Articulated Tracking for Humanoid Manipulation

Christian Rauch

School of Informatics
University of Edinburgh

This thesis is submitted for the degree of
Master of Science by Research

August 2016

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 processional qualification except as specified.

Christian Rauch
August 2016

Acknowledgements

First of all, I would like to thank my supervisor Dr. Maurice Fallon for his guidance throughout this thesis and for creating the working environment in which I could carry out my research. It is a pleasure for me to work with him and discuss research topics in the robot perception group.

Further, a special thanks goes to the postdoctoral researchers and Ph.D. students in the Valkyrie lab for helping me in getting the robot running and supporting me in my data collection.

I finally want to thank Julia for her unending patience during this period and for supporting me on my way.

Abstract

Humanoid manipulator with several degree of freedom are versatile applicable to a variety of tasks by using man-made tools as opposed to task-specific manipulators. This comes with the cost of more complex control over the state of this manipulator and the tool.

This work addresses the problem of articulated manipulator and object tracking using visual feedback from the robot's own perception system. Current research has addressed different aspects of the problem like pose estimation of independent rigid parts, but only a few publications have addressed the specific case of manipulator and object tracking in the case where the state of the manipulator in respect to the observation frame is known.

The main contribution of this work is the investigation of the combined usage of visual perception and prior information from the reported manipulator state for a gradient based optimization approach. This is motivated by the fact that gradient based approaches by design face the issue of local minimums in the objective function.

The evaluation of the gradient based approach using visual perception only, and using additional prior information, will on one side demonstrate the adverse effect of local minima for a manipulation task, and on the other side will show the benefits of exploiting prior information in the very same setting. In this context, the work also addresses the issue of joint position encoder calibration as these values are the foundation of the exploited prior information.

Table of contents

List of figures

Chapter 1

Introduction

1.1 Motivation

Humanoid robots are one of the most versatile robots when it comes to acting in a world with man-made structures and tools. As such, they are a huge area of research involving multiple disciplines. Of special research interest is humanoid manipulation using manipulators that resemble human hands. Manipulators with several degree of freedom (DoF) enable robots to reuse man-made tools for solving different kind of problems as opposed to being restricted to task-specific manipulators. Some of the tasks defined for the DARPA Robotics Challenge 2015 (DRC) [1] intended to advance robotic manipulation in this direction by setting a robot in a rescue scenario that involves the interaction with door handles (Door task), turning valves (Valve task), connecting a hose to a wye (Hose task) and using a drill to cut a hole in a wall (Wall task). Since the DRC rules stated that the robot needed to carry all of its manipulators during each trial, teams were force to use versatile manipulators.

The advantage of having such a generic manipulator comes with the cost of more complex control. Humanoid manipulation does not only involve the grasping of a tool but also the continuous control of this tool for the duration of the task. Since grasped tools are only indirectly connected to the robot's kinematic chain, additional perceptual information is needed to estimate and track the actual state of the manipulated object.

1.2 Problem Definition and Research Aim

Humanoid manipulation is a challenging task because of two main reasons: inaccurate forward kinematics and a changing environment. Inaccurate forward kinematics is mostly caused by improperly calibrated joint encoders and linkage elasticity. Both issues can only be resolved

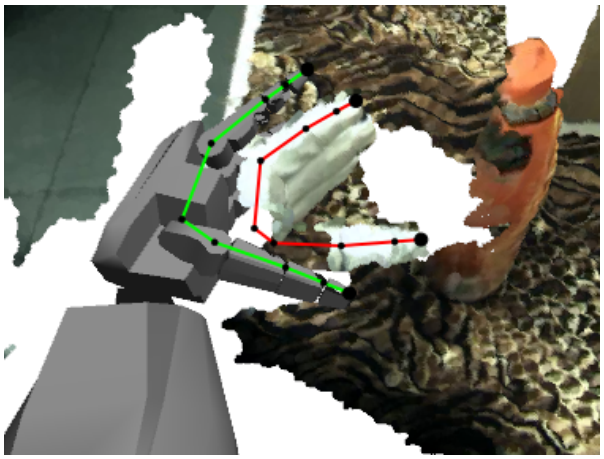


Figure 1.1: Joint calibration issue resulting in offset of reported pose (solid gray mesh, green marker) from perceived hand pose (coloured point cloud, red marker).

to some extent by e.g. modelling joint position offsets [2] and stiffness coefficients [3]. The effect of inaccurate forward kinematics is visualised in figure 1.1. Even small deviations in the reported joint positions propagate through the kinematic chain towards the manipulator and result in an offset that makes open-loop grasping infeasible. By interacting with objects such as a valve, the robot needs to compensate the applied force by a whole body motion which likely changes the pose of the manipulated object in the camera frame.

Both issues are in contrast with open-loop industrial robot applications such as moving parts in an assembly line, where stiff actuators manipulate objects at defined poses. Using visual feedback to estimate the configuration of the manipulator and the pose of the manipulated object could be the key step to enable closed-loop manipulation. In return, visual tracking of highly articulated manipulators and objects raises interesting research questions itself.

Manipulator tracking The main problem of tracking highly articulated manipulators with several degrees of freedom is the large variety of their appearance. Using classical object detection methods on this problem would require an enormous amount of training data containing different configurations at different lighting conditions. In addition, classification results must be provided at real-time. On one side, the problem can be simplified in the context where a robot is observing its own manipulator by using a robot model representation and its joint encoder values. On the other side, the afore mentioned issue of inaccurate forward kinematics reduces the utility of these encoder values. The *primary aim* of this thesis is to investigate the effect of using visual perception and reported joint encoder values in conjunction, to enable tracking when either of the two information is not available or inaccurate.

Object tracking In contrast to the manipulator, we usually have no prior information about the pose of the object up to the point where the manipulator gets in contact with the object. The Amazon Picking Challenge 2015 [4] showed that detection and tracking of ordinary consumer products is still a challenging task. Amongst the 25 products used in the challenge were objects of varying appearance (e.g. non rigid objects), objects with meshed material (such as bins) and objects with reflective and translucent material. Classical machine learning approaches that rely on descriptive keypoint features such as SIFT fail in cases where texture is not available either because of lighting conditions or material properties. The *secondary aim* of this thesis is the investigation of object tracking without texture.

1.3 Outline

The current state of the art will be reviewed in chapter 2 followed by a statement of the contribution of this thesis. The core properties of the DART algorithm, on which this work is based on, will be presented and discussed in chapter 3, which also includes the extensions proposed in this work and the evaluation platform and methods. The extensions to the presented tracking approach will be evaluated in chapter 4 and compared to the base implementation. This work will conclude in chapter 5 with a summarized discussion of the results and propose future work for contributing to the research on articulated manipulator and object tracking.

Chapter 2

Related Work

This chapter is based on a previous literature review and research proposal that have been prepared for this thesis. The literature review has been extended by recent publications and a clarification where this work contributes to the current state of the art.

2.1 Sensor Modalities

2.1.1 2D Vision

Early work in this area has been limited to single 2D perception capabilities. Thus, a common approach was to project known 3D objects into 2D and to compare these with perceived data from images. Chliveros et al. [5], Teulière et al. [6], Choi et al. [7] and Azad et al. [8] rely on an exact 3D model of an object for tracking and use a particle filter to maintain multiple hypotheses of the object's pose. These hypotheses are projected into a 2D plane (prediction step) and evaluated at the observation step. At this observation stage, the predicted edges of the 3D model in the 2D plane are compared with the linear edges and shapes extracted from the actual 2D sensor data. As a positive aspect, these approaches do not require range sensors but they do require detailed knowledge of the object's dimensions beforehand and do not exploit features from colour or texture, excluding Choi et al. which uses SIFT keypoints as initial pose estimation.

Recent template matching approaches explore special properties of boundary edges. Muños et al. [9] for example use HOG features (histograms of oriented gradients) to find corresponding edges between the template and the image. Cao et al. [10] on the other side argue that HOG features lose pixel-level detail and instead use the cross-correlation between the Laplacian of Gaussian between templates and image patch. All template matching approaches have in common that they require a large database with templates of objects in all expected views.

2.1.2 3D Vision

Single Articulated Objects With the advancement in range sensing and especially the availability of consumer depth sensors like the Kinect, similar approaches have been extended to 3D perception. Krull et al. [11] apply a particle filter to estimate the object's pose in single images using RGB-D data by maintaining multiple hypotheses of the object pose. The observation distribution is generated from a discriminative learning method. Using example objects and an example background image, a random forest automatically learns useful features from the RGB-D data during a training phase. Once trained, the probability distribution of object class and coordinate location within the object is obtained for each new image.

The work of Shotton et al. [12] deals with complete body pose estimation and tracking of body parts. Instead of maintaining pose hypotheses of a single object, complete human skeleton hypotheses are evaluated. The observation distribution is generated by a discriminative classifier (random forest) using depth features for classifying 31 body parts. To achieve robustness and flexibility with their approach, a massive amount of training data is synthetically generated by rendering a body mesh at different sizes and with different joint configurations using motion capture. The same combination of random forests and primitive depth features is applied by Widmaier et al. [13] in a robotic manipulator context. However, they skip the intermediate task of classifying parts of the robot and instead predict its joint configuration directly.

A similar discriminative approach is used by Sharp et al. [14] to track articulated human hands. For reinitialisation after tracking loss, the hierarchical distribution of hand poses is learned by a discriminative approach using synthetically generated training data. Synthetic depth images are generated by sampling from a prior distribution of global hand translation and rotations, realistic wrist poses and six predefined finger and thumb states. For finding the optimal hand configuration, they apply a mixture of particle swarm optimization (PSO) and genetic algorithms. In particle swarm optimization, particles represent solutions in the search space whose state is updated by the particle's own best known solution and by the swarm's global best known solution. PSO does not require the gradient of the optimized function. Particles in PSO represent hypotheses as in particle filters, but their state is updated such that at least one particle's state is the global solution, whereas the current global solution of a particle filter is the expectation of the probability distribution formed by all particles (Monte Carlo expectation estimation). At tracking loss, these hypotheses are reinitialized by the prediction of the discriminative method on the new depth input. Each hypothesis of the particle swarm is evaluated by rendering its state and comparing it with the actual perceived sensor input. To form the next generation of the particle swarm, particles with bad hypotheses are randomized and re-initialized.

Combined Object and Manipulator Tracking Pauwels et al. [15] combined pose detection for reinitialization with pose tracking using dense motion and the information from depth data and colour. They rely on sparse SIFT keypoints sampled at equidistant viewpoints to obtain initial pose estimates and to reinitialize the tracker in case of tracking loss. The poses are continuously estimated using the shape and dense motion estimates of the object. This combination of both estimates enables them to cover difficult cases where the objected is blurred by motion or occluded (figure 2.1). The motion is represented as the optical flow on raw image data and on images which are augmented by objects rendered at their current estimated state.

(a) (b) (c)

Figure 2.1: SimTrack: Examples of correct classified object poses. (a) and (b) using *dense* estimation for small and blurry observations, (c) using *sparse* keypoints for occluded observations. (Source: [15], figure 13)

The multi-object tracking in SimTrack was extended for articulated objects [16] by incorporating the pose updates into a kinematic chain with constraints. The pose of each rigid part is estimated independently from each other. The links between parts can be constrained to define free and static joints. The complete update of the kinematic chain takes into account the individual pose updates per part and the movement constraints of parts to each other.

SimTrack was further extended for joint object and manipulator tracking [17], where the manipulator (arm and hand) is considered rigid after the joint values are used to articulate the model. As the approach relies on inaccurate joint encoder values, only the hand (which is assumed to be rigidly connected to the object) is tracked. Further, because the robot's surface is low textured, sparse features cannot be exploited.

The DART framework presented by Schmidt et al. [18, 19] aims to provide a general framework for articulated tracking using depth sensors. DART assumes that the articulated model is given as a set of rigid objects with transformation from the kinematic chain and a local signed distance function that gives the distance of a point to the hull of the object (and is negative inside). The optimal joint configuration that minimizes the distance of the model to

the observed point cloud is recovered by a gradient based optimization approach. Figure 2.2 shows this exemplary for hands and an object in a bimanual manipulation task. Compared to particle filters, this does not maintain explicit hypotheses but optimizes on the distances.

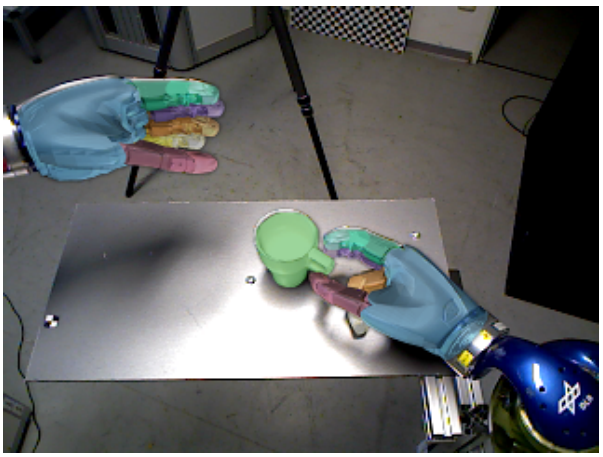


Figure 2.2: DART: Depth based estimation of robot configuration projected on top of the perceived RGB image. (Source: [19], figure 2)

2.1.3 Contact Sensors

Schmidt et al. [19] and Koval et al. [20] showed that contact sensing can be used to estimate the pose of an object or to reject visually proposed poses. Koval et al. [20] manipulated an object by pushing it over a planar surface and estimated its 2D position on the surface and orientation solely by contact sensors. They restrict their sensing to 9 strain gauges on the fingers and the palm of the hand, which only provides boolean information about contact (contact or no contact). By enhancing a particle filter to prevent particle starvation caused by the discrete states of the observation, they are able to estimate the position and orientation despite the low dimensionality of sensing.

As this approach does not scale from 3 DOF (2D position + 1D rotation) to 6DOF poses in 3D, contact sensing is best applied in conjunction with other sensing to eliminate unlikely hypotheses as in [19].

2.1.4 Kinematics and Constraints

The work of Schmidt et al. in [18] is extended in [19] to incorporate physical constraints, namely intersection of the manipulator with the object and contact with the object. If the manipulated object is rigidly connected to a manipulator, the joint encoder values can be an important source to improve the object's pose estimate.

On the other hand, kinematic chains can be used to constrain estimated configurations to physically reasonable and likely values and eliminate unlikely hypotheses [12, 14]. Whereas physical constraints are fixed, e.g. they always hold, kinematic constraints are soft, e.g. unlikely and unreasonable states might still occur.

2.2 Real-Time Capabilities

All of the particle filtering based approaches trade off the quality of the resulting pose estimation, as represented by the particles, with real-time constraints. An increasing number of particles cover a wider range of pose hypotheses but also increase the computation time.

Krull et al. [11] however showed, that using a good proposal distribution can reduce the required amount of particles to achieve similar results compared to particle filters using generic distributions with many more particles.

A general and common solution to meet real-time constraints is the parallelisation of algorithms and offloading of computations to GPUs. Exemplary, the independent state updates for particles in a particle filter [21] or the independent lookup of shortest distances of points to multiple models [18] can easily be parallelized.

2.3 Model-based and Model-free

Tracking of rigid or articulated objects usually requires knowledge of the dimensions of the object or the manipulator beforehand. In contrast, there exist approaches that model manipulated objects and kinematic chains online during manipulation.

Krainin et al. [22] modelled objects online by tracking the manipulator and the object at the same time during a grasping task. They address the issue of relying solely on either the object's appearance or the manipulator's joint values. Using only a single source of information can cause ambiguous and incorrect pose estimates for textureless and symmetric objects with lack of visual distinctive features. By using multiple sources of information they are able to map highly symmetric 3D objects with lack of visual features and in presence of noisy joint values.

The issue of faulty or absent of joint encoder values is addressed by Sturm et al. [23]. In their work, the kinematic chain of an arm is learned from scratch and enables adaptation in cases of malfunction, e.g. if a joint is blocked or its strength decreases over time. Si et al. [24] learn the variety of appearance of articulated objects as a hierarchical composition of parts as AND and OR nodes without providing labels. AND nodes define connection of parts, e.g. torso and limbs, and OR nodes represent different configurations of the same part. As a result, they can provide generic configurable templates for articulated objects.

The availability of accurate CAD models of robots and large online databases of 3D objects [25, 26] reduces the need to create models of common objects and enables the generation of huge synthetic training sets for discriminative approaches.

2.4 Features

Discriminative approaches, that learn the appearance of parts [12, 11, 15] or complete poses [14], rely on manually designed features or keypoints like edges, SIFT [15, 27] or depth features. In the RGB-D domain, where these features often neglect one of the channels, exploiting all the information is beneficial. Ren et al. [28] applied kernel descriptor for RGB-D scene labelling and showed superior performance of RGB-D descriptors compared to RGB or depth only descriptors.

RGB-D Features and Descriptors Learned features not only provide a way to classify 3D objects by machine learning algorithms (e.g. SVM), but also provide interest points for pose estimation by homography.

Gupta et al. [29] combined feature learning on RGB-D data with pixel-wise classification for segmenting objects in depth data. Object region proposals from contours are classified by a linear SVM using RGB-D features learned by a Region-based Convolutional Neural Net (R-CNN). Classified regions are separated in foreground and background pixels using random forest and low-level features such as the pixel coordinate and depth, distance to ground plane and gravity vector. The foreground pixels are allocated to superpixel and classified into one of the many categories.

Blum et al. [30] demonstrated an approach that learns RGB-D features in an unsupervised manner by convolutional k-means (CKM) around SURF interest points. Similar performance can be achieved by the hierarchical matching pursuit (HMP) approach [31] in classification tasks. It was shown that HMP is applicable for pose estimation, achieving appropriable results for weak-symmetrical objects with unambiguous viewpoints.

Joint Detection In the work of Jain et al. [32], the location of joints in 2D image sequences are learned from labelled raw image data using a convolutional neural network (CNN) by only providing simple motion features. These positions can be used to infer a kinematic model of articulated objects.

Tompson et al. [33] applied a CNN to joint detection using depth data and are able to estimate the location of occluded parts without the need to provide joint connectivities from a

kinematic model. They state that the inference of occluded parts is especially enabled by the hierarchical compound features learned in the CNN.

The spatial relation of compound parts is also exploited by Jiu et al. [34]. By learning a pixel-wise segmentation of human body parts incorporating spatial relation, the classification performance is increased with no additional computation costs during testing phase. As with most deep learning applications, they learn from raw depth data and do not require precomputed features.

2.5 Conclusion from State-of-the-Art

The majority of the presented work in the field of articulated tracking applies a discriminative approach to depth sensor data. Because of the high variance in shape and viewpoint of the tracked objects, a very large amount of training data is required to learn a sufficient distribution of articulations. If the articulated objects are known beforehand and a large variety of sensor data is available or can be generated, this is a promising approach.

However, relying solely on either the learned representation of objects or the kinematics of the manipulator fails in cases of low structured objects (highly symmetric, lack of texture and visual features), non-reliable sensor data (occlusions, reflection of sunlight, shadows in depth data, non-IR-reflective surfaces) and noisy or improper calibrated joint encoders.

Tracking and pose estimation of multiple objects is only considered in a small fraction of the presented work. Most of the presented work focuses either on the tracking of rigid objects or the tracking of manipulators; or assumes no interaction of tracked objects.

This thesis contributes towards the optimal integration of prior information from joint position sensing in a setting where a robot is visually observing its own manipulator. We base our work on the optimization approach presented by Schmidt et al. [18], that in its original implementation uses solely depth observations to find an optimal robot configuration.

Chapter 3

Methods

The first half of the following chapter introduces the humanoid robotic platform and its depth perception systems with their individual modalities. The core properties of the applied tracking approach – the Gauss-Newton optimization algorithm and the signed distance function as its objective – will be explained in more detail.

The incorporation of prior information and its effect on the optimization is presented in the second half of this chapter together with the procedure to obtain ground truth data for the evaluation.

3.1 Robotic Platform

The focus of this work is the application of tracking on humanoid robotic platforms where the manipulator is observed in a depth camera frame which is part of the robot's kinematic chain.

3.1.1 Hardware

The applied robot platform is the humanoid robot Valkyrie (figure 3.1) which is developed by the NASA. It is 1.8 m high and weighs about 125 kg. In total, it is actuated by 58 joints: legs (2×6), arms (2×7), hands (2×13), neck (3) and torso (3). Hence, the kinematic chain from the observation frame in the head to the hand frame consists of 10 actuated joints.

The robot will use depth data as its main source of information for estimating the configuration of the manipulator and the object pose. The MultiSense SL stereo camera and LIDAR depth sensor (figure 3.2a), which is mounted inside the head of the robot, provides two sources of depth images: (1) stereo matching at 15 Hz, (2) rotating LIDAR which generates a full scan at ⅙ Hz. An optional structure from light sensor (figure 3.2b), which can be mounted on top of the head, uses the known location of projected infrared (IR) dots to estimate the

Figure 3.1: Humanoid robot Valkyrie (Image credits: NASA)

(a) MultiSense SL

(b) Asus Xtion PRO Live

Figure 3.2: Depth perception systems (Image credits: (a) Carnegie Robotics, (b) Asus)

depth at these locations. The stereo cameras have their IR filter removed to enhance stereo matching by providing distinct keypoints from the IR pattern. The LIDAR data is not used for tracking because of its low update rate. The key properties of both depth sensors are compared in table 3.1. These properties are used for the back-projection to obtain the original 3D points.

3.1.2 Robot Model Representation

Part of this thesis is the adaptation of DART's reference implementation to robot model representations stored in the URDF (Unified Robot Description Format) format. URDF defines at a minimum a tree structure that resembles the kinematic tree of a robot with its joints and links. It can additionally provide, among other things, information about joint limits and the geometric properties of the links. These geometric properties can be defined as primitive shapes like spheres, boxes or cylinders, or alternatively refer to more complex triangle meshes.

The URDF plugin for DART is designed such that all parts of the kinematic chain that are connected to a given root frame are tracked. This enables us to neglect robot parts that are not involved in manipulation. Figure 3.3 shows such a model that is loaded from the `torso` frame upwards in the kinematic tree. Hence, it includes both arms, hands and the head with the sensors. Having all these parts connected through a kinematic chain allows us to track

	MultiSense SL Stereo Camera	*Asus Xtion PRO Live*
depth range	0.4 m - 10 m	0.8 m - 3.5 m
FOV	80° × 80°	58° × 45°
resolution	1024 × 1024	640 × 480
FPS	15 Hz	30 Hz
f_x, f_y	556.183 pxl	528.014 pxl
c_x, c_y	512 pxl	(320 pxl, 267 pxl)
b	0.07 m	0.075 m

Table 3.1: Comparison of depth sensors. FOV: field of view, FPS: frames per second, f: focal length, c: image centre, b: baseline

Figure 3.3: Robot model loaded in DART

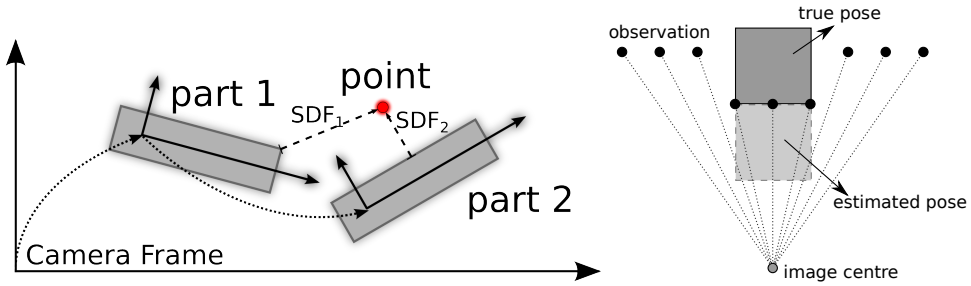
both hands in a bimanual grasping task with respect to the camera frame without invalidating kinematic constraints, such as overrunning joint limits or disconnecting links.

3.2 Signed Distance Function

The description of the signed distance function in this section is a reformulation based on the work by Schmidt et al. presented in [18].

The signed distance function (SDF) gives the shortest distance of a point in the observed point cloud to the robot model. It is signed in the sense that the distance is positive if the point lies outside the model, it is negative if the point is inside the model, and it is zero if the point lies exactly on the surface of the model. DART uses two kind of SDFs: the model SDF and the observation SDF.

Using the model SDF, the optimization assigns observed data to the robot model to find estimates that explain the observation. If a single (global) SDF is used for the articulated robot



(a) Model SDF (b) Observation SDF

Figure 3.4: Local model SDF and observation SDF (Inspired by [18] figures 3 and 6). (a) Two parts (gray blocks) connected by a kinematic chain are observed in the camera frame. The observed data point (red) is assigned to the closest part 2 after querying its distance in each part's local SDF. (b) The estimated pose of a part (light gray box) is blocking the line of sight between the image centre and the observed points. The alternative solution (dark gray box) is more likely selected as it is not violating the free space constraint.

model, this SDF needs to be computed for each individual joint configuration. For this reason, the global SDF is broken down into local SDFs for each rigid robot part. To assign points from the observation to the robot, each data point is queried in each of the robot's local SDFs in parallel (figure 3.4a). A data point is then assigned to the mesh with the smallest absolute distance or it is rejected, if it is too far away from any rigid part.

In addition to considering observed data and the robot model, which is defined as *positive information* in DART, the optimization is also using *negative information*. If a point is observed in 3D, there cannot be anything in between this point and the camera centre (figure 3.4b). The observation SDF uses the distance transform on the observation and the estimated configuration of the model from the model SDF to constrain models to not be in free space.

3.3 Gauss-Newton Algorithm

The *Gauss-Newton* algorithm is a method for finding the optimal set of parameters $\mathbf{p}$ that minimize the sum of squares of non-linear vector-valued objective functions $e(\cdot)$

$$\hat{\mathbf{p}} = \arg\min_{p} \sum_{i=1}^{N} \left[e_i(\mathbf{p})^2 \right]. \tag{3.1}$$

The Gauss-Newton algorithm is derived from the *Newton* algorithm for finding roots and extremal points which itself is based on Taylor series expansion. The Newton method for

finding extremal points of the scalar objective function $f(\cdot)$ is an iterative procedure

$$x_{t+1} = x_t - \frac{f^{(1)}(x_t)}{f^{(2)}(x_t)} \tag{3.2}$$

starting at an initial state x_0 and using the first- and second-order derivatives ($f^{(1)}$, $f^{(2)}$) of the objective function to converge to a local minimum.

For finding the minimum of a vector-valued function $e(\cdot)$ with respect to a parameter vector $\mathbf{p}$, the first- and second-order partial derivatives are used. In particular, the first- and second-order derivatives in equation (3.2) are replaced by the gradient $\nabla e(\mathbf{p})$ and the Hessian $\nabla^2 e(\mathbf{p})$ of the objective function e. The partial derivatives of the vector-valued function $e(\cdot)$ with respect to the parameter vector $\mathbf{p}$ are aggregated in the *Jacobian* matrix $J \in \mathbb{R}^{i \times j}$ with elements

$$J_{i,j} = \frac{\partial e_i(\mathbf{p})}{\partial p_j}. \tag{3.3}$$

Hence,

$$\nabla e(\mathbf{p}) = J^\top e(\mathbf{p}) \tag{3.4}$$

$$H(e) = \nabla^2 e(\mathbf{p}) = J^\top J + \sum_{i=1} e_i(\mathbf{p}) \nabla^2 e_i(\mathbf{p}). \tag{3.5}$$

For small e close to an optimum, the Hessian $H(e)$ can be approximated by

$$H(e) \approx J^\top J \tag{3.6}$$

neglecting the second-order derivatives. This approximation is used in the Gauss-Newton algorithm and the parameter update step then becomes

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \Delta\mathbf{p} \tag{3.7}$$

$$= \mathbf{p}_t - H(e)^{-1} \cdot \nabla e(\mathbf{p}) \tag{3.8}$$

$$= \mathbf{p}_t - \left(J^\top J\right)^{-1} \cdot J^\top e(\mathbf{p}). \tag{3.9}$$

The parameter update $\Delta\mathbf{p}$ is proportionally driven by the gradient of the objective function and scaled by the approximation of the Hessian. Because partial derivatives only consider the local change of the objective function, these gradient based approaches can only converge to local minimums.

3.4 Prior Information

Additional information can be incorporated during the tracking process either indirectly by shaping the input data on which the optimization is operating on, or directly by biasing the objective function. In the latter case, additional objective functions are integrated in the Gauss-Newton algorithm by blending their gradient and Hessian approximation with other objective functions.

The full parameter vector $\mathbf{p}$ in DART contains the 6D pose of the root frame and the N joint positions:

$$\mathbf{p} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & | & q_1 & \cdots & q_N \end{bmatrix} \tag{3.10}$$

with x, y, z for the translation, ϕ, θ, ψ for the orientation in radiant Euler angles, and $q_1 \dots q_N$ the individual joint position values.

The position of each joint in the kinematic chain can be sensed directly through its joint position encoder. We will refer to these internally sensed joint positions as *reported* joint positions. The joint positions that are optimised on the external depth observation will be referred to as *estimated* joint positions.

3.4.1 Known Frame Pose

As mentioned before, we know that the observation frame is part of the kinematic chain and rigidly connected to the head frame. Hence, observing this head frame is impossible.

To prevent movement of the head frame in between optimization iterations, the pose update in the parameter update vector $\Delta\mathbf{p}$ needs to be null. After the gradients and the Hessians of all other objective functions have been blended, the resulting gradient and Hessian is recomputed such that the pose update is removed from the parameter update vector. From equation (3.9) the current parameter update is computed and its pose update is removed:

$$\Delta\mathbf{p} = \left(J^\top J\right)^{-1} \cdot J^\top e(\mathbf{p}) = \begin{bmatrix} \mathbf{0} & | & q_1 & \cdots & q_N \end{bmatrix}. \tag{3.11}$$

This modified parameter update vector is then used to compute a gradient that does not enforce a change in pose:

$$J^\top e = J^\top J \cdot \begin{bmatrix} \mathbf{0} & | & q_1 & \cdots & q_N \end{bmatrix}. \tag{3.12}$$

The final merged Hessian and the new gradient are the Gauss-Newton parameters that are then used to compute the final parameter update.

3.4.2 Reported Joint Positions

The reported joint positions are used in addition to the observation to constrain the optimization to possible solutions that are close to the internally sensed state. This is achieved by blending a joint position based objective function into the SDF based objective function resulting in an objective function whose minimum is globally determined by the reported state and locally determined by depth observations. We will also refer to this process as biasing the objective towards the reported pose.

Two different kind of objective functions have been selected for this purpose which compute a weighted deviation of the estimated joint configuration $\mathbf{q}$ from reported joint configuration $\mathbf{r}$. The deviation is defined as the L2 norm ($\|\cdot\|_2$) of the difference vector $\mathbf{d} = \mathbf{r} - \mathbf{q}$. For all objective functions, r_i and q_i denote the individual joint positions in their corresponding configuration vector.

1. Weighted L2 norm of the estimated joint position deviation:

$$e_1 = w \cdot \|\mathbf{r} - \mathbf{q}\|_2 = \|w \cdot \mathbf{r} - \mathbf{q}\|_2 \tag{3.13}$$

$$= w \cdot \sqrt{\sum_{i=1}^{N} (r_i - q_i)^2} \tag{3.14}$$

   with scalar weight $w > 0$.

2. Sum of squares of individually weighted joints:

$$e_2 = |\mathbf{r} - \mathbf{q}|^\top Q |\mathbf{r} - \mathbf{q}| \tag{3.15}$$

$$= \sum_{i=1}^{N} \left[(r_i - q_i) \cdot \sum_{j=1}^{N} (r_j - q_j) \cdot \omega_{i,j} \right] \tag{3.16}$$

   with the weight matrix $Q \in \mathbf{R}^{N \times N}$ and its elements $\omega_{i,j}$. As an error metric, the weight matrix Q must be *positive-semidefinite*. That is, $\forall \mathbf{x} \in \mathbf{R}^n : x^\top Q x \geq 0$.

Whereas the first objective function (equation (3.14)) uses a common scalar weight for all joint deviations, the second objective function (equation (3.16)) can penalize specific joint deviations and specific linear combinations of deviations. This enables us to enforce a reported state onto the estimated state for specific parts of the robot while keeping other parts free to be optimized by other objective functions.

The gradient and Hessian approximation for the blending process are obtained through the objective function's Jacobian as shown equations (3.4) and (3.6). The respective partial derivatives for the Jacobian are:

1. Weighted L2 norm of the estimated joint position deviation:

$$J_{ji} = \frac{\partial e_1}{\partial q_i} = -w \frac{r_i - q_i}{\|\mathbf{r} - \mathbf{q}\|} \tag{3.17}$$

2. Sum of squares of individually weighted joints:

$$J_{ji} = \frac{\partial e_2}{\partial q_i} = -\left[\sum_{j=1}^{N} d_j q_{i,j}\right] + \left[\sum_{j=1 \neq i}^{N} d_j q_{j,i}\right] \tag{3.18}$$

3.4.3 Observation Filtering

An alternative solution to incorporate the joint position prior into the optimization process is to filter the depth observations of the observed manipulator. Given the camera parameters and a robot model that is articulated by the current joint positions, the robot's parts are projected into the camera image. For this camera image, a binary mask is storing locations that are occupied by one of the robot's parts. This binary mask is further expanded on its borders by a dilation operation with a rectangular kernel of variable size.

An example of such a manipulator mask and the filtered observation is shown in figure 3.5. By expanding the mask, the filtered observation includes more related sensor data from the manipulator itself but also more unrelated sensor data from its surrounding.

3.5 Ground Truth Data Collection

To evaluate the tracking performance and verify the effect of the joint position prior, reference measurements of the robot state are required. The Vicon motion caption system is selected to externally track the pose of IR reflective markers rigidly attached to the robot.

3.5.1 Setup

To measure the transformation from the pelvis frame to the left hand palm frame, reflective markers are rigidly mounted with standoffs at the specific robot frames. The photograph in figure 3.6 shows the highlighted location of the markers on the `pelvis` frame (green) and the `leftPalm` frame (red).

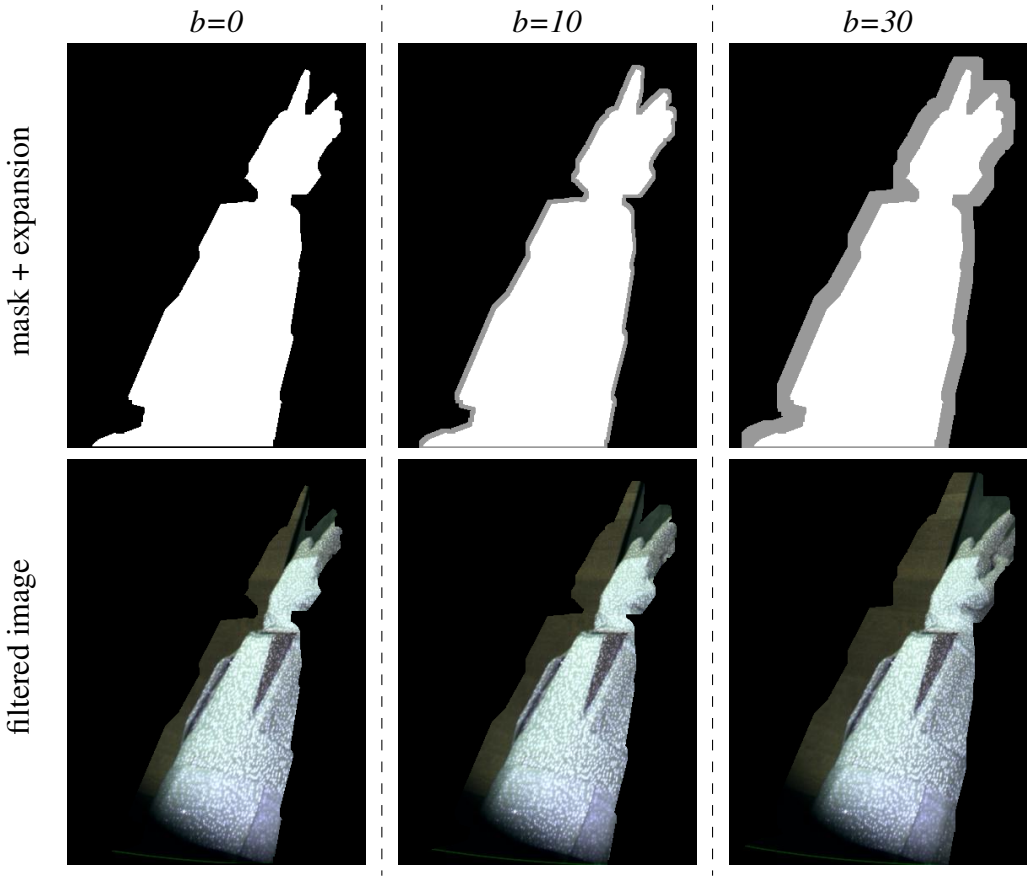


Figure 3.5: Projection of manipulator into camera image and filtering with variable kernel size ($b = \{0, 10, 30\}$pxl). *Top row*: Original mask (left) in white from projected manipulator, expanded border (middle, right) in gray, *Bottom row*: Filtered image at reported manipulator configuration.

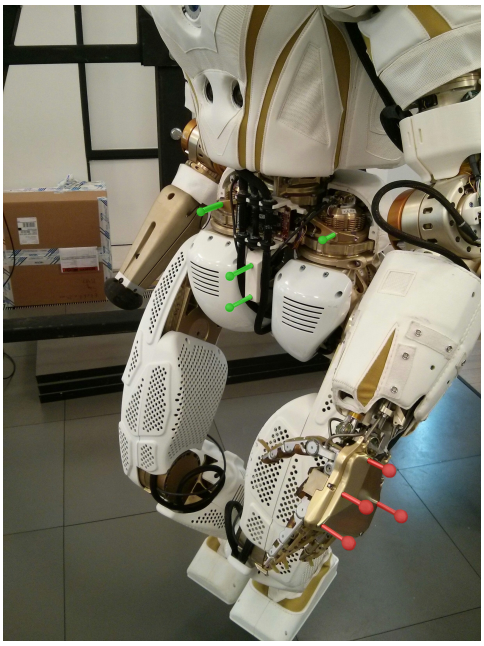


Figure 3.6: Location of Vicon marker on pelvis (green) and left hand (red)

3.5.2 Hand Pose

Each group of markers has its own coordinate system with its origin in the centre. The drawing in figure 3.7 visualizes the transformations between the Vicon world frame, the Vicon marker frames and the robot frames. The transformations $T_{w\to vp}$ and $T_{w\to vh}$ are the reported poses of the Vicon marker in the world frame. To obtain the transformation $T_{p\to h}$, which gives the hand pose in the pelvis frame, we need to find the rigid transformation between the Vicon marker and robot frame ($T_{m\to p}$ and $T_{m\to h}$).

The rigid transformation between marker and robot frames can be obtained by a least-squares optimization on corresponding points in both frames [35]. Given corresponding points m_i in marker frame and x_i in the robot frame, the optimal projection of marker points from the marker frame into corresponding points in the robot frame becomes

$$T_{x\to m} = \arg\min_{R,t} \frac{1}{N} \sum_{i=1}^{N} \|x_i - (R \cdot m_i + t)\|_2 \tag{3.19}$$

where $T_{x\to m}$ denotes the homogeneous transformation matrix containing the rotation matrix R and the translation vector t. The points x_i in the robot frame are here interchangeable for points p_i in the `pelvis` frame and h_i in the `leftPalm` frame. In particular, as result of this optimization we obtain $T_{p\to m}$ and $T_{h\to m}$.

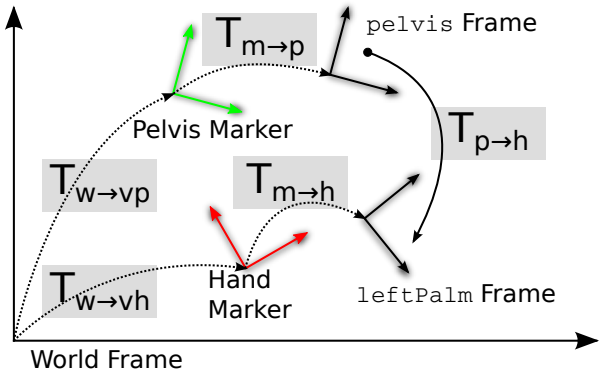


Figure 3.7: Transformations between Vicon markers (*pelvis* - green, *hand* - red) and robot frames

The pose of the `pelvis` and `leftPalm` frame in the Vicon world frame can be expressed by the transformations

$$T_{w\to p} = T_{w\to vp} \cdot T_{m\to p} = T_{w\to vp} \cdot T_{p\to m}^{-1} \tag{3.20}$$

$$T_{w\to h} = T_{w\to vh} \cdot T_{m\to h} = T_{w\to vh} \cdot T_{h\to m}^{-1}. \tag{3.21}$$

The transformations between these robot frames then finally becomes

$$T_{w\to h} = T_{w\to p} \cdot T_{p\to h} \tag{3.22}$$

$$T_{w\to vh} \cdot T_{h\to m}^{-1} = T_{w\to vp} \cdot T_{p\to m}^{-1} \cdot T_{p\to h}$$

$$T_{p\to h} = \left(T_{w\to vp} \cdot T_{p\to m}^{-1}\right)^{-1} \cdot T_{w\to vh} \cdot T_{h\to m}^{-1}. \tag{3.23}$$

Given the transformations of the Vicon markers ($T_{w\to vp}$ and $T_{w\to vh}$) at each time instance, and the rigid transformation between robot frames and their attached markers ($T_{p\to m}$ and $T_{h\to m}$) we can recover the ground truth pose of the hand in the pelvis frame ($T_{p\to h}$) using equation (3.23).

3.5.3 Joint Position Offset Calibration

The availability of a ground truth transformation within the robot kinematic chain enables us to compare the reported hand pose from forward kinematics with the measured hand pose and further, calibrate the joints in the kinematic chain such that this deviation is minimized.

To minimize the joint position deviation and to obtain correction offsets, the same optimization approach as described in [2] is applied. That is, given a correction function that maps a reported joint position value to its corrected value, $f_{corr} : q_{rep,i} \mapsto q_{corr,i}$, and a mapping

from this joint space configuration to the task space pose of the manipulator using forward kinematics, $\phi : \mathbf{q} \mapsto \mathbf{p}$, with the pose $\mathbf{p} \in SE(3)$, the optimal correction parameters can be found by minimising

$$e = \frac{1}{N} \sum_{i=1}^{N} \| \phi \left(f_{corr}(q_{rep,i}) \right) - \mathbf{p}_i \|_2 \tag{3.24}$$

for N samples, respectively poses. The average error is selected over the summed error to have comparable costs between varying lengths of sample sequences.

Two correction functions f_{corr} have been selected. The constant correction

$$f_{corr}(q_{rep}) = q_{rep} + q_{offset} \tag{3.25}$$

assumes a constant offset that is independent from the reported joint position value itself. Here, only one parameter per calibrated joint has to be optimized. A linear correction function

$$f_{corr}(q_{rep}) = m_1 \cdot q_{rep} + q_{offset} \tag{3.26}$$

assumes an additional relation to the reported joint position value itself. In this case twice as many parameters need to be optimised.

To prevent overfitting of the optimized correction parameters on the data set, the optimization result must be applied to a dedicated test set that has not been used for optimization. This is especially the case for more complex correction functions.

Chapter 4

Evaluation

The articulated tracking approach and the incorporation of prior information is evaluated on two different data sets. The data set for the *first experiment* contains depth observations from a manipulator that is moving close to a table and an object. This experiment will be used to compare different approaches to incorporate prior information in the tracking process.

The data set for the *second experiment* contains stereo camera observations of the manipulator at different viewing angles without nearby objects. This data set additionally contains ground truth hand poses that were measured by the Vicon system. The main purpose of this experiment is the comparison of the estimated hand pose with the measured ground truth.

4.1 Experiment 1: Incorporating Prior Information

This section will evaluate the application of prior information in a setting where a robot hand moves close to an object and a table. Two sources of depth information are considered: enhanced stereo matching using IR dot projection (denoted as *stereo*) and structured light (denoted as *xtion* as in Asus Xtion).

4.1.1 Hypotheses

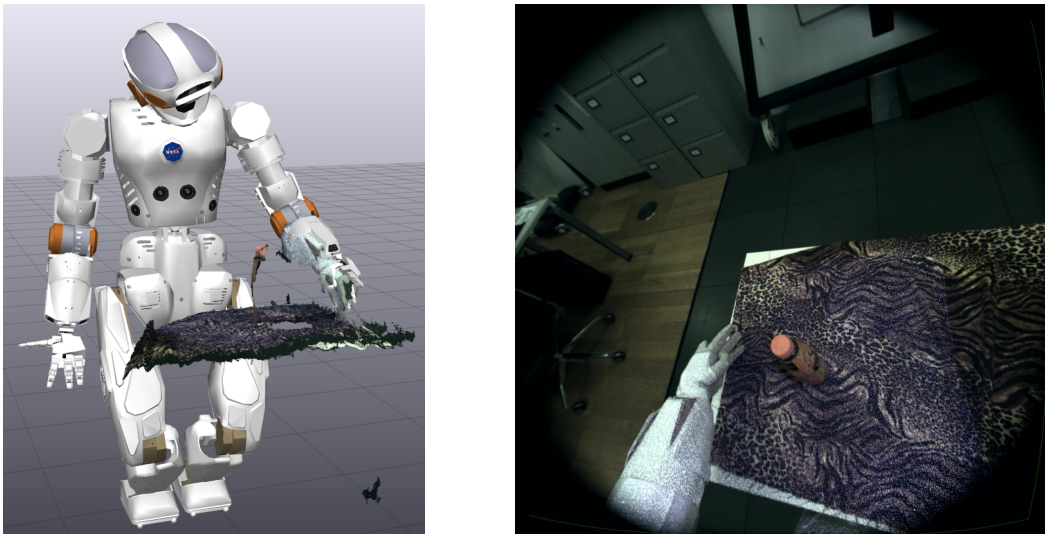
In the base implementation of the assessed tracking approach, its gradient is determined by the signed distance function. Hence, the optimization is driven by the observation once the iteration is initialized with the reported state. Using additional information that effects the gradient can improve the state estimation by driving it away from distracting observations.

Hypothesis 1 *(Distracting sensor readings)*
Distracting sensor readings will impair the tracking performance of the manipulator.

Hypothesis 2 *(Use of prior information)*
Using prior information will reduce the negative effects of distracting sensor readings. Increasing weights will further decrease the dependency on the sensor readings.

4.1.2 Setup

A robot was placed in front of a table with a bottle on it. The left hand was moved from close to the table plane towards the object. Once the object is pushed, the hand moved up and back to its initial position. The setup, as it is perceived by the robots stereo vision in the initial state, is shown in figure 4.1.



(a) Point cloud and reported robot state

(b) View from left camera

Figure 4.1: Experiment 1: Setup with robot in front of table. Initial state where manipulator is observed close to table plane.

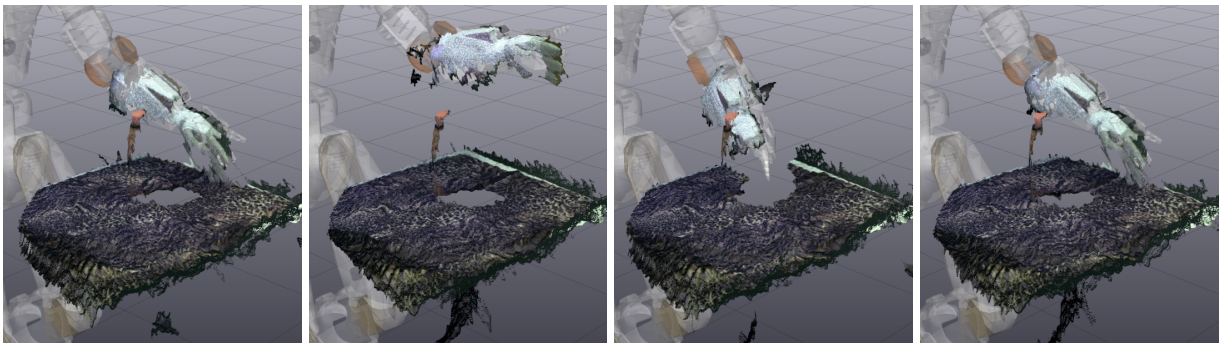
The motion sequence is divided into 5 states which change by 4 movements. The movement phases are defined in table 4.1. Two of the phases define movements of the manipulator away from objects after being in contact with them. These phases are highlighted in the table and in the plots.

The perceived point cloud of the first three states and the final state are shown in figure 4.2. Especially the movement phases 2 and 4, respectively movements starting in states shown in figures 4.2a and 4.2c are of interest.

During the movement, depth data is collected simultaneously from the MultiSense stereo sensor and the Asus Xtion structured light sensor. Hence, the stereo block matching benefits from the distinctive IR dots.

	time (s)	*movement description*
1	0…3	arm resting on table
2	3…10	upward from table
3	10…20	towards object
4	20…30	away from object
5	30…35	downward towards table

Table 4.1: Phases of arm movement in experiment 1. Highlighted rows emphasise movements away from objects like the table or bottle.



(a) start (t=0) (b) moved upwards (t=10) (c) object contact (t=22) (d) end (t=35)

Figure 4.2: Sequence of states between movements in experiment 1 (time in seconds)

4.1.3 Results

For this experiment, we are interested in the deviation of the estimated state from the reported state. This will allow us to (1) see how the estimation on the observed point cloud is deviating from its initial reported pose, and (2) investigate the effect of the prior objective compared to the base signed distance objective. We will refer to this deviation as *error*.

The error in joint and task space is computed as L2 norm (Euclidean distance) of its components and plotted over time. In joint space these components are the left fingers (13 DoF: 3 × *leftIndexFingerPitch*, 3 × *leftMiddleFingerPitch*, 3 × *leftPinkyPitch*, 3 × *leftThumbPitch* and 1 × *leftThumbRoll*) and the left arm (7 DoF: *leftShoulderPitch/Roll/Yaw*, *leftElbowPitch*, *leftForearmYaw*, *leftWristRoll/Pitch*). In task space, the 3D position (x, y, z) and the 3D orientation (*roll*, *pitch*, *yaw*) of the left hand (frame: *leftPalm*) are computed via forward kinematics on the reported and estimated robot configuration.

Joint Prior Objective with Common Weight

The following plots compare the joint and task space errors for the two depth sources for common weights in the range 0 to 5. The common weighting scheme *Weighted L2 norm of*

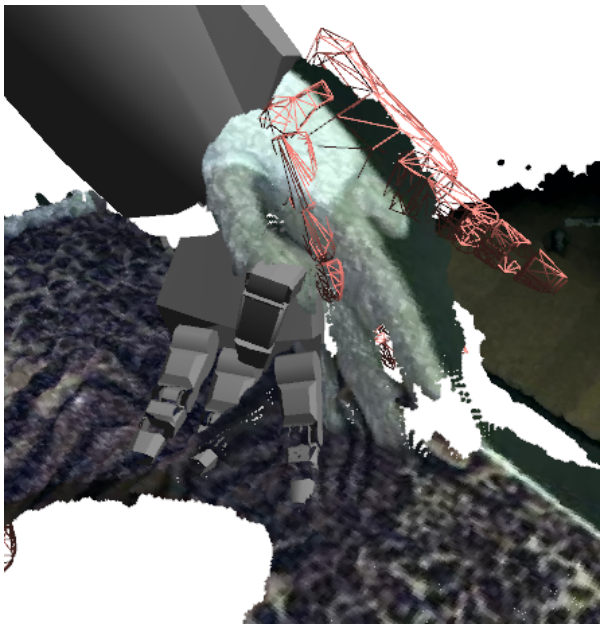


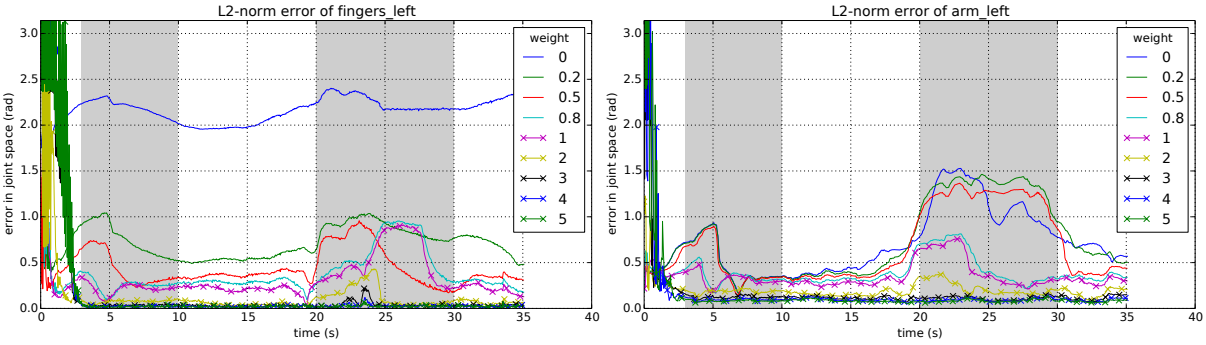
Figure 4.3: Close points from the table plane are associated to the fingers (solid gray mesh). Using solely the signed distance as objective, the gradient based optimization finds a local minimum in regions with dense sensor readings close the initial reported state (red wired mesh).

joint position deviation (objective function equation (3.14)) is applied. A weight of 0 indicates that no prior is used at all.

Stereo Figure 4.4 compares the joint space error for left fingers and arm. For both parts, after the optimization reaches the steady state, the maximum error is reached when no prior information from the reported configuration is used. In this case, the configuration is solely optimized using the distance to the observed point cloud. If the manipulator is close to a large concentration of points, the optimization will relate these readings to the manipulator (figure 4.3). The general trend is that the error decreases with increasing weight. This is not true for the arm movement in the second half of phase 4 ($t = [20, 30]$), where weights of 0.2 and 0.5 result in larger errors compared to no prior.

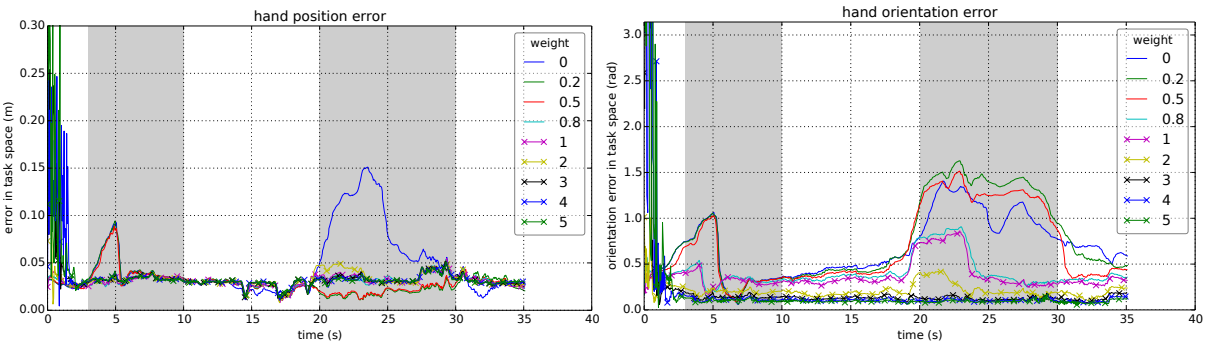
The largest decrease in error can be seen for the finger joints when using already a small weight of 0.2. A similar effect is not present for the arm joints. This is presumably because the robot can only observe the lower part of its arm and hence the arm configuration is mostly determined by its initial state close to the reported state.

As the hand position and orientation only depends on the arm configuration but not the finger configuration, we expect some relation between the joint error of the arm and the pose error on the hand frame. We can see this relation, when comparing the joint space error for the arm in figure 4.4b and the hand pose error in figure 4.5. In particular the error increases in phases where the hand moves upwards and when it moves away from the object (phases 2 and 4). The position error can be reduced significantly when using a prior with low weight (0.2), whereas the orientation error reduces only when using prior weights larger or equal than 0.8.



(a) finger joints (b) arm joints

Figure 4.4: Joint space error for finger and arm joints (Exp. 1, *stereo* data). The finger joint deviation (a) can already be reduced by a weak bias towards the reported pose (0.2, 0.5), while the arm joints (b) require a higher weight (≥ 0.8).

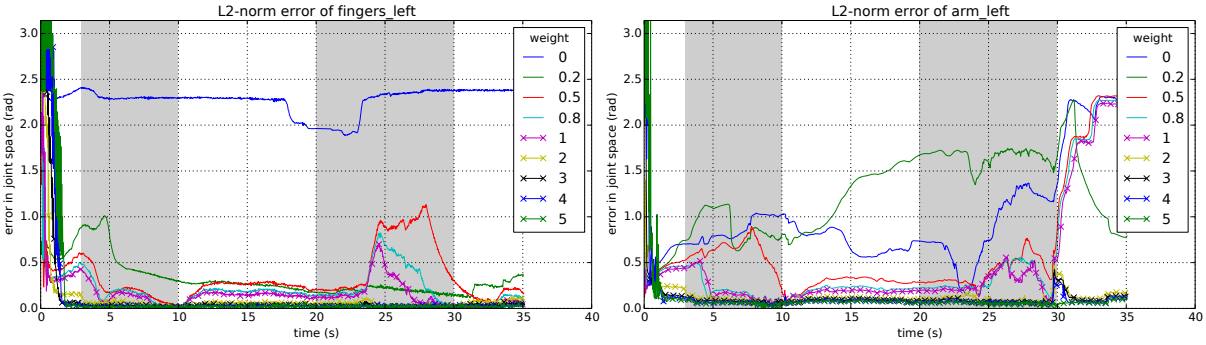


(a) position error (b) orientation error

Figure 4.5: Task space error for left hand pose (Exp. 1, *stereo* data). (a) The hand position error can already be reduced by minimal weighting of the prior objective. (b) Small weights actually increase the error and higher weights of at least 0.8 are required to enforce the reported pose.

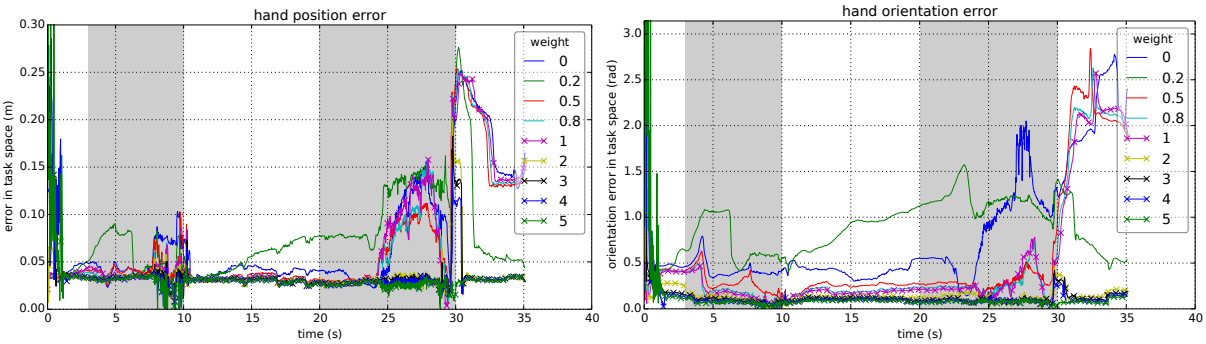
Asus Xtion Using the structured light sensor Asus Xtion, the behaviour of decreasing error with increasing weight is comparable to that seen for the stereo matching sensor. Similar to the stereo sensor (figure 4.4a), the error on the finger joints reduces significantly when using a small weight of 0.2 (figure 4.6a). In contrast to the small errors on the arm joints when using stereo in the phase of moving towards the object (figure 4.4b), the error in this phase when using the Xtion sensor is fairly large for no and low weighted prior (figure 4.6b).

As before, the hand pose error is only effected by the arm joint errors and thus the hand position error shown in figure 4.7a is large in the same moving phase close to the object. For using the structured light sensor, a common weight of at least 2 is required to drive the solution towards the reported hand position.



(a) finger joints (b) arm joints

Figure 4.6: Joint space error for finger and arm joints (Exp. 1, *xtion* data). Comparable to stereo data set, the finger joint deviation (a) is already reduced by small weights.



(a) position error (b) orientation error

Figure 4.7: Task space error for left hand pose (Exp. 1, *xtion* data). The pose error on this data set has a different characteristic than on the stereo data set. Large weights above 2 are required to enforce the reported pose.

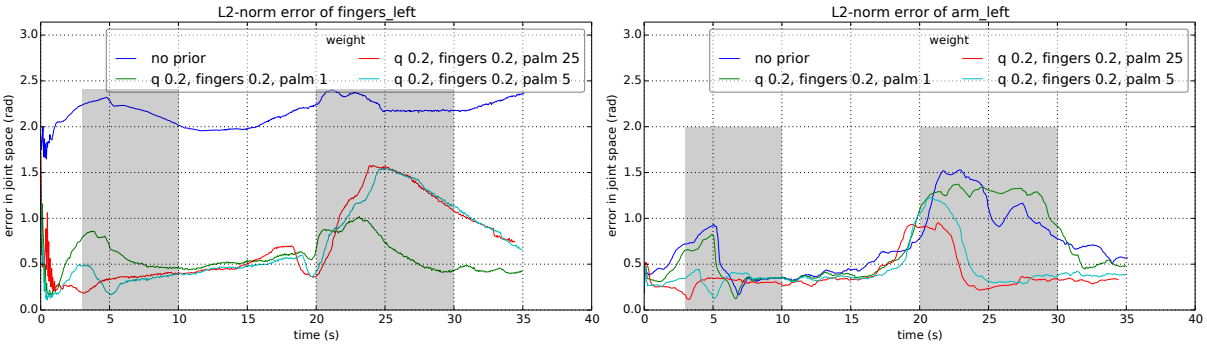
Joint Prior Objective with Individual Weights

Individual weighting is applied to stereo depth data only. This weighting scheme enables us to weight each combination of joint deviations separately as defined by equation (3.16). For simplicity, only single joint deviations are weighted. That is, the weight matrix Q will be a diagonal matrix where only the diagonal elements $q_{i,i}$ will be defined.

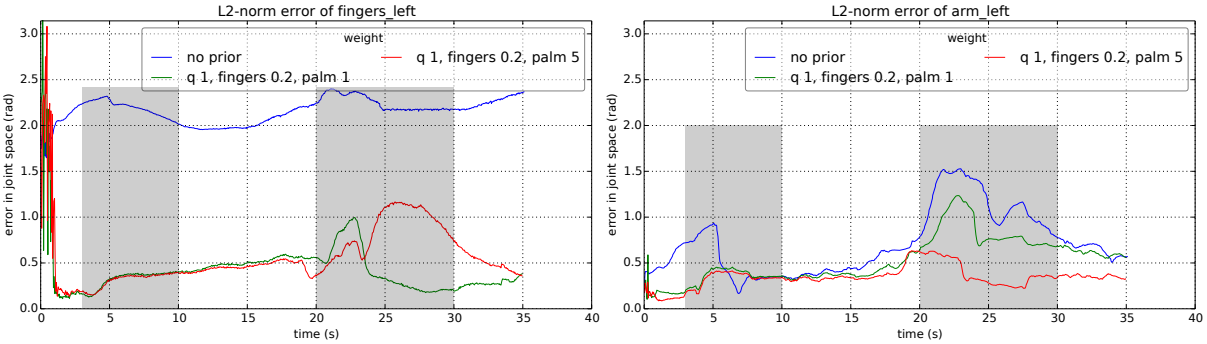
In this scenario, the two palm joints (`leftWristRoll` and `leftWristPitch`) that connect the hand with the arm are weighted with increasing weights, while the remaining joints are kept constant at either low weights (0.2 for finger joints, 0.2 for remaining joints) or kept constant at normal weights (0.2 for finger joints, 1.0 for remaining joints).

Because we specifically weight both palm joints, the joint space error of the finger and the arm joints are effected in different ways (figure 4.8). While the arm joint error reduces with

increasing palm weight, the finger joint error actually increases. The reason for this is that both palm joints, as part of the arm kinematic chain, determine the pose of the hand palm. While the hand pose is biased towards the reported pose by the prior, the fingers do not benefit from this prior information to the same extent.



(a) finger joints (b) arm joints



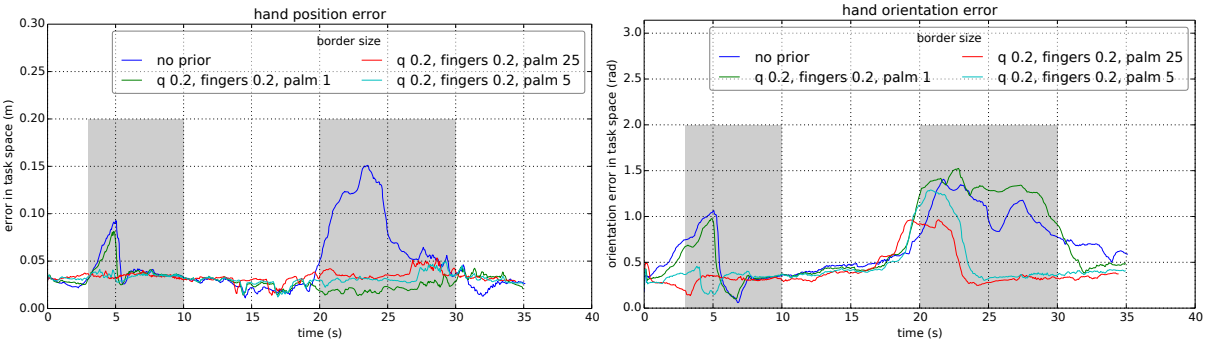
(c) finger joints (d) arm joints

Figure 4.8: Joint space error for individual weighting (Exp. 1). Weights on palm joints are increased (1, 5, 25) while remaining joints are kept constant at low weights (a,b) or at normal weights (c,d). Finger joint errors (a,c) increase with increasing palm weights, while arm joint errors (b,d) are reduced.

The task space error (figure 4.9) is similarly related to the weight of the palm joints as the arm joint error. By particularly enforcing the reported position on two joints, the hand pose error is reduced while avoiding overshooting effects as seen when applying common weights to all joints.

Self-Observation Filtering using Joint Prior

An alternative to using prior information in the optimization is to use prior information to provide the optimization with relevant data. In this sense, the reported state of the robot is used to filter its observation to only contain points close to the predicted location of the manipulator.



(a) position error

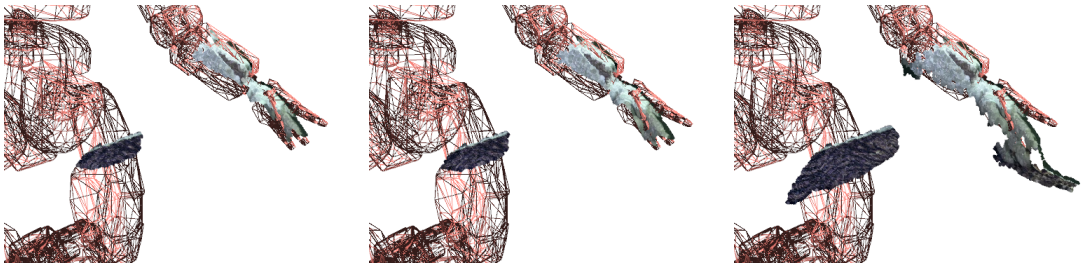
(b) orientation error

(c) position error

(d) orientation error

Figure 4.9: Task space error for individual weighting (Exp. 1). Weights on palm joints are increased (1, 5, 25) while remaining joints are kept constant at low weights (a,b) or normal weights (c,d). The hand pose error is reduced in each case for increasing palm weights.

The filtering of observed data is achieved by projecting the manipulator into the field-of-view of the robot's perception system. Each point in the disparity image that is occupied by one of the robot's projected parts is kept while all remaining values are removed. This filtered region can be further expanded on the borders of the self perceived parts. Figure 4.10 demonstrates the filtered perception on stereo data for three different values for the border expansion.



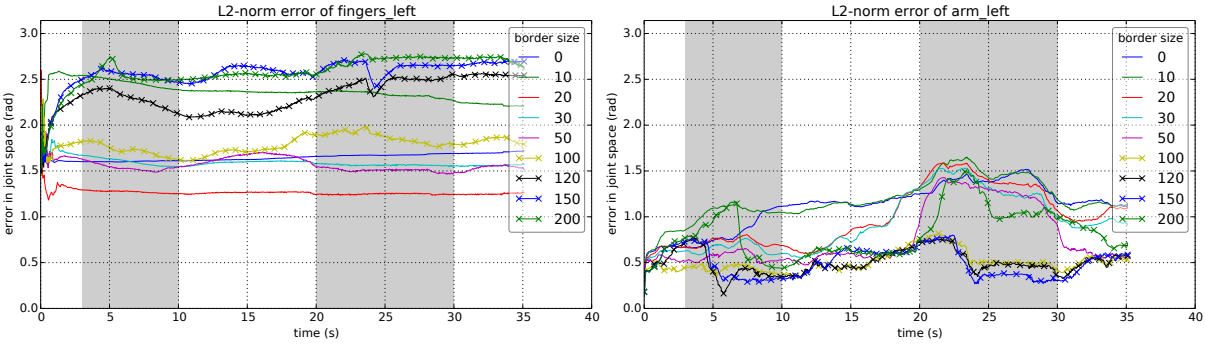
(a) normal mask (b) expanded by 10 pxl (c) expanded by 100 pxl

Figure 4.10: Point cloud filtering by self-observation with different expended regions.

In comparison to the joint position prior applied to the objective function, the joint and task space errors for self-observation filtering are given in figures 4.11 and 4.12. The error is again given as the deviation from the reported pose, however the joint position prior is not used in the optimization and hence, the solution is not directly driven towards the reported configuration.

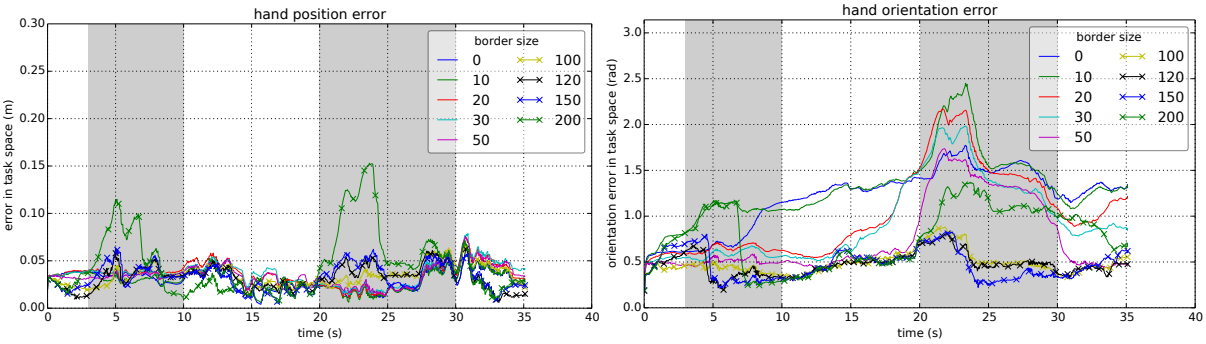
As it can be easily seen for the joint space error in figures 4.11a and 4.11b, there is an individual optimal value for the region expanding that minimizes the error for the finger joints (20) and the arm joints (150). While the finger joint deviation in the optimal expanded case is larger than the joint deviation using small weights (figure 4.4a), the optimal arm region expansion achieves similar results compared to the joint prior with weights around 1 (figure 4.4b).

For large expansions of the self-observation mask (200), the hand position error in figure 4.12a behaves similar to the case where no prior is used on the unfiltered observation (figure 4.5a). Further, in phases of contact with surrounding objects a smaller region expansion is preferable over larger borders. The reason for this is that large self-observation masks will include nearby objects, which again will distract the optimization without prior. A larger region expansion is preferable in phases where the manipulator is moving in free space (time between highlighted phases). The hand orientation error behaves similar to the arm joint error, in particular an optimal region expansion (150) emerges and larger region expansions are favourable in general.



(a) finger joints (b) arm joints

Figure 4.11: Joint space error for filtered perception (Exp. 1). Individual border expansions emerge for minimizing finger joint deviation (20) and for minimizing arm joint deviation (150).



(a) position error (b) orientation error

Figure 4.12: Task space error for filtered perception (Exp. 1). Smaller border expansions are preferable over large borders in phases where the manipulator is close to objects (highlighted).

4.1.4 Interpretation

With the signed distance as the only objective function, the optimization is driven away from the reported state, with which it is initialized, whenever nearby but unrelated depth data is assigned to the robot model. Hence we conclude that hypothesis 1 is true. By augmenting the objective function with an objective related to the reported state as the reference, this effect can be reduced.

Across all weighting schemes and depth sources, we can see that weighting in respect to the deviation from the reported pose: (1) reduces the deviation of finger joint positions over the whole sequence of the trials by already choosing a small prior weight, (2) contributes most in movement phases that follow states where the manipulator is in contact with the scene (table, object). The application of individual weighting of joint deviations showed that by selecting specific joints, the task space error can be reduced while keeping the flexibility of preferring

other objectives for the remaining parts of the robot model. For both assessed weighting schemes, it turned out that weights need to be chosen carefully to prevent overshooting on the reported state and hence the oscillation of the estimate state. Neglecting the issue of oscillating around the optimum, hypothesis 2 is also considered to be verified.

The self-observation filtering of the perception showed that using prior information to shape the input to the optimization also counteracts the issue of distracting observation. By using prior information to filter relevant data, the reported robot state only indirectly improves the optimization result. The decoupling of the optimization from the reported joint values prevents cases of overfitting on the reported state by limiting the bias from prior information.

The assessed prior information is by design only applicable to the robot model. The tracked object remains to be driven by the signed distance function and hence relies on relevant depth data assigned to the object model.

4.2 Experiment 2: Ground Truth Comparison

An experiment was conducted that contains depth data measurements from the manipulator alone, without distracting depth readings nearby. As in the previous experiment, forward kinematics on the reported and estimated joint configuration was used to obtain the pose of the manipulator in task space. Additionally, the Vicon system introduced earlier provides ground truth data of the hand palm pose.

4.2.1 Hypotheses

The experiment has been designed to (1) apply the tracking of the manipulator in a scene without distractions, and (2) to compare the reported and estimated robot configuration with the ground truth poses reported by the Vicon tracking system.

Without distracting sensor readings, we expect that the use of the joint position prior will not have a strong impact on the deviation from the reported robot configuration. If the manipulator is fully observed in the depth data, tracking the manipulator should result in a hand pose close to the true hand pose which itself should coincide with the reported hand pose.

Hypothesis 3 *(No effect of joint position prior)*
There is no significant effect of applying a joint position prior on observations containing solely the manipulator.

Hypothesis 4 *(Matching hand pose)*
Reported and estimated left hand poses agree with the measured hand pose.

4.2.2 Setup

The robot is placed in a nominal position and its manipulator moves from this state into the field of view of the camera (figure 4.13). To observe the manipulator and its fingers from different perspectives, the hand and the fingers are actuated by a certain degree. The data set is therefore divided into two parts: *arm movement* (experiment 2a), where mainly the shoulder and the forearm joints are actuated, and *finger movement* (experiment 2b) where all fingers are actuated at the same time. During the complete experiment, Vicon markers are attached to the pelvis and backside of the palm to track the orientation of both robot frames (figure 3.6).

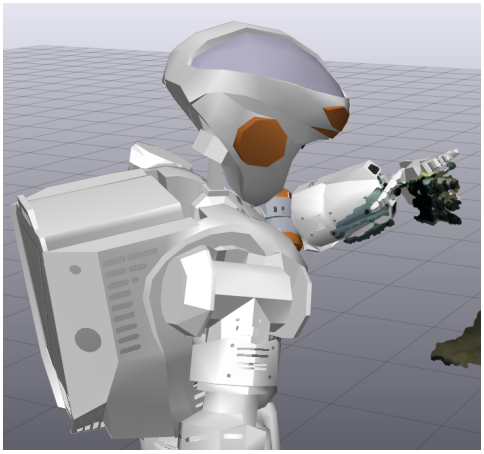


Figure 4.13: Experiment 2: Setup in which a robot is observing its own manipulator.

Experiment 2a: The *arm movement* sequence contains three movement phases (table 4.2) of which 3 final states are depicted in figure 4.14.

	time (s)	*movement description*
1	0…50	no movement, hand in lower camera view
2	50…54	arm movement up
3	85…90	hand palm turning up (forearm joint)
4	125…130	hand palm turning down (forearm joint)

Table 4.2: Phases of arm movement (Exp. 2a)

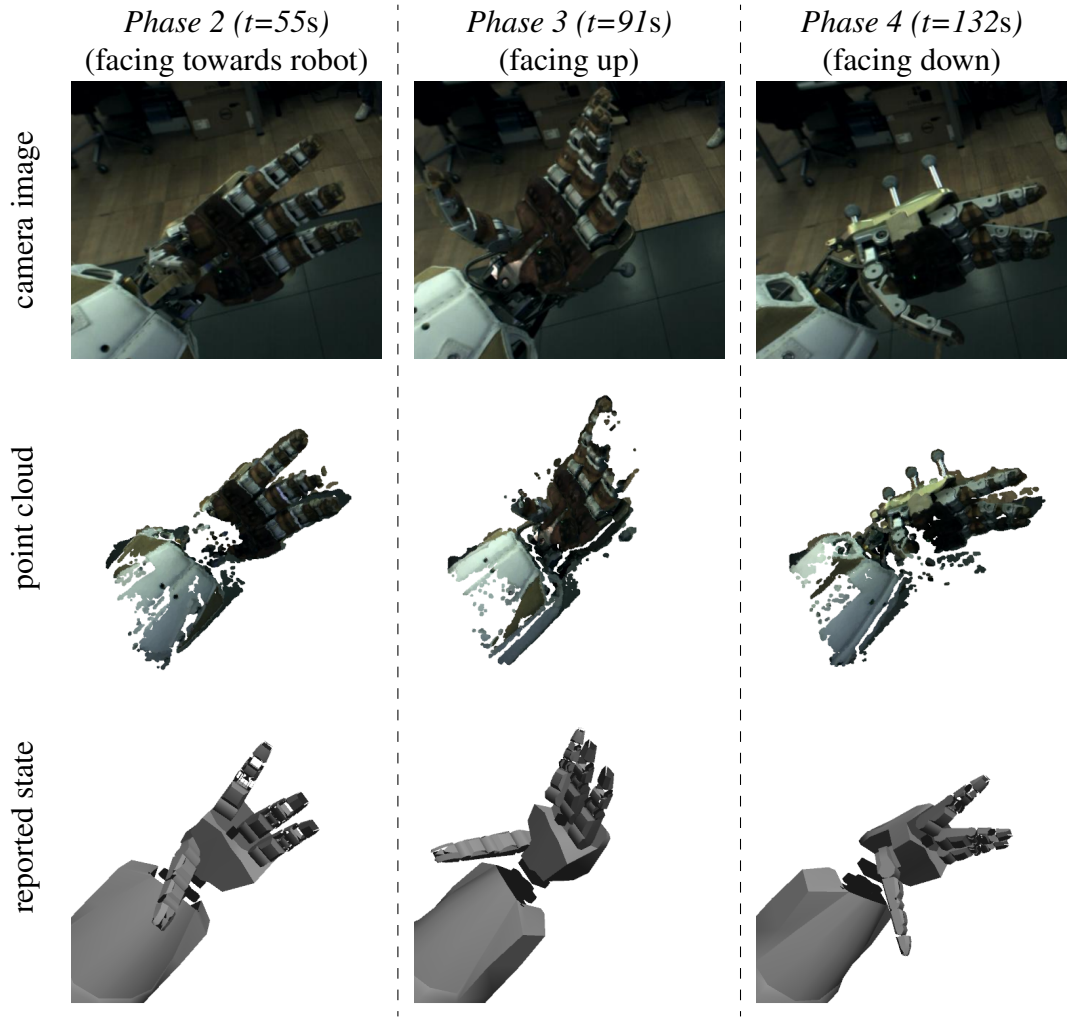


Figure 4.14: Experiment 2a (*arm movement*): Sequence of states in which the manipulator is observed from different viewing angles. *Upper row:* camera image, *Middle row:* point cloud observation, *Lower row:* reported state.

Experiment 2b: The *finger movement* sequence contains 7 movement phases in total (table 4.3) of which 6 states are depicted in figure 4.15.

	time (s)	*movement description*
1	0…30	no movement, palm facing towards camera
2	30…33	fingers closing 25%
3	56…59	fingers closing 50%
4	72…75	fingers opening
5	100…104	hand palm turning up (forearm joint)
6	127…130	fingers closing 25%
7	147…150	fingers closing 50%
8	156…159	fingers opening

Table 4.3: Phases of finger movement (Exp. 2b)

Both data sets show that due to self-occlusion, the manipulator cannot be fully observed within a single observation. In states where the palm is facing down, the Vicon marker can be observed in the point cloud data.

4.2.3 Results

The shown errors are the L2 norm of the joint and task space errors as defined in section 4.1.3. The common weighting scheme *Weighted L2 norm of joint position deviation* (objective function equation (3.14)) is applied for the joint position prior when comparing different weights. Time and durations are given in seconds.

Joint Space Error

The joint space error for both sequences is shown separately for finger and arm joints in figures 4.16 and 4.17. It can be noticed that increasing prior weights reduces the joint position deviation in the case of less distracting observations. However, the effect is not as significant as it is for observations with distractions due to close solutions after proper initialisation.

In states where the palm is facing towards the camera and large continuous hand parts can be observed, the tracked configuration of the arm is already so close to the reported configuration, that an additional joint position prior has no effect. This is the case after phase 2 for the *arm movement* data set ($55 \leq t \leq 84$) and phase 5 for the *finger movement* data set ($t \geq 105$).

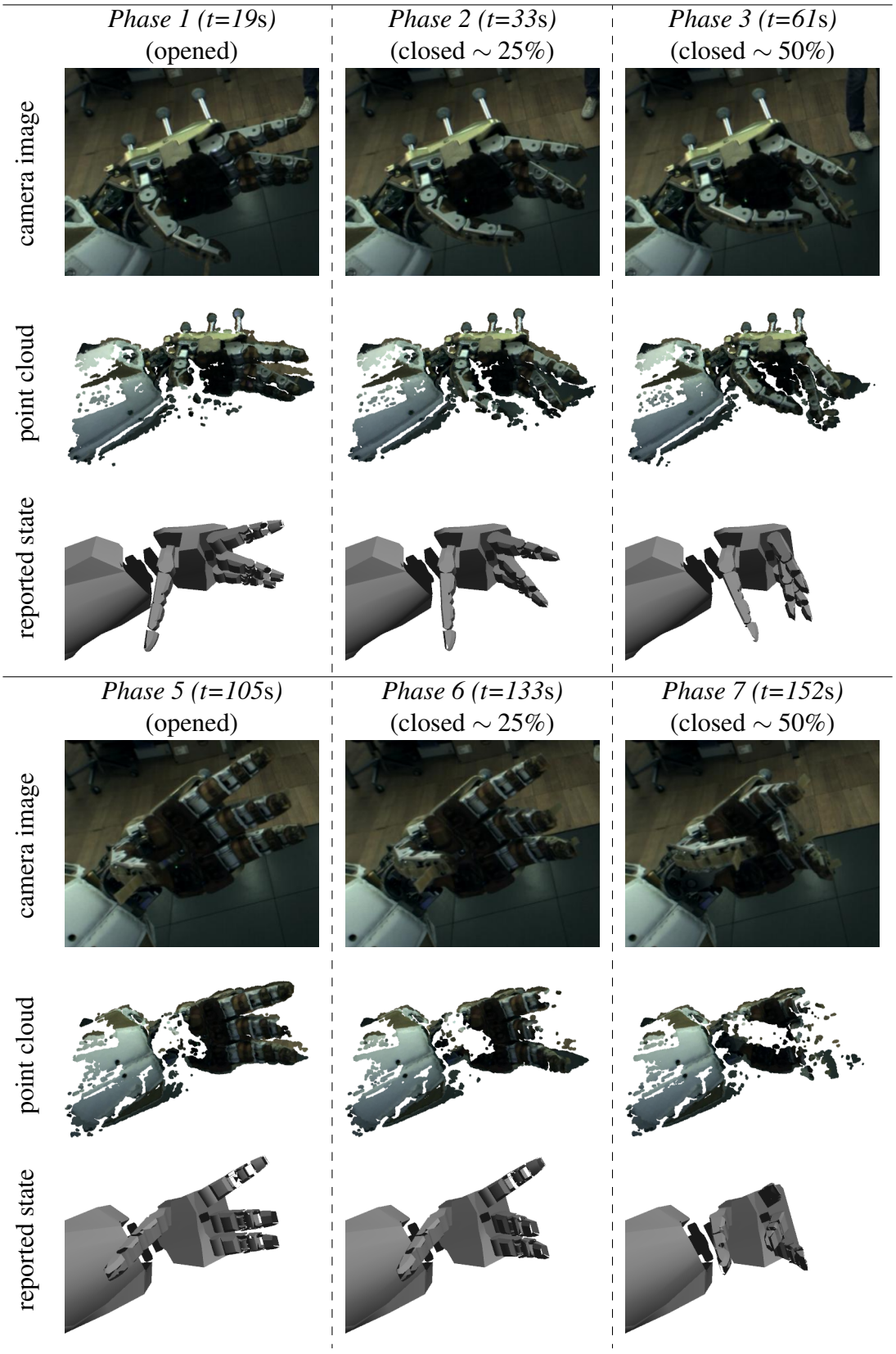
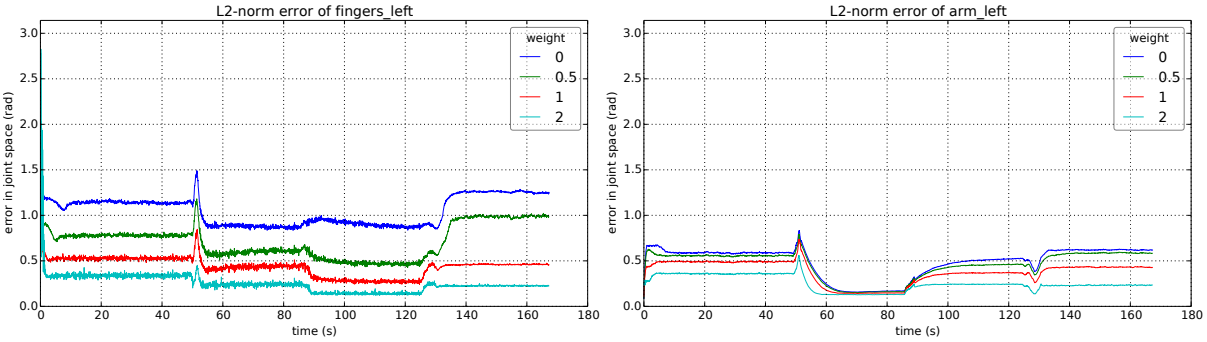
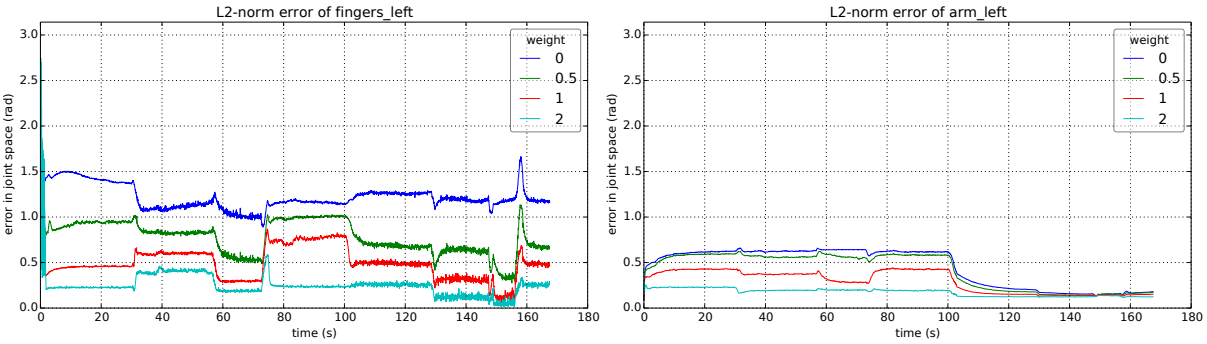


Figure 4.15: Experiment 2b (*finger movement*): Sequence of states in which the fingers are close to certain degree (order from left to right: open, closed ∼ 25%, closed ∼ 50%). *Upper sequence:* Hand palm facing down, *Lower sequence:* Hand palm facing towards robot. In each sequence: *Upper row:* camera image, *Middle row:* point cloud observation, *Lower row:* reported state.



(a) finger joints (b) arm joints

Figure 4.16: Experiment 2a: joint space error



(a) finger joints (b) arm joints

Figure 4.17: Experiment 2b: joint space error

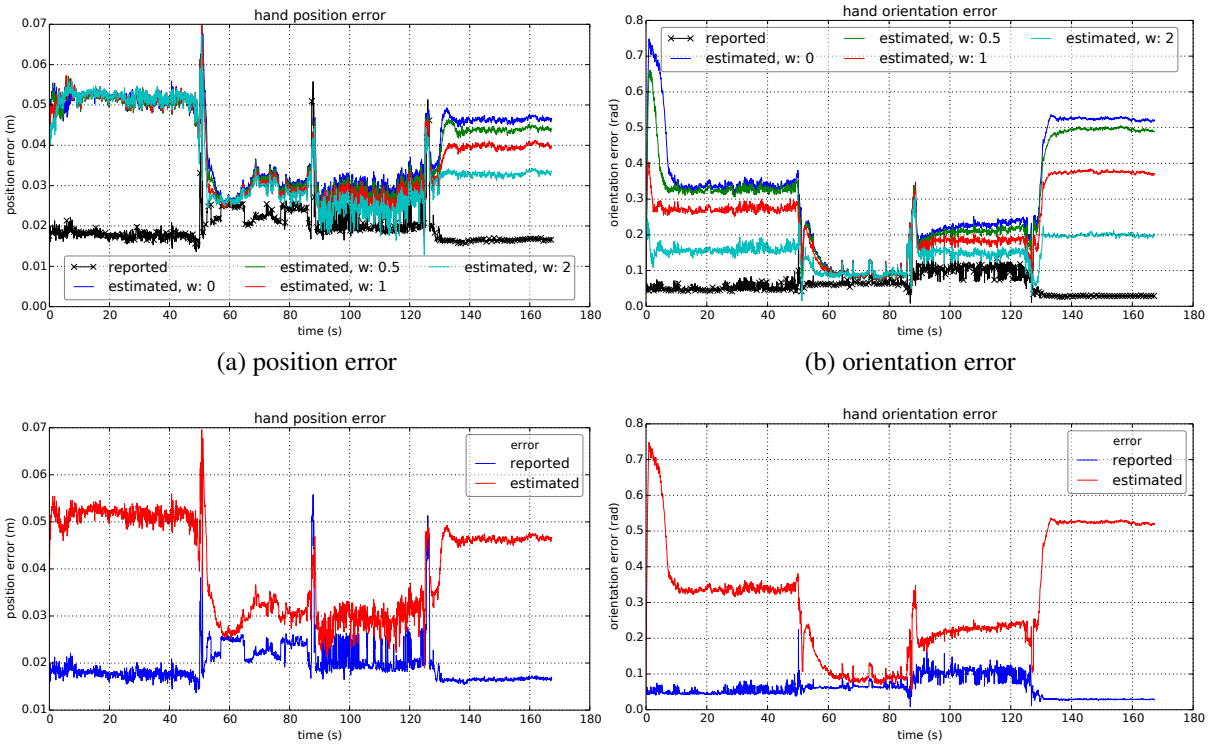
Task Space Error

To evaluate the task space error, we compare the reported and the estimated hand pose with the hand pose measured by the Vicon tracking system. The shown errors are the L2 norm of the 3D pose and 3D rotation difference from Vicon pose to the reported and estimated pose.

The error plots in figures 4.18 and 4.19 compare the effect of prior weighting on the estimated configurations against the reported configuration (which is not effected by the prior). Comparison plots for the task space error on the reported and on the estimated pose without prior are shown separately for a clearer distinction.

The following general statements can be made: (1) neither the reported nor the estimated hand poses agree with the Vicon hand pose, (2) the reported hand pose is closer to the Vicon hand pose than the estimated hand pose for most of the duration.

From both data sets it can be noticed that the estimated pose is closer to the Vicon pose in the same states where the joint deviation from the reported configuration is small. In particular the orientation error drops significantly in these states where large portions of the palm are



(a) position error

(b) orientation error

(c) position error (no prior)

(d) orientation error (no prior)

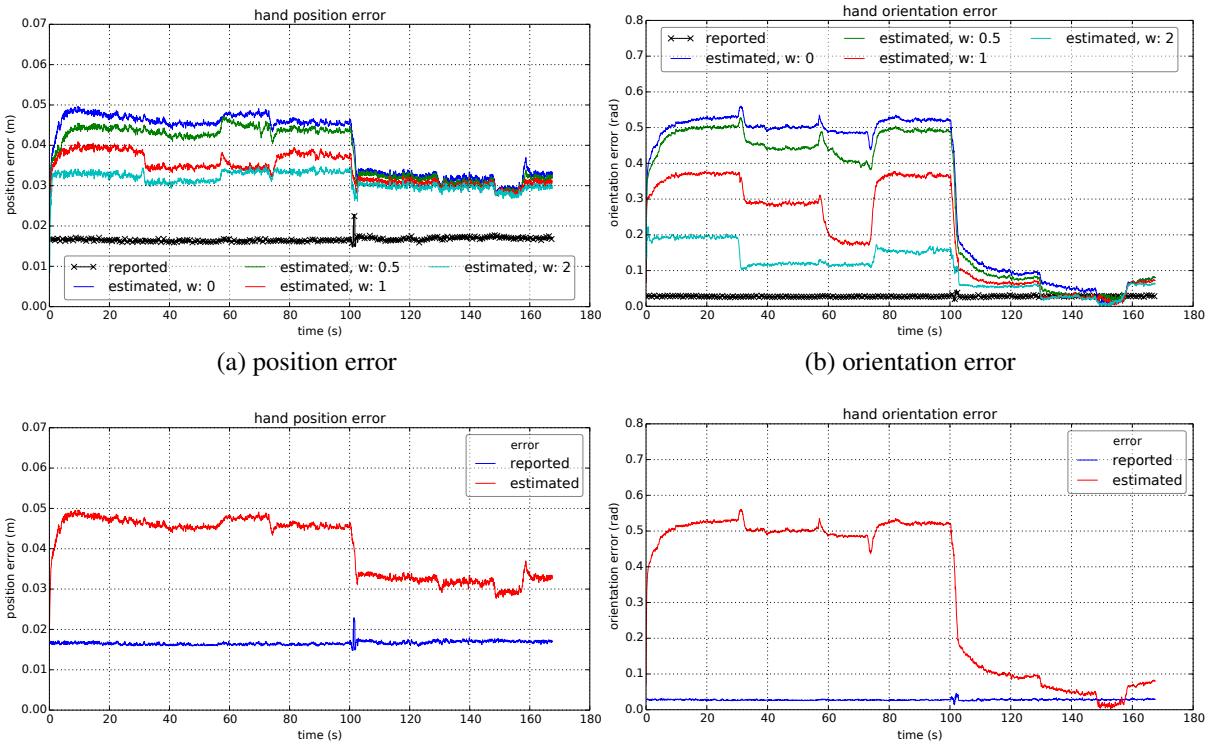
Figure 4.18: Experiment 2a: Hand pose error of the reported and estimated robot state compared to the Vicon hand marker pose.

observable (*arm movement*: $55 \leq t \leq 84$, *finger movement*: $t \geq 105$). In contrast, the error increases when the manipulator is observed from the side, e.g. when fingers are occluding each other (*arm movement*: $t \geq 132$, *finger movement*: $t \leq 104$).

4.2.4 Interpretation

Tracking with manipulator-only observed data showed that there is still an effect of using prior information. However, this effect is not as significant as in distracting scenes and it is minimal in states where large portions of the manipulator can be observed from a single observation. The performance of the tracking and hence the effect of prior information is view dependent. Prior information is still useful for inconclusive observations, e.g. self-occlusion. Despite this, hypothesis 3 is considered as true.

The observation of the Vicon marker in the depth data of the manipulator, e.g. when the hand palm is facing down, is not examined in this evaluation. These markers are required to obtain the ground truth data but they certainly provide readings that could distract the optimization.



(a) position error

(b) orientation error

(c) position error (no prior)

(d) orientation error (no prior)

Figure 4.19: Experiment 2b: Hand pose error of the reported and estimated robot state compared to the Vicon hand marker pose.

An interesting result from this evaluation is that neither the reported nor the estimated hand pose agree with the Vicon hand pose. The measured discrepancy of reported position and Vicon marker position is at least 1.5 cm. Using a strong prior in such scenario thus can only provide an optimal solution to some extent, e.g. as long as the reported configuration is closer to the true configuration than the estimate. Hypothesis 4 is considered as not verified and needs further investigation.

4.3 Joint Calibration

The data set collected in the previous experiment in section 4.2 will be used to investigate the issue of mismatching hand poses of the reported and estimated state from the Vicon hand pose that is considered as ground truth.

4.3.1 Hypothesis

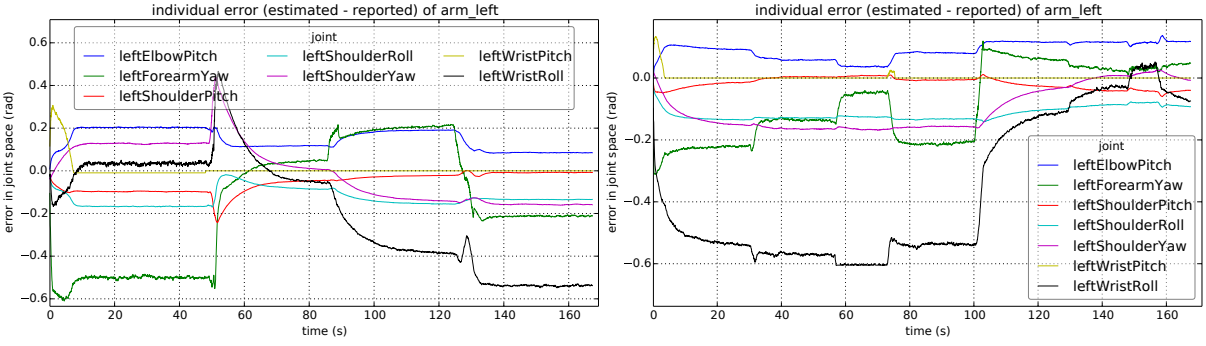
Under the assumption that the tracking obtains the true robot configuration, that is the estimated hand pose agrees with the Vicon hand pose, the joint position discrepancy (1) from reported configuration to configuration at Vicon pose, and (2) from reported configuration to estimated configuration, should be similar.

Hypothesis 5 *(Matching joint offset)*
The individual joint position deviations from reported configuration to Vicon configuration and to estimated configuration are similar.

4.3.2 Results

The individual arm joint position offsets for the arm movement and finger movement data sets (figure 4.20) show that the discrepancy of estimated to reported joints position is not constant over time. Considering that the hand is directly connected to the forearm by two joints (`leftWristRoll/Pitch`), it is interesting to notice that `leftWristPitch` does not have a discrepancy to the reported joint position value, whereas `leftWristRoll` changes significantly over time.

Table 4.4 summarizes the offsets for the estimated (*dart*, averaged over time) and reported (*vicon*, optimized on pose) joint position offsets. As the manipulator tracking considers the kinematic chain from image centre to palm and the Vicon marker pose gives the final transformation for the kinematic chain from pelvis to palm, the arm joints are the common joints whose offsets are considered for comparison. The correlation between the averaged *dart* offsets and *vicon* offsets are given as 0.3366 for the arm data set and 0.1054 for the finger data set.



(a) Arm movement (experiment 2a)

(b) Finger movement (experiment 2b)

Figure 4.20: Individual offsets of reported joint configuration to DART's estimated joint configuration

	joint positions offsets				
	arm data set (Exp. 2a)		finger data set (Exp. 2b)		complete
joint name	*dart*	*vicon*	*dart*	*vicon*	*vicon*
`torsoYaw`		0.00059		−0.00118	0.01319969
`torsoPitch`		0.01166		0.00554	0.01062786
`torsoRoll`		0.00510		0.00195	−0.00483674
`leftShoulderPitch`	−0.05627	0.01258	−0.01713	0.02658	0.00461001
`leftShoulderRoll`	−0.12866	0.00288	−0.11559	−0.01822	−0.0095907
`leftShoulderYaw`	−0.00503	0.01045	−0.09895	−0.01157	0.03021091
`leftElbowPitch`	0.14880	0.00818	0.08986	0.02588	0.00446155
`leftForearmYaw`	−0.16338	0.00166	−0.08339	0.01627	0.00580287
`leftWristRoll`	−0.18549	0.01169	−0.35942	0.01820	0.02037348
`leftWristPitch`	0.00620	0.01953	0.00195	−0.00389	0.02790135

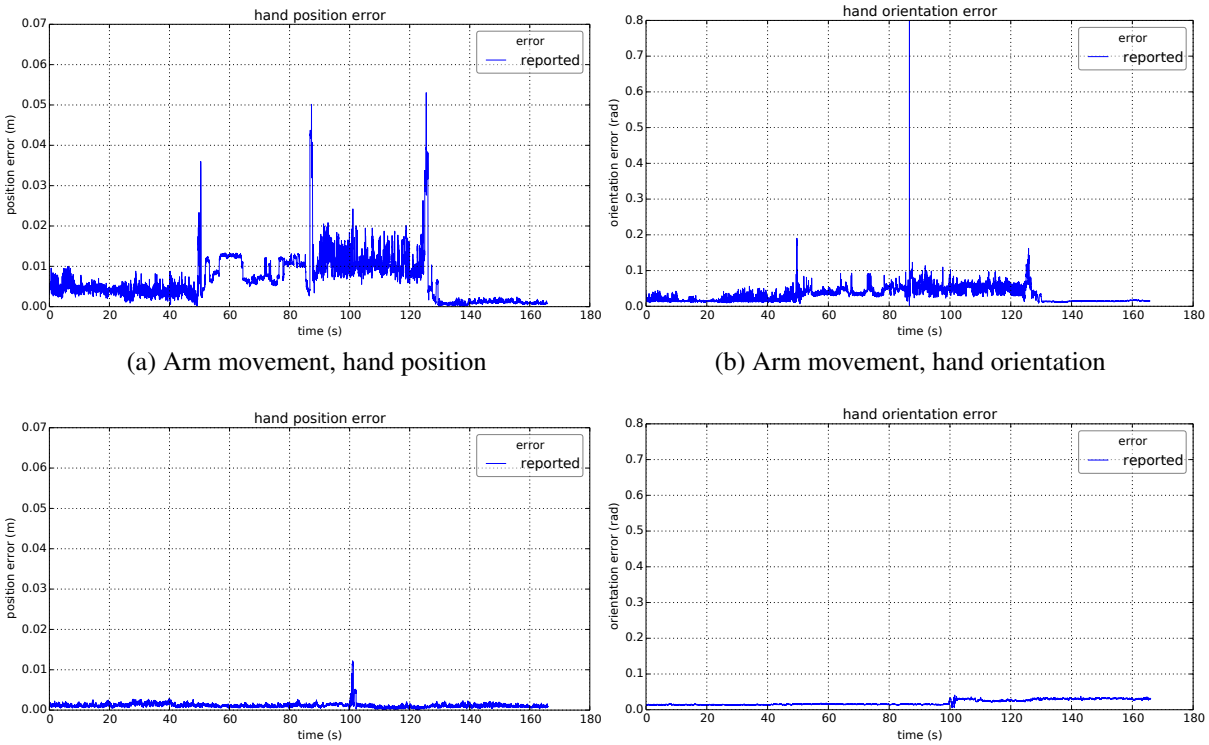
Table 4.4: Joint position offset for kinematic chain *pelvis* to *leftPalm* for reported joint position values. Estimated offsets in column *dart*, reported offsets in column *vicon* for each data set. The last column contains the offsets that have been optimized on the entire data set.

To assess the quality of fitting the correction function $f_{corr}(\cdot)$ onto the *arm movement* and *finger movement* data set, two functions are fitted each on either of the data sets and tested on the other data set. The correction functions that map the reported joint position to the optimal joint position are the constant mapping (equation (3.25)) with one parameter per joint and the linear mapping (equation (3.26)) with two parameters per joint.

The training and test errors are the costs of the objective function given the optimal parameters on the training and test set respectively. The comparison of these training and test errors on both data sets (table 4.5) indicates overfitting on the *finger movement* data set (test error is larger than training error) for both functions. This is likely caused by the small range of arm movement during the *finger movement* data set. Because training and test error are reduced equally when using the more complex linear correction function on the *arm movement* data set, we can conclude that a more complex relation represents the calibration issue more precise without overfitting. A constant joint position offset presumably does not model all underlying, and probably non-linear, issues.

		joint position offset			
data set		constant		linear	
training set	*test set*	*training error*	*test error*	*training error*	*test error*
arm	finger	0.0487	0.0260	0.0447	0.0217
finger	arm	0.00853	0.06859	0.00458	0.06636

Table 4.5: Training and test error for calibration using a constant and linear offset.



(a) Arm movement, hand position

(b) Arm movement, hand orientation

(c) Finger movement, hand position

(d) Finger movement, hand orientation

Figure 4.21: Reported hand pose error after applying joint position offsets that have been optimized on the complete Vicon data set. *Top row*: arm movement data set, *Bottom row*: finger movement data set.

After applying the constant offsets optimized on the entire data set (last column in table 4.4) to the reported pose, the new reported hand pose error is reduced by usually more than 1 cm (compare figures 4.18 and 4.19 with figure 4.21). The effect of applying the offsets are additionally visually verified by comparing the robot state with the the perceived state from LIDAR data of the *arm movement* data set (figure 4.22) and an additional grasping data set (figure 4.23). Note that despite the shown pose in the grasping data set (figure 4.23, left) has not been included in the optimization, the application of the offsets results in a hand pose that is closer to the perceived point cloud (figure 4.23, right).

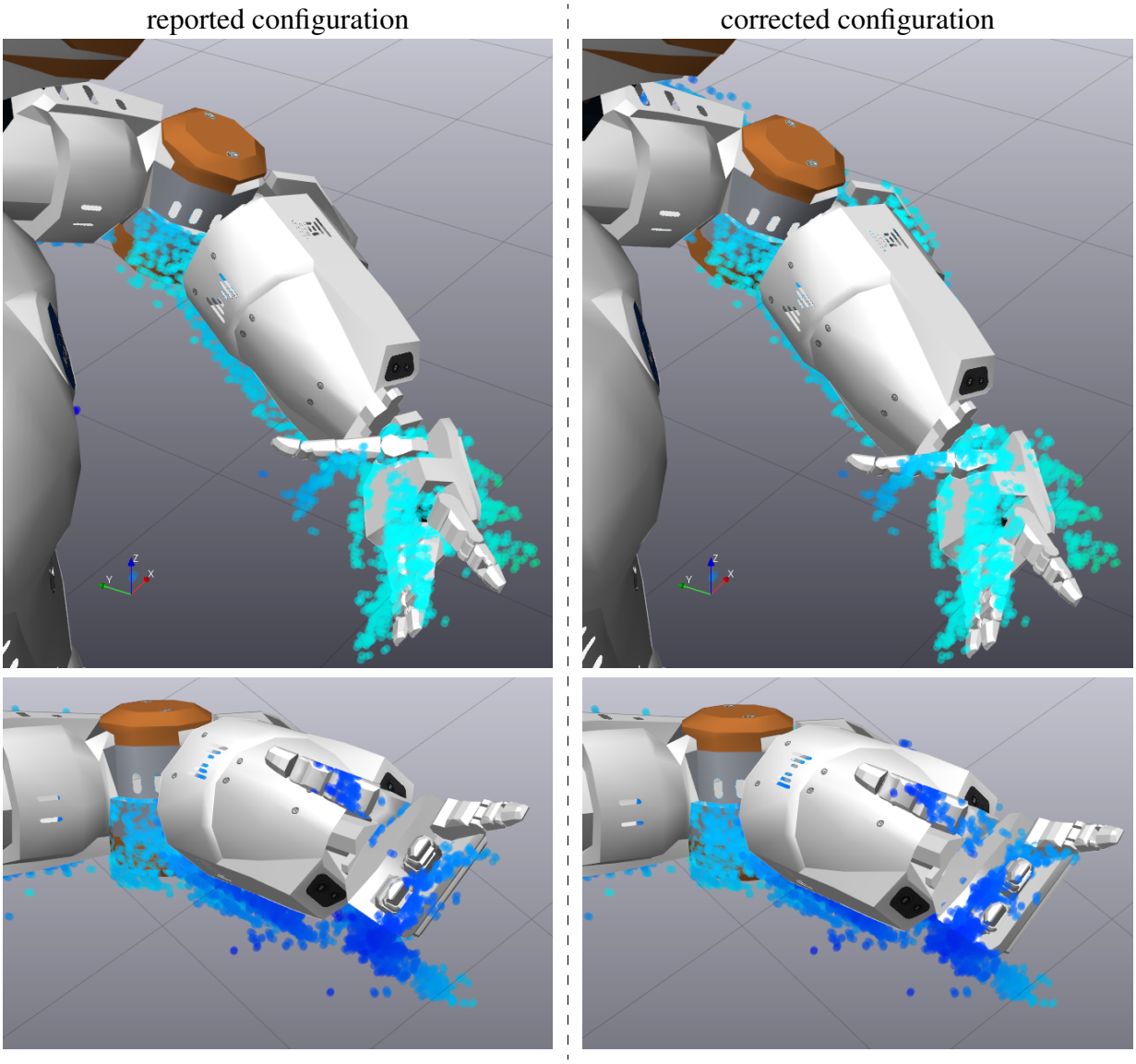


Figure 4.22: Overlaying the robot state before (left) and after (right) applying joint position offsets on LIDAR data from the *arm movement* data set. The corrected pose shows a closer match to the observed state. The colour brightness of the data points indicate their distance to the LIDAR sensor.

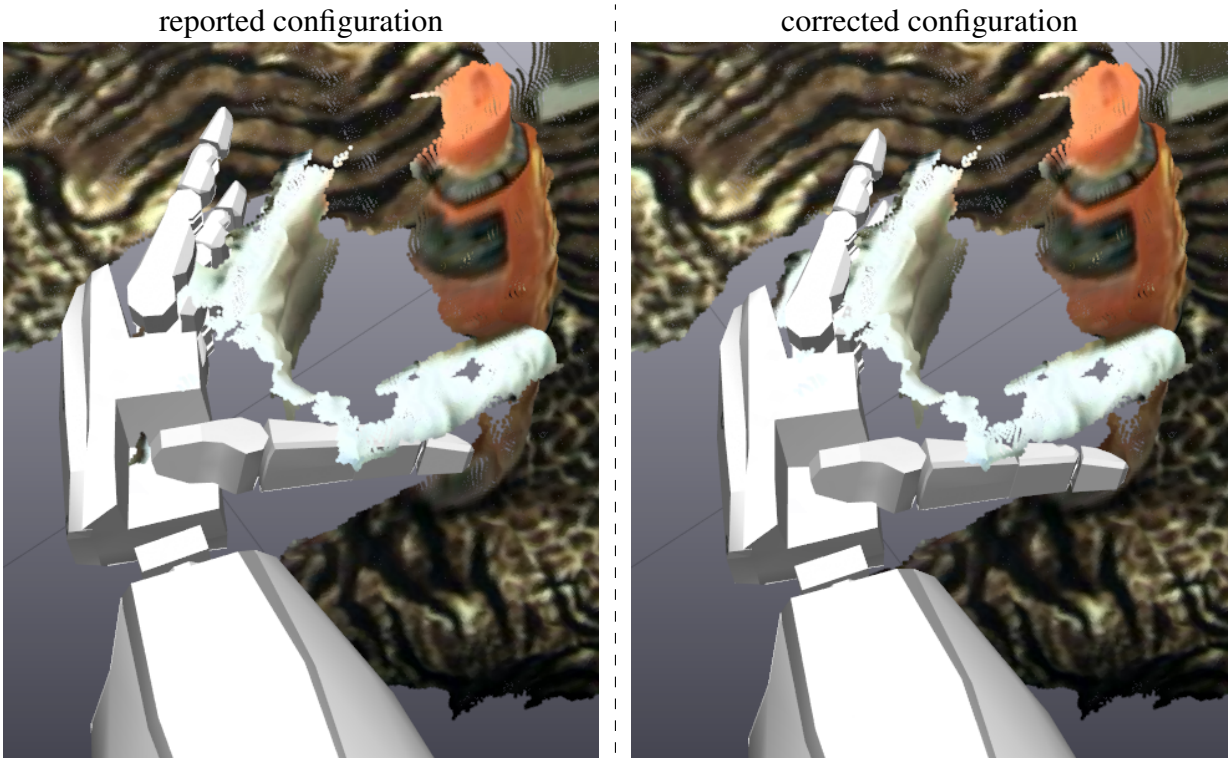


Figure 4.23: Overlaying the robot state before (left) and after (right) applying joint position offsets on stereo depth data from a grasping experiment. Although these poses have not been considered for optimization, the corrected state after applying offsets shows a closer match to the observed state.

4.3.3 Interpretation

The comparison of joint offsets estimated by the tracker and optimized on the reported Vicon marker showed no correlation. Furthermore, time elapse of the estimated offsets revealed no systematic discrepancy from the reported joint position values. Hypothesis 5 is therefore considered as not verified. However, the offset optimization on the ground truth poses has shown that the reported hand pose error can be reduced by a constant individual joint offset.

Chapter 5

Conclusion

5.1 Discussion

This thesis motivated research on articulated tracking for humanoid manipulation by introducing issues that make open-loop manipulation currently infeasible. One of the solutions towards closed-loop manipulation using visual feedback has been presented and investigated.

The presented gradient based approach is generally applicable to articulated models and has been applied to manipulator and object tracking. The advantage of this approach is its simplicity: it only requires a model and raw depth data, and allows the straightforward extension of the objective function. It has been shown, however, that using the signed distance function alone results in local minima caused by depth readings from nearby objects.

Motivated by the application of manipulator tracking when the manipulator is connected by a kinematic chain to the camera frame, the main contribution of this work has been the justification and evaluation of incorporating prior information into the tracking process. Two principally different ways have been presented: (1) biasing the objective function towards the reported joint positions, and (2) filtering the perception to only include relevant information for the optimization process.

The experiments have shown that exploiting prior information is beneficial for acquiring more accurate estimates of the manipulator pose. However, comparisons with the ground truth data also revealed that there is actually an offset between the reported and the measured hand pose. In these circumstances, the application of a prior and the choice of weights must be critically examined. A solution to this dilemma has been presented by optimizing joint position offsets using the measured ground truth hand poses.

5.2 Future Work

5.2.1 Manipulator Tracking

Using the prior information is a step towards more reliable manipulator tracking. But we can only take full advantage of this, if the sensed joint state is as accurate as possible. We further showed that selecting appropriate weights is not a straightforward task. For the robust and useful exploitation of joint position prior in the optimization, the following research questions have been identified.

Calibration The preliminary analysis of joint position offsets should be further investigated to provide valuable information of the true robot state. This should be done in a similar fashion as presented. In particular, different complexity of correction functions need to be evaluated by cross validation on a larger data set containing a wide range of arm configurations.

Additionally, the Vicon marker on the hand provide distinctive key points in the perceived depth data to align the camera frame with the then calibrated hand frame. This should further reduce discrepancies between manipulator and observation frame.

Weights Different weighting schemes and weights have been evaluated. Given the ground truth pose, choosing a weighting scheme and appropriate weights can be automatised by learning the optimal combination of objective functions.

Optimization So far, only the Gauss-Newton method has been used for solving the non-linear least-squares problem. Like all gradient based methods, this optimization method faces the before mentioned problem of local minima. Alternative global optimization approaches like particle swarm optimization or simulated annealing should be investigated. Global optimization methods consider multiple solutions to the optimization problem spread over the whole parameter space. Hence, they do not suffer from locally bounded information.

5.2.2 Object Tracking

Object tracking cannot benefit from prior information provided by the robot and thus suffers most by distracting or missing raw sensor readings. We propose to use depth specific features that have been successfully applied to this task instead of using probably noisy or incomplete raw data. Similarly, the learned combination of primitive depth features can also provide a solution to this problem. In general, the object tracking problem should be considered as a one-shot classification problem using discriminative methods instead of iterative convergence using gradient based methods.

Bibliography

[1] E. Krotkov, "DRC trials task description." http://archive.darpa.mil/roboticschallengetrialsarchive/sites/default/files/DRC%20Trials%20Task%20Description%20Release%2011%20DISTAR%2022197.pdf, 2013.

[2] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, T. Koolen, P. Marion, M. Posa, A. Valenzuela, K.-T. Yu, J. Shah, K. Iagnemma, R. Tedrake, and S. Teller, "An architecture for online affordance-based perception and whole-body planning," *Journal of Field Robotics*, vol. 32, no. 2, pp. 229–254, 2015.

[3] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, "Team ihmc's lessons learned from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

[4] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Lessons from the amazon picking challenge," *arXiv preprint arXiv:1601.05484*, 2016.

[5] G. Chliveros, M. Pateraki, and P. Trahanias, "Robust multi-hypothesis 3D object pose tracking," in *Computer Vision Systems: 9th International Conference, ICVS 2013, St. Petersburg, Russia, July 16-18, 2013. Proceedings* (M. Chen, B. Leibe, and B. Neumann, eds.), (Berlin, Heidelberg), pp. 234–243, Springer Berlin Heidelberg, 2013.

[6] C. Teuliere, E. Marchand, and L. Eck, "Using multiple hypothesis in model-based tracking," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4559–4565, May 2010.

[7] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, 2012.

[8] P. Azad, D. Munch, T. Asfour, and R. Dillmann, "6-DoF model-based tracking of arbitrarily shaped 3d objects," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5204–5209, May 2011.

[9] E. Muñoz, Y. Konishi, V. Murino, and A. 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 2016, Stockholm, Sweden, May 16-21, 2016*, pp. 5623–5630, 2016.

[10] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2441–2448, May 2016.

[11] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother, "6-DOF model based tracking via object coordinate regression," in *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV* (D. Cremers, I. Reid, H. Saito, and M.-H. Yang, eds.), (Cham), pp. 384–399, Springer International Publishing, 2015.

[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, *Machine Learning for Computer Vision*, ch. Real-Time Human Pose Recognition in Parts from Single Depth Images, pp. 119–135. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[13] F. Widmaier, D. Kappler, S. Schaal, and J. Bohg, "Robot arm pose estimation by pixel-wise regression of joint angles," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 616–623, May 2016.

[14] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. 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, (New York, NY, USA), pp. 3633–3642, ACM, 2015.

[15] K. Pauwels, L. Rubio, and E. Ros, "Real-time pose detection and tracking of hundreds of objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[16] K. Pauwels, L. Rubio, and E. Ros, "Real-time model-based articulated object pose detection and tracking with variable rigidity constraints," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 3994–4001, June 2014.

[17] K. Pauwels, V. Ivan, E. Ros, and S. Vijayakumar, "Real-time object pose recognition and tracking with an imprecisely calibrated moving RGB-D camera," in *International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 2733–2740, Sept 2014.

[18] T. Schmidt, R. Newcombe, and D. Fox, "DART: dense articulated real-time tracking with consumer depth cameras," *Autonomous Robots*, vol. 39, no. 3, pp. 239–258, 2015.

[19] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 119–126, May 2015.

[20] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.

[21] C. Choi and H. Christensen, "RGB-D object tracking: A particle filter approach on GPU," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1084–1091, Nov 2013.

[22] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1311–1327, 2011.

[23] J. Sturm, C. Plagemann, and W. Burgard, "Body schema learning for robotic manipulators from visual self-perception," *Journal of Physiology-Paris*, vol. 103, no. 3–5, pp. 220 – 231, 2009. Neurorobotics.

[24] Z. Si and S. C. Zhu, "Learning AND-OR templates for object recognition and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2189–2205, Sept 2013.

[25] M. Firman, "RGBD Datasets: Past, Present and Future," in *CVPR Workshop on Large Scale 3D Data: Acquisition, Modelling and Analysis*, 2016.

[26] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 509–516, May 2014.

[27] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT - the blocks world robotic vision toolbox," in *Proc. ICRA Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[28] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2759–2766, June 2012.

[29] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proceedings of the 13th European Conference on Computer Vision* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 345–360, Springer International Publishing, 2014.

[30] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1298–1303, May 2012.

[31] L. Bo, X. Ren, and D. Fox, *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, ch. Unsupervised Feature Learning for RGB-D Based Object Recognition, pp. 387–402. Heidelberg: Springer International Publishing, 2013.

[32] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "MoDeep: A deep learning framework using motion features for human pose estimation," in *Computer Vision – ACCV 2014* (D. Cremers, I. Reid, H. Saito, and M.-H. Yang, eds.), vol. 9004 of *Lecture Notes in Computer Science*, pp. 302–315, Springer International Publishing, 2015.

[33] J. Tompson, M. Stein, Y. LeCun, Y.un, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (TOG)*, vol. 33, pp. 169:1–169:10, Sept. 2014.

[34] M. Jiu, C. Wolf, G. Taylor, and A. Baskurt, "Human body part estimation from depth images via spatially-constrained deep learning," *Pattern Recognition Letters*, vol. 50, pp. 122 – 129, 2014. Depth Image Analysis.

[35] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 376–380, Apr 1991.

Appendix A

Joint Prior Derivation

This chapter contains the derivation of the joint position prior used for the Jacobian in the Hessian approximation and the gradient for the Gauss-Newton algorithm.

A.1 Weighted L2 Norm of Deviation

A.1.1 Objective Function

$$e = w \cdot \|\mathbf{r} - \mathbf{q}\|_2 \tag{A.1}$$

$$= w \cdot \sqrt{\sum_{i=1}^{N} (r_i - q_i)^2} \tag{A.2}$$

A.1.2 Partial Derivatives

For simplicity we substitute

$$\mathbf{d} = \mathbf{r} - \mathbf{q} \tag{A.3}$$

$$e = w \cdot \sqrt{ss} \tag{A.4}$$

$$ss = \sum_{i=1}^{N} d_i^2. \tag{A.5}$$

Derivation by chain rule:

$$\frac{\partial e}{\partial q_i} = \frac{\partial d_i}{\partial q_i} \cdot \frac{\partial e}{\partial d_i} \tag{A.6}$$

$$\frac{\partial d_i}{\partial q_i} = -1$$

$$\frac{\partial e}{\partial d_i} = \frac{\partial ss}{\partial d_i} \cdot \frac{\partial e}{\partial ss}$$

$$\frac{\partial ss}{\partial d_i} = 2d_i$$

$$\frac{\partial e}{\partial d_i} = \frac{w}{2} \cdot \frac{1}{ss^{-1/2}} \cdot 2d_i = \frac{w \cdot d_i}{\sqrt{\sum_{i=1}^{N} d_i^2}}$$

$$\frac{\partial e}{\partial q_i} = -w \frac{r_i - q_i}{\|\mathbf{r} - \mathbf{q}\|_2} \tag{A.7}$$

A.2 Individually Weighted Deviations

A.2.1 Objective Function

$$e = |\mathbf{r} - \mathbf{q}|^\top Q |\mathbf{r} - \mathbf{q}| \tag{A.8}$$

$$= \sum_{i=1}^{N} \left[(r_i - q_i) \cdot \sum_{j=1}^{N} (r_j - q_j) \cdot \omega_{i,j} \right] \tag{A.9}$$

A.2.2 Partial Derivatives

For simplicity we substitute

$$\mathbf{d} = \mathbf{r} - \mathbf{q}. \tag{A.10}$$

We write the nested sums out

$$\begin{aligned} e =& d_1 \cdot (d_1 q_{11} + d_2 q_{12} + \cdots + d_n q_{1n} + \cdots + d_N q_{1N}) \\ &+ d_2 \cdot (d_2 q_{21} + d_2 q_{22} + \cdots + d_n q_{2n} + \cdots + d_N q_{2N}) \\ &+ \cdots \\ &+ d_n \cdot (d_1 q_{n1} + d_2 q_{n2} + \cdots + d_n q_{nn} + \cdots + d_N q_{nN}) \\ &+ \cdots \\ &+ d_N \cdot (d_1 q_{N1} + d_2 q_{N2} + \cdots + d_n q_{Nn} + \cdots + d_N q_{NN}) \end{aligned} \tag{A.11}$$

and apply the sum and chain rule for derivation:

$$\frac{\partial e}{\partial q_i} = \frac{\partial d_i}{\partial q_i} \cdot \frac{\partial e}{\partial d_i} \tag{A.12}$$

$$\frac{\partial d_i}{\partial q_i} = -1$$

$$\frac{\partial e}{\partial d_i} = \left[\sum_{j=1}^{N} d_j q_{i,j} \right] + \left[\sum_{j=1 \neq i}^{N} d_j q_{j,i} \right] \tag{A.13}$$

$$\frac{\partial e}{\partial q_i} = -\left[\sum_{j=1}^{N} d_j q_{i,j} \right] + \left[\sum_{j=1 \neq i}^{N} d_j q_{j,i} \right] \tag{A.14}$$

The range notation $j = 1 \neq i$ denotes that index j iterates $[1, N]$ but skips index i. If Q is symmetric, the final partial derivative can be written and implemented as a dot product of the vector $\mathbf{d}$ and column- respectively row vectors of Q.

Appendix B

Vicon Frame To Robot Frame Transformations

This chapter aggregates the position of the Vicon markers in each marker frame and robot frame. Coordinates are listed in their corresponding order and given in meters. The transformation from robot frame to marker frame is estimated using least-squares on the objective function defined in equation (3.19).

B.1 Pelvis

B.1.1 Marker Position

Vicon marker in marker frame:

$$M = 0.001 \cdot \begin{bmatrix} 42.43020 & 0.38291 & -75.74550 \\ 36.80328 & 0.23377 & -16.98103 \\ -24.30613 & -119.75513 & 47.44633 \\ -54.92734 & 119.13845 & 45.28020 \end{bmatrix} \tag{B.1}$$

Vicon marker in `pelvis` frame:

$$P = \begin{bmatrix} 0.1674 & 0 & -0.2015 \\ 0.1674 & 0 & -0.14053 \\ 0.1087 & -0.1194 & -0.0717 \\ 0.0837 & 0.1194 & -0.0717 \end{bmatrix} \tag{B.2}$$

B.1.2 Transformation

$$T_{p\to m} = \begin{bmatrix} 0.99610 & 0.02143 & 0.08560 & 0.13180 \\ -0.02129 & 0.99977 & -0.00259 & 0.0 \\ -0.08564 & 0.00075 & 0.99633 & -0.12136 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

B.2 Hand

B.2.1 Marker Position

Vicon marker in marker frame:

$$m = 0.001 \cdot \begin{bmatrix} 8.18111 & -46.51067 & -9.48950 \\ 2.60155 & -14.01778 & 14.08099 \\ -5.18721 & 32.79453 & -37.71204 \\ -5.59546 & 27.73392 & 33.12054 \end{bmatrix} \tag{B.4}$$

Vicon marker in `leftPalm` frame:

$$P = \begin{bmatrix} -0.0592 & 0.103172 & 0.02345 \\ -0.0605 & 0.0629 & 0.02345 \\ -0.06 & 0.054 & -0.046 \\ -0.0605 & 0.0173 & 0.0143 \end{bmatrix} \tag{B.5}$$

B.2.2 Transformation

$$T_{h\to m} = \begin{bmatrix} -0.98248 & -0.18444 & -0.02682 & -0.06005 \\ 0.16571 & -0.79859 & -0.57861 & 0.05934 \\ 0.08529 & -0.57291 & 0.81516 & 0.0038 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6}$$