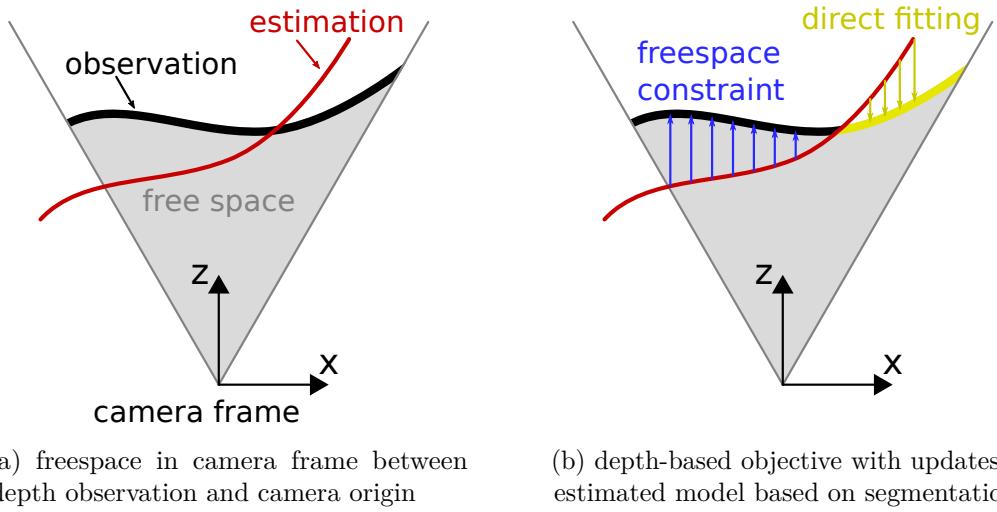


Figure 5.3: Depth-based freespace objective. (a) The location of an estimated visible robot link is constrained to be at or beyond the observed depth readings. (b) This information is used to update the model’s state to move outside of the free space in z-direction (blue), or the model can directly be fitted to the observation if it is known that an observed depth reading belongs to the tracked model (yellow).

Freespace Objective

In a 2D depth image, every pixel stores the z component of the 3D line-of-sight between the camera origin and any visible object in the camera frustum, or 0 for invalid readings. Any estimated state that intersects with these line-of-sight rays is therefore implausible [26]. We can use this information to evacuate the free space between the camera frustum and the observed depth readings by moving the estimated state beyond the depth readings (Figure 5.3).

Instead of computing a 3D signed distance function for the robot meshes as in [67], we propose to solely update the orthogonal distance to the image plane by pixel-wise comparison of the true observed depth image $\mathbf{I}_{D,obs}$ with the estimated depth image $\mathbf{I}_{D,est}$ of the rendered robot model. The signed depth distance $\mathbf{I}_{D,obs} - \mathbf{I}_{D,est}$ is positive if the estimated state is between the camera origin and the true observed depth, i.e. if it intersects with the lines-of-sight, and negative if the estimated model does not violate the freespace constraint.

In the general case of raw depth readings, we assume that the observed depth image also contains readings of untracked objects such as the background, manipulanda and occluders, whereas the estimated depth image only contains the rendered tracked objects. In this general case, only the positive distance (Figure 5.3b, blue) induces an update with respect to the z-direction in the camera frame to satisfy the freespace constraint. In the specific case where observed

pixels of tracked objects are known, we can additionally use the negative distance (Figure 5.3b, yellow) to directly minimise the distance of observed and estimated depth readings of tracked objects.

The objective is only defined for validly observed and estimated depth readings, hence it is only evaluated for the set of valid coordinates $X_v = \{x \mid \mathbf{I}_{D,est}(x) > 0 \wedge \mathbf{I}_{D,obs}(x) > 0\}$. The segmentation \mathbf{I}_S further provides a classification of pixel coordinates into a set of coordinates where the predicted class c matches the tracked object ID t , $X_{c=t} = \{x \mid \arg \max_c \mathbf{I}_S(x) = t\}$, and its disjoint set, $X_{c \neq t} = \{x \mid \arg \max_c \mathbf{I}_S(x) \neq t\}$. These distinct areas of the image plane provide different parts of the freespace objective formulation,

$$e_f(\theta, \mathbf{I}) = \sum_{x \in X_v \cap X_{c \neq t}} \max(0, \mathbf{I}_{D,obs}(x) - \mathbf{I}_{D,est}(\theta, x)) + \sum_{x \in X_v \cap X_{c=t}} |\mathbf{I}_{D,obs}(x) - \mathbf{I}_{D,est}(\theta, x)|, \quad (5.3)$$

to account for coordinates where only the positive distance is minimised ($c \neq t$), and coordinates where the absolute distance is minimised ($c = t$).

5.3.2 Training Data Generation

To generate large amounts of labelled training images without relying on proprioceptive sensing, we propose to synthesise colour and depth images by rendering the robot model at different configurations and in different environments. This has the advantage that we can sample a wider range of possible states than what is typically achieved with normal grasping tasks.

It is crucial that the synthetic training set and the real test set have similar visual properties to minimise the simulation-to-reality gap. For idealised depth images, the visual variety of the image data is dependent on the articulated geometric model, its state and the camera intrinsics. We assume that the model and the camera intrinsics are constant between training and test set. Hence, the variety of the depth image can be expressed by the 6D pose of the robot and its articulation of N joint positions, i.e. $6+N$ dimensions. For RGB colour images, this variety is greatly enlarged by textures and material properties, lighting conditions and the background environment.

Since we are unable to synthesise the real images through a perfectly accurate rendering process, we have to sample training images from a wider distribution around the real properties. However, this sampling of training images has a trade-off between potential overfitting to synthetic properties when sampling from

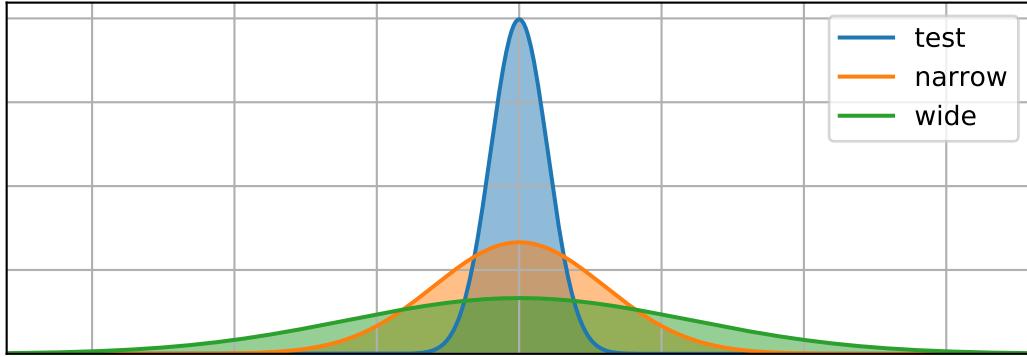


Figure 5.4: Illustration of sampling variance. The variance of kinematic states and visual appearance for real observations (test) is typically very small compared to the entire range of the colour and depth image domain. The training image synthesis has to sample around the real image properties. This is a trade-off between a narrow (orange) sampling with potential overfitting to synthetic properties and an inefficient too wide coverage of the visual appearance (green).

a narrow distribution, and an inefficient wide sampling of images that are too far away from the real images (Figure 5.4).

State Sampling

To efficiently sample training states that densely cover the test states, we propose to sample states within bounds of the expected articulated states during manipulation. This has the drawback that we need to have prior information about the test cases, but it is much more efficient than the naive approach of sampling uniformly from the entire state space of the articulated model. Hence, the training set is biased towards the test set to allow a dense sampling around the test distribution.

Instead of defining limits in the state space of the articulated model, where potentially multiple bounds are required as multiple joint states can map to the same end-effector pose, we propose to first define bounds and sample within those in the task space and then obtain joint states from inverse kinematics.

The task space bounds are defined by transformations between the camera frame and the grasp frame (Figure 5.5): The static transformation between the camera and base frame, and the articulated transformation from the base frame to the grasp frame of the end-effector.

The static camera transform is uniformly sampled within $\pm 5\text{cm}$ and $\pm 0.1\text{rad}$ of the corresponding transform in the test sequences. The translation part of the articulated transform between the base and the grasp frame is sampled by defining a box-volume in the base frame as the expected workspace. The rotation

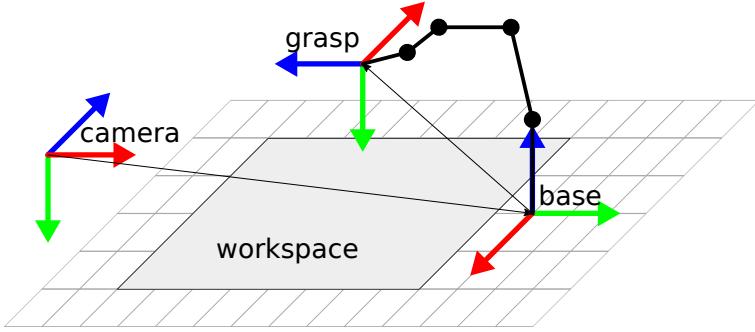


Figure 5.5: Static and articulated transformations in the proposed sampling setup. The camera frame is sampled with respect to the base frame within the bounds of the original transformation in the test sequences. The grasp frame is sampled within an expected workspace and orientation to yield the articulation with respect to the base frame.

part is sampled as a coordinate frame axes from a unit sphere within spherical coordinate system bounds (θ, ϕ) .

The final joint state of the model is obtained by IK on the sampled transformation between base and grasp frame. While the camera frame is not used for the state sampling, it is later used to add distractors within the camera frustum and occluders between the robot and the camera origin.

The coverage of the resulting set of training transformations from camera to grasp frame is visualised in Figure 5.6, alongside the transformations from all test sequences. The sampled frame positions (Figure 5.6a) cover the test positions within the box-volume sample bounds. Similarly, the orientation of the frame axes (Figure 5.6b), cover the test orientations within the selected spherical bounds. Since the task space poses are initially sampled without kinematic limits, and these joint space limits are enforced after the task space sampling, it can be seen that some samples are actually located outside of the original task space bounds.

As a consequence of sampling within defined bounds, a trained network can only be applied to test sequences with the same range of viewpoints and robot states. The real test sequences used for evaluation in later parts of this chapter (Figure 5.12) show the robot reaching from the right into the workspace and the camera view bounds have been chosen accordingly (Figure 5.5). The application to test cases with multiple or wider camera view ranges has to be reflected by sampling the training set from the same distribution of viewpoints and robot states to obtain an accurate and efficient overlap of the training and test distribution.

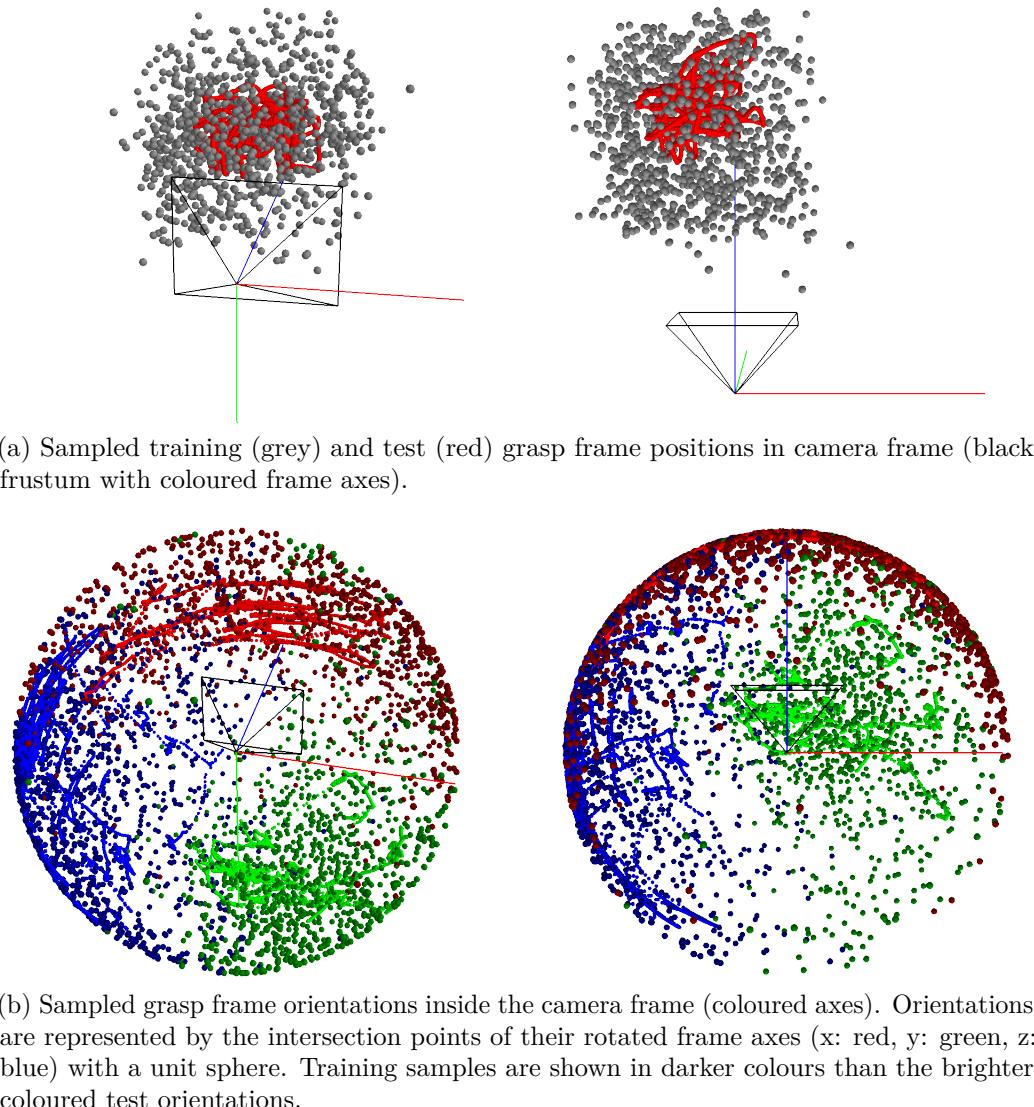


Figure 5.6: Distribution of transformations between camera and grasp frame in the training and test sets. Each sampled transformation is represented by its translation as single point in the camera position frame (a), and by 3 intersection points of its rotated frame axes with a unit sphere (b). The training set distribution (darker coloured points) covers the space of test positions and orientations (brighter coloured points) but also extends much further into areas outside of the test cases.

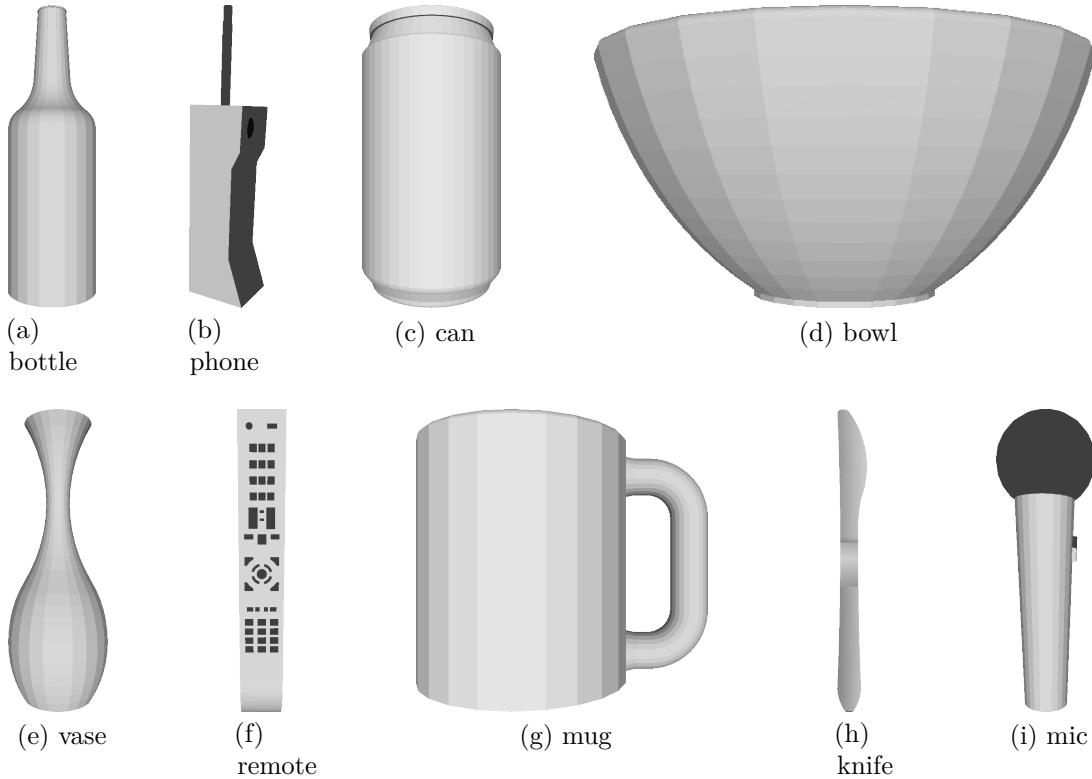


Figure 5.7: Subset of ShapeNet categories.

Image Domain Sampling

The second step in training data generation is the rendering, which transforms our sampled states into colour and depth image pairs. The meshes associated with the visible links of the robot model are transformed via FK on the sampled joint configuration to the camera frame at the sampled camera pose.

Objects To simulate manipulated and occluding objects, we extend the scene with randomly selected meshes from the ShapeNet 3D model database [15]. A subset of medium-sized, graspable, categories were selected (Figure 5.7), with 9735 objects in total. An object mesh that has been selected as manipulandum or occluder is transformed with a random orientation, scale and a position depending on the type. The occluder orientation is sampled from the full range and the position is sampled between camera origin and robot links. The manipulandum is placed at the origin of the grasp frame and orientated and scaled in such a way that the longest extend it vertically aligned, and the shortest extend fits the grasping width of the hand. In addition, the manipulandum transformation is randomly perturbed.

Environment The scene is rendered by transforming all meshes (robot parts and objects) with their sampled transform and 3D projecting the scene using the camera intrinsics of the calibrated sensor in the real tracking scenario.

The depth images are obtained by reading the z-buffer and setting invalid values to 0. For the colour images, the robot is rendered with its original texture as defined by the model. The objects are rendered with randomly selected textures. Further randomisation is added by varying the lighting conditions by randomly sampling light poses and diffuse and specular colours. The background area, which is masked by the initial invalid depth readings, is replaced by randomly sampled colour and depth image pairs (Figure 5.8). We do not add depth specific noise to the rendered robot and object, but the background will contain real noisy depth readings where available.

5.3.3 Optimiser

Derivation of Gradients

The combined keypoint (eq. 5.2) and freespace (eq. 5.3) objective is:

$$\begin{aligned} e(\theta, \mathbf{I}) &= e_k(\theta, \mathbf{I}) + e_f(\theta, \mathbf{I}) \\ &= \frac{1}{N_K} \sum_k^{N_K} \|\text{proj}_{\mathbf{l}_k(\mathbf{I})} \mathbf{p}_k(\theta) - \mathbf{p}_k(\theta)\|^2 \end{aligned} \quad (5.4)$$

$$+ \frac{1}{N_F} \sum_{x \in X_v \cap X_{c \neq t}} \max(0, \mathbf{I}_{D,obs}(x) - \mathbf{I}_{D,est}(\theta, x)) \quad (5.5)$$

$$+ \frac{1}{N_F} \sum_{x \in X_v \cap X_{c=t}} |\mathbf{I}_{D,obs}(x) - \mathbf{I}_{D,est}(\theta, x)| . \quad (5.6)$$

The gradients of this objective can be analytically derived and thus we intend to use a gradient-based solver to most efficiently traverse its state space.

The gradients for e_k (eq. 5.4) w.r.t. θ are directly given by the previously used point-line constraint (eq. 4.16).

The freespace objective e_f is piecewise defined as a positive distance (eq. 5.5) for image segments that are not associated with tracked parts, and as an absolute distance (eq. 5.6) for a direct association of tracked parts via the predicted segmentation.

The gradient of this task is derived from the standard position gradients from the kinematic Jacobian $\frac{\partial \text{FK}(\theta)}{\partial \theta}$. In particular, we want to minimise the distance \mathbf{d} between the two points \mathbf{p}_{obs} and \mathbf{p}_{est} , given by back-projecting x at depths $\mathbf{I}_{D,obs}(x)$ and $\mathbf{I}_{D,est}(x)$. For a single coordinate x in the image, the gradient is

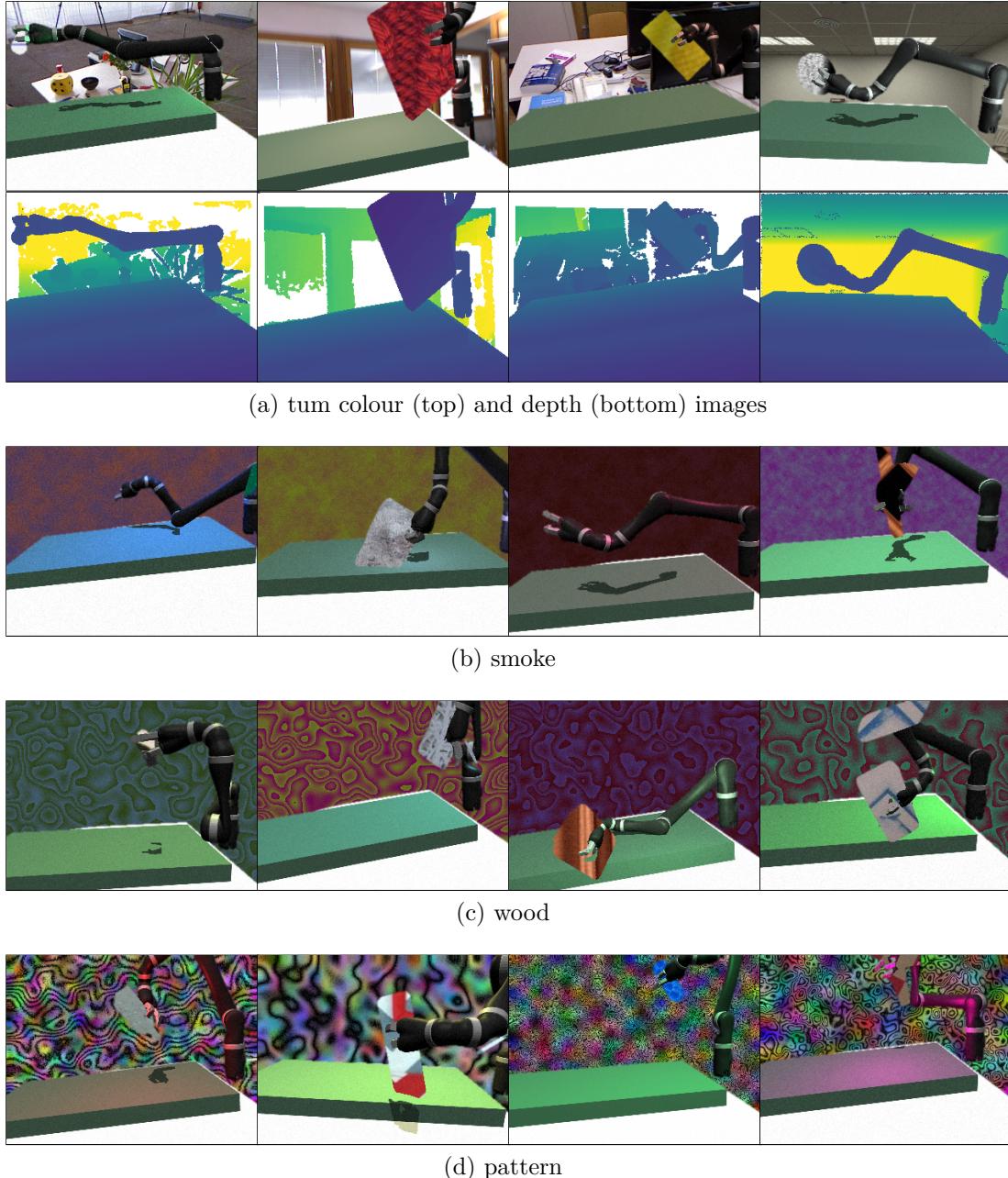


Figure 5.8: Synthesised training images. The articulated robot model is rendered with varying lighting conditions, randomised object models and textures, and real as well as synthetic background patterns. The *tum* background category contains real colour and depth images from the TUM RGB-D SLAM dataset [76]. The depth range from 0 to 4 metres is mapped to colour for visualisation.

defined as:

$$\frac{\partial \|\mathbf{d}_f\|}{\partial \theta} = \begin{cases} \frac{\partial \mathbf{d}}{\partial \theta} \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|} & \text{if } \mathbf{I}_{D,obs}(x) > \mathbf{I}_{D,est}(x) \vee x \in X_v \cap X_{c=t} \\ 0 & \text{otherwise} \end{cases} . \quad (5.7)$$

That is, the gradient is the standard point-to-point task if the estimated part is intersecting with the line-of-sight, or if that image region has been segmented as a tracked part. Otherwise, the estimated state is not updated by the freespace objective. This leads to a discontinuous objective function that combines a step function for the discrete segmentation and an inverted ramp function for the point distance. Similar to the changing associations in Section 3.3.2, this is mitigated by damping in the optimiser. The rendering of $\mathbf{I}_{D,est}$ also provides the association between the point-line distances and the individual links l .

The individual link-specific point-to-line objectives and their gradients:

$$\|\mathbf{d}_l\| = \frac{1}{N_{K,l}} \sum_{k_l}^{N_{K,l}} \|\mathbf{d}_{k_l}\| + \frac{1}{N_{F,l}} \sum_{f_l}^{N_{F,l}} \|\mathbf{d}_{f_l}\| \quad (5.8)$$

$$\frac{\partial \|\mathbf{d}_l\|}{\partial \theta} = \frac{1}{N_{K,l}} \sum_{k_l}^{N_{K,l}} \frac{\partial \|\mathbf{d}_{k_l}\|}{\partial \theta} + \frac{1}{N_{F,l}} \sum_{f_l}^{N_{F,l}} \frac{\partial \|\mathbf{d}_{f_l}\|}{\partial \theta} , \quad (5.9)$$

are combined and stacked into $\phi \in \mathbb{R}^{N_L}$ (eq. 4.20) and $J \in \mathbb{R}^{N_L \times N_J}$ (eq. 4.21).

Optimising Multiple Hypotheses

Due to the high degree of articulation and the visual similarity of states, the objective contains many local minima that do not coincide with the global optimum. Apart from initialising the optimiser close to the optimum, optimisers with multiple hypotheses have been proposed to explore the state space in parallel. Common approaches, like variations of the classical PSO, randomly sample the state space and update individual states using the global and local state of hypotheses. These approaches do not make use of gradients for the state space exploration and typically are motivated by using objectives that do not provide gradients. Hence, these approaches require a large quantity of hypotheses to explore the state space and find minima, and are also difficult to tune.

For this reason, and without the knowledge of an initial state close to the optimum, we propose to evaluate multiple hypotheses and update their search direction using the objective gradients. Tracking multiple potential model state configurations in parallel enables a wider exploration of the objective landscape and the discovery of multiple minima. Using gradients on the other hand enables a much more efficient update of the search direction of individual states.

Inspired by the re-randomisation of particles in [70], we resample states that converged to a local minimum that is not the global minimum. We assume that the distribution of converged states is bimodal (Figure 5.9), with one mode close to the global optimum and one mode taking the opposite kinematic configuration. A single-hypothesis gradient-based approach that takes small update steps will be unable to traverse from one minimum to another, without dramatically increasing the objective error. Moving a kinematic configuration from one minimum to its opposite configuration requires direct resampling.

State Update The structure of the multi-hypotheses optimisation algorithm is described in Algorithm 1. The state hypotheses of the estimated model are randomly initialised by sampling uniformly between joint limits (line 9). The diagonal matrix W (line 13) weights joints by their inverse geodesic distance to the root of the kinematic chain. This effectively penalises updates of joints closer to the root more to mitigate oscillation since those joints will also affect the objective error for all their child joints. The tracking operates continuously on a stream of RGB-D images. Once an image pair is received, the relevant features for the objective are extracted (eq. 5.1). Each estimated state is evaluated by the objective (line 21, incorporating eq. 5.8 and 5.9) on the extracted image features and updated via the Levenberg–Marquardt algorithm (line 25). The Levenberg–Marquardt algorithm uses an adaptive damping λ (line 23) that changes proportional to the objective function error to allow a fast exploration of the state space far way from the optimal solution and to prevent oscillation close to the optimal solution. The Heaviside step function $H(\cdot)$ is hereby used in the step function

$$2H(\Delta e) - 1 = \begin{cases} -1 & \text{if } \Delta e < 0 \\ +1 & \text{if } \Delta e \geq 0 \end{cases} \quad (5.10)$$

to increase or decrease the damping by a factor of 10 depending on the error convergence.

Resampling This approach alone would lead to the individual convergence to multiple local minima (Figure 5.9). To use global information about the objective, we resample states with a high objective error from the distribution of states with low objective error.

The resampling process only considers a subset S_c of states with a gradient updated step $\|\Delta\theta\|$ of less than $\epsilon = 0.05$ (line 27). This subset of converged states ensures that we only resample states in local minima.

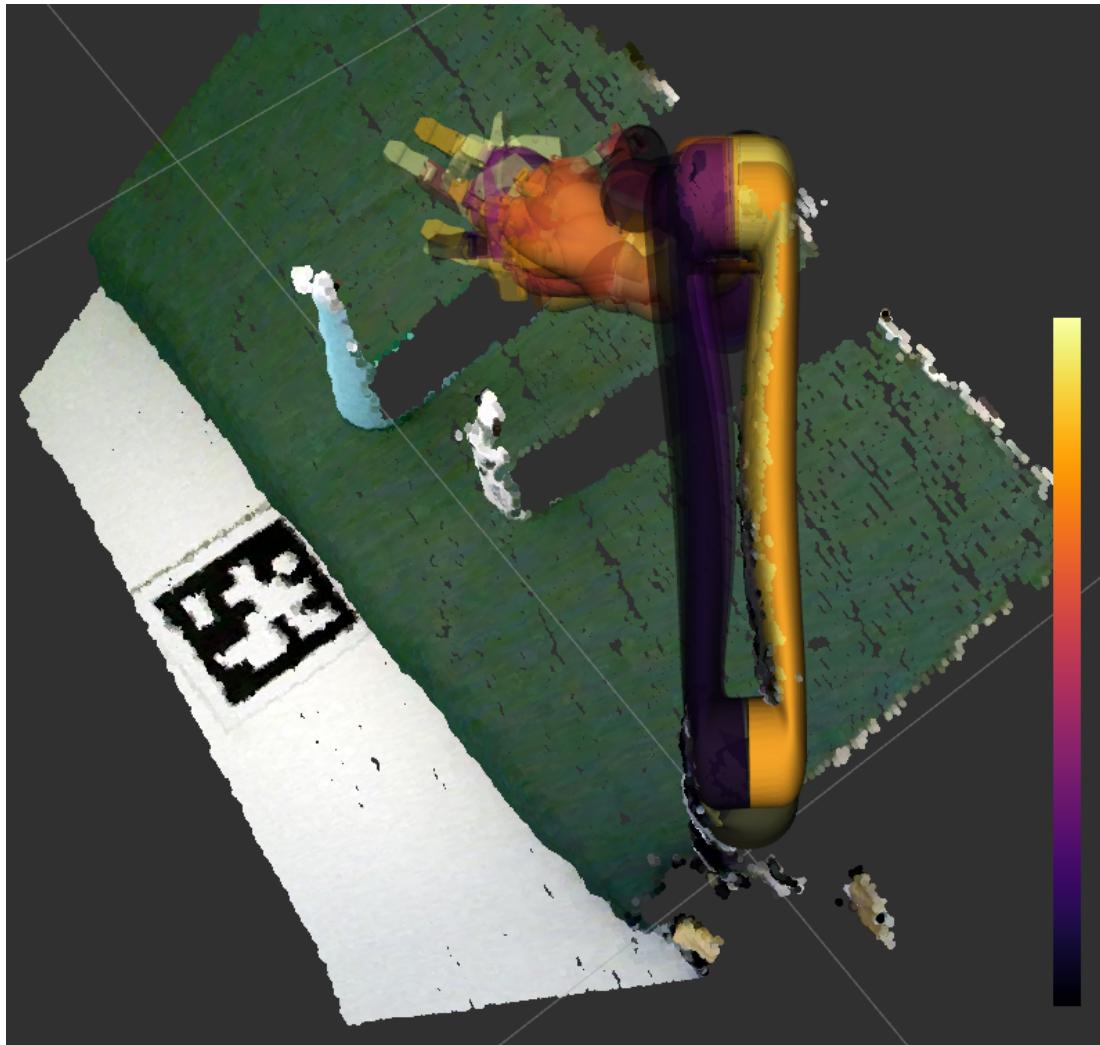


Figure 5.9: Bimodal distribution of converged hypotheses and their objective error (colour legend: bright colours correspond to a low objective error). The scene is observed from the left side of the image. The similar visual appearance of different kinematic configurations leads to two modes of hypotheses that take opposite kinematic configurations. *local minimum*: A set of hypotheses is formed in a minimum closer to the image plane (left side, black to purple colours). These states roughly align with observed keypoints in x-y space, but violate the freespace constraint. *global minimum*: A second set of hypotheses is formed in a minimum further away from the image plane (right side, orange to yellow colours). Hypotheses from this set align better with the observed depth of the robot arm.

The subset of converged states is further partitioned into disjoint sets S_- and S_+ using k -means clustering. The clustering is initialised at the $k = 2$ extrema points of S_c (line 28) and iterates until the cluster partitions do not change between consecutive iterations. The partitioning into two clusters is motivated by our bimodal assumption (Figure 5.9) and provides a set S_- of states with low objective error, which is assumed to represent the global minimum, and the set S_+ of states with high objective error, which is assumed to represent a local minimum.

The distribution of samples in S_- is represented as a multivariate normal distribution $\mathcal{N}(\mu_-, \Sigma_-)$ with the circular mean:

$$\mu_- = \text{atan2} \left(\frac{1}{|S_-|} \sum_{i=1}^{|S_-|} \sin(S_-), \frac{1}{|S_-|} \sum_{i=1}^{|S_-|} \cos(S_-) \right) \quad (5.11)$$

and the circular covariance matrix:

$$\Sigma_- = \frac{1}{|S_-| - 1} \sum_i ((S_- - \mu_-)^\top (S_- - \mu_-)) \quad . \quad (5.12)$$

The circular mean is used to represent the true statistical task space mean of the periodic joint positions. This representation maps the joint positions from their periodic interval to points on the unit circle, computes the arithmetic mean of these points and maps the result back to the periodic interval.

Samples in the local minimum S_+ are resampled from a mixture distribution (line 31): The arm joint positions are sampled from $\mathcal{N}(\mu_-, \Sigma_-)$ and the finger joint positions are sampled uniformly between their joint limits.

The cluster centre of the samples with the lowest objective error, μ_- , is also the solution of this iteration. Finally, the solution of multiple iterations is filtered with a mean-filter to yield the final estimated state, thereby using different filter sizes per joint dimension.

Algorithm 1 Optimiser

```

1:  $P \in \mathbb{N}_1$                                 ▷ number of hypotheses
2:  $D \in \mathbb{N}_1$                                 ▷ articulated dimensions
3:  $\theta \in \mathbb{R}^D$                                 ▷ articulated state
4:  $S \in \{\theta_p \mid p \in [0, P)\}$           ▷ set of optimised states
5:  $W \in \mathbb{R}^{D \times D}$                       ▷ joint space weights
6:  $T \in \mathbb{R}^{D \times D}$                       ▷ task space weights
7: procedure INITIALISE( $P, D$ )  $\rightarrow S$       ▷ randomly initialise states
8:   for  $p \in [0, P)$  do
9:      $\theta_p \leftarrow \{\mathcal{U}(\text{lower}(d), \text{upper}(d)) \mid d \in [0, D)\}$     ▷ sample within limits
10:     $S \leftarrow S \cup \{\theta_p\}$ 
11:   end for
12:    $L \leftarrow \max_{d \in D} |\text{kinematic\_chain}(0, d)|$     ▷ maximum length of kinematic chain
13:    $W \leftarrow \text{diag}(\{L - |\text{kinematic\_chain}(0, d)| \mid d \in [0, D)\})$ 
14: end procedure
15:  $\mathbf{I} \in \mathbb{R}^{W \times H \times 4}$                   ▷ observed RGB-D image
16:  $S_c \in \{\theta_c \mid c \in [0, P_c], P_c \leq P\} \Rightarrow S_c \subseteq S$     ▷ subset of converged states
17: procedure OPTIMISE( $\mathbf{I}$ )  $\rightarrow S$           ▷ optimise on single observation
18:   for  $i \in [0, N)$  do                      ▷ iteration
19:     for  $p \in [0, P)$  do
20:        $\lambda \in \mathbb{R}$                                 ▷ damping
21:        $\phi, J \leftarrow \text{objective}(\theta_p, \mathbf{I})$     ▷ model to observation association
22:        $e_i \leftarrow \phi^\top T \phi$ 
23:        $\lambda \leftarrow 10^{2H(e_i - e_{i-1}) - 1} \lambda$     ▷ adapt damping to error
24:        $J_W \leftarrow T J W$                           ▷ weighted gradients
25:        $\Delta\theta_p \leftarrow -(J_W^\top J_W + \lambda I)^{-1} J_W^\top T \phi$     ▷ Levenberg–Marquardt
26:     end for
27:      $S_c \leftarrow \{\theta_p \mid \|\Delta\theta_p\| < \epsilon\}$     ▷ create subset of converged states
28:      $\{S_-, S_+\} \leftarrow \text{kmeans}(S_c, \{\arg \min_{\theta \in S_c} e(\theta), \arg \max_{\theta \in S_c} e(\theta)\})$ 
29:      $\mu_- \leftarrow \text{mean}(S_-)$ 
30:      $\Sigma_- \leftarrow \text{covar}(S_-)$ 
31:      $S_r \leftarrow \{\theta_p \leftarrow \mathcal{N}(\mu_-, \Sigma_-) \mid \mathcal{U}(\text{lower}(d), \text{upper}(d)) \mid \theta_p \in S_+, d \in [0, D)\}$ 
32:      $S \leftarrow (S \setminus S_+) \cup S_r$                 ▷ replace resampled states
33:      $S \leftarrow \{\theta_p + \Delta\theta_p \mid \theta_p \in S\}$     ▷ update step for all samples
34:      $\hat{\theta}_i \leftarrow \mu_-$                             ▷ estimated state
35:   end for
36:    $\hat{\theta} \leftarrow \frac{1}{N_f} \sum_{f=1}^{N_f} \hat{\theta}_{i-f}$     ▷ mean filtering of state
37: end procedure

```

5.4 Evaluation

This section will evaluate the generalisability of the image synthesis pipeline and the multi-hypotheses tracking pipeline. Sections 5.4.1 to 5.4.3 provide an overview of the synthetic training and the real test setup.

The generalisability of the image synthesis pipeline is evaluated independently for varying input modality (Section 5.4.4), different background settings (Section 5.4.5), and for reduced prediction complexity (Section 5.4.6) with a network architecture like Figure 5.1 with an additional third branch for robot contour prediction. The optimiser and the tracking pipeline are evaluated in Section 5.4.7.

5.4.1 Platform

Our evaluation setup consists of a Kinova Jaco (version 1) robot that is observed by an Asus Xtion PRO LIVE RGB-D structured light sensor (Figure 5.10). Since the camera frame is not part of the kinematic chain, its pose is estimated using an AprilTag marker [58] fixed to the base frame of the robot.

The Jaco robot is a 6 DoF arm with a three finger hand with 2 DoF each (12 articulated dimensions in total). The proximal and distal phalanges of a single finger are actuated together by a single tendon and cannot be actuated individually. The fingers adapt to the shape of the grasped object and their state can therefore not directly be sensed through the tendon state.

In contrast to industrial-grade robots, like the KUKA LWR4, the Jaco is a commodity assistive robot that is meant to be operated close to human operators using a simple end-effector pose controller. It is therefore very lightweight (5.2kg) and compliant, as the less rigid plastics will give way when in contact with a human, but lacks the accurate proprioceptive sensing that is available on specialised industrial robots.

This lightweight design in combination with a lack of accurate proprioception leads a discrepancy between the internally and externally sensed state (Figure 5.11), which makes it necessary to manually label images for quantitative evaluation.

5.4.2 Tracking Sequences

We collected test sequences with varying degrees of distractions from objects in the same cluttered environment (Figure 5.12, Table 5.1). The background and objects in the environment are not part of the synthetic training set (see Figure 5.8). Out of these eight test sequences, two show grasping without occlusion, five show grasping with additional occlusion and one sequence does not show distractions from objects at all.

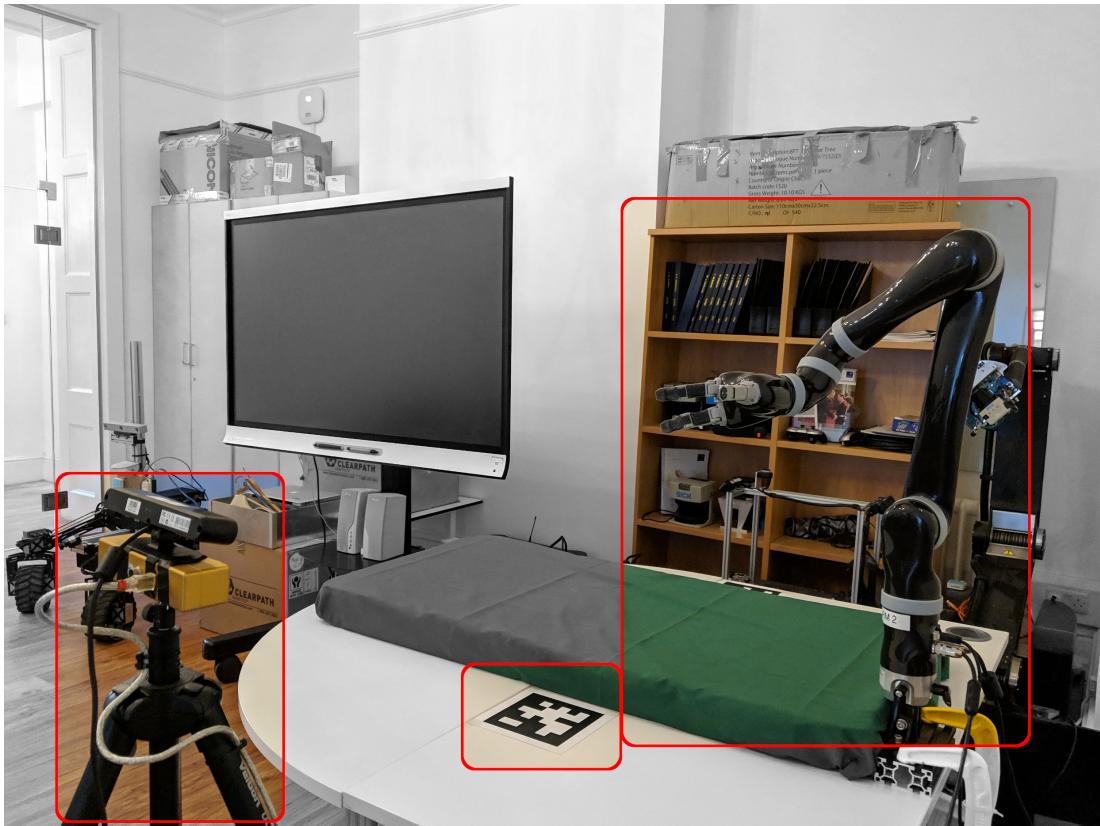


Figure 5.10: Experimental setup with Jaco robot mounted on table (right), Asus Xtion PRO LIVE structured light RGB-D sensor mounted on tripod (left) and an AprilTag marker for camera pose estimation fixed on the table (middle).

sequence	manipulandum	occluder	labelled	duration (s)
gspray	✓			37
gtorch	✓			38
jaco			✓	98
oalu	✓	✓		63
oball_bowl	✓	✓		95
oball_pot	✓	✓	✓	90
ospray	✓	✓	✓	81
owbottle	✓	✓	✓	49

Table 5.1: Test sequence properties.

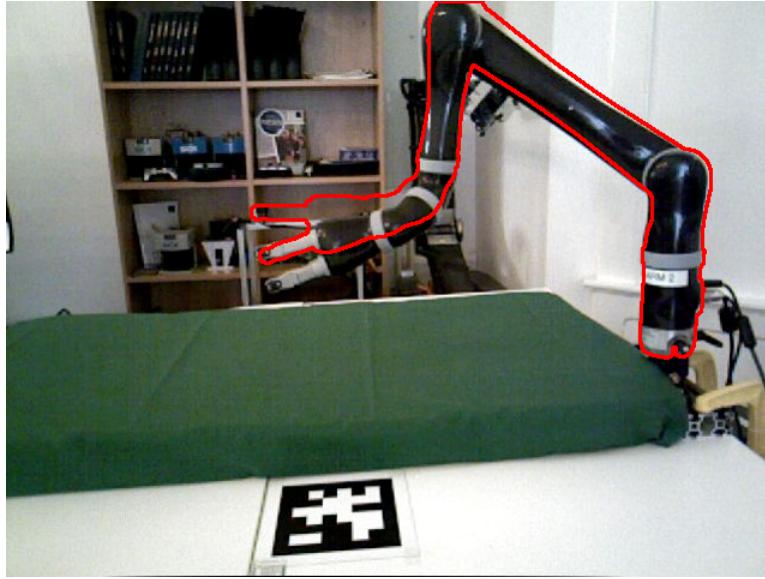


Figure 5.11: Effect of linkage bending. The robot model contours, as reported by FK using the joint encoders, overlaid on top of the observed image, show a significant discrepancy between the internally and externally sensed palm pose. While the internal representation only relies on the joint encoders, the external representation is affected by gravity that pulls the links downwards. Due to the lever effect, this is more prominent the further away a link is from the base frame.

Four of these sequences have been manually labelled (Figure 5.13) to provide ground truth for quantitative evaluation of the tracking error and for evaluation of the generalisability of synthetic training.

5.4.3 Background Image Synthesis

To evaluate the ability to generalise from a synthetic training set to a real test set, we will compare the effect of different background types on the prediction performance on the real test sequences. The background images are only used during training and do not reflect the true background in the real test sequences.

We generate synthetic colour backgrounds with different settings for the colour and structure variation (Figure 5.14). These four categories have been chosen for their distinct colour and texture properties:

tum realistic colour structure of objects in a scene, provides additional depth

smoke smooth texture with gradual change between two colours

wood strong edges and colour changes as proposed in [38]

pattern extreme change and variation of colour and texture

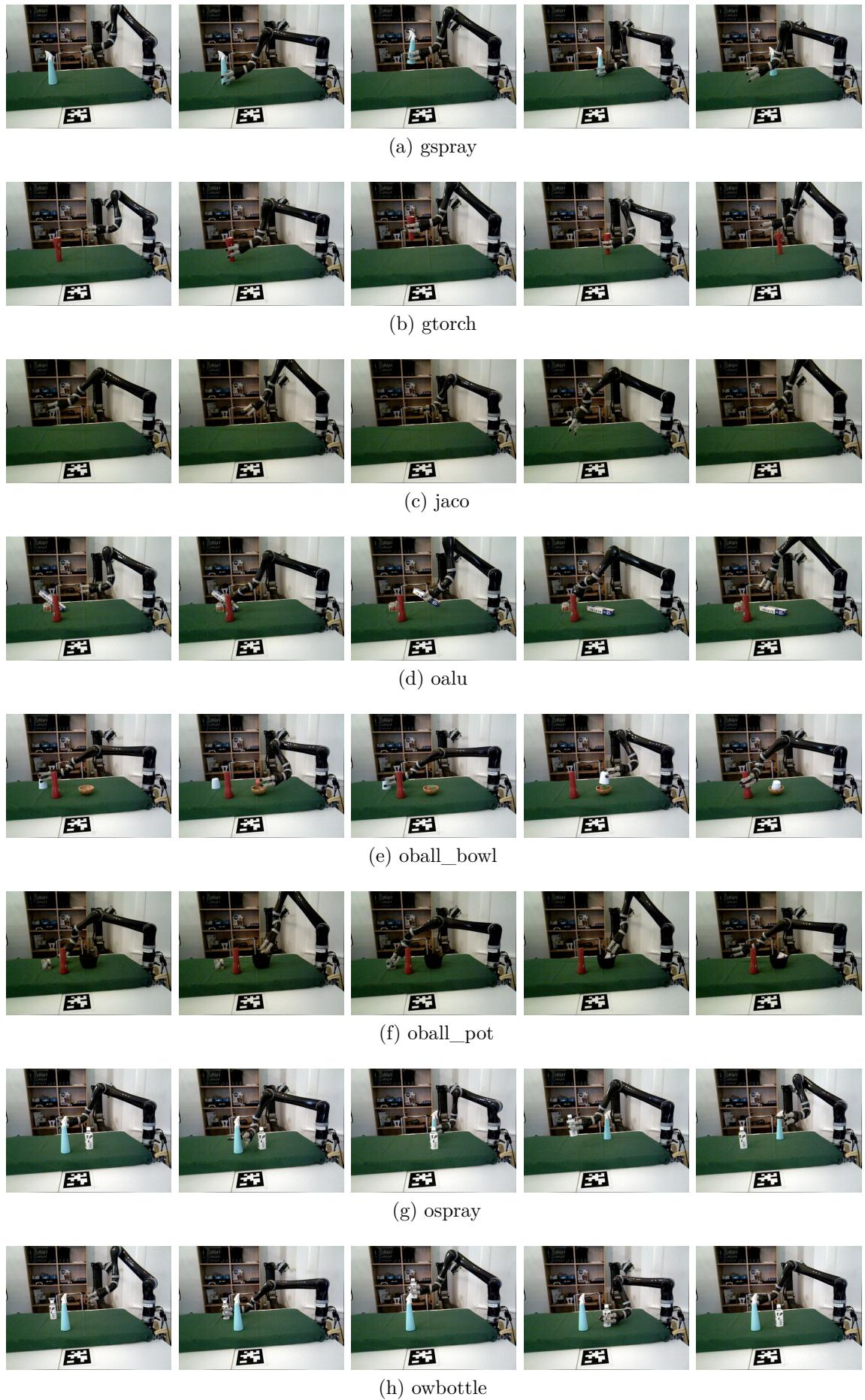


Figure 5.12: Test sequences with varying degrees of distractions.

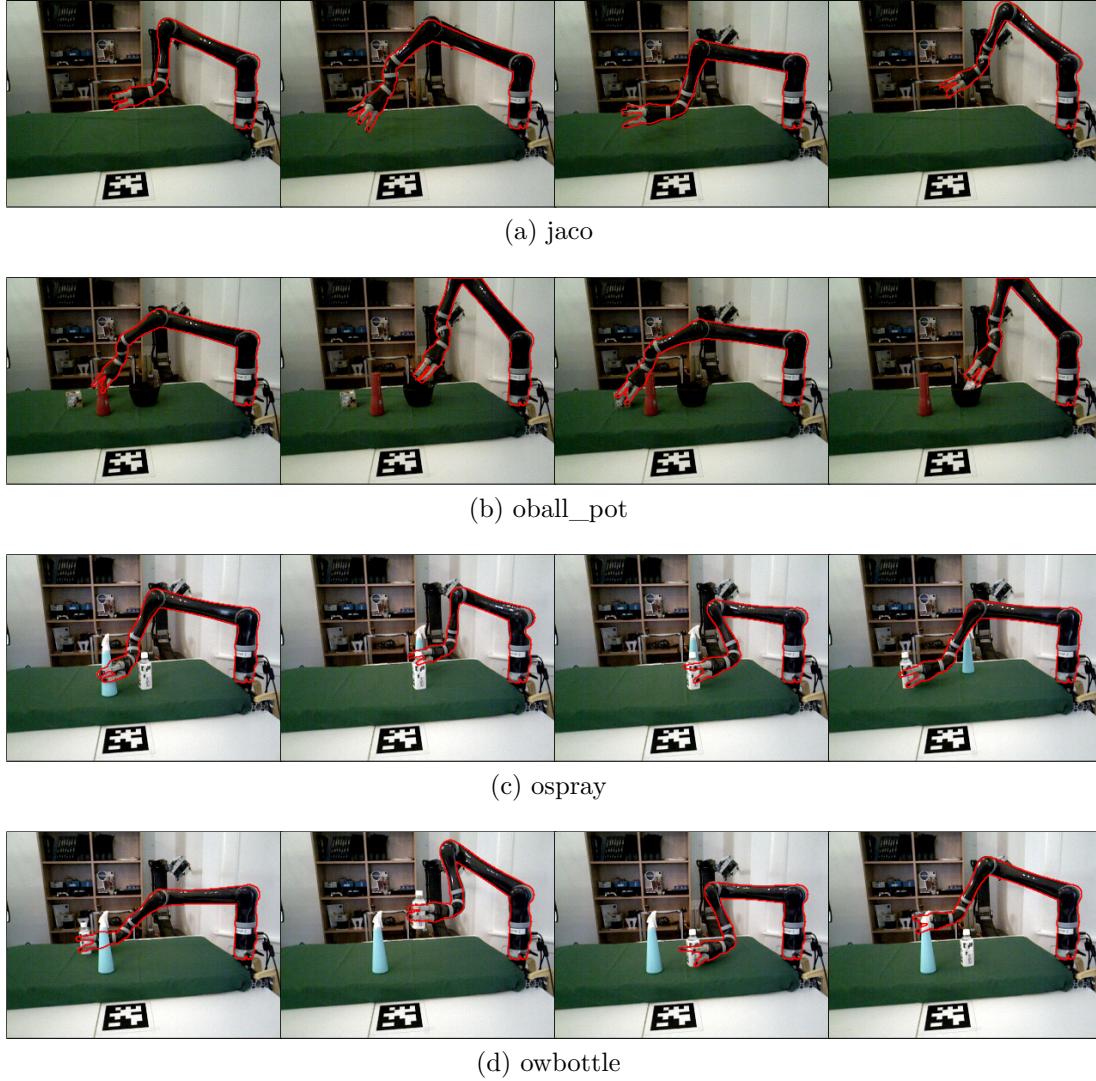


Figure 5.13: Manually labelled real image sequences for quantitative evaluation. The true robot state is overlayed in red contours on top of the colour image.

The *tum* set contains colour and depth image pairs from the synthetic SceneNet [50] and the real TUM-SLAM [76] RGB-D datasets. Originally, these datasets are used for benchmarking Simultaneous Localization and Mapping (SLAM) systems. They contain sequences of camera trajectories in room-sized environments and therefore resemble the environment of the test sequences.

The *wood* background pattern is inspired by [38], where it is used to train a mapping from an image and joint state sequence to joint velocity and gripper actions. While this work concludes that domain randomisation is required for generalisation to real data on a high-level task, it does not attempt to answer the question how realistic the background has to be to improve the actual prediction performance on real data. We use a very similar approach to generate the synthetic backgrounds *smoke*, *wood* and *pattern*, and the additional more realistic

tum background category to investigate their actual prediction performance and generalisability to real data.

5.4.4 Generalisability by Input Modality

The same network architecture is trained with 20k and 100k training images for 62500 iterations on colour-only, depth-only and RGB-D input with the *tum* background. Our hypotheses are (1) that a larger number of sampled states improves the generalisability to synthetic validation and real test sets, and (2) that the combination of colour and depth improves prediction results over using only one of the modalities.

We separately compare the synthetic training and validation loss (Figure 5.15) and its generalisation to the real test loss (Figure 5.16). For quantitative evaluation on real data, we used the manually labelled test sequences *jaco*, *oball_pot* and *ospray* (see overview in Figures 5.12c, 5.12f and 5.12g). Table 5.2 provides an additional overview and comparison of the training, validation and test loss at 62500 iterations.

Synthetic Validation Loss

For the comparison within the synthetic domain in Figure 5.15, we can make two main observations. First, training on the small training set (20k) generally yields a higher validation loss compared to a larger training set (100k). This indicates that it is beneficial to use a larger variety of rendered states. Second, using both modalities – colour and depth images – yields a lower validation loss, whereas depth-only input provides the highest validation loss.

An exception from this general observation is the segmentation and contour validation loss on the network trained with 20k depth-only images. In this setting, extending the depth image with colour information does not improve the prediction performance. These two targets are closely related since the contour of the robot segment class largely coincides with the contour target. The edges of the lower dimensional depth input also directly map to this contour prediction and robot segment contour. We therefore explain this behaviour with the simpler task to map from the lower dimensional depth images to the closely related contours. Adding colour information to this setup only provides redundant information about the contours and therefore impairs the prediction. In comparison, the keypoint heatmaps, which do not directly relate to the edges in the depth images benefit from additional colour information.

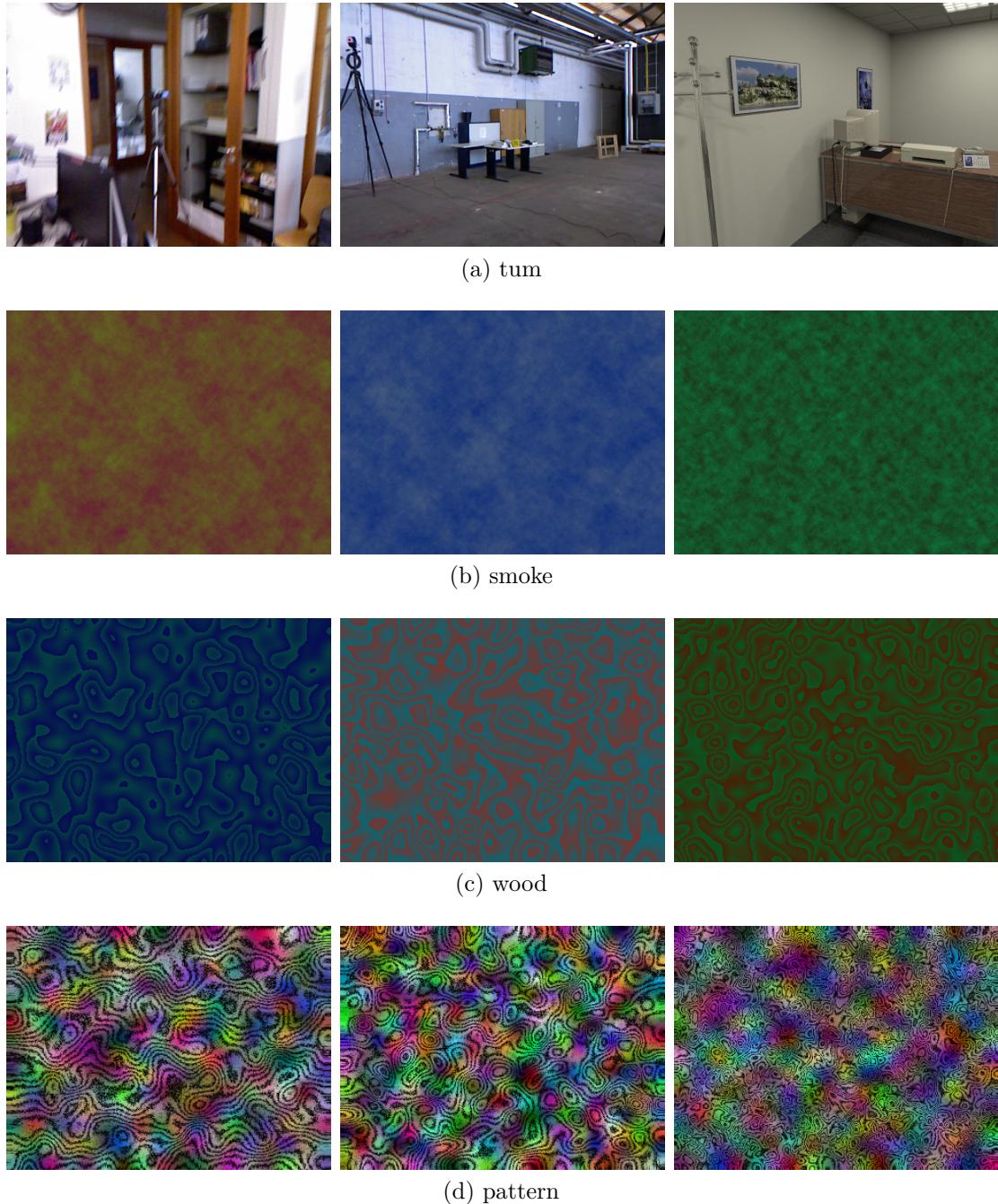


Figure 5.14: Example backgrounds from the RGB-D *tum* category and three synthetic categories of colour-only images with varying colour and texture properties.

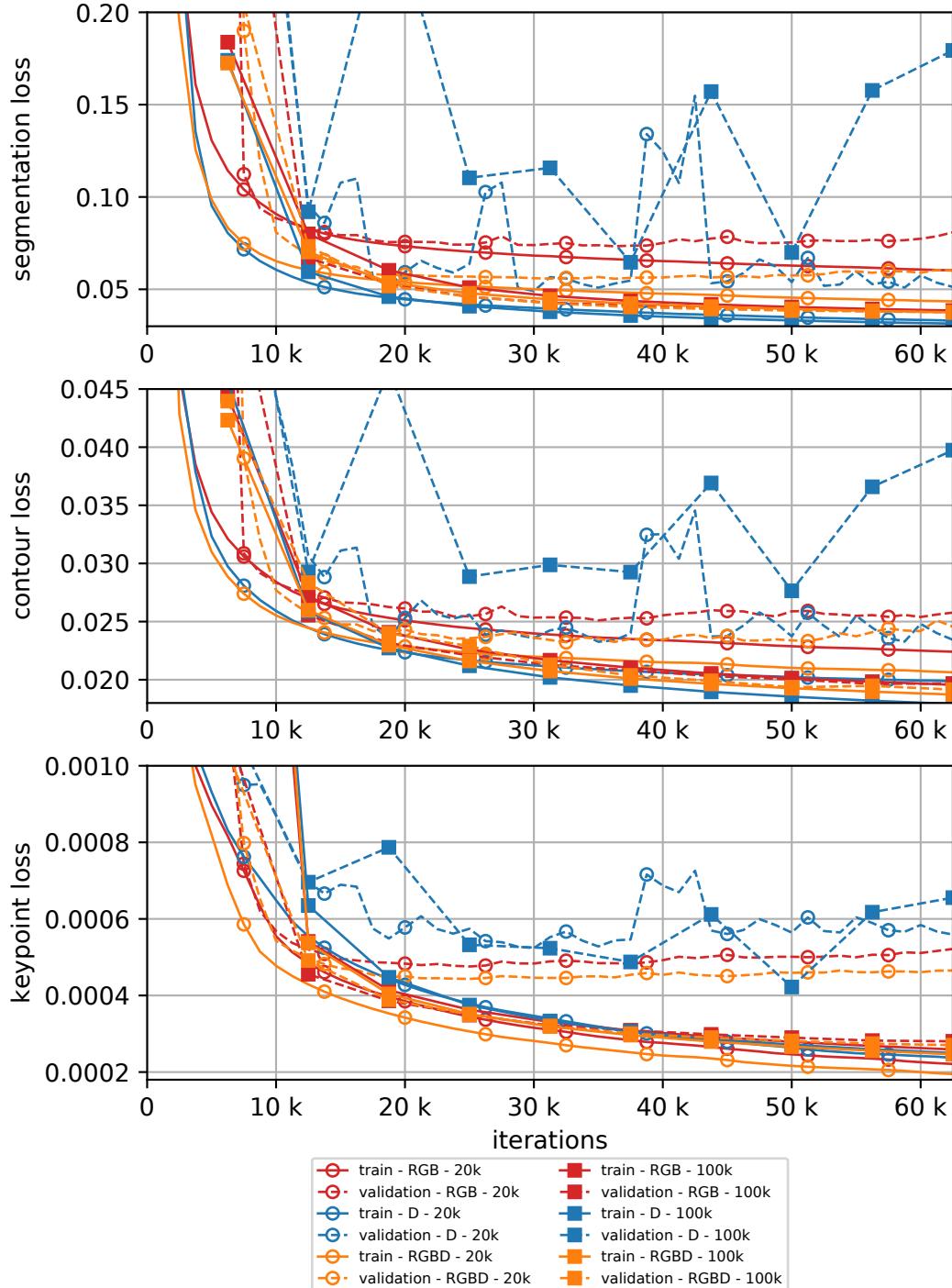


Figure 5.15: Training and validation loss on *synthetic* data for three prediction targets: segmentation, contours and keypoint heatmaps. The validation loss (dashed line) is generally higher for the smaller 20k dataset (circle marker). The combination of colour and depth images (RGBD, orange) provides better results than the individual colour (RGB, red) or depth (D, blue) inputs.

target	input	size	loss					
			synthetic		real			jaco
			training	validation				
segments	colour	20k	0.06029	0.08086	0.38616	0.41195	0.47835	ospray
		100k	0.03849	0.03825	0.93559	0.70842	1.08699	
	depth	20k	0.03305	0.05127	0.05570	0.08628	0.15513	oball_pot
		100k	0.03142	0.17935	0.32086	0.30464	0.40275	
	RGB-D	20k	0.04348	0.06009	0.11044	0.13599	0.16832	jaco
		100k	0.03752	0.03754	0.11190	0.11325	0.15591	
contour	colour	20k	0.02241	0.02577	0.05083	0.04815	0.05059	ospray
		100k	0.01956	0.01968	0.04687	0.04115	0.04634	
	depth	20k	0.01986	0.02343	0.04733	0.04227	0.05044	oball_pot
		100k	0.01792	0.03974	0.06381	0.05880	0.07319	
	RGB-D	20k	0.02064	0.02456	0.04893	0.04442	0.05167	jaco
		100k	0.01872	0.01915	0.03934	0.03389	0.04240	
keypoints	colour	20k	0.00022	0.00052	0.00146	0.00192	0.00151	ospray
		100k	0.00026	0.00028	0.00098	0.00082	0.00094	
	depth	20k	0.00024	0.00056	0.00053	0.00070	0.00092	oball_pot
		100k	0.00025	0.00066	0.00064	0.00066	0.00081	
	RGB-D	20k	0.00019	0.00047	0.00081	0.00088	0.00107	jaco
		100k	0.00025	0.00027	0.00055	0.00048	0.00073	

Table 5.2: Loss of trained models at 62500 iterations for synthetic training and validation set and for three real sequences. Lowest loss per training target is marked in bold. The model that is trained on the larger synthetic colour and depth dataset shows the smallest validation loss on the synthetic validation set, and the smallest test loss for the contour and keypoint targets. Training on synthetic colour-only images does not generalise well to real image sequences.

Real Test Loss

From the application of the synthetically trained network on real images (Figures 5.16 and 5.17), we can observe that the real test loss is overall higher than the synthetic validation loss. This demonstrates the before mentioned simulation-to-reality gap. It is also observable that the colour-only network is overfitting on the synthetic data and badly generalises to the real data for the segmentation and keypoint targets, while the depth-only network overall generalises better. This effect can be explained because the lower dimensional depth data is easier to synthesise than the higher dimensional colour data. Similarly to the synthetic validation loss, the small depth-only network performs better at the segmentation and contour prediction tasks than its larger counterpart.

Nevertheless, the network trained on the larger training set (100k) with both input modalities (RGB-D) provides the lowest loss and best generalisability overall compared to the individual input modalities. Figure 5.18 shows qualitative

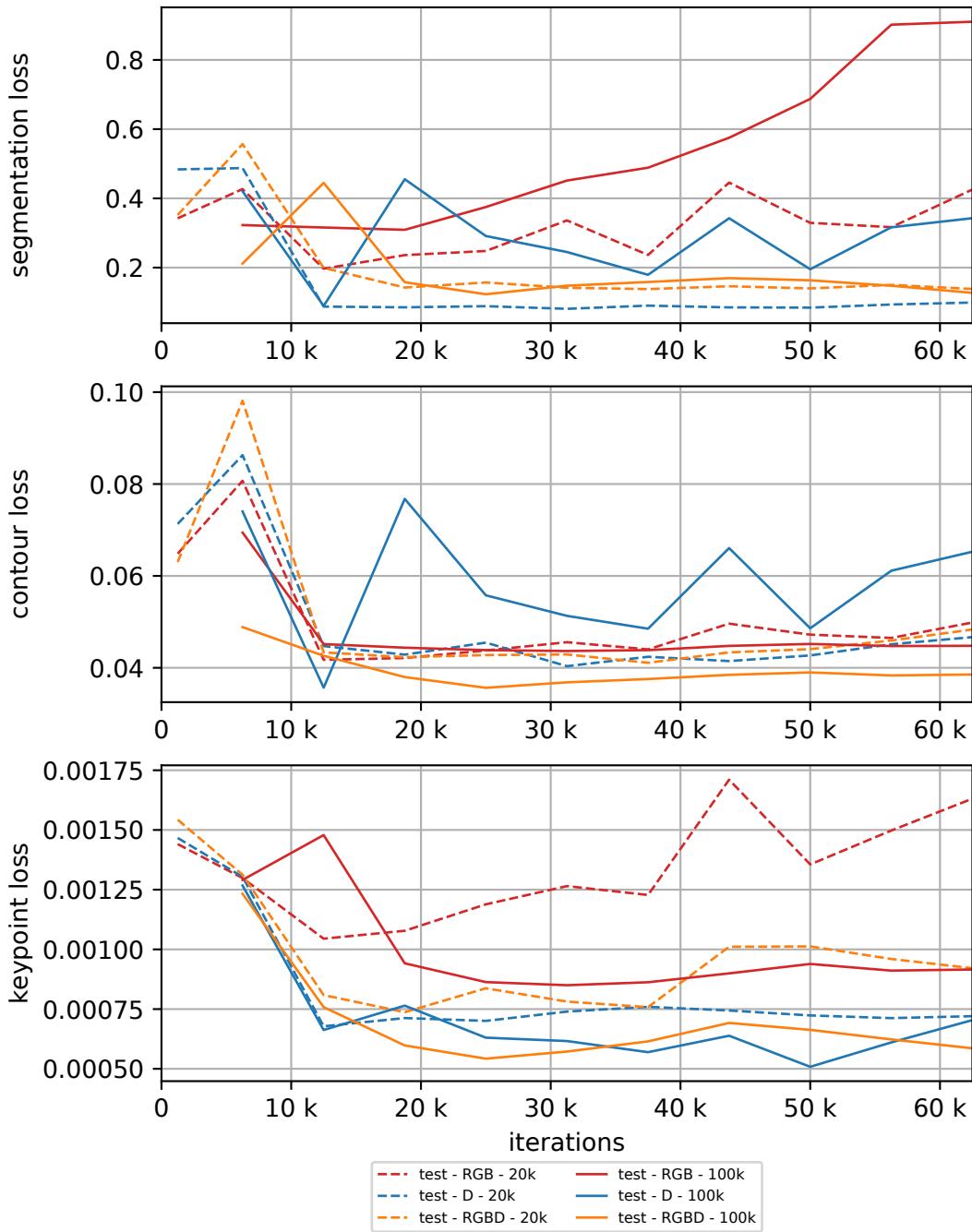


Figure 5.16: Average loss over three *real* test sequences. In comparison with the synthetic validation loss (Figure 5.15), this demonstrates the visual simulation-to-reality gap. The segmentation and keypoint prediction with colour-only input (red) is most complex to synthesise and generalises most poorly to the real test data. Overall, the large training set with a combination of colour and depth input (orange) provides the best prediction performance.

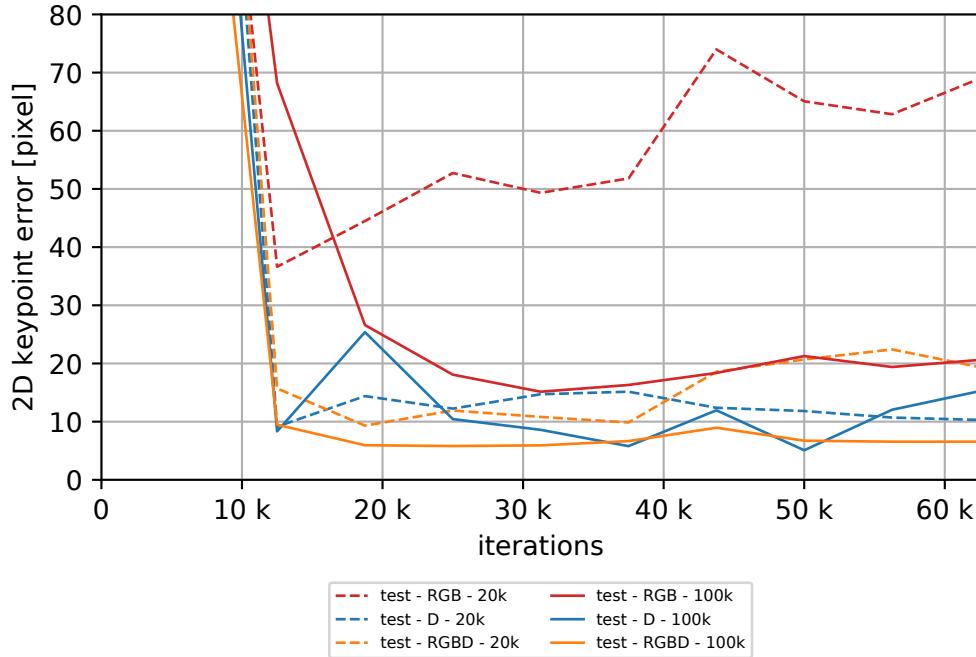


Figure 5.17: Average 2D keypoint localisation error for three *real* test sequences. Similar to Figure 5.16, the colour-only input (red) generalises most poorly and the largest training set with a combination of modalities (solid orange) provides the best results.

segmentation and keypoint prediction results for this trained network on real test sequences.

5.4.5 Generalisability by Background

The background covers a large area of the image data and therefore is crucial in the generalisation from synthetic to real data. To study the effect of different background settings, we train the same three-branch network architecture on the same 100k configurations of sampled states, objects and lighting, but replace the background with a colour image from one of the four background categories as described and shown in Section 5.4.3.

The comparison of the synthetic validation and real test loss in Figure 5.19 shows that while the synthetic validation loss is similar between background settings, it generalises differently to the real test sequences. It can be observed that the randomised texture categories *pattern*, *smoke* and *wood* yield the highest loss and that the realistically structured scenes in the *tum* category yield the lowest loss and therefore better generalise to the real test sequences. This behaviour is consistent across the three test sequences.

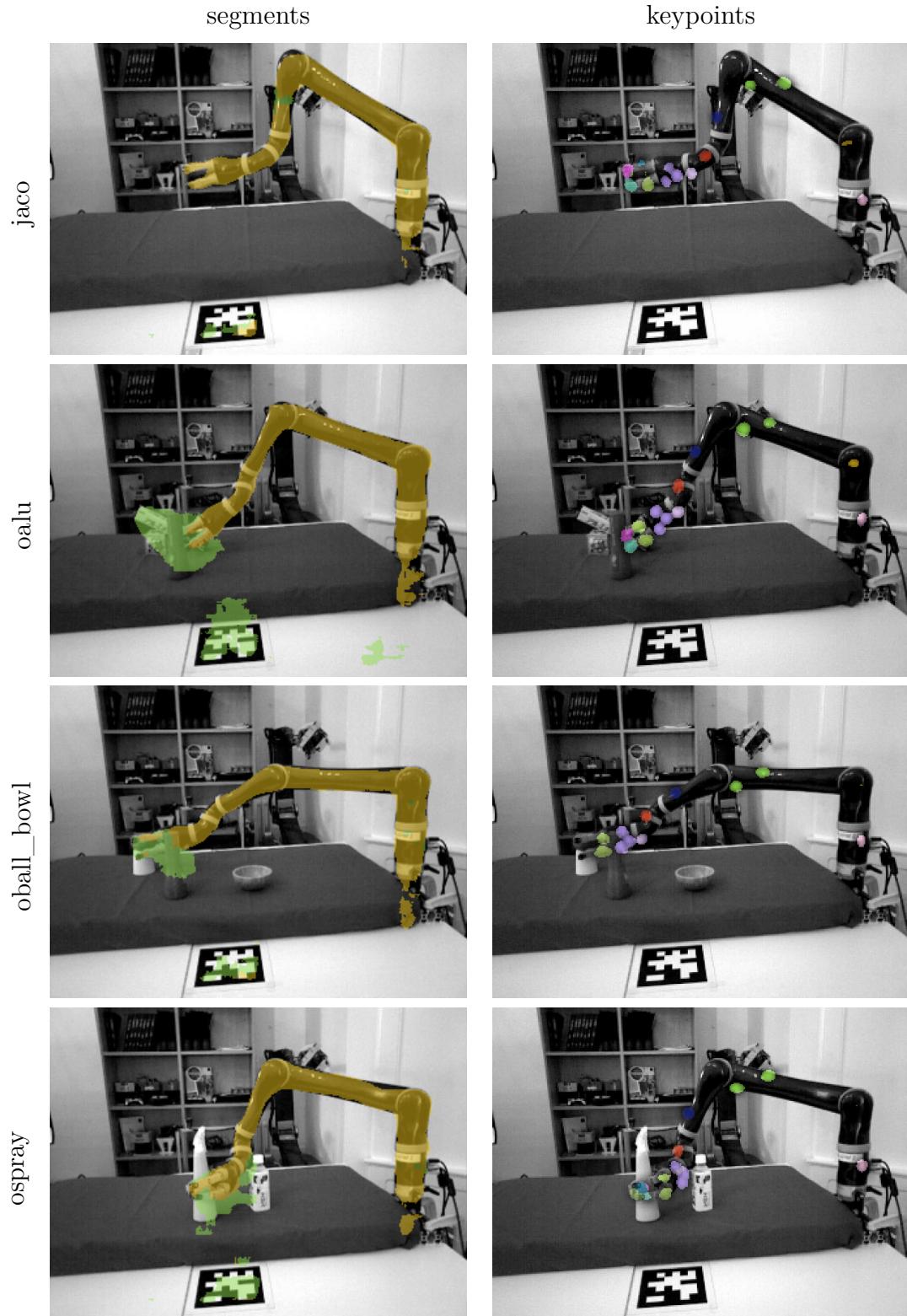


Figure 5.18: Prediction results for selected sequence snapshots. **Left:** Segments for robot (yellow), object (green) and background. **Right:** Keypoint score $s > 0.5$, with colours identifying the individual link.

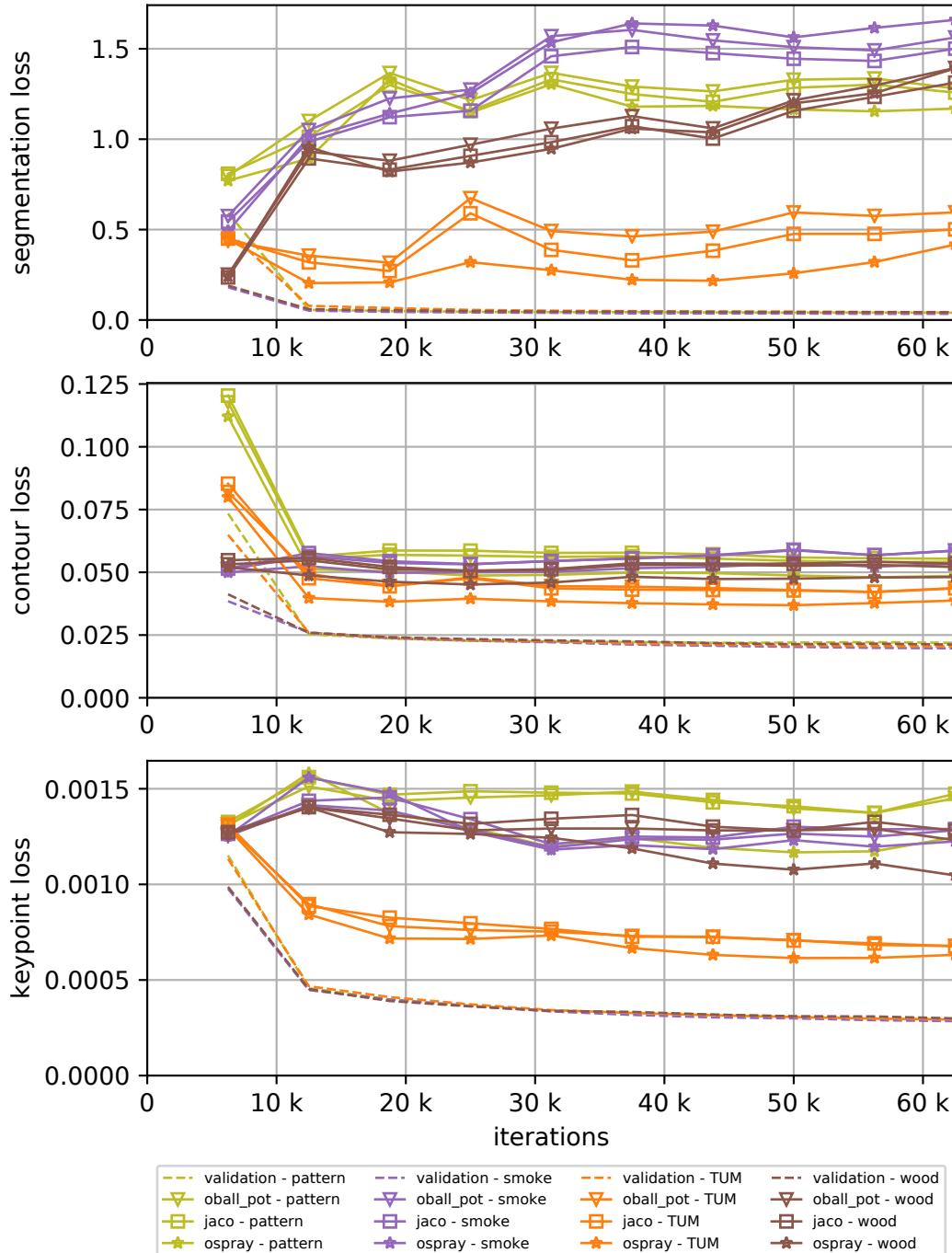


Figure 5.19: Comparison of synthetic validation loss (dashed lines) and real test loss (solid lines) for four different colour background categories (individual line colours). The validation loss is closer to the training loss since the validation set contains the same synthetic background images as the training set. The real test sequences show a significant gap in the loss since the original real background is not part of the training set. The realistic *tum* background (orange) in the training set generalises best to the real test sequences.

We conclude from this that synthetic training data has to contain similar visual colour and textural properties, like those found in real image data, to achieve generalisability of the trained model. Randomising background patterns from a wide distribution of properties harms the generalisation to real data.

5.4.6 Reduced Prediction Complexity

The pipeline presented in Figure 5.1 operates on the full FoV of the camera and predicts the full range of segments and keypoints for the tracking objective. This has the disadvantage that the network needs to process parts at different scales which has implications on the probability of being occluded and the density of keypoints. It follows that small segments, such as the palm and fingers, only cover a fraction of pixels in the image and additionally have a higher density of keypoints. This is a trade-off between the computational performance requirements, related to the input dimensionality, and being able to process all parts at once within a single scene, potentially benefiting from context amongst these parts.

We hypothesise that the dedicated emphasis on small scale parts and the reduction of the prediction targets reduces the task complexity and hence improves the prediction performance in this limited scope. We are specifically interested in improving the keypoint localisation for the palm and fingers by reducing the dimensionality and variety of the input data.

Input and Output Reduction

To analyse this effect, we will vary the input and output dimensionality of the baseline network. Specifically, we reduce the input FoV from the full scene at 320×240 pixel resolution to a 128×128 pixel ROI, $\mathbf{I} \in \mathbb{R}^{128 \times 128 \times 4}$, covering the palm and fingers, as well as any manipulanda and occluders during grasping. This ROI is extracted from the original 640×480 pixel resolution image and hence shows palm and fingers at double resolution of the full FoV image. The dimensionality of the prediction targets is reduced to a single branch for the 3 palm and 6 finger keypoint heatmaps.

The visual variety within the input data can further be reduced by using real sequences from the same environment as the training set. We compare the synthetic 100k baseline training set with a real training set sampled from the labelled sequences *jaco*, *ospray* and *oball_pot* (4027 samples in total) with additional spatial data augmentation via random translation, rotation and flipping.

We evaluate the variation in these settings (input region, prediction target, image source) via the average 2D keypoint localisation error on the additional independently labelled real sequence *owbottle*. The keypoint errors are averaged

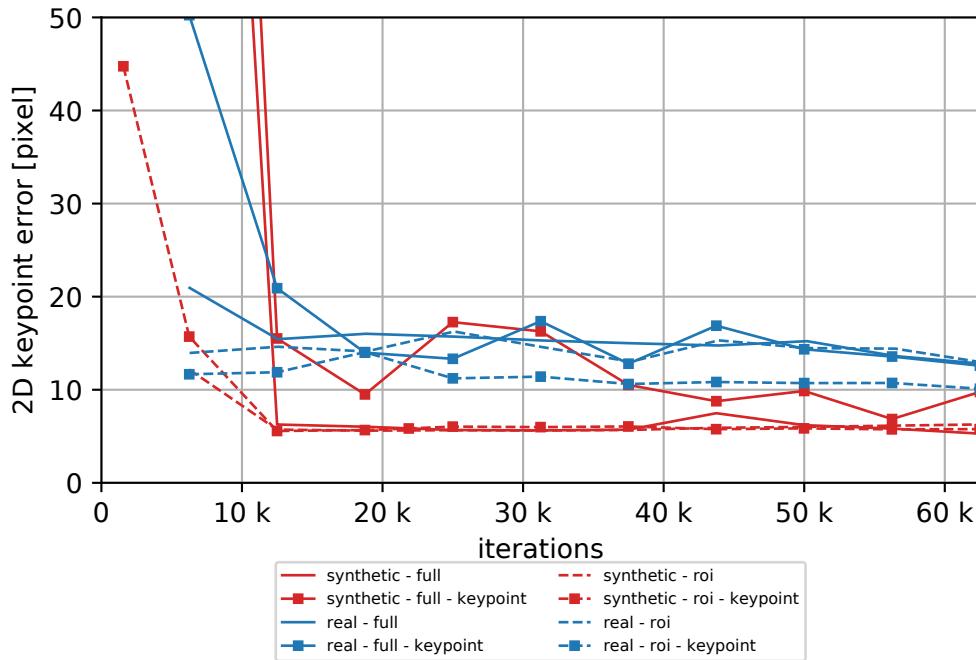


Figure 5.20: Average 2D palm and finger keypoint localisation error on *owbottle* sequence. Overall, the synthetic training set (red) yields lower keypoint errors than the real training set (blue). Constraining the input to a higher resolution area of the hand ROI (dashed line) only improves the keypoint localisation when exclusively predicting these hand keypoints (squared marker).

over the palm and finger keypoints. The pixel distances within the ROI are scaled by 0.5 per dimension to yield the same extent among the robot parts as in the full FoV images, i.e. the extent of a part will have the same pixel size in the 128×128 ROI and the full 320×240 FoV image.

Keypoint Predicton Performance

From the comparison in Figure 5.20 we can observe that, firstly, the synthetic training set yields a lower keypoint prediction error than the corresponding settings for the real training set. Secondly, the emphasis on the hand ROI only improves the keypoint localisation in cases where the prediction task is limited to these keypoint heatmaps.

In conclusion, we argue that a large training set with a wide coverage of the simulated visual and kinematic state space provides a better generalisation to a real test sequence, than a smaller training set with a tight coverage of the real properties. Further, the reduction in complexity of the prediction task by decreasing the input as well the target output dimension only shows an improvement on real training data. Forcing the network to learn from a large variety and dimensionality of

input to predict multiple related targets outperforms any attempts to reduce the complexity of the task. Neither the reduction of the input size (from full FoV to hand ROI) and variety (wide synthetic to tight real) nor the reduced dimension of prediction targets (multiple branches to hand keypoint heatmaps) improves the keypoint localisation compared to the baseline with full input and output dimensionality.

5.4.7 Optimiser

The complete tracking pipeline, that combines the extraction of semantic keypoints and segments from RGB-D images for the keypoint and freespace tracking objective (Section 5.3.1) and the multiple-hypotheses optimiser (Section 5.3.3) is evaluated via the pose tracking error, where labelled ground truth is available, and via the depth synthesis error against the reported state.

Pose Tracking Error

As in the previous chapters, the pose tracking error is defined as the transformation that needs to be applied to the estimated forearm and palm pose, to obtain the labelled reference pose. The estimated state is then the filtered mean of the estimated distribution (Algorithm 1, line 36). The pose tracking error is evaluated on the labelled sequences *jaco*, *oball_pot*, *ospray* and *owbottle*.

The main property of the proposed optimiser is the use of multiple parallel hypotheses and their resampling to avoid getting stuck at local minima. To evaluate the robustness of parallel tracking and resampling, we run the optimiser with three different amounts of hypotheses and 11 seeds. By running the tracking with different seeds for the initialisation of hypotheses, we are able to evaluate the robustness of convergence to the global optimum.

Figures 5.21 to 5.24 show the distribution of pose tracking errors for the four labelled sequences, as their position and orientation components, with the mean and standard deviation of this distribution as solid line and translucent area. A high standard deviation typically indicates that the distribution of states has not converged yet and that there are kinematically opposite configurations within the distribution. This effect is particularly observable for the forearm pose with only 2 hypotheses (Figures 5.21 to 5.24, blue shaded area in the bottom left plot). Convergence of the full distribution on these four sequences is only achieved by 5 or more hypotheses.

Figure 5.25 additionally shows how the average tracking performance improves with an increasing amount of hypotheses. While the palm position distribution converges to 2 to 3 cm with only 2 hypotheses, the before mentioned bimodal

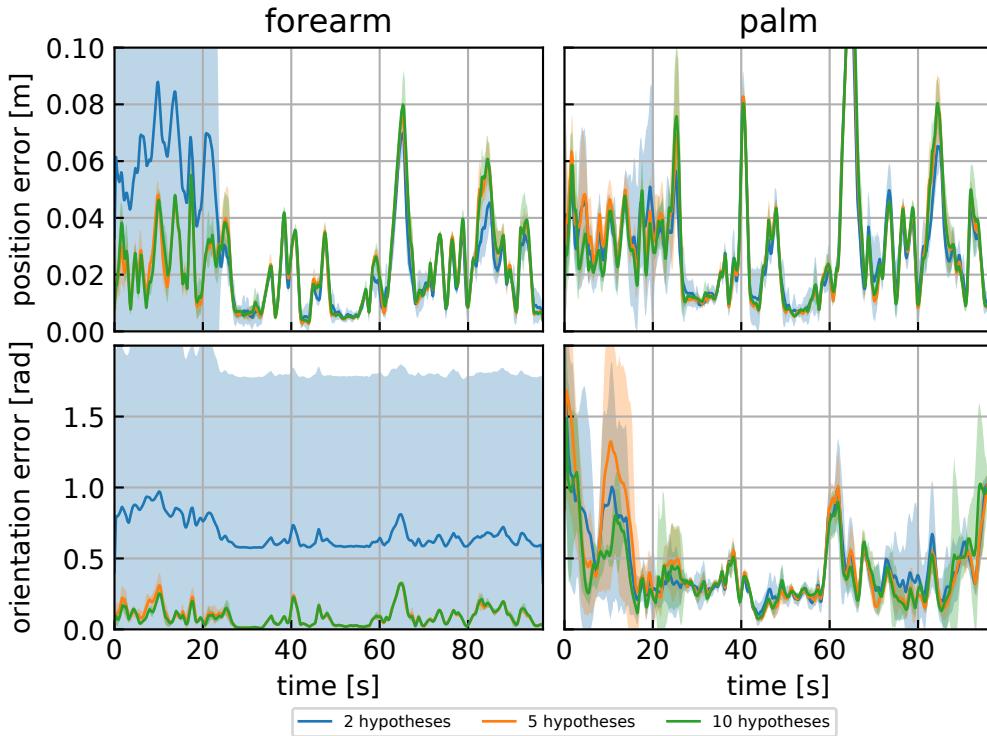


Figure 5.21: Pose tracking error distribution on *jaco* sequence.

distribution of configurations is indicated by large forearm orientation error with 2 hypotheses. This bimodal distribution can be explained by the local minima in the objective function (Figure 5.9). Smaller sets of hypotheses are more likely to get stuck entirely in a single local minimum, without the ability to make use of the proposed resampling strategy. The pose error distributions for the four sequences demonstrate that the proposed multi-hypotheses optimiser with gradients and resampling makes efficient use of as few as 5 samples to reliably explore the objective function and can converge to a minimum close to the true model configuration.

An additional qualitative overview of the tracking performance, as the mode of the distribution, is given in Figure 5.26 and shows that the estimated state aligns well with the observed robot.

Convergence Properties

To evaluate the optimiser independently from the performance of the trained network, the tracking objective and optimiser are applied to the ground truth keypoint heatmaps and segmentation of a synthetic dataset that contains 1000 images with randomly sampled robot state configurations, backgrounds and textured object models. The optimiser is initialised separately for each of the

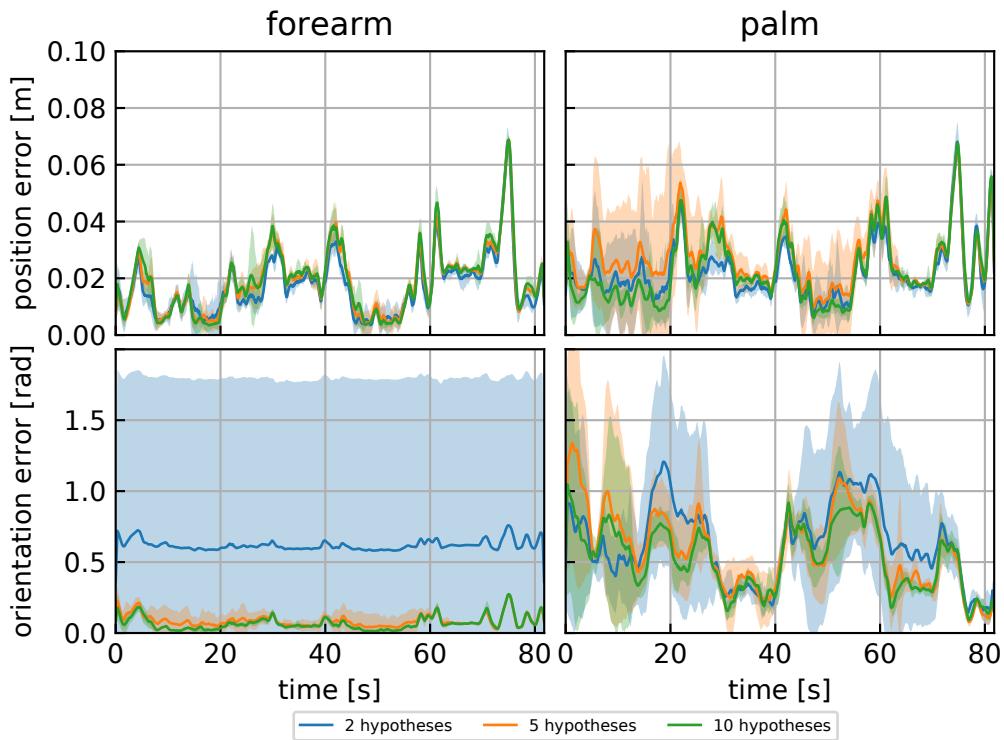


Figure 5.22: Pose tracking error distribution on *oball_pot* sequence.

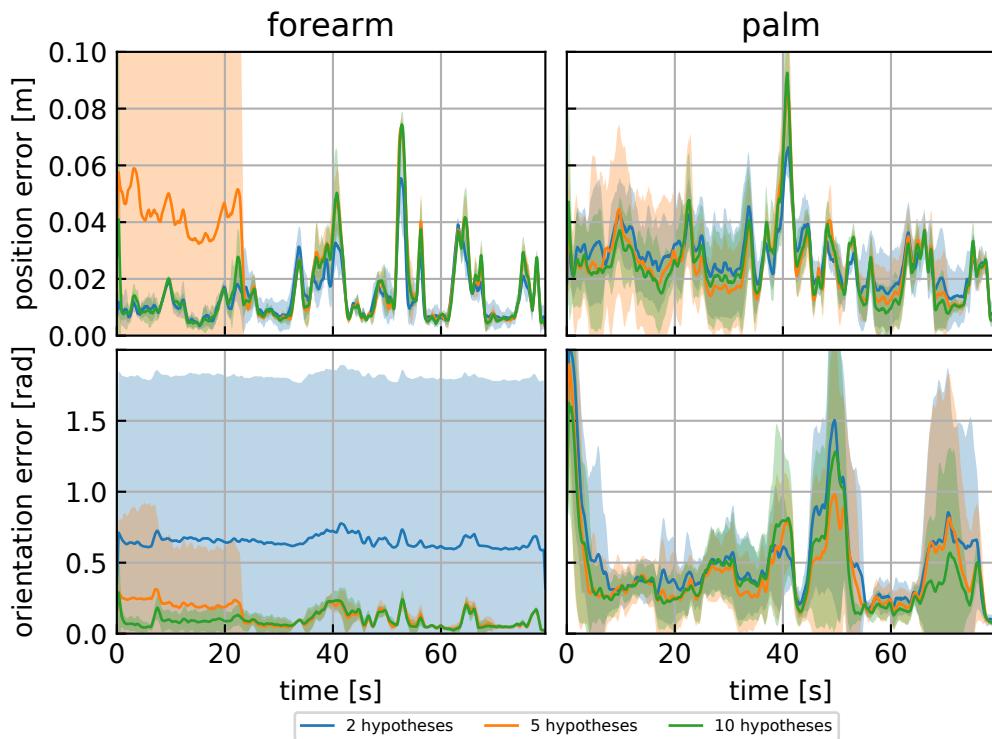


Figure 5.23: Pose tracking error distribution on *ospray* sequence.

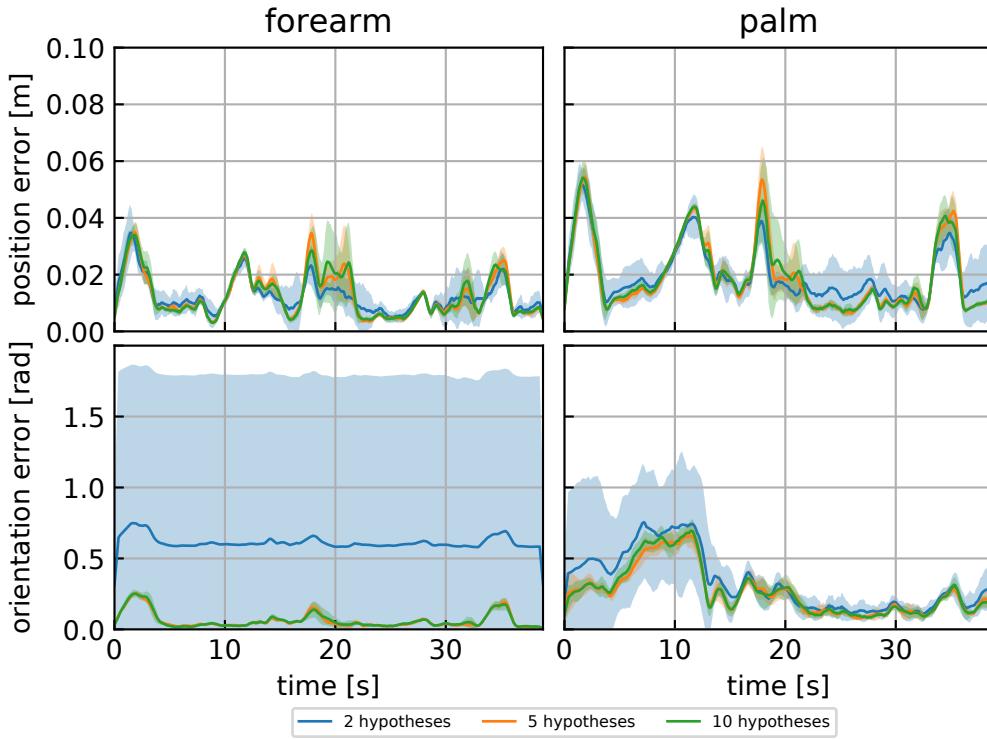


Figure 5.24: Pose tracking error distribution on *owbottle* sequence.

1000 images samples and optimises 10 hypotheses for 100 iterations on the true association for the tracking objective.

Figure 5.27 shows the distribution of forearm and palm pose errors and the cumulative histogram for the position and orientation error. Similar to previous observations, the distribution shows a wider spread of the orientation error than the position error. This can be explained that mismatching positions tend to induce a higher objective error than mismatching orientations. Overall, around 75% of forearm and 50% of palm poses converge to less than 1cm and $\frac{1}{16}\pi$ rad position and orientation error.

Depth Image Synthesis Error

In the following, we will compare the reported and estimated robot state by their depth image synthesis error. The depth image synthesis error is the discrepancy between the real observed depth image $\mathbf{I}_{D,obs}$ and the synthesised depth image $\mathbf{I}_{D,syn}$ given by a rendered robot state θ :

$$d_{err} = \frac{1}{|X_s|} \sum_{x \in X_s} |\mathbf{I}_{D,syn}(x) - \mathbf{I}_{D,obs}(x)|^2 \quad (5.13)$$

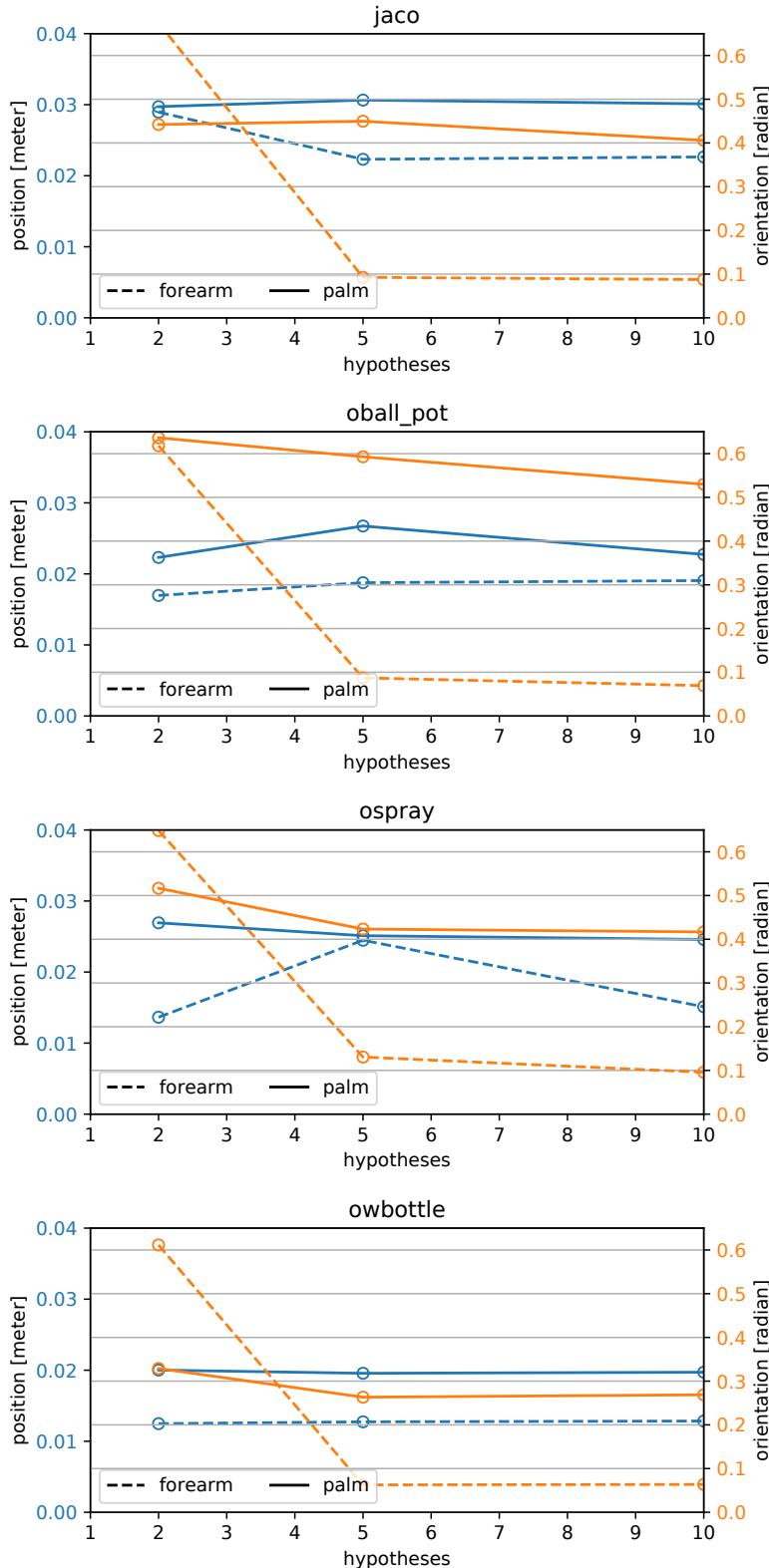


Figure 5.25: Average position (blue) and orientation (orange) error for a forearm (dashed line) and the palm (solid line) link over duration of sequences. While the average position error (blue) below 3cm indicates tracking success, the high forearm orientation error (orange) with only 2 hypotheses suggests that the underlying distribution partially converged to the wrong mode.

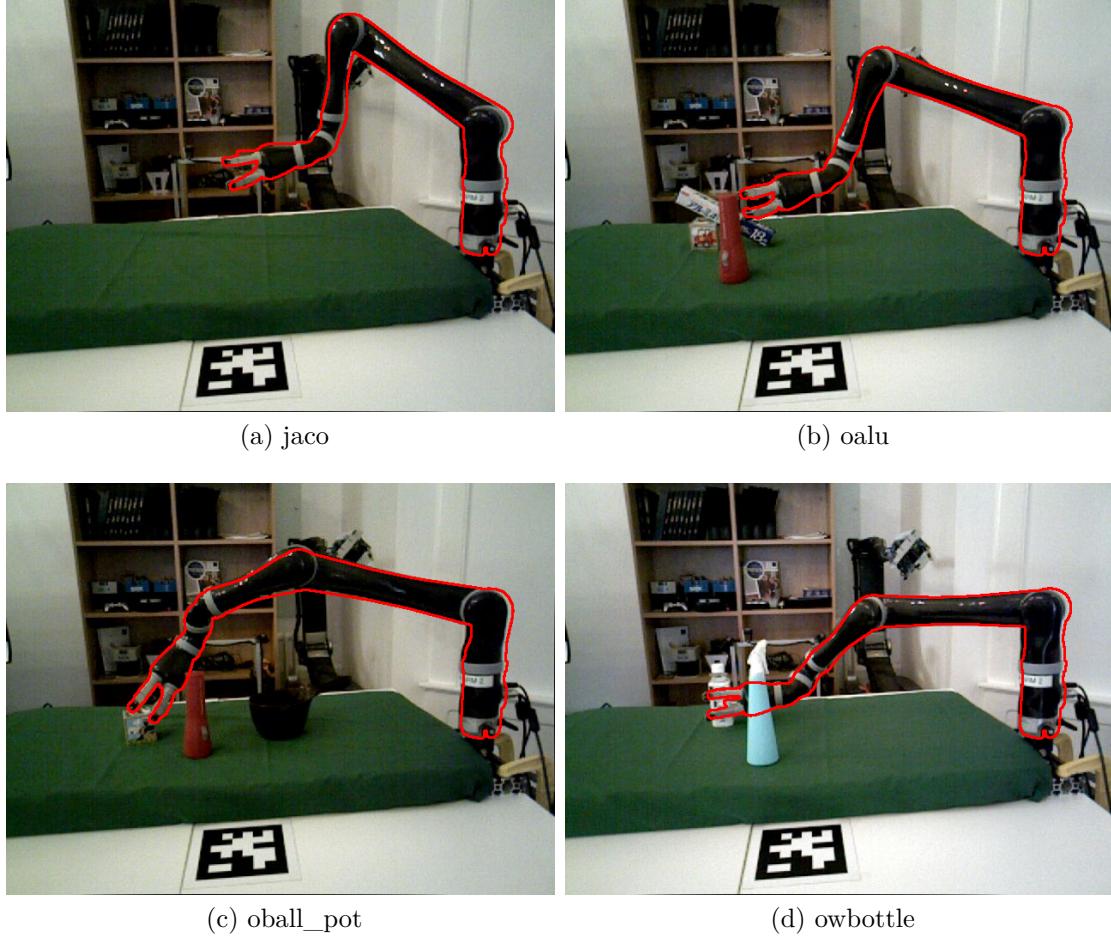


Figure 5.26: Observed robot with superimposed contour edges (red) of estimated state. Contour edges on objects, e.g. the bottle in (d), visualise the estimated state behind an occlusion.

with X_s as the set of valid synthesised pixel coordinates

$$X_s = \{x \mid \mathbf{I}_{D,syn}(x) > 0\} \quad . \quad (5.14)$$

To this end, the depth image synthesis error provides a measure of how well a robot state explains the observed depth.

Table 5.3 provides an average of this measure over all sequences and shows that the estimated state θ_{est} is typically closer to the observed state, than the state θ_{rep} that is provided by the joint position encoders. Figure 5.28 additionally shows that this improvement is constant over time.

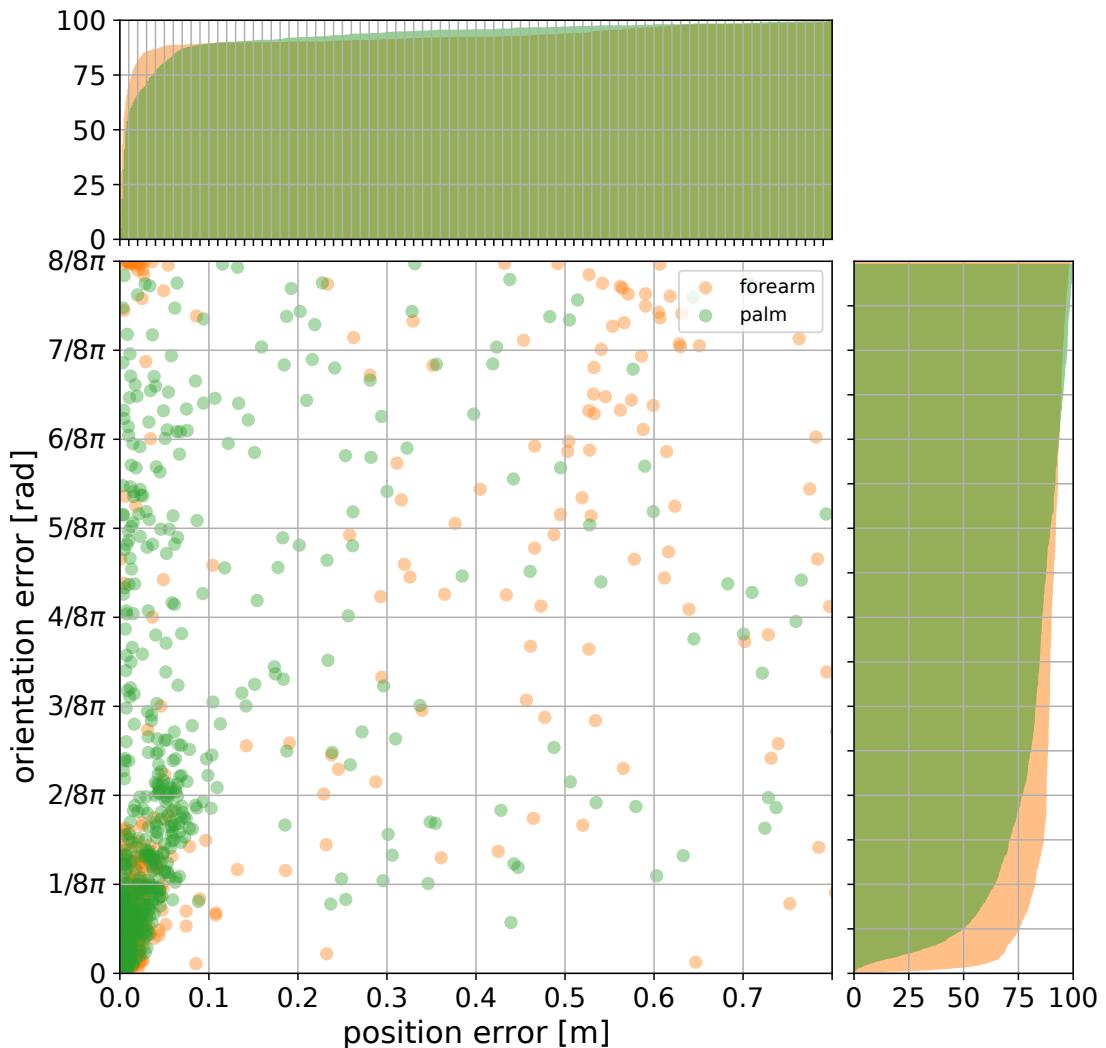


Figure 5.27: Distribution of converged forearm (orange) and palm (green) pose errors on 1000 synthetic images with ground truth keypoints and segmentation. Around 75% of forearm poses and 50% of palm poses converge within 1cm and $\frac{1}{16}\pi$ rad of pose error.

sequence	reported	estimated
gspray	0.56	0.37
gtorch	0.57	0.36
jaco	0.57	0.38
oalu	0.60	0.32
oball_bowl	0.68	0.38
oball_pot	0.46	0.35
ospray	0.56	0.37
owbottle	0.58	0.38

Table 5.3: Depth image synthesis error (eq. 5.13) in metres, averaged over the whole sequence. The *estimated* state provides a lower error than the originally *reported* joint position encoder state. Hence, the estimated state better explains the observed depth image.

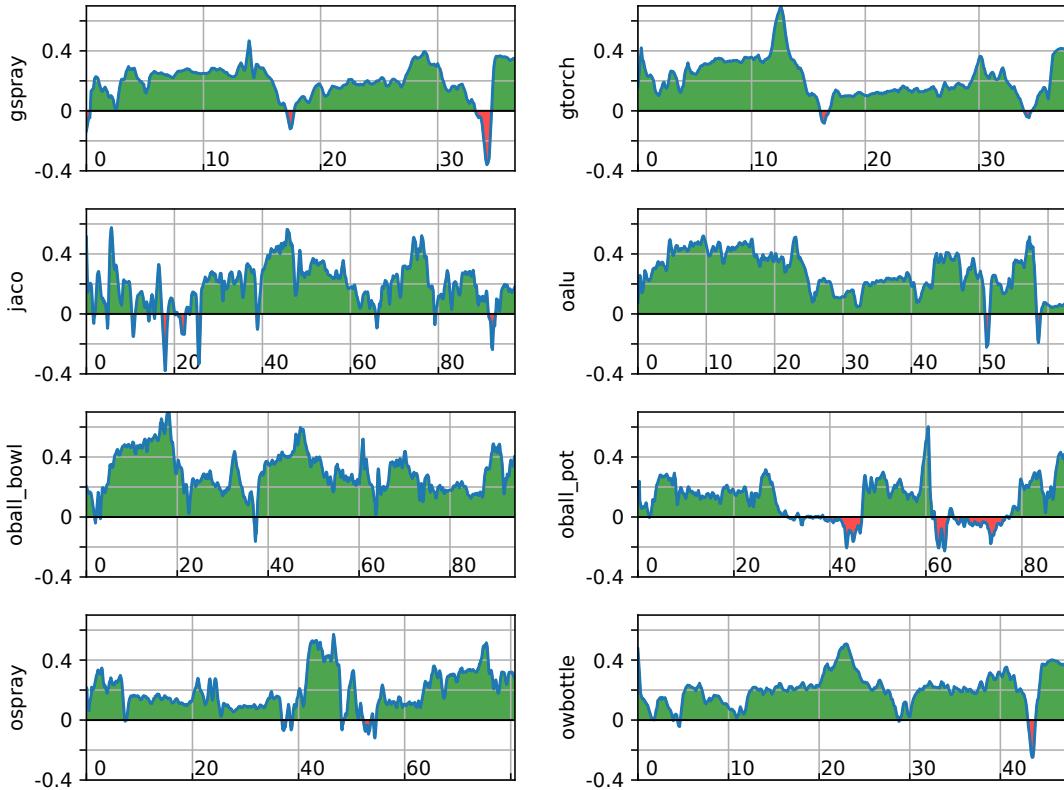


Figure 5.28: Improvement of the depth image synthesis error by the estimated state, $d_{err}(\theta_{rep}) - d_{err}(\theta_{est})$ (y-axis), plotted against the sequence timestamp (x-axis). Typically, the estimated state improves the depth synthesis error (green) compared to the reported state. Only a few time intervals show a disimprovement (red) by the estimated state.

particles	optimisation		total Hz
	seconds	Hz	
1	0.012	81.1	29.8
3	0.040	24.6	14.7
5	0.063	15.8	9.7
7	0.073	13.6	7.2
10	0.093	10.6	5.3

Table 5.4: Runtime performance of tracking with varying number of particles. *Optimisation*: duration of objective evaluation and optimisation. *Total*: Final measured time for model-fitting on single RGB-D image pair including prediction, optimisation and overhead through the underlying message passing framework.

Runtime

Table 5.4 gives an overview of the runtime performance of the full tracking pipeline and how it scales with varying number of particles. The tracking pipeline consists of the predictor and the optimiser, which run in parallel in two dedicated threads.

The RGB-D sensor provides colour and depth images at 30Hz. From this image pair, the predictor extracts segments and keypoints (eq. 5.1) within 17ms. The optimiser sequentially evaluates each particle on the objective (Algorithm 1) within 12ms per particle. Since all particles are evaluated on the same segments and keypoints, the prediction network is a constant factor, while the duration of model-fitting is linearly related to the number of particles.

5.5 Conclusion

In this chapter we relaxed some of our previous assumptions on the environment and the initial optimiser state to make the proposed tracking approach more widely applicable and robust to cluttered environments.

We proposed a pipeline for sampling a large variety of kinematic states and rendering them in a variety of lighting conditions and environments. This pipeline can be fully automated and easily adapted to other robot models without the need to physically access the robot. The generalisability of this image synthesis pipeline to real sequences was evaluated under different conditions for the input modality and background images.

We found that, although synthetic colour images alone generalise worse to real images than synthetic depth images, the combination of both modalities as RGB-D image pairs generalises best and provides the best prediction performance compared to the individual modalities. The comparison between a large synthetic and a smaller real training set revealed that synthesised training images are

capable to outperform manually labelled real data, despite the presence of the simulation-to-reality gap. This indicates that a large variety of kinematic states and visual properties is more important than having real but few training samples. However, these visual properties need to simulate the same properties found in the real application cases.

We proposed a parallel exploration of the search space with gradient-based search directions which combines the advantages of particle-based optimisers, avoiding convergence to a single local minimum, with the optimal update steps of gradient-based optimisers. We demonstrated that this optimiser in combination with the keypoint and freespace objective yields a robust tracking approach that converges on average within 3cm of the desired optimum over multiple trials and sequences. This was achieved despite strong background clutter and visual distractions from occlusions of the tracked manipulator. Further, our approach operates on the raw data without modelling the environment, and without assuming explicit knowledge about the manipulated and occluding objects.

The resampling strategy enables an efficient use of as few as 5 hypotheses for a large articulated kinematic chain such as the Jaco with 12 DoF. Compared to the PF approach in [43], which uses 70 particles to estimate a rigid object pose, and the PSO approach in [70], which uses 100 particles to estimate an articulated hand state, our proposed approach uses less particles per tracked state dimension.

Similarly to [70, 79], we would like to extend the optimiser to partially resample hypotheses from a distribution that is derived from the input image, to further improve the efficiency of sampled hypotheses. Such a resampling distribution could be provided by a method similar to what was proposed in Chapter 4.

Chapter 6

Conclusion

6.1 Summary

The aim of this thesis was to develop methods for articulated tracking that make minimal assumptions about available sensors and prior knowledge about the environment. Chapter 1 motivated the application of articulated tracking to robot manipulator state estimation using only visual sensing, with the explicit aim of enabling tracking of occluded manipulators during grasping. This required the development of these methods in such a way that they would be robust to visual distractions caused by manipulated objects, occlusions and the environment.

The articulated state estimation problem was formulated as an optimisation problem, where the solution can be found by minimising the distance between the true observed state and the estimated state. Approaching this problem involved investigating these three areas:

objective finding a robust representation of the distance between the observed image and an estimated state that is differentiable,

synthesis generating labelled colour and depth image data for supervised training of data-driven distance representations that generalise to real image data,

optimisation finding an optimisation approach that efficiently explores the state space of the distance representation and reliably converges to the optimum.

Chapter 2 provided an overview of related methods and how these are applied in state-of-the-art work. We concluded that:

- model-based approaches provide valuable prior knowledge about the kinematic, geometric and visual properties of the tracked object and facilitate enforcing of plausibility constraints,

- data associations establish correspondences in the 2D image or 3D observation frame and are therefore differentiable with respect to the estimated state using the inverse model,
- data-driven approaches provide the discrimination of points in the distance representation that is required to establish correspondences that are robust to unmodelled objects and environments,
- gradient-based optimisation approaches efficiently explore the local state.

While state-of-the-art work individually used these properties, they often had to make assumptions about the availability of proprioceptive sensing or the actual models of manipulated and occluding objects, or they did not consider occlusions at all, causing these approaches to break in the presence of visual distractions.

6.2 Contributions

We addressed some of the shortcomings of state-of-the-art work in Chapters 3 to 5 with respect to the main lines of research: robust objectives, robust image synthesis and training, and robust optimisation.

Chapter 3 introduced the concept of discriminative model-fitting. This was motivated by experiments on generative model-fitting objectives, which demonstrated the fragility of indirect non-discriminative data associations with respect to poor proprioception and visual distractions. We contributed a data-driven objective to resolve ambiguities in the data association and demonstrated that this mitigates effects of local minima in the objective.

While the depth image segmentation can provide a direct association between an area of pixels to specific parts of the tracked robot manipulator, it does not discriminate between these pixels and therefore leaves ambiguity. Furthermore, it neglects any unmodelled observations, such as untracked manipulanda or occluders, and would therefore fail in manipulation scenarios. To account for this, we additionally contributed a training strategy that samples randomised occlusion pixels during training and demonstrated that this makes the depth segmentation more robust to occlusions.

At this stage, the gradient-based optimiser still relied on initialisation from joint position encoder readings for the very first image of a sequence. The tracking objective only relied on depth images, which are simpler to synthesise and generalise more easily to real sensor readings, and had to handle typical structured light sensor effects like missing depth readings and unsharp edges.

Chapter 4 addressed the problem of optimiser initialisation and the indirect associations of pixels within segments. To remove the requirements on proprioception for initialising the optimiser, we contributed an optimiser initialisation from a predicted distribution and showed that this approach is superior to uniform initialisation as used by some state-of-the-art approaches. However, the use of the predicted state distribution was limited to the initialisation and the optimiser would thereafter use a single state hypothesis with the same implications on convergence towards local minima.

The data-driven tracking objective evolved around keypoints for a more direct association of the image to the model, which was further augmented with intensity edge information. This combined objective enabled tracking with strong distractions from multiple manipulated and occluding objects at the same time.

Chapter 5 built on findings of the previous work on input modality and multi-state optimisation. The objective combined colour and depth as input to a data-driven semantic keypoint and segment extractor. This required an extension of the image synthesis pipeline to colour, through which we contributed insight into the problem of generalisability of synthetic colour and depth training images to real test sequences.

The initial idea of using multiple possible states to avoid local minima at initialisation of an optimiser was developed further into a gradient-based multi-hypotheses optimiser which continuously maintains a distribution of possible states. This contributed an optimiser for articulated kinematic chains that is robust to random initialisation and local minima in the objective and at the same time requires fewer particles than traditional particle-based approaches.

6.3 Discussion and Future Work

6.3.1 Objective

Different representations of model-fitting objectives have been presented and evaluated in this thesis. The focus on data-driven methods enabled a free choice of the intermediate representation, which was reflected by the evolution of the objective from segmentation, to keypoints and edges, and finally the combination of keypoints and segmentation.

We found that the latter combination of keypoints and segmentation provided the strongest constraints in 3D space to fit a model to an observation and thus recover its state. The deliberative choice of using 2D and 3D point correspondences enabled the use of analytic gradients for an efficient exploration of the state space.

The choice of objectives and the combination of their individual strengths is a design decision which requires domain knowledge. This thesis explored these designs, but did not answer questions about the optimal choice of meta-parameters, such as the amount and placement of keypoints, the sub-segmentation of parts or the choice of the truncation of distances.

Specifically for keypoints, unsupervised approaches can provide an automatic way to learn the optimal location of meaningful keypoints [37], and remove the need to manually select keypoints in the local link frames. A logical continuation of the presented work on combining individual keypoint and segmentation objectives would be the use of dense features [68]. Dense features combine the pixel-level discrimination of keypoints with the density of segmentation into a single representation. We assume that these features are less generalisable to the currently employed model and increase the requirements on colour image synthesis.

6.3.2 Image Synthesis

We found that the extension of depth image synthesis to the colour domain is a non-trivial problem. The choice of pose coverage and visual variety is a trade-off and greatly impacts the generalisability of a synthetically trained model to real data. Articulated models of robotic manipulators largely focus on kinematic and geometric correctness for planning and control, but neglect visual accuracy. Additional effort has to be taken to align the visual representation of those models with the observation. It was found that a synthesis of random background colours and textures is not sufficient and that more realistic backgrounds are required.

The simulation-to-reality gap has implications on the accuracy of predictions and thus limits the accuracy of the estimated state. This was especially evident for tracking the small finger links that are also most likely affected by occlusions. While our approach robustly tracks the arm and hand, it has limited abilities to accurately track fingers.

Solving the image synthesis problem is necessary to provide the large amount of data required for data-driven methods. Going further, solving the image synthesis problem immediately enables access to arbitrary labels, that are otherwise difficult to obtain manually.

Large realistic high-fidelity environment datasets like Matterport3D [14], Gibson Virtual Environment [87] and the Replica-Dataset [75] became available in recent years and facilitate image synthesis for training. The problem of finding proper visual representations of models can be approached by differentiable rendering [46] to recover optimal texture properties from real images.

6.3.3 Models

Methods developed in this thesis were trained and developed in such a way that they did not require an explicit model of the manipulated or occluded objects, or the environment in which they were applied. This was a deliberative choice to become independent of prior knowledge about the grasping scene.

However, the proposed methods heavily relied on articulated models of the tracked object. These models provided the backward relations that are required to obtain gradients to explore the objective, and they provided the forward relations that were used for image synthesis during tracking and training. The use of articulated models was justified by the application to robotic manipulators, where kinematic and geometric models are widely available.

By design, our approach is limited to applications where a kinematic, geometric and visual model is available. Specifically the data-driven objectives are model-specific and have to be retrained for new applications.

Other application domains, such as tracking the articulation of manipulated objects where models may not be available would benefit from the simultaneous online modelling and tracking of observed objects.

6.3.4 Optimisation

The problem of robust tracking concerns the design of the objective and the choice of optimiser. Even with the augmentation of the objective with discriminative information, we could not arrive at a convex objective that would converge from arbitrary initial states to the global minimum. This is a predominant issue for articulated models with many dependent chained transformations along a kinematic path. The choice of initialisation is critical to avoid local minima.

This thesis eventually approached this issue from within the solver by considering many possible solutions to the objective optimum. We found this an elegant way to mitigate the problem of initialisation and local minima to some extent.

The resampling approach assumed a bimodal distribution of states in minima of the objective. This choice was motivated by the kinematic properties and the visual appearance of the manipulator model but has limited applicability to alternative kinematic structures, such as star-like hand structures.

Extensions of this work could investigate more general approaches to cluster objective minima and further populate the distribution with random states to enable recovering from collapses to local minima that may arise later in tracking.

Bibliography

- [1] Paulo Abelha, Frank Guerin, and Markus Schoeler. “A Model-Based Approach to Finding Substitute Tools in 3D Vision Data”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 2471–2478. DOI: [10.1109/ICRA.2016.7487400](https://doi.org/10.1109/ICRA.2016.7487400).
- [2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3686–3693. DOI: [10.1109/CVPR.2014.471](https://doi.org/10.1109/CVPR.2014.471).
- [3] Pedram Azad, David Münch, Tamim Asfour, and Rüdiger Dillmann. “6-DoF Model-based Tracking of Arbitrarily Shaped 3D Objects”. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. May 2011, pp. 5204–5209. DOI: [10.1109/ICRA.2011.5979950](https://doi.org/10.1109/ICRA.2011.5979950).
- [4] Charles A. Baird. “Quasilinearization and the Methods of Finite Difference and Initial Values”. In: *Journal of Optimization Theory and Applications* 6.4 (Oct. 1970), pp. 320–330. DOI: [10.1007/BF00925380](https://doi.org/10.1007/BF00925380).
- [5] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. “Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 2593–2602. DOI: [10.1109/ICCV.2017.281](https://doi.org/10.1109/ICCV.2017.281).
- [6] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. “Robot Arm Pose Estimation through Pixel-Wise Part Classification”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 3143–3150. DOI: [10.1109/ICRA.2014.6907311](https://doi.org/10.1109/ICRA.2014.6907311).
- [7] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. “Learning 6D Object Pose Estimation Using 3D Object Coordinates”. In: *Computer Vision – ECCV 2014*. 2014, pp. 536–551. DOI: [10.1007/978-3-319-10605-2_35](https://doi.org/10.1007/978-3-319-10605-2_35).

- [8] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. “Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 3364–3372. doi: [10.1109/CVPR.2016.366](https://doi.org/10.1109/CVPR.2016.366).
- [9] Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. “Smart Particle Filtering for High-dimensional Tracking”. In: *Computer Vision and Image Understanding* 106.1 (Apr. 2007), pp. 116–129. doi: [10.1016/j.cviu.2005.09.013](https://doi.org/10.1016/j.cviu.2005.09.013).
- [10] Jonathan Brookshire and Seth Teller. “Articulated Pose Estimation via Over-parametrization and Noise Projection”. In: *Proceedings of Robotics: Science and Systems*. July 2014. doi: [10.15607/RSS.2014.X.009](https://doi.org/10.15607/RSS.2014.X.009).
- [11] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698. doi: [10.1109/TPAMI.1986.4767851](https://doi.org/10.1109/TPAMI.1986.4767851).
- [12] Zhe Cao, Yaser Sheikh, and Natasha Kholgade Banerjee. “Real-time scalable 6DOF pose estimation for textureless objects”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 2441–2448. doi: [10.1109/ICRA.2016.7487396](https://doi.org/10.1109/ICRA.2016.7487396).
- [13] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. “Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 1302–1310. doi: [10.1109/CVPR.2017.143](https://doi.org/10.1109/CVPR.2017.143).
- [14] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Niebner, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. “Matterport3D: Learning from RGB-D Data in Indoor Environments”. In: *2017 International Conference on 3D Vision (3DV)*. Oct. 2017, pp. 667–676. doi: [10.1109/3DV.2017.00081](https://doi.org/10.1109/3DV.2017.00081).
- [15] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. “ShapeNet: An Information-Rich 3D Model Repository”. In: *CoRR* abs/1512.03012 (2015). arXiv: [1512.03012](https://arxiv.org/abs/1512.03012).
- [16] Georgios Chliveros, Maria Pateraki, and Panos Trahanias. “Robust Multi-hypothesis 3D Object Pose Tracking”. In: *Computer Vision Systems*. 2013, pp. 234–243. doi: [10.1007/978-3-642-39402-7_24](https://doi.org/10.1007/978-3-642-39402-7_24).

- [17] Changhyun Choi and Henrik I. Christensen. “RGB-D Object Tracking: A Particle Filter Approach on GPU”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 1084–1091. DOI: [10.1109/IROS.2013.6696485](https://doi.org/10.1109/IROS.2013.6696485).
- [18] Antonio Criminisi and Jamie Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Publishing Company, Incorporated, 2013. DOI: [10.1007/978-1-4471-4929-3](https://doi.org/10.1007/978-1-4471-4929-3).
- [19] Pierre Del Moral. “Nonlinear filtering: Interacting particle resolution”. In: *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics* 325.6 (1997), pp. 653–658. DOI: [10.1016/S0764-4442\(97\)84778-7](https://doi.org/10.1016/S0764-4442(97)84778-7).
- [20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2009, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [21] Karthik Desingh, Shiyang Lu, Anthony OPIPARI, and Odest Chadwicke Jenkins. “Factored Pose Estimation of Articulated Objects using Efficient Nonparametric Belief Propagation”. In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 7221–7227. DOI: [10.1109/ICRA.2019.8793973](https://doi.org/10.1109/ICRA.2019.8793973).
- [22] David Eigen and Rob Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 2650–2658. DOI: [10.1109/ICCV.2015.304](https://doi.org/10.1109/ICCV.2015.304).
- [23] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Distance Transforms of Sampled Functions”. In: *Theory of Computing* 8.19 (2012), pp. 415–428. DOI: [10.4086/toc.2012.v008a019](https://doi.org/10.4086/toc.2012.v008a019).
- [24] Peter R. Florence, Lucas Manuelli, and Russ Tedrake. “Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation”. In: *Proceedings of the 2nd Conference on Robot Learning*. Vol. 87. Oct. 2018, pp. 373–385. URL: <http://proceedings.mlr.press/v87/florence18a.html>.
- [25] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. “Virtual Worlds as Proxy for Multi-Object Tracking Analysis”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 4340–4349. DOI: [10.1109/CVPR.2016.470](https://doi.org/10.1109/CVPR.2016.470).

- [26] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. “Real-Time Human Pose Tracking from Range Data”. In: *Computer Vision – ECCV 2012*. 2012, pp. 738–751. doi: [10.1007/978-3-642-33783-3_53](https://doi.org/10.1007/978-3-642-33783-3_53).
- [27] Cristina Garcia Cifuentes, Jan Issac, Manuel Wüthrich, Stefan Schaal, and Jeannette Bohg. “Probabilistic Articulated Real-Time Tracking for Robot Manipulation”. In: *IEEE Robotics and Automation Letters* 2.2 (Apr. 2017), pp. 577–584. doi: [10.1109/LRA.2016.2645124](https://doi.org/10.1109/LRA.2016.2645124).
- [28] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. “Vision meets robotics: The KITTI dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237. doi: [10.1177/0278364913491297](https://doi.org/10.1177/0278364913491297).
- [29] Philip E. Gill and Walter Murray. “Algorithms for the Solution of the Nonlinear Least-Squares Problem”. In: *SIAM Journal on Numerical Analysis* 15.5 (1978), pp. 977–992. doi: [10.1137/0715063](https://doi.org/10.1137/0715063).
- [30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. “Deep Learning”. In: MIT Press, 2016. Chap. Convolutional Networks, pp. 326–366. ISBN: 0262035618. URL: <http://www.deeplearningbook.org>.
- [31] Daniel Grest and Volker Krüger. “Gradient-Enhanced Particle Filter for Vision-Based Motion Capture”. In: *Human Motion – Understanding, Modeling, Capture and Animation*. 2007, pp. 28–41. doi: [10.1007/978-3-540-75703-0_3](https://doi.org/10.1007/978-3-540-75703-0_3).
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [33] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, and Pascal Fua. “Gradient Response Maps for Real-Time Detection of Textureless Objects”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (May 2012), pp. 876–888. doi: [10.1109/TPAMI.2011.206](https://doi.org/10.1109/TPAMI.2011.206).
- [34] ASUSTeK Computer Inc. *Xtion PRO LIVE*. URL: https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/.

- [35] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. “Depth-Based Object Tracking Using a Robust Gaussian Filter”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 608–615. doi: [10.1109/ICRA.2016.7487184](https://doi.org/10.1109/ICRA.2016.7487184).
- [36] Vladimir Ivan, Yiming Yang, Wolfgang Merkt, Michael P. Camilleri, and Sethu Vijayakumar. “EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control”. In: *Robot Operating System (ROS): The Complete Reference (Volume 3)*. Ed. by Anis Koubaa. Springer International Publishing, 2019, pp. 211–240. doi: [10.1007/978-3-319-91590-6_7](https://doi.org/10.1007/978-3-319-91590-6_7).
- [37] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. “Unsupervised Learning of Object Landmarks through Conditional Image Generation”. In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 4016–4027. URL: <https://papers.nips.cc/paper/7657-unsupervised-learning-of-object-landmarks-through-conditional-image-generation>.
- [38] Stephen James, Andrew J. Davison, and Edward Johns. “Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task”. In: *Proceedings of the 1st Conference on Robot Learning*. Vol. 78. Nov. 2017, pp. 334–343. URL: <http://proceedings.mlr.press/v78/james17a.html>.
- [39] James Kennedy and Russell C. Eberhart. “Particle Swarm Optimization”. In: *Proceedings of ICNN’95 - International Conference on Neural Networks*. Vol. 4. Nov. 1995, 1942–1948 vol.4. doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [40] Hyunseok Kim and Dongjun Suh. “Hybrid Particle Swarm Optimization for Multi-Sensor Data Fusion”. In: *Sensors* 18.9 (2018). doi: [10.3390/s18092792](https://doi.org/10.3390/s18092792).
- [41] Kinova inc. *KINOVA JACO® Assistive robot User Guide*. 2018. URL: https://www.kinovarobotics.com/sites/default/files/UG-007_KINOVA_Jaco_Assistive_robot_User_guide_EN_R02.pdf.
- [42] Philip Krejov, Andrew Gilbert, and Richard Bowden. “Guided optimisation through classification and regression for hand pose estimation”. In: *Computer Vision and Image Understanding* 155 (2017), pp. 124–138. doi: [10.1016/j.cviu.2016.11.005](https://doi.org/10.1016/j.cviu.2016.11.005).
- [43] Alexander Krull, Frank Michel, Eric Brachmann, Stefan Gumhold, Stephan Ihrke, and Carsten Rother. “6-DOF Model Based Tracking via Object Coordinate Regression”. In: *Computer Vision – ACCV 2014*. 2015, pp. 384–399. doi: [10.1007/978-3-319-16817-3_25](https://doi.org/10.1007/978-3-319-16817-3_25).

- [44] Kenneth Levenberg. “A method for the solution of certain non-linear problems in least squares”. In: *Quarterly of Applied Mathematics* 2 (1944), pp. 164–168. DOI: [10.1090/qam/10666](https://doi.org/10.1090/qam/10666).
- [45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. 2014, pp. 740–755. DOI: [10.1007/978-3-319-10602-1_48](https://doi.org/10.1007/978-3-319-10602-1_48).
- [46] Matthew M. Loper and Michael J. Black. “OpenDR: An Approximate Differentiable Renderer”. In: *Computer Vision – ECCV 2014*. 2014, pp. 154–169. DOI: [10.1007/978-3-319-10584-0_11](https://doi.org/10.1007/978-3-319-10584-0_11).
- [47] Siddharth Mahendran, Haider Ali, and René Vidal. “3D Pose Regression Using Convolutional Neural Networks”. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2017, pp. 2174–2182. DOI: [10.1109/ICCVW.2017.254](https://doi.org/10.1109/ICCVW.2017.254).
- [48] Alexandros Makris, Nikolaos Kyriazis, and Antonis A. Argyros. “Hierarchical Particle Filtering for 3D Hand Tracking”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2015, pp. 8–17. DOI: [10.1109/CVPRW.2015.7301343](https://doi.org/10.1109/CVPRW.2015.7301343).
- [49] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. “What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?” In: *International Journal of Computer Vision* 126.9 (Sept. 2018), pp. 942–960. DOI: [10.1007/s11263-018-1082-6](https://doi.org/10.1007/s11263-018-1082-6).
- [50] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. “SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 2697–2706. DOI: [10.1109/ICCV.2017.292](https://doi.org/10.1109/ICCV.2017.292).
- [51] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. “Real-Time Hand Tracking Under Occlusion from an Egocentric RGB-D Sensor”. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Oct. 2017, pp. 1284–1293. DOI: [10.1109/ICCVW.2017.82](https://doi.org/10.1109/ICCVW.2017.82).

- [52] Enrique Muñoz, Yoshinori Konishi, Vittorio Murino, and Alessio Del Bue. “Fast 6D Pose Estimation for Texture-less Objects from a single RGB image”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 5623–5630. DOI: [10.1109/ICRA.2016.7487781](https://doi.org/10.1109/ICRA.2016.7487781).
- [53] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked Hourglass Networks for Human Pose Estimation”. In: *Computer Vision – ECCV 2016*. 2016, pp. 483–499. DOI: [10.1007/978-3-319-46484-8_29](https://doi.org/10.1007/978-3-319-46484-8_29).
- [54] Phuong D.H. Nguyen, Tobias Fischer, Hyung Jin Chang, Ugo Pattacini, Giorgio Metta, and Yiannis Demiris. “Transferring Visuomotor Learning from Simulation to the Real World for Robotics Manipulation Tasks”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 6667–6674. DOI: [10.1109/IROS.2018.8594519](https://doi.org/10.1109/IROS.2018.8594519).
- [55] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. “Generalized Feedback Loop for Joint Hand-Object Pose Estimation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–15. DOI: [10.1109/TPAMI.2019.2907951](https://doi.org/10.1109/TPAMI.2019.2907951).
- [56] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. “Hands Deep in Deep Learning for Hand Pose Estimation”. In: *Proceedings of the 20th Computer Vision Winter Workshop*. 2015, pp. 21–30. DOI: [10.3217/978-3-85125-388-7](https://doi.org/10.3217/978-3-85125-388-7).
- [57] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. “Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints”. In: *2011 IEEE International Conference on Computer Vision (ICCV)*. Nov. 2011, pp. 2088–2095. DOI: [10.1109/ICCV.2011.6126483](https://doi.org/10.1109/ICCV.2011.6126483).
- [58] Edwin Olson. “AprilTag: A robust and flexible visual fiducial system”. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. May 2011, pp. 3400–3407. DOI: [10.1109/ICRA.2011.5979561](https://doi.org/10.1109/ICRA.2011.5979561).
- [59] Valerio Ortenzi, Naresh Marturi, Rustam Stolkin, Jeffrey A. Kuo, and Michael Mistry. “Vision-guided state estimation and control of robotic manipulators which lack proprioceptive sensors”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2016, pp. 3567–3574. DOI: [10.1109/IROS.2016.7759525](https://doi.org/10.1109/IROS.2016.7759525).
- [60] Karl Pauwels, Vladimir Ivan, Eduardo Ros, and Sethu Vijayakumar. “Real-time Object Pose Recognition and Tracking with an Imprecisely Calibrated Moving RGB-D Camera”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2014, pp. 2733–2740. DOI: [10.1109/IROS.2014.6942936](https://doi.org/10.1109/IROS.2014.6942936).

- [61] Karl Pauwels, Leonardo Rubio, Javier Díaz, and Eduardo Ros. “Real-Time Model-Based Rigid Object Pose Estimation and Tracking Combining Dense and Sparse Visual Cues”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2013, pp. 2347–2354. doi: [10.1109/CVPR.2013.304](https://doi.org/10.1109/CVPR.2013.304).
- [62] Vagelis Plevris and Papadrakakis Manolis. “A Hybrid Particle Swarm-Gradient Algorithm for Global Structural Optimization”. In: *Computer-Aided Civil and Infrastructure Engineering* 26 (Jan. 2011), pp. 48–68. doi: [10.1111/j.1467-8667.2010.00664.x](https://doi.org/10.1111/j.1467-8667.2010.00664.x).
- [63] Christian Rauch, Timothy Hospedales, Jamie Shotton, and Maurice Fallon. “Visual Articulated Tracking in the Presence of Occlusions”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018, pp. 643–650. doi: [10.1109/ICRA.2018.8462873](https://doi.org/10.1109/ICRA.2018.8462873).
- [64] Christian Rauch, Vladimir Ivan, Timothy Hospedales, Jamie Shotton, and Maurice Fallon. “Learning-driven Coarse-to-Fine Articulated Robot Tracking”. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 6604–6610. doi: [10.1109/ICRA.2019.8794359](https://doi.org/10.1109/ICRA.2019.8794359).
- [65] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018, pp. 4510–4520. doi: [10.1109/CVPR.2018.00474](https://doi.org/10.1109/CVPR.2018.00474).
- [66] Tanner Schmidt, Katharina Hertkorn, Richard Newcombe, Zoltan Marton, Michael Suppa, and Dieter Fox. “Depth-Based Tracking with Physical Constraints for Robot Manipulation”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. May 2015, pp. 119–126. doi: [10.1109/ICRA.2015.7138989](https://doi.org/10.1109/ICRA.2015.7138989).
- [67] Tanner Schmidt, Richard Newcombe, and Dieter Fox. “DART: dense articulated real-time tracking with consumer depth cameras”. In: *Autonomous Robots* 39.3 (2015), pp. 239–258. doi: [10.1007/s10514-015-9462-z](https://doi.org/10.1007/s10514-015-9462-z).
- [68] Tanner Schmidt, Richard Newcombe, and Dieter Fox. “Self-Supervised Visual Descriptor Learning for Dense Correspondence”. In: *IEEE Robotics and Automation Letters* 2.2 (Apr. 2017), pp. 420–427. doi: [10.1109/LRA.2016.2634089](https://doi.org/10.1109/LRA.2016.2634089).
- [69] SCHUNK GmbH & Co. KG. *SDH2 Operating Manual*. 2018. URL: <https://schunk.com/fileadmin/pim/docs/IM0023480.PDF>.

- [70] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. “Accurate, Robust, and Flexible Real-Time Hand Tracking”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. 2015, pp. 3633–3642. DOI: [10.1145/2702123.2702179](https://doi.org/10.1145/2702123.2702179).
- [71] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. “Real-Time Human Pose Recognition in Parts from Single Depth Images”. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2011, pp. 1297–1304. DOI: [10.1109/CVPR.2011.5995316](https://doi.org/10.1109/CVPR.2011.5995316).
- [72] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. “Hand Keypoint Detection in Single Images Using Multiview Bootstrapping”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 4645–4653. DOI: [10.1109/CVPR.2017.494](https://doi.org/10.1109/CVPR.2017.494).
- [73] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. “Fast and Robust Hand Tracking Using Detection-Guided Optimization”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3213–3221. DOI: [10.1109/CVPR.2015.7298941](https://doi.org/10.1109/CVPR.2015.7298941).
- [74] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. “Real-Time Joint Tracking of a Hand Manipulating an Object from RGB-D Input”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 294–310. DOI: [10.1007/978-3-319-46475-6_19](https://doi.org/10.1007/978-3-319-46475-6_19).
- [75] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. “The Replica Dataset: A Digital Replica of Indoor Spaces”. In: *CoRR* abs/1906.05797 (2019). arXiv: [1906.05797](https://arxiv.org/abs/1906.05797).
- [76] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. “A Benchmark for the Evaluation of RGB-D SLAM Systems”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots*

- and Systems (IROS)*. Oct. 2012, pp. 573–580. doi: [10.1109/IROS.2012.6385773](https://doi.org/10.1109/IROS.2012.6385773).
- [77] Wenyu Sun and Ya-Xiang Yuan. “Solving Nonlinear Least-Squares Problems”. In: *Optimization Theory and Methods: Nonlinear Programming*. Springer US, 2006. Chap. Nonlinear Least-Squares Problems, pp. 353–383. doi: [10.1007/0-387-24976-1_7](https://doi.org/10.1007/0-387-24976-1_7).
- [78] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. “Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 3325–3333. doi: [10.1109/ICCV.2015.380](https://doi.org/10.1109/ICCV.2015.380).
- [79] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. “Efficient and Precise Interactive Hand Tracking through Joint, Continuous Optimization of Pose and Correspondences”. In: *ACM Transactions on Graphics (TOG)* 35.4 (July 2016). doi: [10.1145/2897824.2925965](https://doi.org/10.1145/2897824.2925965).
- [80] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. “Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks”. In: *ACM Transactions on Graphics (TOG)* 33.5 (Sept. 2014). doi: [10.1145/2629500](https://doi.org/10.1145/2629500).
- [81] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. “Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation”. In: *International Journal of Computer Vision* 118.2 (June 2016), pp. 172–193. doi: [10.1007/s11263-016-0895-4](https://doi.org/10.1007/s11263-016-0895-4).
- [82] Julien P. C. Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip H. S. Torr, Shahram Izadi, and Cem Keskin. “Learning to Navigate the Energy Landscape”. In: *2016 International Conference on 3D Vision (3DV)*. Oct. 2016, pp. 323–332. doi: [10.1109/3DV.2016.41](https://doi.org/10.1109/3DV.2016.41).
- [83] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. “Convolutional Pose Machines”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 4724–4732. doi: [10.1109/CVPR.2016.511](https://doi.org/10.1109/CVPR.2016.511).

- [84] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. “ElasticFusion: Dense SLAM Without A Pose Graph”. In: *Proceedings of Robotics: Science and Systems*. July 2015. doi: [10.15607/RSS.2015.XI.001](https://doi.org/10.15607/RSS.2015.XI.001).
- [85] Felix Widmaier, Daniel Kappler, Stefan Schaal, and Jeannette Bohg. “Robot Arm Pose Estimation by Pixel-wise Regression of Joint Angles”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 616–623. doi: [10.1109/ICRA.2016.7487185](https://doi.org/10.1109/ICRA.2016.7487185).
- [86] Manuel Wüthrich, Peter Pastor, Mrinal Kalakrishnan, Jeannette Bohg, and Stefan Schaal. “Probabilistic Object Tracking using a Range Camera”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 3195–3202. doi: [10.1109/IROS.2013.6696810](https://doi.org/10.1109/IROS.2013.6696810).
- [87] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. “Gibson Env: Real-World Perception for Embodied Agents”. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018, pp. 9068–9079. doi: [10.1109/CVPR.2018.00945](https://doi.org/10.1109/CVPR.2018.00945).
- [88] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes”. In: *Proceedings of Robotics: Science and Systems*. June 2018. doi: [10.15607/RSS.2018.XIV.019](https://doi.org/10.15607/RSS.2018.XIV.019).
- [89] Yi Yang and Deva Ramanan. “Articulated Human Detection with Flexible Mixtures of Parts”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (2013), pp. 2878–2890. doi: [10.1109/TPAMI.2012.261](https://doi.org/10.1109/TPAMI.2012.261).
- [90] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. “Deep Kinematic Pose Regression”. In: *Computer Vision – ECCV 2016 Workshops*. 2016, pp. 186–201. doi: [10.1007/978-3-319-49409-8_17](https://doi.org/10.1007/978-3-319-49409-8_17).
- [91] Christian Zimmermann and Thomas Brox. “Learning to Estimate 3D Hand Pose from Single RGB Images”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4913–4921. doi: [10.1109/ICCV.2017.525](https://doi.org/10.1109/ICCV.2017.525).