Generative 3D Hand Tracking with Spatially Constrained Pose Sampling

Konstantinos Roditakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Professor *Antonis Argyros*

The work reported in this thesis has been conducted at the Computational Vision and Robotics (CVRL) laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology–Hellas (FORTH) and has been financially supported by a FORTH-ICS scholarship, including funding by the European Commission through projects WEARHAP (FP7-ICT-2011-9) and Co4Robots (H2020-731869).

University of Crete
Computer Science Department

Generative 3D Hand Tracking with Spatially Constrained Pose Sampling

Thesis submitted by
Konstantinos Roditakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Konstantinos Roditakis

Committee approvals: _____
Antonis Argyros
Professor, Thesis Supervisor

_____
George Papagiannakis
Associate Professor, Committee Member

_____
Xenophon Zabulis
Principal Researcher, Committee Member

Departmental approval: _____
Antonis Argyros
Professor, Director of Graduate Studies

Heraklion, November 2017

Generative 3D Hand Tracking with Spatially Constrained Pose Sampling

Abstract

Estimating the 3D pose and full articulation of a human hand based on visual information remains a challenging task that recieved a lot of attention by the research community, particularly during the last decade. The main challenges arise from the dimensionality of the problem, the rapid hand motion and self-occlusions that occur in the majority of observed poses. Reliable, robust and accurate solutions can facilitate the development of industrial and consumer-level applications. There exist practical hand motion scenarios where the hand parts are spatially constrained. In these scenarios, hand part locations can be inferred implicitly from data-driven detectors or interaction with the environment.

In this thesis, we investigate such scenarios, and we consider this type of spatial constraints. We present a method for 3D hand tracking that efficiently exploits spatial constraints in the form of end effector (fingertip) locations. An end-effector target can be either a specific 3D point or a 3D region, and the number of constrained fingertips may vary through frames. The proposed method follows a generative, hypothesize-and-test approach and uses a hierarchical particle filter to track the hand.

The current state of the art methods consider these spatial constraints in a soft manner and can not guarantee that the resulting estimate will satisfy them. We tackle this issue by enforcing spatial constraints during the hand pose hypothesis generation phase. In that direction, we developed a simple and fast finger articulation sampling method that is based on the concept of Reachable Distance Space (RDS).

The main contributions of this work are the following: (a) We extend the original RDS formulation to generate finger articulations that respect both the hands' joint limits and the end effector constraints, (b) we introduce the C-HMF framework by tightly integrating our constraints-aware sampling strategy to the Hierarchical Model Fusion (HMF) framework. If spatial constraints are absent at certain frames, our proposed C-HMF framework can seamlessly fall back to the original HMF method. Each hypothesis is evaluated by measuring the discrepancy between the rendered 3D model and the available observations.

Several error metrics are employed to extensively evaluate our methodology on challenging, ground truth-annotated sequences that contain severe hand occlusions. Quantitative and qualitative results demonstrate that the proposed approach significantly outperforms state of the art in hand tracking accuracy and robustness. Additionally, we demonstrate that our methodology is robust to fingertip detection noise.

By exploring more densely the space of feasible solutions, we require the evaluation of much fewer hand hypotheses, all of which satisfy the given constraints.

Along with the proposed light-weight sampling strategy, our methodology is suitable to cope with the performance requirements of applications requiring a real time solution to the problem of 3D hand tracking.

Τρισδιάστατη Παρακολούθηση Ανθρώπινων Χεριών που Υπόκεινται σε Χωρικούς Περιορισμούς

περίληψη

Η εκτίμηση της θέσης και της πλήρους αρθρωτής κίνησης ενός ανθρώπινου χεριού με βάση οπτική πληροφορία παραμένει ένα δύσκολο πρόβλημα το οποίο έχει μελετηθεί εντατικά από την ερευνητική κοινότητα. Οι κυριότερες προκλήσεις προκύπτουν από τη μεγάλη διαστατικότητα του προβλήματος, την γρήγορή κίνηση των χεριών και τις παρατηρούμενες αυτό-επικαλύψεις. Αξιόπιστες, ανθεκτικές και ακριβείς λύσεις για το πρόβλημα μπορούν να διευκολύνουν την ανάπτυξη πολλών εφαρμογών. Σε πολλές από αυτές, υπάρχουν περιορισμοί ως προς τη θέση και την κίνηση των τμημάτων του χεριού τα οποία μπορούν να εκτιμηθούν από ανιχνευτές κατευθυνόμενους από δεδομένα (data-driven detectors) ή εμμέσως εξαιτίας της αλληλεπίδρασης των χεριών με το περιβάλλον.

Σε αυτή την εργασία, διερευνούμε τέτοια σενάρια και λαμβάνουμε υπόψιν αυτό το είδος χωρικών περιορισμών. Παρουσιάζουμε μια μέθοδο για την τρισδιάστατη παρακολούθηση του χεριού η οποία εκμεταλλεύεται αποτελεσματικά χωρικούς περιορισμούς της 3Δ θέσης των ακροδαχτύλων. Πιο συγκεκριμένα, τα ακροδάχτυλα μπορεί να έχουν γνωστή 3Δ θέση ή να βρίσκονται σε μία γνωστή περιοχή. Ο αριθμός των περιορισμένων δακτύλων μπορεί να μεταβάλλεται χρονικά κατά τη διάρκεια μιας ακολουθίας. Η προτεινόμενη μέθοδος ακολουθεί μια γενετική προσέγγιση που δημιουργεί και αξιολογεί υποθέσεις (generative hypothesize-and-test approach) και που χρησιμοποιεί ένα ιεραρχικό φίλτρο σωματιδίων (hierarchical particle filter) για την παρακολούθηση του χεριού.

Οι καλύτερες γνωστές μέθοδοι για την επίλυση του προβλήματος λαμβάνουν υπόψη χωρικούς περιορισμούς με χαλαρό τρόπο και συνεπώς δεν μπορούν να εγγυηθούν ότι η τελική λύση θα τους ικανοποιήσει. Αντιθέτως, στην παρούσα εργασία, αντιμετωπίζουμε αυτό το ζήτημα επιβάλλοντας τους χωρικούς περιορισμούς κατά τη φάση δημιουργίας υποθέσεων της άρθρωσης του χεριού. Προς αυτή την κατεύθυνση αναπτύξαμε μια απλή και γρήγορη μέθοδο δειγματοληψίας αρθρώσεων των δακτύλων που βασίζεται στην έννοια του Χώρου Προσβασιμότητας βάση Απόστασης – ΧΠΑ (Reachable Distance Space - RDS).

Οι κύριες συνεισφορές αυτής της εργασίας είναι οι εξής: (α) Επεκτείνουμε την αρχική διατύπωση του ΧΠΑ (RDS) για να παράγουμε αρθρώσεις των δακτύλων που σέβονται τόσο τα κινηματικά όρια των συνδέσμων του χεριού όσο και τους περιορισμούς των τελικών τελεστών, (β) εισάγουμε το αλγοριθμικό πλαίσιο C-HMF που ενσωματώνει την παραπάνω στρατηγική δειγματοληψίας με το πλαίσιο Ιεραρχικής Συγχώνευσης Μοντέλων - ΙΣΜ (Hierarchical Model Fusion - HMF) . Εάν απουσιάζουν χωρικοί περιορισμοί σε συγκεκριμένα καρέ της ακολουθίας, το προτεινόμενο πλαίσιο C-HMF ανάγεται στην αρχική μέθοδο ΙΣΜ - HMF . Κάθε υπόθεση αξιολογείται εκτιμώντας την ασυμφωνία μεταξύ του παραγόμενου 3Δ μοντέλου και των διαθέσιμων παρατηρήσεων.

Χρησιμοποιήθηκαν αρκετές μετρικές σφάλματος για την εκτενή ποσοτική και ποιοτική αξιολόγηση της μεθοδολογίας μας σε ακολουθίες οι οποίες περιέχουν αρκετές αυτοεπικαλύψεις. Τα ποσοτικά και ποιοτικά αποτελέσματα καταδεικνύουν ότι η προτεινόμενη προσέγγιση υπερτερεί σημαντικά σε σχέση με τις βέλτιστες γνωστές μεθόδους σε ακρίβεια εκτίμησης της θέσης και της αρθρωτής κίνησης του χεριού και στην ευρωστία.

Ευχαριστίες

I thank Prof. Antonis Argyros for introducing this challenging topic. I thank him for his trust, support and guidance during all years at FORTH.

I thank and give credits to Alexandros Makris. His support and understanding were essential for making this thesis a successful one.

Thanks goes to the rest of FORTH, CVRL and CSD members that supported me during my master studies.

Special thanks to my family and my friends for their patience.

Contents

List of Figures

Chapter 1

Introduction

The main task of computer vision is to understand the world through images and to infer its properties, such as shape, illumination, and appearance. Computers process digital images which is a numeric representation of the observed world. These images contain low-level information that is not straightforward to interpret and infer high-level information. The human visual system can detect and interpret information from images and build a representation of the surrounding environment. For humans, this is an essential task and they can perform it seemingly. Computer vision systems aim to simulate the human visual system functions and obtain same-level or even greater abilities.

This work discusses the problem of observing and understanding human actions in a non-invasive manner. More specifically it focuses on observing the human hands. This is an interesting and challenging topic in the field of computer vision. Humans use hands as manipulators and as a tool of communication and expression. Understanding hand motion can facilitate the development of industrial and consumer-level applications. These applications typically require reliable and robust estimation of the hand pose. Challenges of the problem arise due to its high dimensionality (degrees of freedom), visibility limitations (e.g., self-occlusions, occlusions from the interacting objects) and fast motion. The following sections of this chapter introduce in detail the hand tracking problem and its challenges that this thesis investigates.

1.1 Description of the problem

We focus on the problem of estimating the 3D pose and full articulation of human hands based on visual information. 3D hand tracking consists of estimating the 3D position and orientation of the palm and the joint angles. Research in vision-based 3D hand tracking targets primarily the scenario in which a bare marker-less hand performs unconstrained motion in front of a camera system. Practical scenarios exist where the hand motion is constrained to perform certain actions, or parts of the hand are constrained to remain in certain areas. In this thesis, we investigate

such scenarios where hand motion constraints can be exploited to facilitate robust and accurate hand tracking.

The observations are in the form of single-view image sequences, acquired from a calibrated RGB-D sensor. The visual observations are markerless, to avoid interference with the observed scene.

The type of constraints that we investigate are spatial constraints in the form of end effector (fingertip) locations. A fingertip location constraint can be either a specific 3D point or a 3D region, and the number of constrained fingertips may vary through frames. In this thesis we assume that the spatial constraints are known a priori. In practice, the fingertip locations can be detected explicitly using a fingertip detector or implicitly, if a hand interacts with an object and the contact points are stationary and known.

1.2 Importance of the problem

Between the instances of the generic problem of articulated object tracking, the hand tracking is of particular interest. Hands are the main actuators of humans and essential for executing complex everyday tasks. Understanding the hand motion can provide essential information for understanding human activity.

The accurate estimation and tracking of hand articulations provides the basis for many applications like human computer interaction (HCI), computer aided design (CAD), human robot interaction and robot control, medical applications, activity analysis, and sign language recognition. Recent advances in technologies of virtual reality (VR), haptics, and robotics have strengthened the interest for vision systems that can estimate the full, time-varying pose of human hands in real time.

The interest of the relevant research community is focused on the case of marker-less tracking of the human hand(s). This is because marker-less hand tracking is not invasive and poses far fewer restrictions to any application domain. Recent solutions that use less invasive and low-cost setups demonstrate the potential for introducing consumer products to the market. Nevertheless, marker-based tracking remains useful in application domains that require high accuracy and robustness.

Apart from the practical utility, hand pose estimation and tracking are interesting at a theoretical level as well. The human visual system can effortlessly perform the task. Understanding the algorithmic solutions to hand pose estimation may aid in advancing our understanding of the inner workings of the human brain. Methodologies applied to the instance of the hand tracking problem can potentially be adapted to more general problems of articulated tracking or different class of computer vision problems.

1.3 Problem challenges

Tracking a human hand either in isolation or in interaction with objects is a challenging computer vision problem. The hand is an articulated object with more than 20 DOFs. This results in a problem with a large number of parameters to be estimated. To simplify the estimation problem, the hand can be treated as an articulated object with a hierarchical structure and a tree-like connectivity. All fingers are connected to a common base, the palm, which moves as a whole rigidly. However, difficulties still arise from the similar appearance and articulation structure of the fingers which introduces ambiguities that are difficult to resolve.

Existing hand models provide a simplified approximation of the human hand, both in terms of kinematic structure and appearance. A wide variability across individuals in bone size, flesh shape, and anatomical constraints. The bare hand has consistent chromatic appearance but can vary significantly between subjects. The issues mentioned above raise the question, if it is practical to formulate a generic hand model that does not require special adaptation for each observed individual.

Different types of difficulties arise from the particular demands of each application. Many practical applications require tracking the hand in a cluttered background and arbitrary lighting conditions. Other applications, require to track the human hand while it interacts with the environment (not in isolation). These requirements raise the issues of how to perform hand detection and segmentation in a cluttered environment, and how to handle occlusions that happen during interaction. In single view sequences, most observations include hand poses with self-occluded fingers. Depending on the camera placement, such as egocentric videos, the hand may dissapear from the observation frame and re-appear in random locations. In other cases, the user may be located far away from the camera, which results to noisy and low resolution observations.

This work consider most of the listed difficulties. To do so, we adopt an existing tracking framework that can deal with the dimensionality of the problem. We propose an extension of that framework which can consider spatial (fingertip) constraints. Our proposed extension can achieve robust and accurate tracking in challenging single view sequences.

Chapter 2

Literature Overview

Several approaches have been proposed that address the hand tracking challenges that where reported in section 1.3. Related work try to tackle the problem in several scenarios: (i) tracking a single hand performing unconstrained motion in isolation, (ii) tracking a hand manipulating/interacting with object(s) or operating in a cluttered non-stationary environment, (iii) tracking multiple hands that either operate independently or strongly interacting with each other. Additionally, we can classify each methodology by the type of input that they consider. There are methodologies that rely on RGBD sensors, multi-camera setups, monocular/stereo RGB cameras or infrared input. Most of the latest work is based on the rich point cloud information that depth sensors provide. Tracking from monocular sequences is the most challenging setup because the provided information is the most limited. In the meantime, monocular tracking is the most promising for introducing hand tracking to consumer-level applications with low-cost setups. Multi-camera and marker-based systems can acquire rich information to solve the hand tracking problem accurately, but are impractical for consumer-level applications due to their cost and/or the induced restrictions in mobility.

Moeslund et al. [18] provides a review of research to the general problem of visual human motion capture and analysis. A review that is specific to the problem of human hand motion estimation is provided in [6].

Section 2.1 provides an overview of systems that perform marker-less hand tracking. Section 2.2 and its accompanying sub sections provide an overview of systems that exploit richer (not markerless) information. Section 2.3 introduces the direction that we follow in this work with respect with the bibliography. Our methodology is able to perform both marker-less hand tracking and exploit the fingertip location priors in an efficient way.

2.1 Tracking approaches for bare hands

This sub section reports marker-less hand tracking methods. These approaches can be characterized as top-down (generative), bottom-up (discriminative) and, hybrid.

2.1.1 Generative methods

Generative methods use parametric articulated models to generate hypotheses. They employ rendering techniques to generate synthetic images and features. Subsequently, they compare them with the observation by utilizing a similarity measure (objective function or energy function) to quantify the discrepancy. They rely on trackers/optimizers to iteratively estimate the hand state that best explains/fits the available observations. The hypothesize-and-test methodology that is followed by the Particle Swarm Optimization (PSO) algorithm [23] and the Particle Filters (PF) [3] has proven particularly suitable for this high-dimensional problem. To tackle the high dimensionality, certain methods [37, 15] create hypotheses hierarchically by exploiting the tree-like kinematic structure of the hand. Typically, local search is performed, seeded by an initial pose which is potentially close to the true solution. Generative methods assume temporal continuity, between frames, to achieve tracking and handle observations with ambiguities and self-occlusions. On the contrary, they are unable to perform single-shot pose estimation and suffer to recover from tracking failures. Optimizer based methods exploit the temporal continuity by accepting only the best estimate at each frame (single hypothesis tracking). Bayesian filtering based methods maintain a set of hypothesis during tracking (multiple hypothesis tracking) to be more robust to tracking loss.

Model-based methods can be adapted easily to different tracking scenarios. For tracking an articulated object with different kinematic structure, generative methods need only to be tuned with the appropriate model. By adding penalization terms or tuning the existing parameters of the objective function of a generative method, it is possible to adapt to different problem requirements or penalize undesirable solutions. As an example, Oikonomidis [23] avoids to produce implausible hand poses with self-collision by using a penalization term that considers finger inter-penetration. Makris and Argyros [13] proposed a model-based approach to jointly solve the pose tracking and shape estimation problem from depth measurements in an on-line framework.

Furthermore, they can directly adapt to different computational and accuracy requirements by adjusting the allowed computational budget (number of available hypotheses evaluations). Despite the relatively high computational requirements of model-based methods, the operations that they perform are usually parallelizable hence implementations that exploit that (e.g. using GPGPUs) are able to achieve real-time performance.

2.1.2 Discriminative methods

Discriminative (data-driven) approaches [44, 11, 37, 8, 21, 48, 47, 35, 38, 5, 29, 30] learn a mapping between image features and the pose space. They require training on large training sets which can be either generated synthetically or captured using real-data.

Synthetic datasets can model, up to a certain level, the appearance of the hand and noise characteristics of real-data. Synthetic datasets aim to densely cover the whole pose space which grows exponentially with the number of joints. The main advantage of synthetic data is that joint (or hand parts) annotations are obtained along with the generation of the synthetic poses. F. Mueller et al. [19] introduce a synthetic RGB-D dataset (*SynthHands*) which contains realistic rendered data for male and female hands, both with and without interaction with objects. The hands and foreground object were synthetically generated along with real object textures and background images (depth and color).

Real-world datasets are captured by recording human hands. Marker-based data capture is not preferred to avoid scene invasiveness and the induced range restriction of the hand motion. Real-world datasets are accompanied with inaccurately annotated data. To reduce the annotation inaccuracy, a manual or semi-auto refinement strategy is followed [20]. Real-world datasets are limited in quantity and coverage of hand pose space, viewpoint and hand shape variability, mainly due to the required laborious work to capture and difficulty to annotate them. Shanxin Yuan et al. [49] introduce a large-scale hand pose dataset (*BigHand2.2M*), collected using a RGB-D sensor along with electromagnetic tracking units. The hand poses were automatically annotated by using an inverse kinematics method that considers the magnetic data. The *BigHand2.2M* dataset includes approximately 290K frames captured from an egocentric view.

On the training phase of discriminative methods, visual features are extracted from each of the training samples. The training phase produces a database or a regressor that is capable to associate each hand pose with the extracted image features. At runtime, features are extracted from the observation and fed as input to the pre-trained regressor. The reported solution is usually hand joint locations. Each joint is treated independently and do not model the complex dependencies between other joints and kinematics of the hand. The resulting estimates are not constrained by hand anatomy or physics. Thus, the obtained hand pose estimates might be implausible.

The main advantages of discriminative methods is that at run time they are more efficient compared to generative ones and are able to perform single shot hand pose estimation. However, the output pose granularity is relatively coarse. They have a fixed accuracy that depends on the density of sampling of the 3D hand pose space. They are not flexible to adapt to different problems, even with minor changes, since their performance critically relies on the training set of labeled frames. They require to be retrained with new data which is a time and computationally consuming process.

2.1.3 Hybrid methods

Hybrid methods [25, 26, 40, 39, 36, 32, 31, 41, 33, 34, 2, 43] attempt to retain the advantages of both the discriminative and generative strategies.

Typically, they employ a discriminative component to arrive at a coarse solution which is then refined by a generative component. The discriminative component aids the generative component in two ways. First, the detected coarse solution aids the fitting step of the generative component to converge faster to a local minima. Second, the coarse solution aids the recovery from failures which generative methods suffer from. On the other hand, the generative component uses an anatomically consistent model to estimate a refined solution that does not violate kinematic constraints. The discriminative component detects hand parts relying on a set of image features. The detected parts are then either incorporated into the objective function of the generative component as soft constraints [25] or as a seed to the optimization [40].

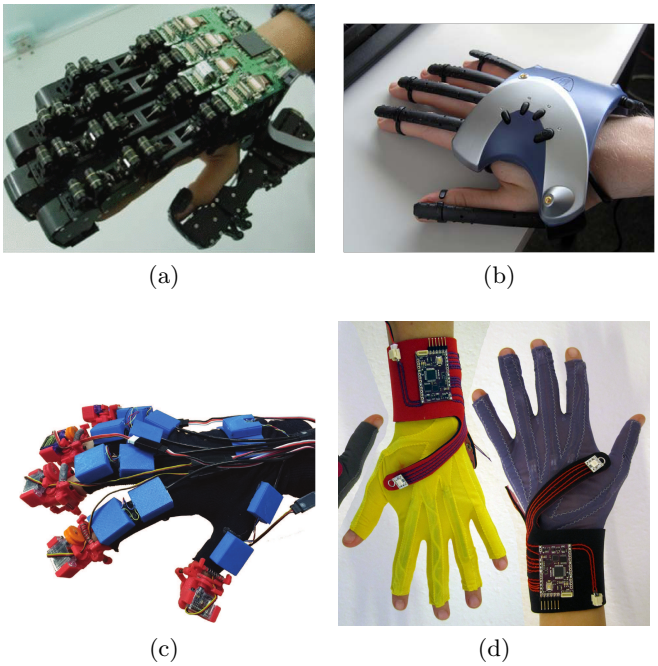
2.2 Tracking approaches that exploit richer-than-markerless information

While marker-less tracking of bare hands is the more general formulation of the problem and as such, the most interesting one, several works have dealt with the problem of richer-than-markerless tracking. There is a variety of solutions, such as optical motion capture systems (mocap), exoskeletons, fabric-integrated sensors and color gloves. These techniques sacrifice the ease of deployment and configuration, low-cost, complexity, and invasiveness of setup for high accuracy and robust tracking. They usually rely on grounded devices and/or structured environments. They modify the appearance of the human hand or augment it with sensors.

2.2.1 Instrumented gloves (non-visual)

Instrumented gloves require the users to wear additional sensing equipment to facilitate hand articulation estimation. These techniques do not require cameras, the only rely on a localization device (e.g. inertial measurement unit) to estimate global orientation and position. Gloves can operate at higher frame rates than camera-based trackers and and have demonstrated precise capture of input for real-time control. Unlike vision-based tracking systems, the sensing glove do not suffer from occlusion problems and lighting conditions. Despite the aforementioned advantages, these type of setups may prevent the users to have a natural interaction with the environment and/or they can be restrictive to hand movement.

Data gloves with exoskeletons (Figure 2.1a, b) allow accurate estimation of the hand pose thanks to their rigid structure and high quality sensors. These systems are typically expensive, bulky and unwieldy.



(a) (b)

(c) (d)

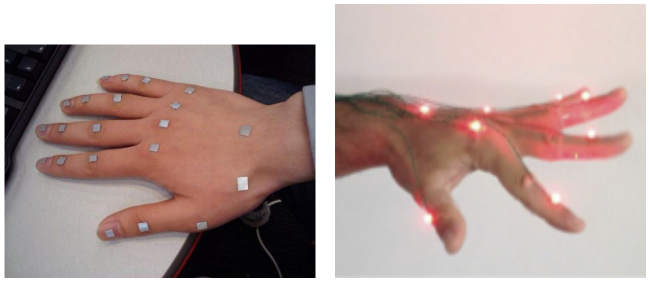
Figure 2.1: Demonstration of instrumented glove solutions. (a) A bulky exoskeleton (b) P5 Data glove (c) GESTO MARG-based glove [12] (d) Fabric-integrated data glove.

Inertial Measurement Units (IMUS) based gloves (Figure 2.1c) augment the human hand with inertial and magnetic sensors, placed in strategic locations. They are less invasive than exoskeletons/data gloves and the data from the sensors can be transmitted wirelessly to a computer. One major drawback is that they suffer from drift over time. Tommaso Lisini Baldi et al. [12] proposed a MARG based glove solution for hand pose estimation along with the ability to provide force feedback by placing cutaneous haptic devices on fingertips (Figure 2.1c). To estimate the hand pose from the MARG sensors, they use Gauss-Newton method (GN) combined with a complementary filter.

Fabric-integrated data gloves (Figure 2.1d) use piezoresistive, fiberoptic, magnetic, and hall-effect sensors. They exploit the conductive properties of the fabric materials to estimate the degree of flexion of fingers. Fabric-integrated systems sacrifice accuracy for being low cost.

2.2.2 Sensor fusion (visual and non-visual)

Sensor fusion techniques attempt to combine information from two non-homogeneous modalities. There are several works, mainly in body tracking, which employ visual and IMU input. Typically, the complementary modalities mutually reinforce one another during inference. The inertial sensors facilitate the pose estimation



(a) (b)

(c)

Figure 2.2: Motion capture setups. (a) hand with retro-reflective markers, (b) hand with LED markers connected by wires, (c) Multi-camera setup of commercial Vicon system.

without rotational and occlusion ambiguities that are contained in visual data. In the meantime the visual are used to refine the coarse or incomplete solution that result from non-visual sensors. The refined, from the visual input, solutions are then applied to correct the global positional and rotational drift artifacts of the inertial sensors.

M. Trumble and colleagues [42] propose fusion approach that is applied to body tracking. Their setup use 8 x 1080p60 RGB cameras and 13 IMU sensors. Multiple views incorporated into a fully 3D convolutional neural network for video-based pose prediction. They augment their pipeline with two extra neural models to integrate temporal continuity and fusion with the inertial data.

To successfully combine two non-homogeneous modalities it takes temporal synchronization between the measurements and spatial calibration between their coordinate frames.

2.2.3 With explicit joint annotations (marker-based)

A straight forward solution to the hand pose estimation problem is to place distinctive markers on the hand of the subject and use them afterwards to estimate

(a) (b)

Figure 2.3: Glove designs of color glove bases solutions. (a) Glove design proposed from Wang and Popovic [45]. (b) Glove design of Roditakis and Argyros [27]

the hand pose. Existing commercial marker-based systems (also called motion capture), such as Vicon (Vicon Motion Systems U.K.) and Optitrack (Natural-Point Inc., USA) follow this approach to solve the hand tracking problem with high precision and accuracy. Optical-Passive motion capture systems use retro-reflective markers that are tracked by infrared cameras. It is the most flexible and common method used in the industry. Optical-Active motion capture systems use LED markers connected by wires to a suit. A battery or charger pack must also be worn by the subject.

Typically, these systems use multi-camera setups to cope with the self-occlusions. Figure 2.2c show the multi-camera setup of a Vicon commercial system. Markers can be either used extensively to cover most of hand joints (Figure 2.2a) or be positioned at strategic points for minimum interference (Figure 2.2b). There is a wide range of solutions to solve the IK problem in order to obtain the hand pose based on the marker positions [1]. Analytical methods work efficiently for simple kinematic chains. The Jacobian methods suffer from the computational complexity and singularity cases. Newton methods and data-driven methods. Finally, the heuristic methods have proven suitable for the IK problem because of their simplicity and speed. Cyclic coordinate descent (CCD) [4] and Forward and Backward Reaching IK (FABRIK) [1] are two heuristic solvers that work well in practice.

2.2.4 Less-invasive setups to solve ambiguities

This type of methods [45, 27] don't use expensive setups so they can be adopted in cost efficient consumer products. They modify the appearance of the hand in less invasive manner. This is achieved by using a thin color glove with a simpler distinctive design that does not explicitly annotate joint locations but aids the accuracy.

Wang and Popovic [45] proposed a data-driven method for hand tracking that uses a single RGB camera and an ordinary cloth glove (Figure 2.3a) which is

imprinted with a custom pattern. Their method tackles the 3D hand tracking problem as a nearest neighbor database search problem. Their results show that they can track the hands with fairly good accuracy.

In previous work [27] we proposed an extension of a model-based, hypothesize and test approach that uses RGB-D data. We used a color glove (Figure 2.3b) with six distinctive colors to segment the hand parts (fingers and palm). Results demonstrated that this extension could robustly track the hands even in low frame rates where the hand motion is observed more dexterous.

2.3 Direction of our approach

Considering the presented literature review, in this work we propose a modification of a generative state of the art method [17] for marker-less 3D hand tracking. Our modification takes advantage of the available spatial constrains on fingertips.

Existing generative and hybrid methods are able to incorporate such priors. However, they do so in a soft manner. More specifically, this is achieved by introducing an error term in the objective function they optimize, which quantifies how far a candidate solution is from satisfying these constraints. The contribution of this error term is then aggregated with all other error terms during optimization. This has two important, negative implications:

1. At the end of the optimization, it is not guaranteed that the solution satisfies the given constraints.

2. During hypothesize and test, a lot of computational effort is wasted in evaluating hypotheses that do not satisfy the available constraints.

In this work, we address these problems. We present a generative hand tracking method that efficiently exploits the available spatial constraints by considering them during the hypotheses generation stage. To best exploit the information about the end effector locations, we rely on the concept of *Reachable Distance Space* (RDS) [46]. RDS provides a fast method to generate hypotheses that respect the constraints. For the kinematic structure of the hand, our proposed methodology can sample hand poses, solve the IK problem in one shot and cope with far away targets. This way, we can significantly narrow the search on the high dimensional pose space. RDS-based sampling is used to extend the Hierarchical Model Fusion particle filter (HMF) [15, 14] to estimate the hand pose. HMF decomposes the hand's state according to the kinematic hierarchy (palm plus five fingers) and thus integrates nicely with the RDS provided hypotheses that also concern specific hand parts (fingertips).

The main claim of this thesis is that careful design and implementation of the steps of a model-based approach can lead to robust, full DoF hand tracking systems that perform close to real-time achieving accuracy in the order of millimeters.

Chapter 3

Foundation on Kinematics

Kinematic chain or linkage system refers to an assembly of rigid bodies (links) connected by joints to provide constrained articulated motion. In computer animation literature, links are named bones since they refer to parts of the skeleton of the animated character. Two consecutive links or bones form a kinematic pair. The intermediate joint in a kinematic pair is the component that restricts the relative motion between two connected links. In most cases, links are grouped hierarchically, each link is associated with a parent link and/or several child links. The link (or joint) that is not associated with any parent link is called the root (base) of the kinematic chain. All joints that are not associated with any child links are called end-effectors and essentially are the end-points of a kinematic chain.

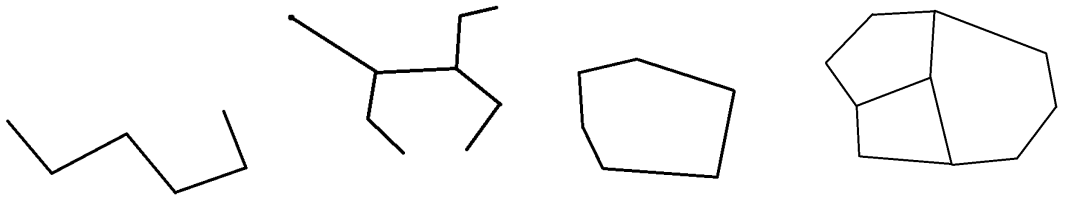
Each kinematic chain and its complexity can be categorized by two factors.

- The topology of its connected links.
- The type of constraints that are assigned at the joints.

Section 3.1 reports common topologies of linkage systems that operate either on 2D or 3D space. Section 3.2 reports common constraints that are used in Robotics and Computer Animation literature. Additionally, sections 3.1 and 3.2 include comments on how each joint parametrization relate to the parametrization of the hand model. Section 3.3 state the two methodological approaches that we employ to shape poses in space.

3.1 Types of kinematic chains

Single-end effectors / Single chain linkages:
It is the simplest type of kinematic chain. It consists of consecutive links that are connected with joints (Fig. 3.1(a)). The most common industrial robots are serial manipulators where the first joint of the chain acts as the base and the last joint acts as the manipulator/end-effector.
Multiple-end effectors / Tree-like graspers:



(a) Chain (b) Tree-like robot (c) Closed chain (d) Multiple loop closed chain

Figure 3.1: Examples of common kinematic chains that are used in robotics and computer graphics

It is a multi-body kinematic chain with a hierarchical topology and includes more than one endpoints (Fig. 3.1(b)). The main part can be a rigid body or a subpart of the chain that is assumed to move as a whole rigidly. The main part is considered to be the base of the chain. Each end-effector is attached to a serial chain or single link that is connected to the main part. More that one links can share the same joint.

We can model the hand as a tree-like kinematic chain. The palm can act as the main rigid part and each sub finger as a single-end effector chain. Another example is the human body where the torso acts as the main part, and the limps are single linkages with one end-effector.

Closed loop chain systems (multiple or single):
Closed chain robots are a generalization of linkages in which chains of links may form one or many loops (Fig. 3.1(c)(d)). Loops in a kinematic chain can be formed in different ways. The native structure of the kinematic chain may demand to form such a topology. For instance, the problem requirements may demand two endpoints to occupy the same point. An alternative requirement can be to keep a fixed distance between two joints. A manipulator (hand) that interacts with an object in known touch points forms a loop in the chain by keeping the distance between its end-effectors fixed.

3.2 Types of constraints on kinematic chains

3.2.1 Biomechanical kinematic constraints

Most kinematic chains are comprised of joints having biomechanical constraints, which provide natural restrictions on their motion. Joint kinematic constraints are mainly characterized by the number of parameters that describe the motion space. A joint is defined by its position and orientation and, in the most general cases, it is a joint of 3 $DoFs$ parametrized using polar coordinates.

Ball-and-socket joint: This is one of the most common joints found in human modeling. It is a joint in which a ball moves within a socket to allow rotary motion

in every direction within certain limits. This type of rotation can be factored into two components: one "simple rotation", named rotational (2 DoFs), that moves the bone to its final direction vector and another called orientational (1 DoF), which represents the twist around this final vector.
Hinge/Planar: These are 1 $DoFs$ articulated joints. The relative motion between two links is restricted to operate on the 2D plane. Linkages connected by adjacent planar joints are coplanar and for chains comprised of only planar joints the entire chain will be coplanar and will operate on 2D.

3.2.2 Spatial constraints

Spatially constraints require certain parts of kinematic chain to remain in a certain area, to maintain contact or a particular clearance from each other.
Joint placement constrains: A joint must remain within the specified boundary. A particular case is **end-effector restriction** constraint which requires the end-effectors to satisfy a spatial constraint.
Closure constrains: This constraint requires a kinematic chain to form closed loops. It can be either a single loop or multiple loops. Two overlapping (end-effector or intermediate) joints in the final configuration of the system result to closed loops. Additionally, closed loops can be formed by keeping a fixed distance between two joints. The fixed distance between the two joints acts as a virtual link which virtually closes the loop.

3.2.3 Self collision avoidance

Collision detection is essential for producing plausible poses in computer animation, physically based modeling and robotics. For instance, in human-like models, body segments often collide with others or the main body. The kinematic constraints of a model do not natively consider the self-collision, so additional techniques are required to restrict its configuration space to avoid self-collision.

3.3 Shaping poses of kinematic chains

Forward kinematics: Is the problem of determining the position and orientation of the final links of the kinematic chain given the joint variables. This problem has a unique solution that can be computed directly.
Inverse kinematics: Is the problem of finding a set of joint variables that will place the final link in a given position and orientation (end-effector restriction constrain). There are typically several solutions for positions and orientations within the workspace ,but these cannot be easily found for an arbitrary kinematic chain. For simple kinematic chains a closed-form solutions exists. Almost all industrial robot arms are designed so that the inverse kinematic problem can be solved directly. For complex kinematic chains, various methods have be developed

which try iteratively to find solutions that minimize the the distance between the end-effector and the target.

Chapter 4

Algorithmic Tools

Section 4.1 describes the main components of generative frameworks and section 4.1.1 presents the baseline method which we build upon our methodology. Section 4.2 introduces the method that we build upon to generate samples for finger articulation which respect to spatial constrains.

4.1 Generative framework for hand tracking

Kyriazis el al. [10] in their technical report presents a generic computational framework that addresses the computational requirements of model-based 3D tracking. Model-based 3D tracking may involve the recovery of the 3D position and pose of a rigid object or the full-articulation of complex objects such as the human body or hand. The applicability of the framework has been validated through its application to various instances of the problem. Actually, works that are presented in [13, 16, 24, 9] follow the proposed paradigm and are able to achive state-of-the-art accuracy and real-time perfomance. They follow a hypothesized-and-test approach that can quantify the compatibility of a hypothesis with the observations and can initialize a search procedure to to find the best scoring hypothesis. Figure 4.1 illustrates the high-level architecture of a model-based framework. Each white box (with blue outline) shows the main component which is assigned a specific role and require different computational resources.

- Search / Hypothesis generation

  This component is the backbone of a model-based framework. It iteratively produces hypotheses which require evaluation. Depending on the score of each hypothesis, it generates new hypotheses for the sake of finding the best hypothesis that explains the observation. In most cases, the search procedure is treated as an optimization instance which exploits the temporal continuity. The baseline method and our methodology use a variant of Particle Filter which can follow the hypothesize-and-test approach.

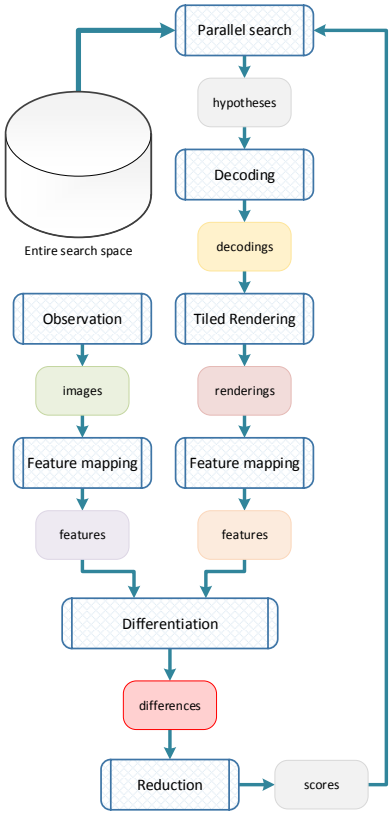


Figure 4.1: GPU-powered computational framework for model-based vision, proposed from Kyriazis el al. [10]

The search space is defined by the parametrization of a model. In the hand tracking problem, the search space is the global position and orientation, and the angles that parametrize the hand articulation. So the search procedure operates on the angle space. Operating on the angle space offers the opportunity to model joint limits constraints seamlessly. The bounds of parameters of search space model the corresponding joint limit constraints of the hand model.

The computational cost of generating parameter values is the most lightweight since it involves low-cost numerical operations. The dominant time-consuming process is the hypothesis evaluation phase. It requires transforming parameter values to features that explain the observation and numerical quantification of the discrepancy. Evaluating a single hypothesis at a time can be prohibitively expensive for achieving real-time performance. There exists search heuristics which permit the parallel evaluation of hypotheses. State of the art frameworks exploit this property for accelerating the hypothesis evaluation phase which is the main performance bottleneck.

- Input preprocessing:

  This component involves transforming raw input to features that are comparable with each hypothesis. The raw input may be visual or non-visual information of the scene. The visual data (RGB or Depth Frames) can be acquired from a single or multiple camera systems. For visual data, the most common preprocessing operations are background subtraction, edge detection, distance transform from silhouettes, skin color detection, color glove segmentation.

  Different modalities can provide non-visual information per frame. An example is inertial data from an IMU/MARG sensor which requires a preprocessing step (filtering) to limit its drift artifacts. In the hand tracking scenarios of this work, the non-visual cues are spatial constraints which we assign in each fingertip, and it does not require a preprocessing step.

  This step exploits the spatio-temporal continuity of the tracking problem. Regularly this is done by reducing the raw input to a 3D-bounding box around the previous frame solution. This step is not computationally intensive, and commonly it is performed in CPU.

- Hypothesis features generation:

  This component involves transforming the model parameters to features that are comparable with the observation. In the case of articulated tracking, hypothesis features may be synthetic images that emulate the observed scene or any other non-visual information about the hypothesized pose. Widely generated features, in hand tracking, is the 3D locations of the joints, silhouette, foreground and depth maps of hand. To produce such features, it involves shaping the pose to space by performing forward kinematics and assigning the generation of synthetic images to a rendering pipeline.

  This step is computationally intensive and can be dealt by composing hypotheses in batches with the aid of parallel architectures. Forward kinematics are lightweight enough to be performed on a multi-core CPU. The Rendering process is computationally intensive and require GPU rendering techniques that are reported in [10].

- Hypothesis evaluation (objective function)

  The Objective function measures the degree of matching between a hypothesized model pose and the observations. It employs the preprocessed observation features and the generated features of a hypothesis. It may perform additional computation of feature maps which facilitate the quantification of the discrepancy. Common operations between map include *differentiation* and *reduction* which result to a distance measure. The design of the objective function and its feature requirements dictates the computational requirements of the whole framework. The search procedure may be able to exploit properties of resulting search surface. Gradient-decent optimization

schemes can utilize first-order differentiable objective functions for the sake of faster convergence.

4.1.1 HMF-Framework for 3D hand-tracking

The baseline method [15] is a Bayesian tracking framework that seeks to estimate the posterior distribution of the hand's state. It is an adapted version of Hierarchical model fusion framework (HMF) [14], which is a particle filter (PF) variant that decomposes the initial problem into smaller and simpler problems and efficiently addresses the implications of the high dimensionality. The HMF uses several auxiliary models that are able to provide information for the state of the main model which is to be estimated. In the hand tracking problem the main model is a full 26-DOF model of the hand configuration. The hand model parametrization (subsection 4.1.1.2) and auxiliary model specifications are loaded at the initialization of the framework. The filter relies on a likelihood model that measures the discrepancy between a rendered hypothesis and the observation. The proposed approach accepts marker-less visual input to track the pose and full articulation of a human hand performing unconstrained 3D motion. By estimating the probability density function of the hand's state posterior it has increased robustness to observation noise and compares favorably to existing methods that can be trapped in local minima resulting in track loses.

4.1.1.1 Visual input and preprocessing

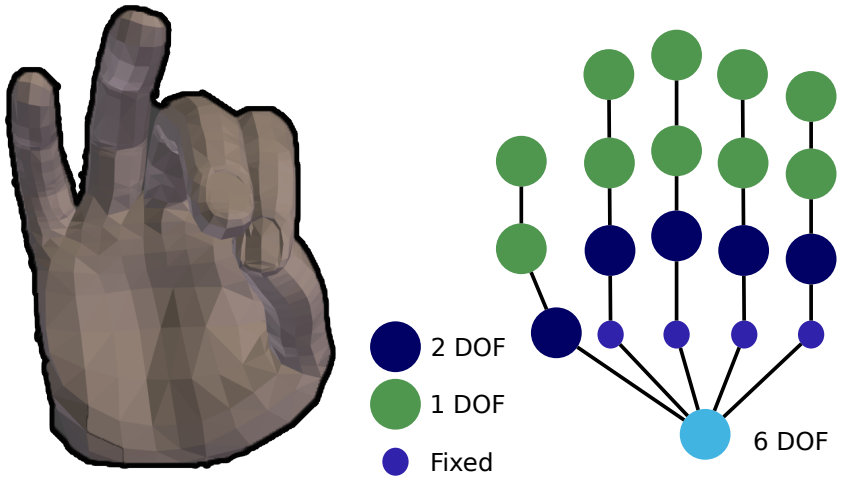
The raw input to the framework is RGB-D frames coming from a commodity sensor. The work of [15] accepts the depth images and applies skin color detection to segment the hand region from the image. In this work we accept as foreground, each non-zero depth pixel from the raw depth map. By experimentation with the baseline method, we noticed that skin color detection may lead to lower quality segmentation when illumination varies significantly. For that reason, we do not process the RGB color frames to infer the hand region.

Additionally, the preprocessing step of the baseline uses the estimated hand position in the previous frame as reference and keeps only the observations that are within a predefined 3D-bounding box around it. The observation $\mathbf{z}$ consists of the cropped 2D depth and foreground map. We denote the observations as $\mathbf{z} = \{\mathbf{z_d}, \mathbf{z_f}\}$ correspondingly.

4.1.1.2 Hand model and parametrization

The method uses a parametric 3D hand model l (Fig. 4.2a) that can be articulated in 3D space. A given hypothesis about the hand configuration provides a hypothesis about the 3D location of every point of the hand model. Its parameterization makes it anatomically consistent, and its appearance is visually realistic.

The hand model consists of a palm and five fingers. The kinematics of each finger are modeled using four parameters, two for the base angles and two for the



(a) (b)

Figure 4.2: The employed 3D hand model: (a) hand geometry, (b) hand kinematics.

remaining joints. Both the finger base and hinge joints use Euler-angle representation. The global position of the hand is represented by a fixed point on the palm and the global orientation by a quaternion representation. This results in a 27 parameter representation that encodes the 26- DOF. It is anatomically consistent by specifying appropriate bounds on the parameters of each joint.

In the work of [15], the appearance consists of a primitive-based model which a crude approximation of the real hand. To achieve better performance, we use a more accurate hand mesh (1597 vertices) animated using a skeleton consisting of 20 bones. We animate each vertice by performing linear blend skinning [7] .

4.1.1.3 Bayesian tracking with state decomposition

The baseline framework follows the Bayesian approach for tracking that were proposed in [14]. By $\mathbf{x_{0:t}}$ we denote the state sequence $\{\mathbf{x_0}...\mathbf{x_t}\}$ and accordingly $\mathbf{z_{1:t}}$ the set of all measurements $\{\mathbf{z_1}...\mathbf{z_t}\}$ from time step 1 to t. The tracking consists of calculating the posterior $p\left(\mathbf{x_{0:t}}|\mathbf{z_{1:t}}\right)$ at every step, given the measurements up to that step and a prior, $p\left(\mathbf{x_0}\right)$.

The HMF framework follows the divide and conquer strategy to update the high dimensional hand state $\mathbf{x}_t$ at each frame, using several auxiliary models and one main model. We use one auxiliary model for the palm (with 6-DOFs for its 3D position and orientation) and one for each finger (with 4-DOFs for the joint angles), as shown in Fig.4.3a. Each of the auxiliary models estimates the state of a hand part. The main model contains all the hand pose parameters and is used to estimate the output state. The purpose of the main model is to combine and fine tune the poses estimated by the auxiliary models.

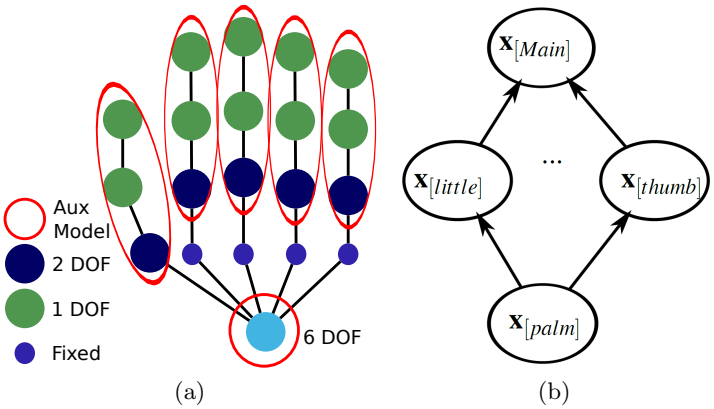


Figure 4.3: State decomposition of hand model with auxiliary models: (a) total of 6 auxiliary models (palm and 5 fingers), (b) update hierarchy.

The auxiliary models are organized in a hierarchy so that each one is able to provide information on the state of its parents in this hierarchy. As shown in Figure 4.3b, we use a hierarchy with 3 levels. The top level contains the main model, the middle level contains the finger auxiliary models, and the bottom level contains the palm auxiliary model.

We define the full state $\mathbf{x}_t$ at a time step t as the concatenation of the sub-states that correspond to the M auxiliary models and the main model $\mathbf{x}_{[0:M]t}$. Using the mentioned above state decomposition, the posterior can be expressed as:

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) \prod_i p(\mathbf{z}_t|\mathbf{x}_{[i]t}) p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})), \tag{4.1}$$

where $Pa(\mathbf{x}_{[i]t})$ denotes the parent nodes of $\mathbf{x}_{[i]t}$ (see Fig. 4.3b). In Eq.(4.1) we make the approximation that the observation likelihood is proportional to the product of individual model likelihoods $p(\mathbf{z}_t|\mathbf{x}_{[i]t})$:

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \prod_i p(\mathbf{z}_t|\mathbf{x}_{[i]t}) \tag{4.2}$$

4.1.1.4 HMF algorithm

To efficiently approximate the posterior given the above state decomposition, we use a particle filter that updates the sub-states. The algorithm approximates this posterior by propagating a set of particles for each model (auxiliary and main) using the importance sampling technique in the same manner as [14]

The steps at time t given the previous estimate are shown in Algorithm 1. The input of the algorithm is the set of N weighted particles from the previous time step $\{\mathbf{x}^{(n)}_{[0:M]t-1}, \mathbf{w}^{(n)}_{t-1}\}^N_{n=1}$ and the current observations z_t. The algorithm sequentially

Algorithm 1 HMF Hand Tracking Algorithm

Input: $\{\mathbf{x}^{(n)}_{[0:M]t-1}, \mathbf{w}^{(n)}_{t-1}\}^N_{n=1}$, $\mathbf{z}_t$.
for each model $i = 0$ to M **do**
  for each particle $n = 1$ to N **do**
    Sample $\mathbf{x}^{(n)}_{[i]t}$ from $p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})^{(n)})$ (Section 4.1.1.5).
    Update its weight $\mathbf{w}^{(n)}_t$ using $p(\mathbf{z}_{[ren]t}|\mathbf{x}^{(n)}_{[i]t})$ (Section 4.1.1.6).
  Normalize the particle weights.
  Resample the particle set according to its weights.
Output: $\{\mathbf{x}^{(n)}_{[0:M]t}, \mathbf{w}^{(n)}_t\}^N_{n=1}$.

updates the sub-states by sampling from the **dynamic model** (see Section 4.1.1.5) that corresponds to the i-th sub-state. Subsequently it updates the weights with the i-th factor of the **observation likelihood** (see Section 4.1.1.6).

The **normalization step** of the algorithm modifies the weights to sum up to one.

The **re-sampling** step randomly chooses particles according to their weights so that particles with low weights are discarded, and particles with high weights are selected multiple times. The output of the algorithm is the current weighted particle set $\{\mathbf{x}^{(n)}_{[0:M]t}, \mathbf{w}^{(n)}_t\}^N_{n=1}$. The **track estimate** of the algorithm at each step is the particle, of the main model particles, with the maximum weight.

4.1.1.5 Model dynamics

The state evolution of each model exploit the state of the updated parent models:

$$p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})) = N(\mathbf{x}_{[M]t}; Pa(\mathbf{x}_{[i]t}), \Sigma_i), \tag{4.3}$$

The model hierarchy is demonstrated in Figure 4.3. $Pa(\mathbf{x}_{[i]t})$ denotes the parent nodes of state model $\mathbf{x}_{[i]t}$. $N(y; m, \Sigma)$ denotes the normal distribution over y with mean m and covariance matrix Σ. The above distribution encodes the fact that the main model is expected to be around the estimated position of its parts.

Specifically for the auxiliary model in the bottom of the hierarchy, we define the state evolution using the main model at the previous time step.

$$\begin{aligned} p(\mathbf{x}_{[bottom]t}|Pa(\mathbf{x}_{[bottom]t})) &\equiv p(\mathbf{x}_{[palm]t}|\mathbf{x}_{[M]t-1}) \\ &= N(\mathbf{x}_{[palm]t}; a(\mathbf{x}_{[M]t-1}, palm), \Sigma_{palm}), \end{aligned} \tag{4.4}$$

where the operator $a(\mathbf{x}_{[M]t-1}, palm)$ gives the part of the state of the main model $\mathbf{x}_{[M]t-1}$ that corresponds to the palm auxiliary model.

4.1.1.6 Observation likelihood

The hypothesis evaluation step of the baseline method is the calculation of the **observation likelihood** $p(\mathbf{z}|\mathbf{x})$. For a specific time step t and a selected auxiliary model i it is denoted as $p\left(\mathbf{z_t}|\mathbf{x_{[i]t}}\right)$. For the sake of clarity the following paragraph, we drop the subscripts that define the time and the model number.We thus refer to the state of a hypothesis by $\mathbf{x}$ and observation by $\mathbf{z}$.

To calculate the observation likelihood for a given hypothesis of an auxiliary or of the main model we perform the following:

1. Use the preprocessed observation $z = \{z_d, z_f\}$ (See section 4.1.1.1).

2. Perform rendering of a hypothesis $\mathbf{x}$ to generate $r = \{r_d, r_f\}$.

3. Compute $P_i = \{z_f \wedge r_f\}$, $P_u = \{z_f \vee r_f\}$, $\lambda = |P_i| \setminus |P_u|$ from z_f, r_f

4. Compute the dissimilarity measure:.

$$D(\mathbf{z}, \mathbf{x}) = \lambda \frac{\sum_{p \in P_i} min(|z_{d,p} - r_{d,p}|, d_M)}{d_M \mid P_i \mid} + (1 - \lambda). \tag{4.5}$$

5. The likelihood is given by the function:

$$p(\mathbf{z}|\mathbf{x}) = \exp\left\{-\frac{D^2(\mathbf{z}, \mathbf{x})}{2\sigma_l^2}\right\}. \tag{4.6}$$

Rendered hypothesis $r = \{r_d, r_f\}$ are synthetic depth and foreground 2d maps analogous with the observation $\{z_d, z_f\}$. P_i are the set of pixels that are labeled as foreground in both the observation and the hand model defined as $P_i = \{z_f \wedge r_f\}$ and P_u be the set of pixels that are labeled as foreground in either the hand model or the observation $P_u = \{z_f \vee r_f\}$. We denote λ as the ratio of the number of elements of these two sets: $\lambda = |P_i| \setminus |P_u|$.

The function $D(z, x)$ evaluates the discrepancy between a hypothesis $\mathbf{x}$ and the observation $\mathbf{z}$. This ranges from 0 for a perfect match to 1 for a mismatch. The intuition for this definition is that we weight by the clamped depth difference the part of the pixels that overlap in the model and observation (P_i) whereas the rest of the pixels influence negatively the total distance. The clamping threshold d_M is required so that a few pixels with big depth differences should not influence an otherwise reasonable fit. Pixels that have depth difference above d_M are considered mismatched. The definition for the distance guarantees that these mismatched pixels will penalize $D(z, x)$ with the maximum value thus in exactly the same way as the pixels that are not in P_i. This is justified because in both these cases the corresponding 3D observation and hand model points are considered to be far from each other. Using $D(z, x)$ the likelihood function is then given equation 4.6.

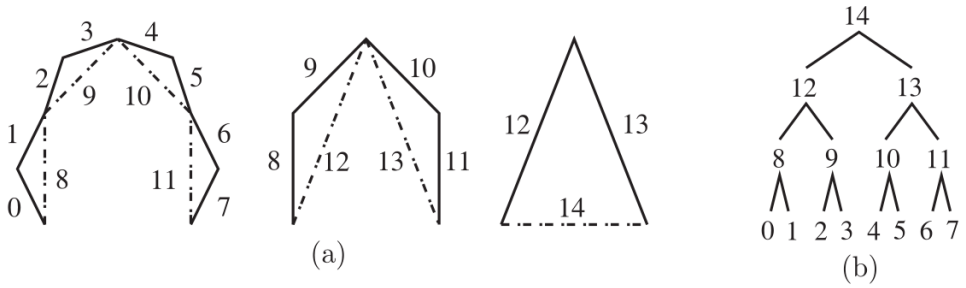


Figure 4.4: Articulated linkage with eight links (labeled 0–7). Two consecutive links form a parent vlink (dashed lines). This repeats to form a hierarchy (b) where the parents in one level become the children in the next level. (b) The tree represents the entire reachable distance hierarchy where the nodes correspond to the vlinks.

4.2 Sampling spatially constrained poses in Reachable Distance Space

Authors of [46] introduce the concept of **Reachable Distance Space (RD-Space)** which is an alternative space that encodes spatial constraints between joints in a kinematic chain. The **reachable distance** of a kinematic chain is the range of distances that its end effector can reach with respect to its base. Sampling in the RD-Space guarantees that the resulting pose will satisfy the specified spatial constraints.

The authors of [46] mainly apply and evaluate their methodology in probabilistic motion planning for robots with a various number of links. Apart from motion planning, they demonstrate an application of their method which a serial chain can solve the IK problem, but at the same time, it can produce random poses. The RDS sampling method [46] can efficiently cope with kinematic chains of 1-DOF planar joints. Therefore, it is suitable for sampling the pose of each finger. We built-upon their proposed application, and we integrate the RDS sampling scheme to the hand tracking problem.

4.2.1 Reachable Distance Space formulation

The method considers a kinematic chain with several links. To enable spatial constrains-aware sampling, they redefine the chain's degrees of freedom and constraints into a new set of parameters, called reachable distance space (RD-space). They compute a hierarchical data structure, called **reachable distance tree (RD-tree)**, by recursively partitioning the original system into smaller sub-systems. In a kinematic chain, two consecutive links define a **virtual link (vlink)** or **sub-chain c**. The links (children) that form a link (parent) can be actual links or previously constructed vlinks.

The reachable distance of a vlink (sub-chain) is the distance between the endpoints of the sub-chain it represents. It has different values for different configurations. The range of those values is the **reachable range (RR)** of the vlink (sub-chain). The RR of a parent can be calculated from the RRs of its children. Let l_{min} and l_{max} be the minimum and maximum allowable values of the vlink's length.

(a) For an actual link that is not prismatic, then $l_{min} = l_{max}$. RR is simply the range of its length $RR = [l_{link}, l_{link}]$.

(b) For a vlink l with two children, a and b, a triangle is formed. Let $RR_a = [a_{min}, a_{max}]$ be the RR of the first child and $RR_b = [b_{min}, b_{max}]$ be the RR of the second child. The RR of the parent vlink l (or subchain c) is $[l_{min}, l_{max}]$ where

$$l_{min} = \begin{cases} max(0, b_{min} - a_{max}), & a_{min} < b_{min}, \\ 0, & a_{min} = b_{min}, \\ max(0, a_{min} - b_{max}), & a_{min} > b_{min}, \end{cases} \tag{4.7}$$

$$l_{max} = a_{max} + b_{max} \tag{4.8}$$

(c) When considering joints with angle limits in 2D, the RR of a *vlink* that is comprised of two actual links is given by the distance between its endpoints for the cases of minimum and maximum joint angle.

The *RD-Tree* is constructed by recursively joining the links of a chain into *vlinks* until a single root *vlink* is constructed (Fig. 4.4).

Given the RRs of any two links in the same triangular sub-chain, Equations (4.7) and (4.8) can calculate the RR of the remaining link to satisfy the triangle inequality: $|a - b| \leq |c| \leq |a + b|$. If the RR of a vilink in a subchain is updated, we use equations (4.7) and (4.8) to update accordingly the RR of adjacent vlinks. We define the **available reachable range (ARR)** as the subset of the RR that satisfies the spatial constraints with respect to the rest of the sub-chain.

4.2.2 Sampling procedure

Given a computed RD-Tree, the sampling procedure is performed in 3 steps.

1. Recursively sample vlink lengths from their ARR. (section 4.2.2.1)
2. Sample the orientation of each sub-chain. (section 4.2.2.2)
3. Compute appropriate joint angles from the vlink lengths. (section 4.2.2.3) and orientations.

Algorithm 2 Sample lengths in RD-Tree

Input: A sub-chain c. Let $c.arr$ be c's ARR, $c.left$ and $c.right$ be c's children, and $c.len$ be the length of c's vlink. Let p be c's parent and s be c's sibling.

1: Update $c.arr$ from $p.arr$ and $s.arr$ (from Equations (4.7) and (4.8) where link l is c, link a is p, and link b is$s$ in the equations).
2: Randomly sample $c.len$ from $c.arr$.
3: Set $c.arr$ to $[c.len, c.len]$.
4: **if** c has children then **then**
5: $\quad Sample(c.left)$
6: $\quad Sample(c.right)$

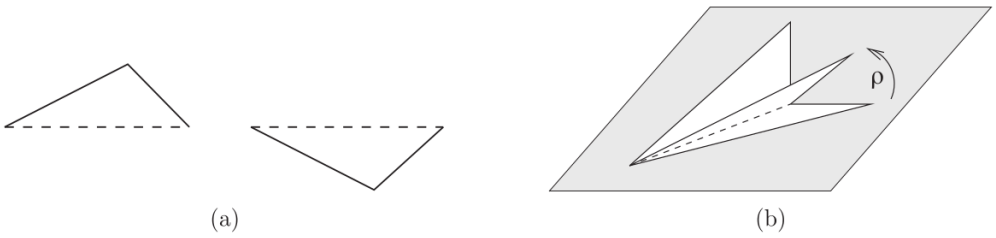


Figure 4.5: In two dimensions, the same vlink represents two configurations: (a) concave triangle and a convex triangle. (b) In three dimensions, the same vlink represents many configurations with different dihedral angles ρ. (Here ρ is the angle between the sub-chain's plane and its parent's plane.)

4.2.2.1 Recursively sample link lengths

RDS sampling is performed by recursively sampling the lengths of the *vlinks* of the RD-Tree, starting from the root *vlink* and descending the hierarchy. After a *vlink* length is sampled, the available RRs of its sibling and children *vlinks* are restricted and have to be recalculated. Algorithm 2 describes this recursive sampling strategy.

4.2.2.2 Sample link orientations

Each sub-chain forms a triangle, and there are multiple configurations with the same vlink length: there are two in two dimensions (i.e. concave and convex, Figure 4.5 (a)), and there are many in three dimensions depending on the dihedral angle between its triangle and its parent's triangle (see Figure 4.5 (b)). Thus, we also sample the orientation of the vlink. This orientation sampling is done after all of the vlink lengths are sampled.

4.2.2.3 Back to joint angles

$$\theta = arccos\Big(\frac{l_a^2 + l_b^2 - l_c^2}{2l_a l_b}\Big) \tag{4.9}$$

The sampling process results in a set of lengths, one for each vlink, for a configuration that satisfies the spatial constraints. Then, we can compute the joint angles between vlinks using only basic trigonometry functions instead of more expensive inverse kinematics solvers. Consider the joint angle θ between links a and b. Links a and b are connected to a vlink c to form a triangle. Let l_a , l_b , and l_c be the lengths of links a, b, and c, respectively. The joint angle can be computed using the law of cosines (equation 4.9).

4.2.3 Application of RDS: restricted end-effector sampling

Reachable distance formulation can be used to efficiently sample configurations of an articulated linkage when its end effector required to remain within a specified boundary, such as a work area or a safe zone. Consider a serial kinematic chain (or a manipulator) with a single end-effector e and a fixed base b. The RR of the kinematic chain is the range of distances from r to e under all chain configurations. Its end effector is restricted to remain inside the box or **target area T**. To sample such pose:

1. Randomly sample a point t in T.
2. Calculate the distance d_t between the base r and t.
3. Set the ARR of the root $vlink$ of RD-Tree to $ARR_{root} = [d_t, d_t]$.
4. Perform sampling in the RD-Space (section 4.2.2).
5. Compute the new position of end-effector e by shaping pose of kinematic chain.
6. Compute and apply the rotation θ to the base b in order to place end-effector e in the location of t.

The above sampling procedure guarantees to generate a constraint-satisfying sample in one shot if the target t is within the working area of the manipulator.

Chapter 5

Methodology

This chapters presents the methodology and main contributions of this thesis. Section 5.1 introduces the Constrained-HMF (C-HMF) method. We extend the HMF framework [15] by tightly integrating our RDS-based, constraints-aware sampling strategy and is shown to achieve state of the art hand tracking accuracy, while requiring the evaluation of much less hand hypotheses, all of which satisfy the given constraints. Subsection 5.1.4 presents our constraints-aware sampling strategy. We employ RDS to consider explicitly spatial and kinematic constraints at the hand pose hypothesis generation phase. In that direction, we developed a simple and fast method to consider the finger joint limits, extending the original RDS formulation [46] and, thus, rendering it suitable for the real-time performance requirements of the hand tracking problem.

5.1 Constrained-HMF framework

5.1.1 Overview

The state of the hand model is estimated using an adapted version of the HMF tracking framework [15, 14] denoted as C-HMF. C-HMF follows the hypothesize and test approach. The generated hypotheses satisfy both the hand's kinematic constraints (motion model, joint limits) and the available end-effector target constraints. This way, all hypotheses are valid, and sampling efficiency is greatly enhanced; therefore fewer particles are required to achieve the same tracking accuracy. An end-effector target can be either a specific 3D point or a 3D region. In the latter case, we randomly pick a specific 3D point within this region. Not all finger end-effectors are required to be associated with target constraints at any frame. For unconstrained fingers, we generate pose hypotheses that only respect the hand's kinematic constraints (motion model, joint limits).

5.1.2 Extending the HMF framework (C-HMF)

Algorithm 3 C-HMF Hand Tracking Algorithm

Input: $\{\mathbf{x}^{(n)}_{[0:M]t-1}, \mathbf{w}^{(n)}_{t-1}\}_{n=1}^{N}$, $\mathbf{z}_t$.
for each model $i = 0$ to M **do**
  for each particle $n = 1$ to N **do**
    Constraints Aware Sample $\mathbf{x}^{(n)}_{[i]t}$ from $p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})^{(n)})p(\mathbf{z}_{[trg]t}|\mathbf{x}^{(n)}_{[i]t})$.
    Update its weight $\mathbf{w}^{(n)}_t$ using $p(\mathbf{z}_{[ren]t}|\mathbf{x}^{(n)}_{[i]t})$.
  Normalize the particle weights.
  Re-sample the particle set according to its weights.
Output: $\{\mathbf{x}^{(n)}_{[0:M]t}, \mathbf{w}^{(n)}_t\}_{n=1}^{N}$.

The C-HMF formulates the hand-tracking problem identically as the baseline method (section 4.1.1). It follows the Bayesian approach for tracking. Tracking amounts to calculating the posterior $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ at every step, given the measurements up to that step and a prior, $p(\mathbf{x}_0)$. By $\mathbf{x}_{0:t}$ we denote the state sequence $\{\mathbf{x}_0...\mathbf{x}_t\}$ and by $\mathbf{z}_{1:t}$ the set of all measurements $\{\mathbf{z}_1...\mathbf{z}_t\}$ from time step 1 to t.

The C-HMF follows the divide and conquer strategy to update the high dimensional hand state $\mathbf{x}_t$ at each frame, using several auxiliary models and one main model. The auxiliary models are organized in a hierarchy so that each one can provide information on the state of its parents in this hierarchy. We use the same 3 level hierarchy as the baseline method. The top level contains the main model, the middle level contains the finger auxiliary models, and the bottom level contains the palm auxiliary model. We define the full state $\mathbf{x}_t$ at a time step t as the concatenation of the sub-states that correspond to the M auxiliary models and the main model $\mathbf{x}_{[0:M]t}$ Using the state decomposition, the posterior can be expressed as:

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) \prod_i p(\mathbf{z}_t|\mathbf{x}_{[i]t}) p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})), \tag{5.1}$$

By $\mathbf{x}_{0:t}$ we denote the state sequence $\{\mathbf{x}_0...\mathbf{x}_t\}$ and by $\mathbf{z}_{1:t}$ the set of all measurements $\{\mathbf{z}_1...\mathbf{z}_t\}$ from time step 1 to t.

To efficiently approximate the posterior given the above state decomposition, we use a particle filter that updates the sub-states. The algorithm approximates this posterior by propagating a set of particles for each model (auxiliary and main) using the importance sampling technique.

In order to incorporate spatial constraints, we modify the basic components of the baseline filter accordingly. The basic components of the filter are the **state evolution dynamic model** and the **proposal distribution** that is used to sample, and the **observation likelihood**.

The proposal distribution, described in detail in Sec. 5.1.4, generates particles that respect the dynamic model, and the end-effector position constraints when

available. We augment the observation likelihood, described in Sec. 5.1.3, with a target likelihood component.

The state estimate for each frame $\bar{\mathbf{x}}_{[M]t}$ is given by the main model particle with highest weight. The steps of the algorithm are summarized in Algorithm 3.

5.1.3 Observation likelihood

The observation likelihood has two components:

1. The rendering component $p(\mathbf{z}_{[ren]t}|\mathbf{x}_{[i]t}^{(n)})$ compares a hypothesized, rendered hand model and the RGB-D image as in [15]. The result of that comparison is a distance D_{ren} (normalized in [0, 1]) that takes into account the silhouette and depth match of the rendered hypothesis and the actual observations. The rendering likelihood is calculated as an exponential function of D_{ren}:

$$p(\mathbf{z}_{[ren]}|\mathbf{x}) = \exp\left\{-\frac{D_{ren}^2(\mathbf{z}_{[ren]}, \mathbf{x})}{2\sigma_{ren}^2}\right\} \tag{5.2}$$

2. The target likelihood component $p(\mathbf{z}_{[trg]t}|\mathbf{x}_{[i]t}^{(n)})$ is an exponential function of the average distance D_{trg} between the end-effector targets and the hypothesized end-effector positions with standard deviation σ_{trg}.

The total likelihood is given as the product of these two components.

5.1.4 Constrain-aware hypotheses generation

Several techniques are integrated to generate constraints-aware hypotheses for each C-HMF sub-model (auxiliary and main) at each time step t.

The palm auxiliary model is updated first, according to the C-HMF hierarchy. For the particles of the finger auxiliary models, we sample from a proposal distribution $q(\mathbf{x}_{[finger]t}|\mathbf{x}_{[palm]t}, \mathbf{z}_{[trg]t}) = p(\mathbf{x}_{[finger]t}|\mathbf{x}_{[palm]t})p(\mathbf{z}_{[trg]t}|\mathbf{x}_{[i]t}^{(n)})$ which is conditioned on the updated palm sub-state and the finger target likelihood. This proposal generates valid kinematic samples that satisfy the end-effector target of that finger if available (see Section 5.1.4.2). In this step, only the finger joint angles are modified and fingertip targets can be reached only if the corresponding palm pose is appropriate. For fingers with no associated end effector constraints, we sample from the palm-conditioned term of the proposal distribution $p(\mathbf{x}_{[finger]t}|\mathbf{x}_{[palm]t})$.

Finally, the main model samples are generated from a proposal distribution that is conditioned on the updated palm and finger pose provided by the auxiliary models and takes into account all the available end-effector target constraints (see Section 5.1.4.2).

Algorithm 4 Constrain-Aware Palm Sampling

Input: $x_{[palm]}$, $p_{efs_{[1:F]}}$, p_{root}, $\mathbf{z}_{trg_{[1:F]}}$.
//Step 1, Unconstrained sampling
Sample $x_{[palm]}$ from HMF dynamic Model.
//Step 2, Fit to constrains
if $\mathbf{z}_{trg} \neq \emptyset$ **then**
  Update p_{efs}, p_{root} from new $x_{[palm]}$
  Add noise to p_{efs}
  $P = [p_{root}, p_{efs_{[1]}}, p_{efs_{[2]}}, ..., p_{efs_{[F]}}]$
  $Q = [p_{root}, \mathbf{z}_{trg_{[1]}}, \mathbf{z}_{trg_{[2]}}, ..., \mathbf{z}_{trg_{[F]}}]$
  $(R, t) = SVD(P, Q)$ (Least Squares Fitting)
  //Update palm particle
  Apply rigid transformation (R,t) to particle $x_{[palm]}$
Output: $x_{[palm]}$.

5.1.4.1 Rigidly fitting a sampled hand pose in order to satisfy constrains

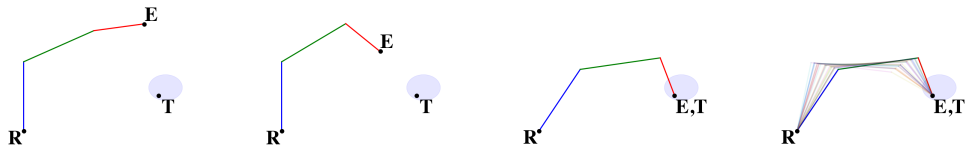
This step samples palm poses in regions where all available target constraints can be reached by the fingers. To do so, we apply a technique which performs 2 steps which are demonstrated in algorithm 4.

First, we use the dynamic model (Gaussian distribution) of the palm auxiliary model to update an existing palm particle $x_{[palm]}$.

Second, we perform a fitting step to ensure that the targets constraints remain within the reachable area of the hand. The fitting step is performed if target constraints exist for the current frame t. We perform fitting in the least squares sense by computing a rigid transformation with **Singular Value Decomposition**. To compute it we utilize two sets of points P,Q. P includes the root joint x_{root} and the end effector positions $p_{efs_{[1..F]}}$ that are assigned with a target. Q includes the root joint p_{root} and the available target positions $\mathbf{z}_{trg_{[1:F]}}$. We append the root joint in both sets, to avoid transformations that exceedingly relocate the hand's root joint. Additionally, we add noise to the end-effector positions of the first set. This enables us to achieve hand reachability without resulting to palm poses that are highly biased to the latest hand articulation estimations.

5.1.4.2 Sampling finger articulation in the Reachable Distance Space (RDS)

We present the method that we follow to generate samples for the finger joints given the palm position and orientation. The generated samples respect the hand dynamics and the end effector target constraints. The procedure has two steps which are illustrated in Fig. 5.1:



(a) (b) (c) (d)

Figure 5.1: An illustration of the RDS-based sampling process. (a) A simple model of a finger, consisting of three links. R denotes the base of the finger, E the end effector and T the finger end effector target position picked from a target region (blue area). (b) RDS sampling defines the hinge joint angles so that $|RE| = |RT|$. (c) A rotation at the joint base brings E at T. (d) Different solutions in step (b) result in different finger configurations that respect the end effector constraints.

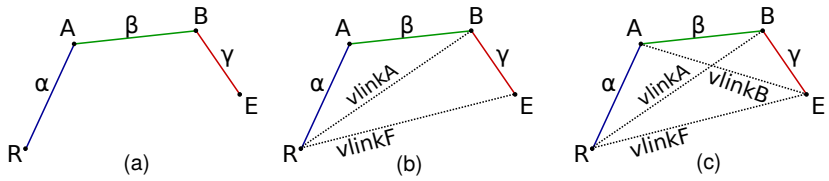


Figure 5.2: RD-Tree construction example on a 3-link chain (finger): (a) Initial chain, (b) RD-Tree, (c) RD-Tree augmented with *vlinkB*.

1. Sample a finger articulation (proximal inter-phalangeal joint, and distal inter-phalangeal joint) in the Reachable Distance Space which satisfies the target distance constraint. This step is detailed in the rest of the section.

2. Orient the finger by modifying its base (metacarpophalangeal) joint so that its end-effector lies in the line defined by the base-joint and the target. In this step, we consider the joint-angle limits of the finger base.

5.1.4.3 Finger RD-tree construction

The RD-Tree of a finger is visualized in Fig. 5.2. Because all fingers of the hand model consist of 3 bones, we construct an RD-Tree for each finger by applying the following. We construct *vlinkA* from the actual links α, β and we calculate its initial RR_A from triangle RAB and joint A angle limits. We construct the root vlink, *vlinkF*, from *vlinkA* and γ. The minimum and maximum joint A and B angles define the RR_F of *vlinkF*.

Assuming that the end-effector target and the finger base position are set, the target distance (length of the root *vlinkF*) is determined. Therefore, the direct application of the recursive RDS sampling procedure reduces to sampling a single distance for *vlinkA*. Sampling *vlinkA* will always satisfy the limits of joint A since it is comprised of actual links. However, the joint limits of joint B are not guaranteed.

In practice, the majority samples in RD-space that try to satisfy target distances near the minimum RR of the root *vlink* violate the limits of joint B.

5.1.4.4 Incorporating joint limits in the sampling scheme

We propose an alternative sampling procedure that is able to incorporate hinge joint limits. For a target (root *vlinkF*) distance, we seek to restrict the RR_A of *vlinkA* to a range that sampled distances will not force joint B to violate its joint limits. To do so, we augment the RD-Tree with an additional vlink. Specifically, *vlinkB* is constructed from the actual links β, γ and its initial RR_B is calculated from the triangle ABE and joint B angle limits (see Fig. 5.2(c)). Given this configuration, samples are drawn by the following steps:

(a) Update the RR_A of *vlinkA* from the lengths of link γ and root *vlinkF* (triangle RBE).

(b) Update the RR_B of *vlinkB* from the lengths of link α and root *vlinkF* (triangle RAE).

(c) Update the RR_A of *vlinkA* from minimum and maximum RR_B lengths since these lengths uniquely determine *vlinkA* length.

(d) Sample in RDS the updated RR_A.

(e) Compute hinge joint angles from vlink distances.

This process guarantees that sampling in this updated Reachable Distance Space will result in configurations that do not violate any of the finger's hinge-joints limits since by construction RRs respect the limits and the subsequent steps do not expand them.

Chapter 6

Constrained Hand Tracking Scenarios

In this chapter, we assess the work of this thesis in various 3D hand tracking scenarios where the motion of the hand is constrained. In all the following cases, the constraints are available as fingertip placement constraints. However, the type of hand motion, number of constrained fingertips and availability of constraints vary depending on the tracking scenario. We assess the tracking performance against the baseline **HMF**, **HMF-SP** which essentially is a HMF extension that consider soft constrains (see section 6.2), and our proposed **C-HMF** framework.

We perform both quantitative and qualitative experiments. In both kinds of experiments, we compiled our challenging sequences which enable us to emphasize the advantages of our method. For the **qualitative evaluation** of the methods, we used real data obtained by a single RGB-D sensor. More specifically, we used a Kinect1 sensor configured to acquire depth frames registered to the RGB image, in VGA resolution and with acquisition rate 30 fps. For **quantitative evaluation**, we generated synthetic data since real-world annotated data are difficult to obtain. We followed a common practice in the field [23, 22], that is, to first track real sequences and then use the tracking result as the basis for generating ground-truth annotated synthetic sequences by means of rendering. The model we use to produce the synthetic frames is identical to the model that each method use for tracking in our experiments.

The following sections in this chapter, describe each experiment setup in detail and reports the findings of our evaluation.

6.1 Evaluation criteria

We use several error metrics to assess the performance of the methods:

- $\mathbf{E_j}$ measures the average distance between corresponding phalanx endpoints over a sequence. This is the most common error metric that is used in

bibliography. Since it is an average of all joints, it provides a generic overview of the tracking accuracy.

- $\mathbf{E_{ee}}$ measures the average distance between corresponding end-effectors. End-effectors are the joints which their location provide important information about the hand pose and type of interaction with the environment. There exists manipulation scenarios in VR that the location of fingertip joints is used to infer if the hand is manipulating an object and the type of manipulation. Other tracking scenarios, such robot tele-operation or medical applications, may require high-precision and low-jerkiness trajectories of the fingertip locations through the sequence. We investigate this alternative error metric which we believe that is more suitable for these type of applications.

- $\mathbf{E_{trg}}$ measures the average distance only for the end-effectors that have been associated with constraints. By using this metric, we can assess that our proposed framework can in practice satisfy the constraints while estimating the full pose parameters of the hand.

- **C** is the ratio of the frames of the sequence for which the maximum position error of phalanx endpoints is below a certain threshold.

The methods that we use are not deterministic. We repeat each experiment several times and we report the median error.

6.2 Comparison with soft constrains (HMF-SP)

We extended the baseline **HMF** method to consider target positions as soft constraints. We do this by augmenting the observation likelihood with a **target constraint likelihood** $p(\mathbf{z}_{[ren]}|\mathbf{x})$. The augmented likelihood is defined as the weighted average of the rendering and the target constraint likelihoods:

$$p(\mathbf{z}|\mathbf{x}) = lp(\mathbf{z}_{[trg]}|\mathbf{x}) + (1-l)p(\mathbf{z}_{[ren]}|\mathbf{x}) \tag{6.1}$$

l is the weight factor which can act as the main tunable parameter for achieving different accuracy. The standard deviation parameters for both likelihood components σ_{ren}, σ_{trg} are set to 0.005. In following sections we denote this method as **HMF-SP**.

6.3 Hand motion constrained in stationary touch points

In this scenario, it is assumed that a hand moves while some of the fingertips lie at known points on a planar surface. We provide three such sequences:

- **ALLFNG** where all the fingertips are constrained to remain at specific locations. The hand motion is quite limited. For qualitative evaluation, we

acquired 554 frames of real data. For quantitative evaluation, we generated a synthetic sequence of analogous motion that consists of 254 frames.

- **IDXMDL** where the index and middle finger are constrained and the rest can move freely. For qualitative evaluation, we acquired 631 frames of real data. For quantitative evaluation, we generated a synthetic sequence of analogous motion that consists of 631 frames.
- **IDXTHM** where the index and the thumb are constrained and the rest can move freely. For qualitative evaluation, we acquired 481 frames of real data. For quantitative evaluation, we generated a synthetic sequence of analogous motion that consists of 319 frames.

To estimate the contact points in real data, we placed sticker markers on the planar surface, and we performed manual annotation. The subject was asked to perform a hand choreograph while the hand fingertips lie on the corresponding markers.

6.3.1 Quantitative evaluation

Figure 6.1 plots the obtained results for all error metrics (rows) and provided sequences (columns). In all cases, the proposed **C-HMF** method (red curve) outperforms the baseline HMF variant as well as **HMF-SP**. **HMF-SP** performs better **HMF**, but the performance gain is not that significant. The discrepancy in accuracy between the proposed and the rest of the evaluated methods increases if we consider only the end effectors with constraints (4th row) compared to all end effectors (3rd row) and all hand joints (2nd row). However, the results of the 2nd row suggest that **C-HMF** does not only improve the estimation of the 3D hand end effectors alone, but the full articulation of the hand. Additionally, the 4th row of Figure 6.1 demonstrate that our proposed method can satisfy the given constrains. For all provided sequences. It maintains an error of $2mm$ which in practice is close to zero.

6.3.2 Qualitative evaluation

Sample results are shown in Figures 6.2, 6.3, 6.4 . The results concern the sequences **ALLFNG**, **IDXMDL** and **IDXTHM** respectively. We compare the proposed **C-HMF** to the **HMF** method. All utilize 160 particles. In **ALLFNG** sequence, all fingers have known contact points with a planar surface and hand motion is quite limited. In **IDXMDL** and **IDXTHM** sequences two fingers have known contact points with a planar surface while the rest can move freely. The results show that **C-HMF** estimates accurately the articulation of the constrained fingers even when they are partially, or even almost fully occluded. Furthermore, the constrained fingers provide anchor points for the palm whose pose is, therefore, better approximated. The state estimation for the rest of the fingers benefits from the better palm pose estimation.

6.4 Free hand motion with provided fingertip locations

In this scenario, a detector provides the fingertip positions at each frame. We assume that the detector is a component that acts independently and provides its output detections as input to our method. The number of the detected fingertips, as well as the accuracy of the detection, vary. It is essential to consider these artifacts since they represent the limitations of detectors that use non-invasive and simple camera setups. Our method can cope with noisy detections by sampling a fingertip target from an area around the noisy detection. We expect that our method is robust to this type detections and does not require the unrealistic assumption of noise-free detections.

6.4.1 Quantitative evaluation

One sequence is provided for this scenario, **FREEHM**. It consists of 326 frames where the hand performs unconstrained motion. We simulated the limitations of a fingertip detector, that is, inaccurate detection of positions and missed detections. We used the **FREEHM** sequence and its accompanying ground truth in two different experiments.

First experiment: We assessed the tolerance of **C-HMF** to errors in the estimation of the target constraints. To do so, in each frame constrained 5 fingers and we added Gaussian noise to the true positions of the fingertips. We considered the performance of **C-HMF** running with 40 particles, as well as of the baseline **HMF** method for two different computational budgets, that is 40 particles (**HMF-40**) and 200 particles (**HMF-200**). For each experiment configuration, we repeated the corresponding experiment five times. Figure 6.5 plots the obtained results for all error metrics. It can be verified that for noise-free data, the **C-HMF** has 4 times smaller error that **HMF** when they both run with 40 particles. Even if the budget of **HMF** is increased to 200 particles (5× budget of **C-HMF**), **C-HMF** maintains half the error. In order to match the performance of **HMF-40** and **HMF-200**, the standard deviation of the error in the estimation of the constraints should reach $20mm$ and $10mm$, respectively.

Second experiment: We assessed the tolerance of **C-HMF** both to errors in the estimation of the target constraints and missed detections. We constrained 2, 3, 4 and 5 of the 5 fingers. In each frame of the sequence, the actual ids of constrained fingers were selected randomly and independently. To each true fingertip position, we added Gaussian noise of $8mm$ standard deviation. We run **C-HMF** five times for each different number of constrained fingers using 40 particles and measured E_j. E_j varied between **7.5mm (5 constrained fingers)** and **10.0mm (2 constraint fingers)**, showing that as the number of constraints increase, the accuracy in hand tracking also increases. Results demonstrate that **C-HMF** does not require the fingertip detections to remain stationary in order to benefit the pose estimation.

6.5 Implicitly providing fingertip locations from a tracked object

In this scenario, the fingertip constraints are implicitly provided by tracking a rigid object while being manipulated by the hand. The fingertip locations can be inferred if hand-object contact points are known as prior, and the 3D pose of an object is estimated. The pose estimation of a rigid object is a lower dimensional (6-D) problem than the hand tracking problem (26-D). In some cases, tracked objects contain texture or visual cues that can facilitate robust estimation with minimal computational resources. The inferred touch points can act as strong priors/fingertip constraints for the CHM-F Framework. These priors can facilitate the robust tracking in challenging cases of dexterous motion, occlusions from interactions, and ambiguities that result from a cluttered background.

6.5.1 Qualitative evaluation

Sample results are shown in Figure 6.6. The results concern the **CALIB** sequence which consists of 641 frames. In this experiment we compare the proposed **C-HMF** to the **HMF** method. Both methods utilize 160 particles. In **CALIB** sequence, the hand interacts with a rectangular cardboard with a calibration pattern imprinted on it. The index and thumb finger manipulate the cardboard through a handle at known contact points. The contact points were manually annotated in the object space of the cardboard. At each tracked frame, we used the calibration pattern along with least squares fitting to estimate the 3D pose of the cardboard. Then we inferred the 3D locations of the contact points and applied them as constraints in the C-HMF method.

The results show that **C-HMF** estimates accurately the articulation of the constrained fingers in all frames. The applied constraints benefit the estimation of the palm and free fingers. The **HMF** can estimate the hand pose with a fairly good accuracy, but artifacts appear in challenging interactions. The flat surface of the handle of the cardboard introduces ambiguities in depth information. For **HMF**, this results to imprecise estimation of the index and thumb finger. The **CALIB** contains a dexterous translation of the hand along with the cardboard. For the **HMF** method, this type of motion results to inaccurate pose hand estimation or tracking failures which are hard to recover. The **C-HMF** can robustly and accurately track this challenging sequence, by incorporating strong priors that result from the pose estimation of the cardboard.

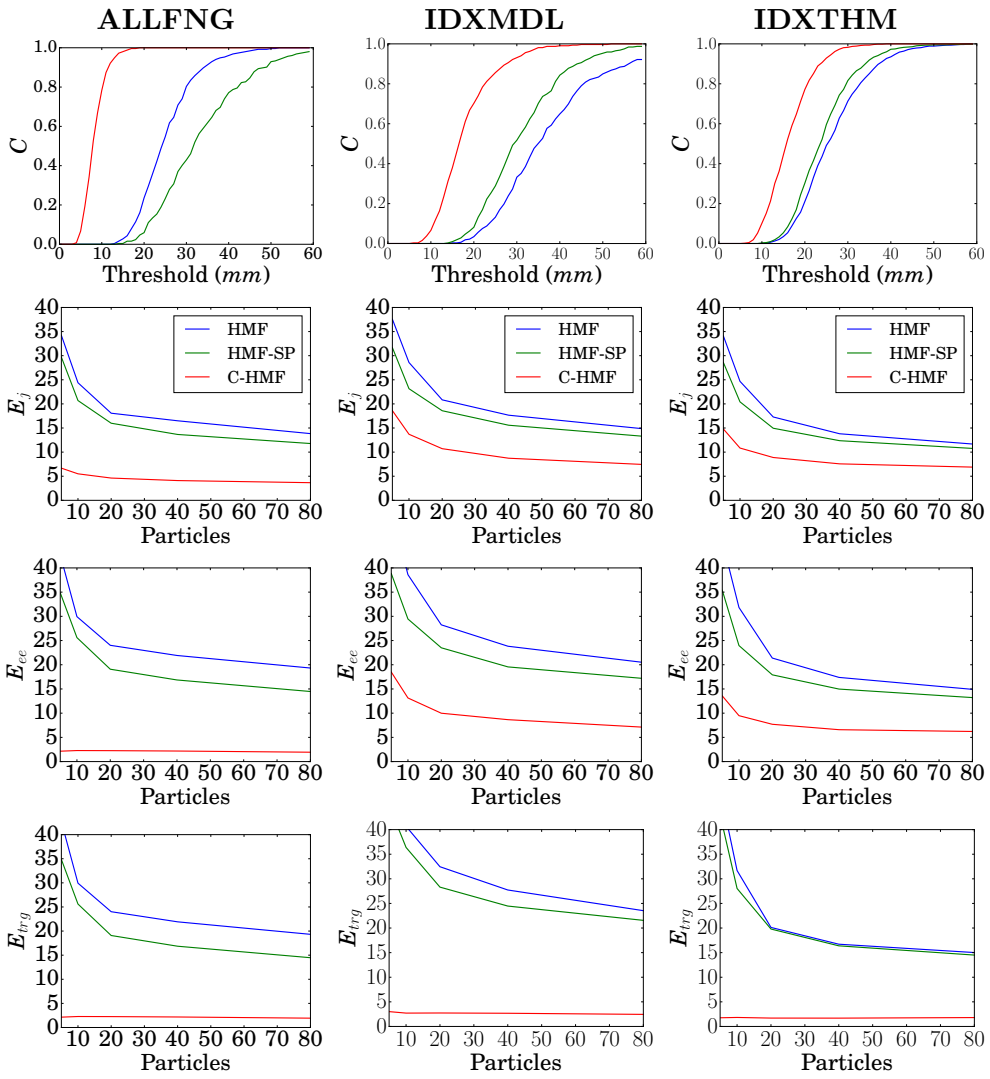


Figure 6.1: Error plots for the **C-HMF** (proposed, red) in comparison to **HMF** and **HMF-SP**. Figure rows correspond to the different error metrics: C, E_j, E_{ee}, E_{trg}. Columns correspond to different sequences, from left to right: **ALLFNG**, **IDXMDL**, **IDXTHM**

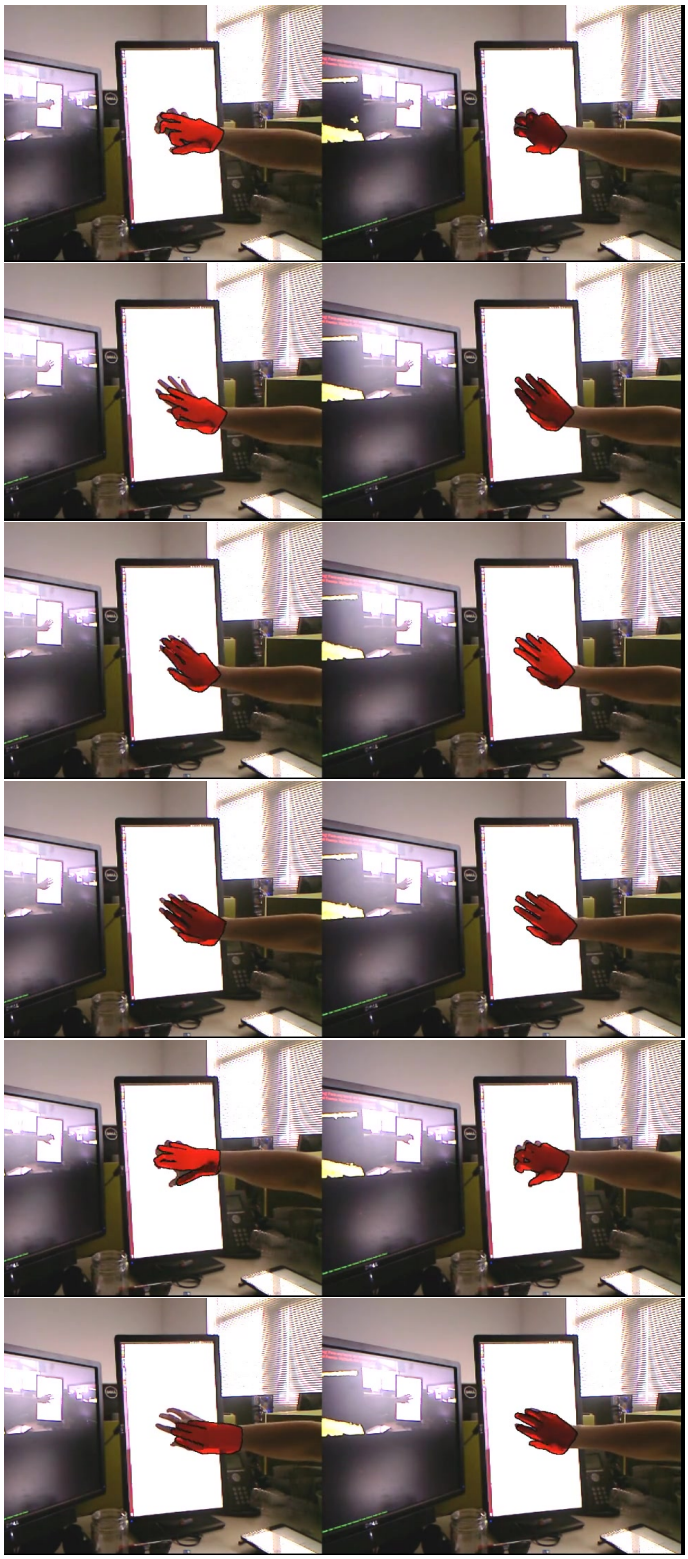


Figure 6.2: Qualitative results on the scenario with known fixed contact points, **ALLFING** sequence.
Left: **HMF** (baseline), Right: **C-HMF** (proposed).
Both methods use 160 particles

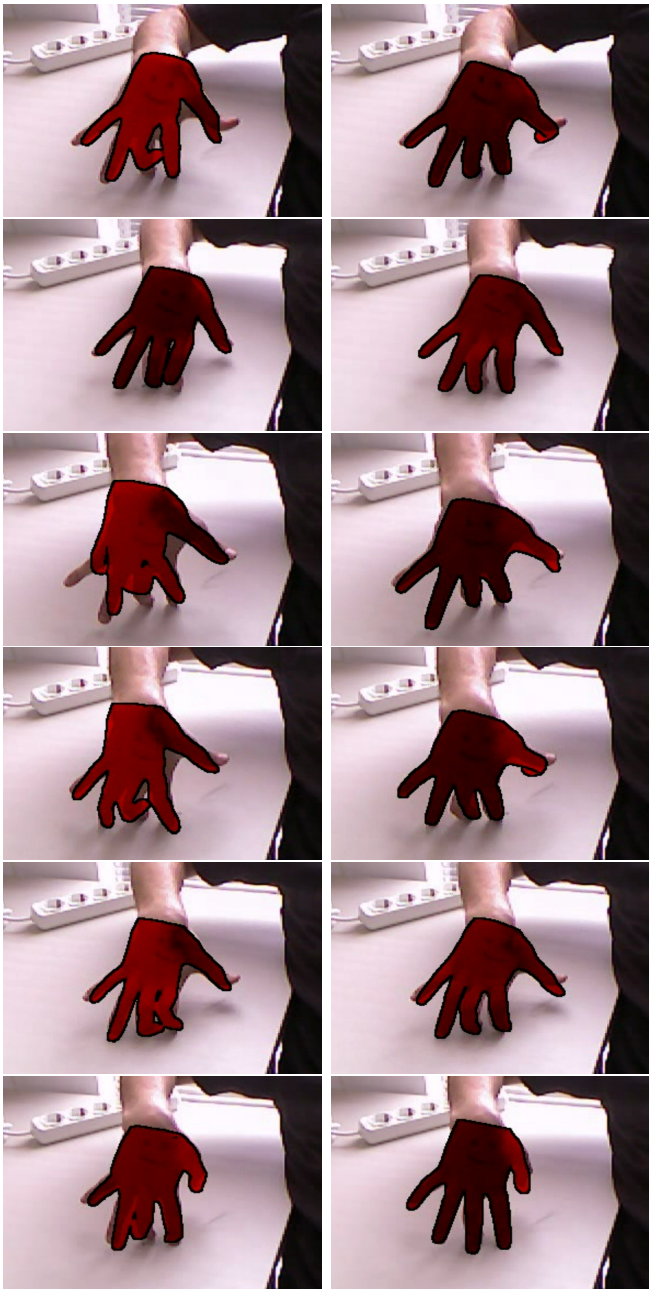


Figure 6.3: Qualitative results on the scenario with known fixed contact points, **IDXMDL** sequence.
Left: **HMF** (baseline), Right: **C-HMF** (proposed).
Both methods use 160 particles.

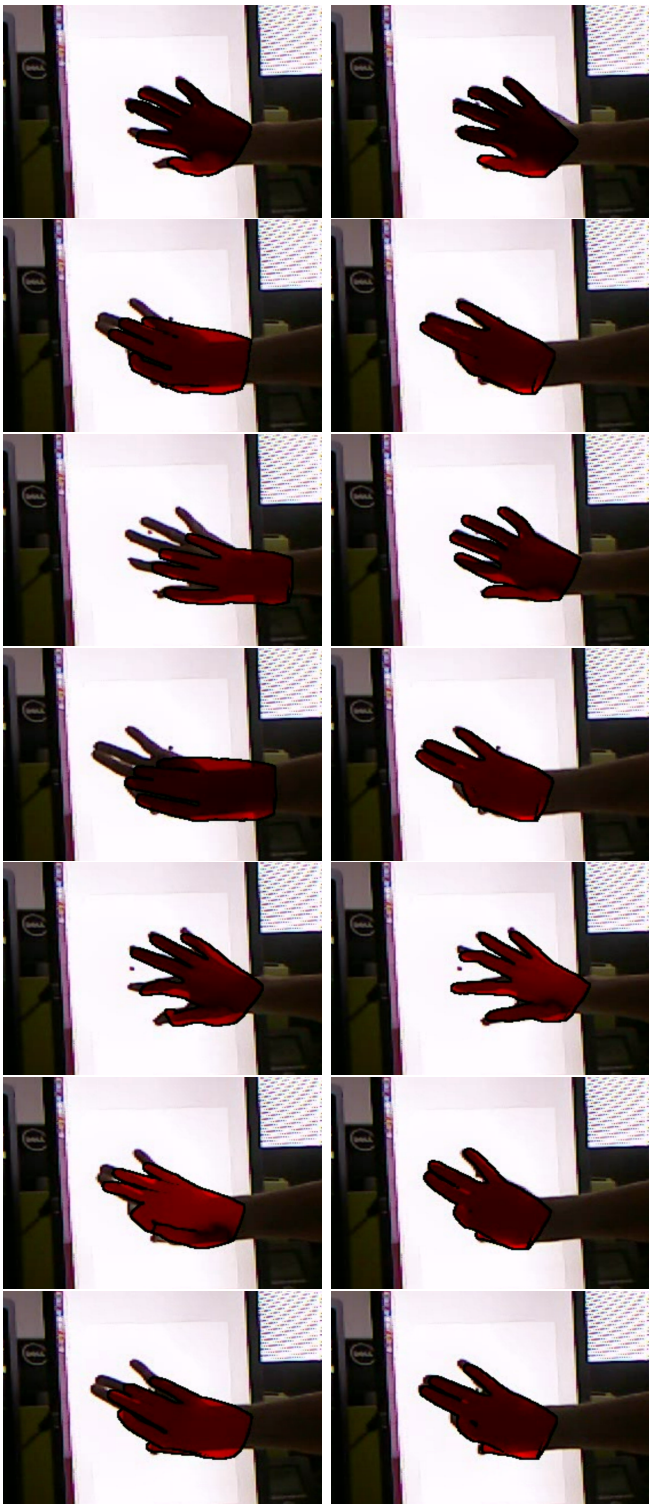


Figure 6.4: Qualitative results on the scenario with known fixed contact points, **IDXTMH** sequence.
Left: **HMF** (baseline), Right: **C-HMF** (proposed).
Both methods use 160 particles

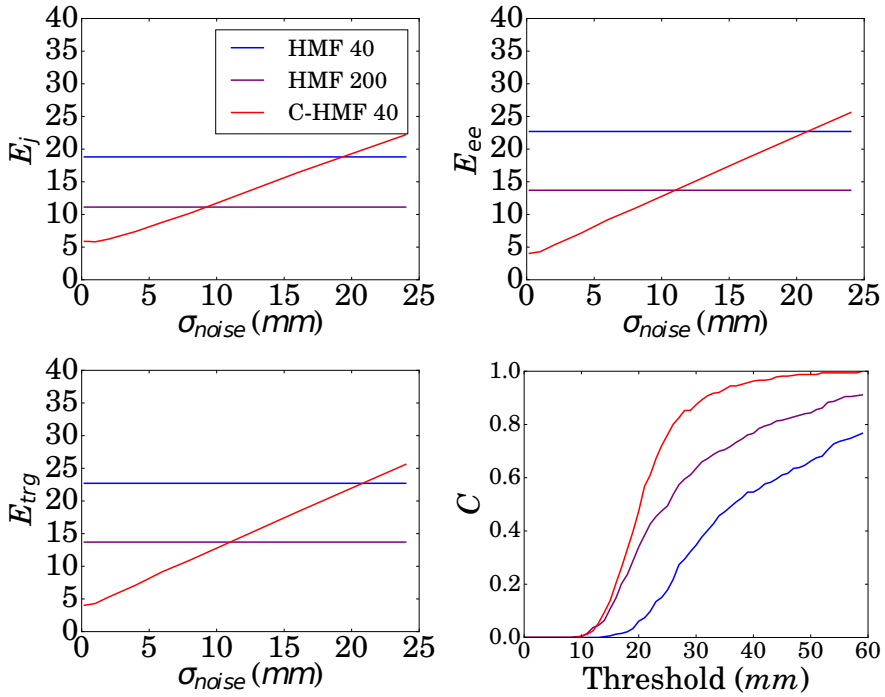


Figure 6.5: Error plots for the **C-HMF** (proposed, red) in comparison to **HMF** with 40 and 200 particles, for different levels of noise on the 3D position of the end effectors for the **FREEHM** dataset. Columns correspond to the different error metrics

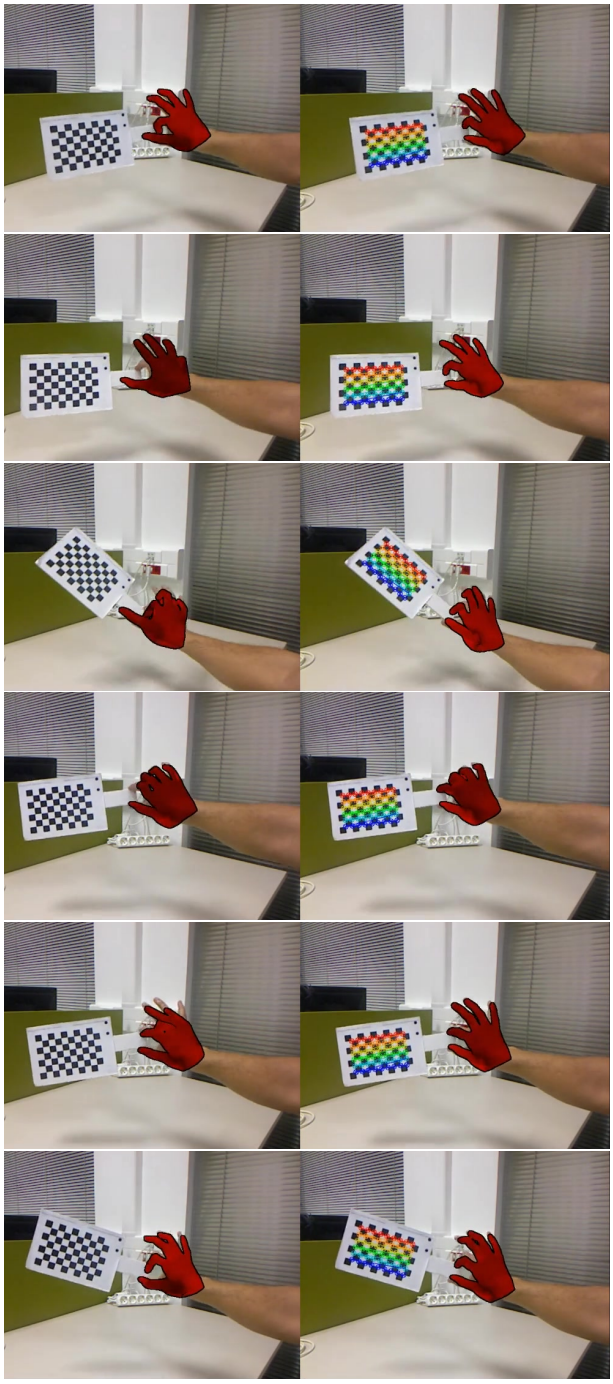


Figure 6.6: Qualitative results on the scenario with inferred contact points from a tracked object, **CALIB** sequence.
Left: **HMF** (baseline), Right: **C-HMF** (proposed).
Both methods use 160 particles

Chapter 7

Discussion

In this work, we proposed a novel 3D hand tracking method that explicitly considers constraints on the 3D locations of fingertips. Such constraints arise often, both in free hand motion and in hands interacting with other objects. Existing 3D hand tracking methods exploit such constraints in a soft manner, i.e., by considering them in the objective function they optimize. To the best of our knowledge, our approach is the first hypothesize-and-test method that samples and evaluates candidate hand poses that are guaranteed to satisfy the available constraints. This was achieved by exploiting the hierarchical structure of the hand model and state decomposition functionality of the HMF framework. This strategy, enabled us to develop a fast and simple constraints-aware sampling method. We used single view RGB-D input to tackle challenging sequences. These sequences included challenging viewpoints with severe self-occlusions. Extensive experiments on ground truth annotated data sets have shown that our proposed framework can significantly outperform the baseline approach that uses soft constraints or no constraints at all. Futhermore, our methodology can handle fingertip detection uncertainty. We proved the robustness of our method by investigating the tracking accuracy under various fingertip detection noise levels. The proposed constraints-aware sampling explores more densely the space of feasible solutions. As a result, increased hand tracking accuracy is achieved with a lower number of hypotheses evaluations. The time complexity of the sampling scheme is constant and does not depend on the location of the target with respect to the base. Our methodology is suitable to cope with the real-time performance requirements of the hand tracking problem.

7.1 Impact

The presented work resulted in the following publication:

- K. Roditakis, A. Makris and A.A. Argyros, *Generative 3D Hand Tracking with Spatially Constrained Pose Sampling*, In British Machine Vision Conference (BMVC 2017), BMVA, London, UK, September 2017 [28]

Contributions made to European projects:

- **WEARHAP** (FP7-ICT-2011-9)
- **Co4Robots** (H2020-731869).

7.2 Future work

Future research will focus on extending the type of employed constraints beyond end-effector constraints. This will include applying restrictions at intermediate joints or even more complex constraints, such as loop closure constraints. Another future direction would be to extend the methodology to apply to other articulated structures in particular human bodies. Furthermore, we will consider an evaluation of data-driven fingertip detectors with different characteristics to incorporate them into our framework. The proposed method has also the potential to replace the generative part of other existing hybrid methods and improve their accuracy. Another direction is an in-depth investigation of hand-object interaction scenarios. The current framework does not track the object jointly with the hand and does not use any type of segmentation or visibility model. Finally, the current implementation of this work has been developed in Python and was evaluated off-line on captured data. A natural subsequent step would be the production of an optimized c++ based code-base which will result in a system that operates at real-time frame rates.

Bibliography

[1] Andreas Aristidou and Joan Lasenby. FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011.

[2] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012.

[3] Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. Smart particle filtering for high-dimensional tracking. *CVIU*, 2007.

[4] Adrian A. Canutescu and Roland L. Dunbrack. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12(5):963–972, 2003.

[5] Teófilo Emídio de Campos and David W Murray. Regression-based hand pose estimation from multiple cameras. In *CVPR*, 2006.

[6] Ali Erol, George Bebis, Mircea Nicolescu, Richard D Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 2007.

[7] Ladislav Kavan and Jiří Žára. Spherical blend skinning: A real-time deformation of articulated models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, I3D '05, pages 9–16, New York, NY, USA, 2005. ACM.

[8] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*, 2012.

[9] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *CVPR14*, 2013.

[10] Nikolaos Kyriazis, Iason Oikonomidis, and Antonis A Argyros. A gpu-powered computational framework for efficient 3d model-based vision. Technical Report TR-420, 2011.

[11] Peiyi Li, Haibin Ling, Xi Li, and Chunyuan Liao. 3D Hand Pose Estimation Using Randomized Decision Forest with Segmentation Index Points. In *ICCV*, 2015.

[12] T. Lisini Baldi, S. Scheggi, L. Meli, M. Mohammadi, and D. Prattichizzo. Gesto: a glove for enhanced sensing and touching based on inertial and magnetic sensors for hand tracking and cutaneous force feedback. *IEEE Trans. on Human-Machine Systems*, 2017.

[13] Alexandros Makris and Antonis A Argyros. Model-based 3d hand tracking with on-line shape adaptation. In *BMVC*, 2015.

[14] Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis. A hierarchical feature fusion framework for adaptive visual tracking. *Image and Vision Computing*, 2011.

[15] Alexandros Makris, Nikolaos Kyriazis, and Antonis A. Argyros. Hierarchical particle filtering for 3D hand tracking. In *CVPRW*, 2015.

[16] Alexandros Makris, Nikolaos Kyriazis, and Antonis A Argyros. Hierarchical particle filtering for 3d hand tracking. In *CVPR*, 2015.

[17] Alexandros Makris, Clémentine Prieur, Théo Vischel, Guillaume Quantin, Thierry Lebel, and Rémy Roca. Stochastic tracking of mesoscale convective systems: evaluation in the West African Sahel. *Stochastic Environmental Research and Risk Assessment*, pages 1–11, jul 2015.

[18] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90 – 126, 2006. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour.

[19] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, October 2017.

[20] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. *CoRR*, abs/1605.03389, 2016.

[21] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015.

[22] I. Oikonomidis, M.I.A. Lourakis, and A.A. Argyros. Evolutionary Quasi-Random Search for Hand Articulations Tracking. In *CVPR*, 2014.

[23] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, 2011.

[24] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012.

[25] Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof, and Antonis A. Argyros. Hybrid One-Shot 3D Hand Pose Estimation by Exploiting Uncertainties. In *BMVC 2015*, 2015.

[26] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *CVPR*, 2014.

[27] Konstantinos Roditakis and Antonis A Argyros. Quantifying the effect of a colored glove in the 3d tracking of a human hand. In *International Conference on Computer Vision Systems (ICVS 2015)*, pages 404–414, Copenhagen, Denmark, July 2015. Springer.

[28] Konstantinos Roditakis, Alexandros Makris, and Antonis A Argyros. Generative 3d hand tracking with spatially constrained pose sampling. In *British Machine Vision Conference (BMVC 2017)*, London, UK, September 2017. BMVA.

[29] Grégory Rogez, James S Supancic, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *CVPR*, 2015.

[30] Grégory Rogez, James S Supancic, and Deva Ramanan. Understanding everyday hands in action from rgb-d images. In *ICCV*, 2015.

[31] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *ACM Human Factors in Computing Systems*, 2015.

[32] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, 2015.

[33] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, 2016.

[34] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*, 2013.

[35] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[36] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, 2015.

[37] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV*, 2015.

[38] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013.

[39] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM TOG*, 2016.

[40] Jonathan Taylor, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, Jamie Shotton, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, and Julien Valentin. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics*, 2016.

[41] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM TOG*, 2014.

[42] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC17*, 2017.

[43] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2015.

[44] Chengde Wan, Angela Yao, and Luc Van Gool. Direction matters: hand pose estimation from local surface normals. *arXiv:1604.02657 [cs]*, 2016.

[45] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *ACM TOG*, 2009.

[46] Xinyu Tang, Shawna Thomas, Phillip Coleman, Nancy M. Amato, Xinyu Tang, Shawna Thomas, Phillip Coleman, and Nancy M. Amato. Reachable Distance Space: Efficient Sampling-Based Planning for Spatially Constrained Systems. *The International Journal of Robotics Research*, 2010.

[47] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *ICCV*, 2013.

[48] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *The European Conference on Computer Vision (ECCV)*, 2016.

[49] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. *CoRR*, abs/1704.02612, 2017.