Tracking and Calibration for a Ball Catching Humanoid Robot

Oliver Birbach

Kumulative Dissertation
zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften – Dr.-Ing. –

Vorgelegt im Fachbereich 3 (Mathematik und Informatik)
Universität Bremen

22. November 2012

Datum des Promotionskolloquiums: 27. März 2013

Gutachter

Prof. Dr. Udo Frese (Universität Bremen)
Prof. Michael Beetz, Ph.D. (Universität Bremen)

Abstract

In recent decades, the only impact of robotics on real-world applications has been confined to the execution of predetermined, repetitive tasks in controlled industrial environments. Although recent advances in all fields of robotics research have led to the development of a first generation of highly actuated, multi-sensory equipped machines, they still fall short of the range of activities humans are capable of. With the goal of having robots operate autonomously in everyday domestic environments, it is certainly necessary that human-like dynamics can be performed to a certain degree. To foster research in this direction, it is therefore often proposed to engage robots in sporting benchmark activities as these dynamic tasks are demanding for the robot's mechanical, sensory and computational capabilities and also require a high quality of integration.

This dissertation is part of the work in making a humanoid robot perform such a dynamic task, namely enabling DLR's mobile humanoid robot *Rollin' Justin* to catch up to two simultaneously thrown balls, where each ball is caught with one of its hands. To be more specific, this thesis is concerned with the perception system. Despite being a clearly defined task with easily assessable performance even for non-specialists, it is still demanding and underlines the challenges for realizing dynamic tasks in general. The challenges are: Obtain the trajectory of the thrown balls with the necessary accuracy to move the arms to the right position at the right time; handle unmodeled shaking of the robot caused by the dynamic nature of the task; avoid computational latencies while processing sensor signals to ensure proper execution within the short duration of the ball flight.

From a perception point of view, this requires solving two separate problems. Firstly, for meaningful evaluation of the input data, the geometric relationships between all sensing and actuation components of the robot have to be determined through calibration. Secondly, detection, tracking, and prediction of the ball during flight have to be performed in an accurate manner while considering that the robot's cameras also move. Of course, this has to be performed in real-time.

Based on these requirements, this thesis contributes an automatic and self-contained method for calibrating all relevant sensors involved in the task. The highlights of the developed procedure are that it requires no external tools and no human assistance while achieving an accurate calibration. Furthermore, besides implementation of state-of-the-art approaches for tracking balls, a general tracking scheme is proposed that integrates detection and tracking in a fully probabilistic manner. Finally, besides contributions to the task of robotic catching, this thesis further covers the work of porting the obtained methods to a ball playing entertainment robot and additional calibration problems.

All presented methods and algorithms have been evaluated on the respective robots and were presented at trade fairs, public institute events and numerous lab demonstrations. Thus the methods have contributed to the development of sporting activities with humanoid robots and in doing so have extended the state of the art in service robotics.

Zusammenfassung

Der praktische Einfluss von Robotern beschränkte sich in den vergangenen Jahrzehnten weitestgehend auf das Ausführen von vordefinierten, sich wiederholenden Tätigkeiten in beaufsichtigten, industriellen Umgebungen. Trotz des rasanten Fortschritts in allen Teilgebieten der Robotik und der Entwicklung einer ersten Generation von vielseitig aktuierten, mit verschiedensten Sensoren ausgestatteten Maschinen sind diese noch weit entfernt davon, die Breite der menschlichen Aktivitäten abzubilden.

Um jedoch autonom in alltäglichen, z.B. häuslichen, Umgebungen zu agieren, ist es sehr wahrscheinlich notwendig, dass die Roboter menschenähnliche Eigenschaften hinsichtlich der Bewegungsdynamik aufweisen. Ein häufiger Vorschlag, um Forschung in diese Richtung zu lenken, ist es, Roboter in sportliche Aktivitäten einzubeziehen, da diese besondere Anforderungen an die mechanischen, sensorischen und rechnerischen Fähigkeiten stellen sowie eine hohe Qualität der Integration erfordern.

Die vorgelegte Dissertation ist Teil der Bemühungen mit einem humanoiden Roboter solch eine dynamische Aufgabe zu realisieren, nämlich mit dem mobilen humanoiden Roboter *Rollin' Justin* des DLR bis zu zwei gleichzeitig geworfene Bälle zu fangen, jeden jeweils mit einer Hand. Dabei konzentriert sich diese Arbeit auf die Wahrnehmungskomponente. Diese klar festgelegte Aufgabe erlaubt es nicht nur Laien die erbrachte Leistung auf einfache Weise zu beurteilen, sie ist zugleich anspruchsvoll und unterstreicht in unmittelbarer Weise die Herausforderungen für dynamische Aufgaben: Berechnung der Flugbahn der geworfenen Bälle mit der notwendigen Genauigkeit, um die Bälle an der richtigen Position zum richtigen Zeitpunkt zu fangen; Berücksichtigung von nicht modellierten Schwingungen des Roboters aufgrund der Dynamik; Vermeidung von rechnerisch verursachten Verzögerungen während der Verarbeitung der Sensordaten, um eine einwandfreie Durchführung während der kurzen Flugdauer zu garantieren.

Für die Wahrnehmung des Ball fangenden Roboters bedeutet dies die Entwicklung von Lösungen zu zwei Problemen. Zum einen müssen für eine aussagekräftige Auswertung der Eingangsdaten die geometrischen Zusammenhänge aller Sensor- und Aktuatorkomponenten des Roboters durch eine Kalibrierung bestimmt werden. Zum anderen ist es notwendig, die Bälle während des Fluges mit der nötigen Genauigkeit unter Berücksichtigung der Bewegung des Roboters zu erkennen, zu verfolgen und vorherzusagen. Dabei muss letzteres in Echtzeit geschehen.

Angelehnt an diese Anforderungen legt diese Arbeit ein automatisches und in sich abgeschlossenes Verfahren vor, um alle benötigten Sensoren und Aktuatoren zu kalibrieren. Bei dieser Prozedur ist besonders hervorzuheben, dass keine zusätzlichen Hilfsmittel und keine menschliche Hilfestellung für eine sorgfältige Kalibrierung benötigt werden. Des Weiteren wird das Verfolgen von Bällen mit aktuellen Verfahren realisiert und ein neues Verfahren vorgestellt, welches Erkennen und Verfolgen in einer probabilistischen Art und Weise integriert. Neben den Beiträgen zum Ballfangen behandelt diese Arbeit auch den Transfer der Verfahren auf einen ballspielenden Unterhaltungsroboter sowie die Behandlung weiterer Kalibrierungsprobleme.

Die vorgestellten Verfahren und Algorithmen wurden mit den verwendeten Robotern experimentell evaluiert und auf Messen, öffentlichen Veranstaltungen sowie Labordemonstrationen der Öffentlichkeit vorgeführt. Die Verfahren haben zur Realisierung einer sportlichen Aktivität mit einem humanoiden Roboter beigetragen und auf diese Weise den Stand der Technik für Serviceroboter im Allgemeinen erweitert.

Contents

Chapter 1

Introduction

For centuries, humans have been fascinated by the idea of creating machines similar to existing life. Although robots have replaced the human worker at repetitive tasks in industrial scenarios for many years, they do not have yet reached the capability and autonomy to accomplish everyday human tasks. This becomes even clearer when considering what variety of dynamic activities humans are able to perform, e. g. when doing sports.

In this thesis, parts of the work in making a humanoid robot perform such a dynamic activity are studied. In detail, DLR's mobile humanoid robot *Rollin' Justin* is instructed to catch a thrown ball with one of his arms and is even instructed to catch two balls at the same time, one with each of his arms. The latter is even difficult for humans and underlines the main challenges which will be considered in this thesis: keeping track of multiple balls robustly and accurately (object tracking), distinguishing robot motion and motion of the thrown balls (tracking of egomotion), and to be aware of the perception and actuation setup (robot calibration).

1.1 Dynamic Tasks as a Challenge for Robotics

The development of autonomous systems that share the same physical space as humans is an active field in robotics research. As an alternative to constructing special machines, which are only able to fulfill one dedicated task, the development of a robot, which is able to handle any given task like humans, is desired. Currently, humanoid robots have become the form of choice and have been successfully deployed for basic operation in domestic settings. Despite this remarkable progress, the developed systems still do not match the broad range of activities humans are capable of. As these robots are likely to use human driven machinary and tools, and engage in physical interaction with humans, it is advisable to also have, at least up to a certain degree, human-like dynamics. For this to be achieved, several open challenges in robotics still exist such as the development of powerful mechatronic hardware allowing dynamical motion or the ability to process relevant data at high rates for reactive behavior in complex scenarios, just to name two.

To address these challenges, contests emerged which embed humanoid robots in competitive environments. Although these contests usually require the accomplishment of a series of specialized tasks in standardized testbeds, they, by their design, are expected to benefit research broadly. A popular example of such a contest is the *RoboCup* initiative which was founded with the idea in mind "to pursue and analyze technical issues involved in a humanoid to play soccer" [KITANO and ASADA, 1998] and its extension

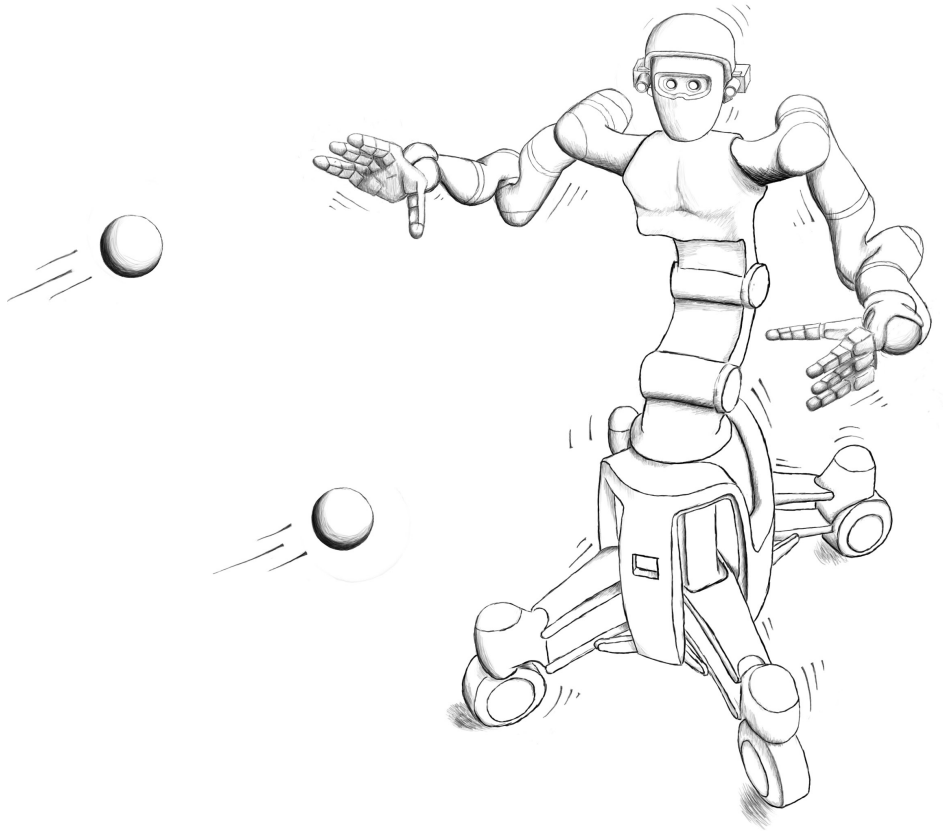


Figure 1.1: Sketch of DLR's *Rollin' Justin* catching two balls which were thrown towards the robot, each ball with one of his hands. This work focuses on tracking multiple balls, distinguishing own motion and motion of the balls, and calibrating the robot's setup.

RoboCup@Home [VAN DER ZANT and WISSPEINTNER, 2006] with the goal of promoting domestic service robots. Others include, although for driverless vehicles, the series of DARPA Grand Challenges [BUEHLER et al., 2007, BUEHLER et al., 2010] and the upcoming DARPA Robotics Challenge where humanoid robots will operate in disaster scenarios.

1.2 Ball Catching as a Robotic Testbed

To help robots establish themselves in dynamic activities, a similar specialized task is pursued in this thesis, namely the catching of thrown balls. In detail, a mobile humanoid robot is instructed to catch up to two thrown balls with his arms. See Fig. 1.1 for an illustration of this dynamic task using DLR's *Rollin' Justin*.

Being able to reliably catch a ball is not an easy task, neither for humans nor for robots. While humans spend a reasonable amount of time in their early life to master catching, realizing this dynamic and skillful manipulation task for a robot requires careful implementation of perception, control and planning algorithms. While for most robotic tasks implementing one of these is already challenging in itself, the required integration of all three makes this task particularly demanding. Hence, robotic ball catching qualifies to be an appropriate testbed for the evaluation of different key robotic technologies.

In addition, the value of realizing a ball catching robot is not limited to roboticists. In contrast to other research topics, *every* human has experience in ball catching. Therefore, everyone is also able to assess the performance of the robot and compare it to human capabilities. This gives even non-specialists the ability to assess the current state of the art of robotics and allows them to see for themselves how much work is still left for robotics research to match human capabilities.

It should not be left unmentioned that ball catching is currently one of the few opportunities for humans to interact with a robot. In general, human robot interaction, especially physical interaction, is still in its early stages of development which is mostly caused by the lack of safety mechanisms. The proposed ball catching activity therefore provides a safe and entertaining way to directly engage humans with a robot.

Engaging robots in such kinds of activities has a long history. In 1989, Andersson [ANDERSSON, 1989] introduced a ping-pong playing robot based on stereo vision and a robot arm. Although limited by the vision system's predictive capabilities and the robot's dynamics, it was fully functional on a small table and is considered to be one of the first robots in such a dynamic activity. Robotic ball catching was first successfully studied by Hove and Slotine for a two degree of freedom (DOF) robot arm [HOVE and SLOTINE, 1991] and later for a four DOF arm [HONG and SLOTINE, 1995] using actively controlled stereo vision. While these approaches made extensive use of special hardware for vision, the seminal work of Frese et al. [FRESE et al., 2001] introduced a system for using common computing hardware for stereo vision processing and catching with a seven DOF lightweight robot arm. The complementary work on optimal trajectory optimization was studied by Bäuml et al. [BÄUML et al., 2010] for an arm of this kind. Ball catching was realized in a non-prehensile way by using a balancing controller to keep the ball on a plate instead of utilizing a gripper or catching tool such as a basket [BÄTZ et al., 2010].

So far, ball catching using a humanoid robot has been implemented twice. The humanoid robot *Saika* [NICHIWAKI et al., 1997, KONNO et al., 1997] equipped with twelve DOF was able to catch a ball thrown from around 2 m away with a basket attached to one of its arms. A stereo vision system mounted in the robot's head detected the colored ball and the robot was rigidly mounted to the ground. Riley and Atkeson [RILEY and ATKESON, 2002] presented ball catching experiments using a 30 DOF humanoid robot equipped with a baseball glove. Here, an external stereo vision system with a rather wide baseline was looking for color-coded balls. The work presented here differs from these two as it makes use of both available humanoid arms, employs robotic hands instead of catching tools, handles the moving cameras while the robot moves, allows arbitrary colored balls to be used, and enables vision processing on an embedded system in the robot instead of relying on unlimited external computing resources.

Furthermore, the sole task of detecting and tracking the ball has been established for broadcast or umpire purposes. In 1997, the *FoxTrax* hockey puck tracking system was presented by Cavallaro [CAVALLARO, 1997]. An infrared emitting puck is tracked using multiple cameras allowing the viewers to easily follow the fast moving puck in an augmented TV image. Similarly, Guéziec [GUÉZIEC, 2002] introduced a vision-based baseball tracking system. It gives the audience an indication whether a pitched ball qualified as regular thrown ball or not. In recent years, the *Hawk-Eye* system [OWENS et al., 2003] for tracking the ball in cricket or tennis games has become popular. It is used in cricket's adjudication process by helping the umpire resolve difficult decisions or by tennis players to

challenge perceived erroneous calls by line judges. Usually, a computer generated replay of the scene is given, showing the tracked trajectory of the ball and the resulting computer generated decision. For the sport of soccer, Beetz et al. [BEETZ et al., 2006, BEETZ et al., 2007] developed a camera-based observation system, called *ASPOGAMO*, that tracks the ball and the players and is used for the analysis of matches and multi-agent activity. Finally, the author of this thesis conducted prior work [9]. Here, a human observed flying balls on a soccer field while wearing a helmet equipped with a camera and an inertial measurement unit with the goal of accurately predicting the ball's trajectory.

1.3 Challenges

Based on this setup, a set of challenges has been identified in [2]. These need to be addressed properly in order to successfully perform the catching of balls. These challenges are:

1. **Low Latency:** The flight time of a thrown ball is around 1 s from a distance of around 5 m. Combined with the limited dynamic performance of the robot, this leaves no room for complicated methods but rather demands low latency implementations. Only these enable a reactive catching behavior that covers most of the robot's workspace. This is not only true for the perception module but also holds for the ensuing planning stage.

2. **High Precision in Space and Time:** For successful catching, the robot configuration must match the catch point along the trajectory at the intended time of catch precisely. This is especially true for the hand closing command. Any deviation from the actual time of catch would lead to the ball hitting the outside of the hand or a ball bouncing out of the hand.

3. **Moving Camera System:** The vision system's challenge is to track the ball and precisely predict its upcoming trajectory. This has to happen in two modes. First, when the robot is inactive the cameras are static. Second, once a trajectory is projected to be catchable, the robot follows the ball due to controlled head movement such that the object stays in the field of view as long as possible to obtain the precise measurements near the robot.

4. **Not Completely Cancelable Vibrations:** Unfortunately, unwanted vibrations propagating through the whole structure are excited during robot movement, as a result of elasticities in the lightweight design of DLR's *Rollin' Justin*. This affects the head-mounted cameras and negatively impacts the perception's system performance if not appropriately considered. Furthermore, because these vibrations are only partially cancelable by control algorithms, the precision with which the hand can be positioned is impaired as the planning stage cannot anticipate these in time.

5. **Partly Observable Kinematic State:** In addition to these observable vibrations, the robot also encounters partly observable states. Appearing at the torso and the mobile platform, these have mechanical origins and show up as dynamic movements. Special care must be taken in handling these phenomena.

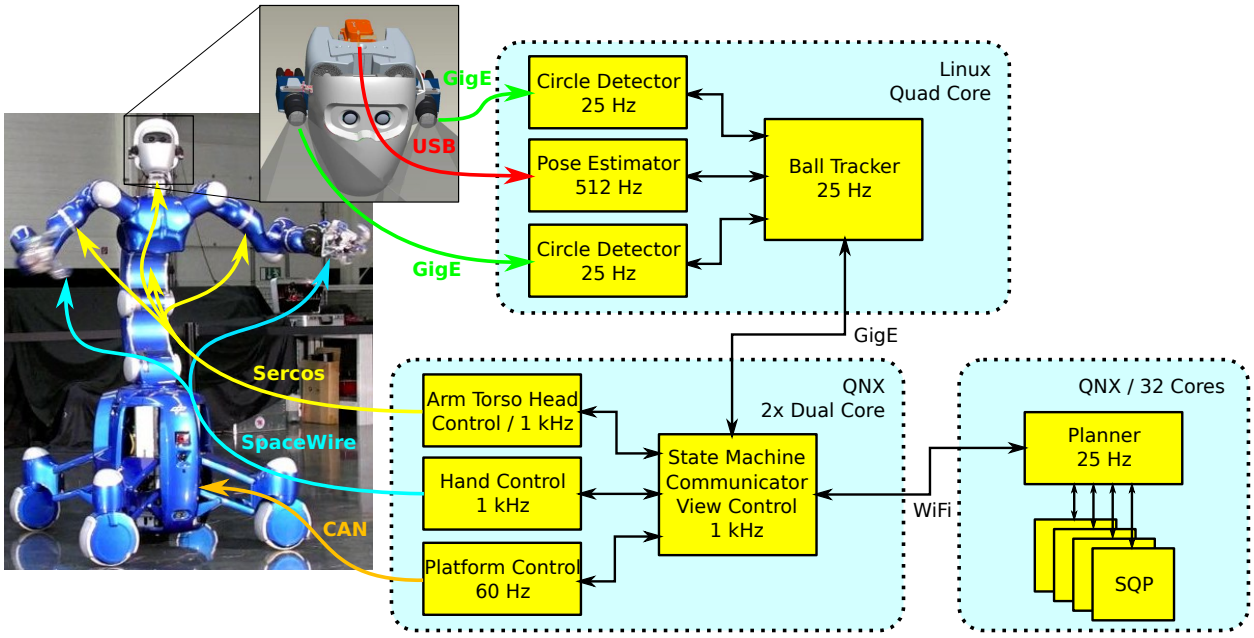


Figure 1.2: Architecture of the ball catching system. The robot's sensors and actuators are coupled (shown as colored arrows) to the robot's embedded computing hardware (two left light blue boxes). The involved components and their interactions are depicted as yellow boxes and arrows, respectively. Every component's processing cycle rate is given where the ones receiving sensor data correspond to the readout rate and the ones passing data to actuators correspond to the control cycle rate. The rightmost box is a remote computing resource comprising the planning stage.

6. **Limited Computing Resources and Communication Bandwidth:** As illustrated in Fig. 1.2, attention is required what data to compute at which stage and what kind of information needs to be exchanged between components. To operate completely wirelessly, it is required that all computations that demand high bandwidth or low latency have to be performed on the robot's embedded hardware.

7. **No Globally Synchronized Clocks and Communication Latencies:** The system consists of various sensors and actuators running at their respective readout rate. These are connected by various bus architectures to the system's different computing entities, each of which has its own clock. For proper integration of measurement data, a consistent time synchronization is required to correctly assign all physical sensor and actuator events with a high precision time stamp that is valid system-wide.

The latter two challenges address system integration requirements. As this thesis focuses on perception, the focus will be on the challenges one to five throughout this work.

1.4 Used Robotic Platforms

The main robotic platform used for task of ball catching is DLR's mobile humanoid robot *Rollin' Justin*. It consists of an omnidirectional platform based on four individually moveable wheels [BORST et al., 2009]. The upper body consists of a torso with four degrees

Figure 1.3: In-game snapshot of a ball playing entertainment robot. The robot perceives the ball using a pair of cameras hidden in in the eyes and is able to return balls thrown towards the robot using a bat.

of freedom (DOF), of which one is passive, and two LWR-III arms [HIRZINGER et al., 2002] with seven DOF each. Each arm is equipped with a DLR-Hand-II [BUTTERFASS et al., 2001] which has twelve DOF distributed over four fingers. Completing the robot's anthropomorphic design, the head is controlled by a pan-tilt unit.

The head is equipped with a pair of Prosilica GC1600 GigE cameras at a rather short baseline of 20 cm. These cameras provide a synchronized pair of 1616×1220 px sized images at a frame rate of 25 Hz, each exposed for a period of 1.5 ms to reduce motion blur. Each camera is equipped with an 8 mm lens (Schneider Kreuznach CNG 1.4/8) allowing a field of view of 47° horizontally and 36° vertically, which ensures that the angular extent of the scene is sufficient for a variety of ball trajectories. The robot is further equipped with a head-mounted XSens MTi inertial measurement unit (IMU) which provides linear acceleration and angular velocity information at a rate of 512 Hz.

The complete system including the robot and all involved computing resources is depicted in Fig. 1.2. All sensor readings arrive at a quad-core embedded system running Linux. Processing is performed at the point of arrival and includes extraction of circle features from camera images, pose filtering from IMU data, and, based on this data, tracking of the ball trajectory. Each time new trajectory information is obtained, this data is passed to a pair of embedded dual-core systems running QNX. These machines share the task of keeping track of the system's catching state, provide communication channels between components through the *agile Robot Development* (aRD) concept [BÄUML and HIRZINGER, 2008], provide tools for visual inspection of catching experiments and contain the control components. From there, data is passed wirelessly to a remote 32-core cluster, which performs elaborate planning of the catching configuration based on the current robot state through parallelized sequential quadratic programming (SQP) optimization. Once a solution has been found, it is passed back, the robot's catching state is adjusted and the

obtained configuration is commanded to the actuators through their respective transfer buses.

Beyond ball catching, the developed methods have been transferred to a minimal ball playing entertainment robot which is shown in Fig. 1.3. Here, a bat with an approximate length of 0.85 m is used to return balls thrown towards the robot. Although a pair of synchronized cameras is used for tracking, no IMU is required due to the immobility of the robot. The bat is actuated by only two DOF. This setup allows motion to be computed in an *ad hoc* fashion based on the obtained ball trajectories.

1.5 Contributions

The realization of a perception system for a ball catching humanoid robot is central to this dissertation. Therefore, the main contributions are the following:

- A novel tracking algorithm called fully probabilistic multiple target tracking (FPMTT) is proposed that integrates tracking and response-based detection for continuous trajectory estimation. Being formulated as a batch-mode Bayesian approach, it performs global estimation of the trajectory given all collected image evidence so far. For handling of multiple targets, track limit determination and a method for assuring mutual exclusion are presented. Experimental evaluation is performed with respect to ground truth and results are compared to other tracking approaches. While the realized task is ball tracking, the algorithm itself is generic and is the first approach integrating response-based object localization from images and tracking based on a physical motion model for trajectory estimation. This contribution deals with challenges one and two.

- A calibration approach for determining all relevant parameters of the setup introduced in Sec. 1.4 is presented. The novelty of this approach is that the robot collects all necessary sensor data by itself through automated motion and evaluates it in an automated manner. No external calibration tools are employed and therefore no human assistance is required. A co-developed least squares estimation framework is introduced that allows the direct setup and efficient solving of this class of problem. This contribution addresses the second challenge and, to the knowledge of the author, is the first procedure to calibrate a robot in such an automated and self-contained manner.

- The extension of IMU-based head pose estimation to not only complement the cameras, but also to obtain the unobservable torso and platform motion is discussed. The concept of a self-contained catching device is introduced, which corresponds to an isolated head-arm system which is somehow moved by the robot. This contribution deals with challenges three to five.

- The realization of ball tracking using classical multiple hypothesis tracking (MHT) in combination with the unscented Kalman filter (UKF) and a single target tracking model. In addition, a Gaussian mixture probability hypothesis density filter (GM-PHD) is implemented in combination with a prior initialization routine for solving

a specific problem of initializing uncertain states from measurements. Both algorithms are compared to each other and to ground truth. This work is the first to employ the PHD filtering scheme in robotics.

Further contributions include derivative work of the above, or constitute contributions outside of ball catching:

- Tracking and calibration approaches have been realized for the ball playing entertainment robot. While the MHT/GM-PHD has been ported directly, the calibration approach is greatly simplified to account for the entertainment robot's reduced complexity.
- Initial work and results of optimal experimental design of the automatic and self-contained calibration approach is presented.
- Two additional calibration procedures are introduced concerned with the problem of determining a camera's pose relative to a RoboCup soccer field and recovering the parameters of a Microsoft Kinect sensor.

1.6 Outline

In this *thesis by publication*, the following chapters summarize the previously published work in a concise manner. This summary consists of the following chapters:

Chapter 2 discusses multiple-target tracking methods and their application to robotic ball catching. The underlying single target model for tracking a ball is introduced including the approach of estimating the head pose through the employed IMU. The presented tracking methods include multiple hypothesis tracking (MHT), probability hypothesis density (PHD) filtering, and the proposed approach called fully probabilistic multiple target tracking (FPMTT).

Chapter 3 addresses the problem of calibrating multiple sensors mounted on a humanoid robot. In detail, a textbook-style and an automatic self-contained approach will be employed for calibrating the setup illustrated in Fig. 1.2. Furthermore, additional calibration work outside of ball catching will be presented briefly.

Chapter 4 concludes this dissertation by summarizing the presented work and giving an outlook on potential future work based on the insights from this thesis.

Chapters 2 and 3 follow the same structure. Each of these chapters starts with a general introduction followed by a chapter-related motivation. An elaborate review of related work is conducted before a summary of the compiled publications is presented. Finally, the chapters close with a summary reiterating the contributions and giving an outlook on the transferability of the proposed approaches.

After the summary, a list of the publications of the author and the references follows. The appendix is composed of a list of released software before the compiled publications are reprinted.

Chapter 2

Multiple Target Tracking

Multiple target tracking (MTT) has the goal of estimating individual states of an unknown number of possibly moving objects from sensor input. Depending on the sensing setup and observation features, observations might be noisy, missing, or be false-alarms. Furthermore, handling of creation and deletion of targets is desired. This chapter presents work on multiple target tracking as part of a computer vision application for the task of robotic ball catching. After a discussion of the state of the art tracking methods, it is shown how a classic MTT approach, namely multiple hypothesis tracking (MHT), and a newly emerged technique known as probability hypothesis density (PHD) filtering have been employed for the above mentioned task. In the final section, a novel tracking approach is presented where detection and tracking are not considered as separate stages but are combined in a single optimization stage.

2.1 Motivation

Admittedly, tracking multiple balls using stereo cameras does not pose an inherently difficult multiple target tracking problem. Due to the known dynamics (and thus good predictability) of the ball trajectory, association of measurements to the ball, even when considering clutter, is rarely ambiguous.

As can be directly derived from the challenges listed in Sec. 1.3, the difficulty for the algorithm focuses on accuracy of the estimated trajectory (Challenge 2), keeping computational latency low (Challenge 1), and handling of the unavoidable vibration of the robot (Challenges 3 through 5).

One might argue that using separate single-target trackers initialized by a heuristic track starting mechanism would solve the problem efficiently. While such an *ad hoc* solution might be feasible, it was discarded and instead full-fledged multiple target solutions were sought as these thoroughly cover all aspects of common tracking problems in a methodically sound way.

2.2 Related Work

Multiple target tracking has found widespread use in multiple disciplines which has resulted in a vast amount of published work. To keep this section concise, only methods that are closely related to approaches addressed in this dissertation will be listed. After

a review of classical recursive Bayesian filtering and its siblings, so-called batch-mode (Bayesian) tracking approaches, where some kind of global tracking approach is established, will be considered. Finally, work addressing tracking of flying balls from camera images is presented which includes approaches employed in other robotic ball catching setups.

2.2.1 Recursive Bayesian Filtering

The concept of Bayesian filtering has become the standard procedure when confronted with the problem of estimating the state of an unobservable system over time through measurements. The goal of the filtering process is to recursively update the system's state estimate $\hat{x}_t$ at time t through (usually noisy) measurements $z_1 \dots z_t$, starting from an initial state x_0. Unfortunately, the ideal Bayes filter is computationally intractable as computation of the update requires costly evaluation of integrals depending on the class of the chosen distributions.

To tackle this problem, approaches approximating the underlying distribution have been established for quite some time. Instead of analytical modeling, the use of a weighted set of samples that approximates the posterior distribution in combination with importance sampling gave rise to the particle filter (PF), initially proposed in [GORDON et al., 1993] as the bootstrap filter. The number of samples in this approach serves as a trade off between computational cost and accuracy of estimation making it suitable for wide range of applications, e. g. in computer vision [ISARD and BLAKE, 1998] or mobile robot localization [FOX et al., 1999]. See also Arulampalam et al. [ARULAMPALAM et al., 2002] and Cappé et al. [CAPPÉ et al., 2007] for introductory texts.

Another approximation is the Kalman filter [KALMAN, 1960], see also Ho's and Lee's Bayesian formulation [HO and LEE, 1964], which is a closed form solution for one special case of the underlying model: the dynamic and the measurement model are now assumed to be linear transformations with additive independent zero-mean Gaussian noise. Based on these assumptions, Kalman recursion generates an updated Gaussian posterior at time t from a Gaussian posterior of the previous step $t - 1$. While this is only valid for linear Gaussian systems, nonlinearity can be handled through further approximation. The extended Kalman filter (EKF) [ANDERSON and MOORE, 1979] is an approximation acquired through (usually first order) Taylor series expansion. The unscented Kalman filter [JULIER and UHLMANN, 2004, WAN and VAN DER MERWE, 2000] generates a set of so-called sigma points from mean and covariance of the state, which are recombined to a Gaussian after propagation through the nonlinear functions. The Kalman filter is a recursive implementation of Gauss' method of least squares and their relationship was reviewed by Sorenson [SORENSON, 1970].

While these two major approaches allow estimation of the state from noisy measurements, special care has to be taken for tracking a target in the presence of false alarms and non-existent measurements. The key problem here is to decide which measurements should be integrated with a target's state, a task known as data association. In the case of considering a varying number of objects, a sophisticated approach known as multiple hypothesis tracking (MHT) [REID, 1979] became popular in multiple target tracking. As the name suggests, the tracker maintains a set of hypotheses, where each hypothe-

sis represents a unique mapping of measurements to targets or false alarms. When new measurements arrive at each time step, these hypotheses are systematically expanded by updating its states in a Kalman filtering framework. As this leads to exponential growth, elaborate strategies have to be employed to limit the number of hypotheses and keep the tracker computationally feasible, such as the approach of Cox and Hingorani employing Munkre's algorithm [COX and HINGORANI, 1996], or the MCMC data association approach in [OH et al., 2009]. Besides its use in the tracking community, applications of MHT in robotics include tracking of people using a laser range finder by Luber et al. [LUBER et al., 2011b, LUBER et al., 2011a]. See also [BLACKMAN, 2004] for an introductory text to MHT and confer Sec. 2.5 for a more detailed explanation of MHT as it is applied to the case of ball tracking.

Also, particle filters capable of tracking multiple targets have been developed, where two classes of approaching such an algorithm have to be distinguished: separate or joint representation of the target configurations in the posterior distribution. One of the first approaches was realized by Cai et al. [CAI et al., 2006] where particle sets for each target are generated and data association to measurements is achieved through finding the most likely nearest neighbor assignment. Such a separate approach is prone to errors when object interaction occurs, which can be handled by keeping the target configuration in one joint particle set such as presented in [KHAN et al., 2005]. For successful sampling in a possibly high-dimensional state space, traditional sampling turned out to be inefficient and a MCMC sampling step was introduced. Von Hoyningen-Huene and Beetz [VON HOYNINGEN-HUENE and BEETZ, 2009] provide a Rao Blackwellized resampling particle filter where the posterior is approximated as a Gaussian and hence use Kalman filtering in the prediction step. Data association is achieved through smart resampling while robustness is increased through a fixed-lag target estimation scheme.

In recent years, a conceptually different approach to Bayesian filtering of multiple targets emerged. Instead of explicitly modeling associations between measurements and objects through hypotheses, states and measurements are modeled as random finite sets (RFS) which then allow formulation of a Bayesian approach directly. Again, computational infeasibility of the involved integrals make it unsuitable for most applications, but a first moment approximation known as the PHD filter was proposed by Mahler [MAHLER, 2003, MAHLER, 2007b], which recursively updates a posterior multiple target intensity over time. This intensity behaves like a distribution in state space as its peaks indicate the objects of interest, but unlike a distribution, the integral is not one but the number of expected objects. As these filters make use of particle filtering and Kalman filtering as the underlying single-target filter, implementations of the PHD filter exist and represent the intensity as particles known as the SMC-PHD filter [MAHLER, 2007b], or as a mixture of Gaussians known as the GM-PHD filter [VO and MA, 2005, VO and MA, 2006]. While these filters still enumerate associations of measurements to states, they do *not* enumerate the different possibilities to select a subset of states as a hypothesis such as MHT does and are therefore regarded as computationally attractive. So far, the use of this concept outside the tracking community is rather limited and the filtering scheme has been mostly employed in computer vision applications for tracking feature points [IKOMA et al., 2004], people [WANG et al., 2006, WANG et al., 2007], faces, people and vehicles [MAGGIO et al., 2007] or for tracking objects in aerial images [POLLARD et al., 2009]. For a derivation of the PHD filter using infinitesimal sized bins instead of random finite sets, see Erdinc's

and Bar-Shalom's article on the bin-occupancy filter [ERDINC et al., 2009]. Please confer Sec. 2.6 for detailed treatment of this filter applied to ball tracking.

2.2.2 Batch-Mode (Bayesian) Tracking

As the success of these recursive approaches depends on the detector's ability to handle difficult detection situations, recent efforts in tracking have focused on global tracking schemes, mostly employing a batch-mode Bayesian approach. Originating from the computer vision community, these approaches try to resolve ambiguities, such as temporary occlusions, at a global level, i. e. by looking at measurements from multiple time steps at once. In [YAN et al., 2006] single target trajectories were generated as the concatenation of so-called tracklets containing true positive measurements for the case of tracking a tennis ball from broadcast video. Approaches tracking multiple pedestrians include the approach in [ZHANG et al., 2008] where the data association problem was encoded in a cost-flow network and solved by finding the min-cost flow using the push-relabel method. Using an equivalent representation, Pirsiavash et al. [PIRSIAVASH et al., 2011] provide greedy solutions by means of dynamic programming. By hierarchical linking of measurements to trajectories Brendel et al. [BRENDEL et al., 2011] ensure mutual exclusivity through a maximum weight independent set formulation.

Going one step further, approaches coupling detection and tracking/data association have become an active field of research. Leibe et al. [LEIBE et al., 2007] and Ess et al. [ESS et al., 2009] achieve combined detection and trajectory estimation through optimization by means of a quadratic boolean problem (QBP). Modeled as a continuous energy minimization problem, Andriyenko [ANDRIYENKO and SCHINDLER, 2011, ANDRIYENKO et al., 2012] combined tracking of multiple targets with a constant-velocity dynamic model. The latter approaches have in common that they introduce an analytical distribution based on the detector output. Wu et al. [WU et al., 2012] coupled the same two tasks in a joint objective function, which includes data association based on network flows and sparsity driven detections. Leveraging motion in cases where appearance-based detection is employed was proposed by Collins [COLLINS, 2012].

As it will be seen later, a similar idea going one step further will be pursued: Using detector responses available *directly* at image-level, the states of the ball linked through the known physical motion model over time will be estimated.

2.2.3 Tracking Approaches Specific to Flying Balls

In the past, other robotic ball catching systems did not employ any multiple target approaches as catching was limited to one arm. Using either color information [RILEY and ATKESON, 2002, SMITH and CHRISTENSEN, 2007, BÄTZ et al., 2010] or segmentation with respect to a reference image [FRESE et al., 2001], measurements were extracted from the camera images. Estimating the ball's position and velocity from image features is performed either using an EKF [FRESE et al., 2001, SMITH and CHRISTENSEN, 2007] or by fitting a parabola to camera measurements [HOVE and SLOTINE, 1991, RILEY and ATKESON, 2002, BÄTZ et al., 2010].

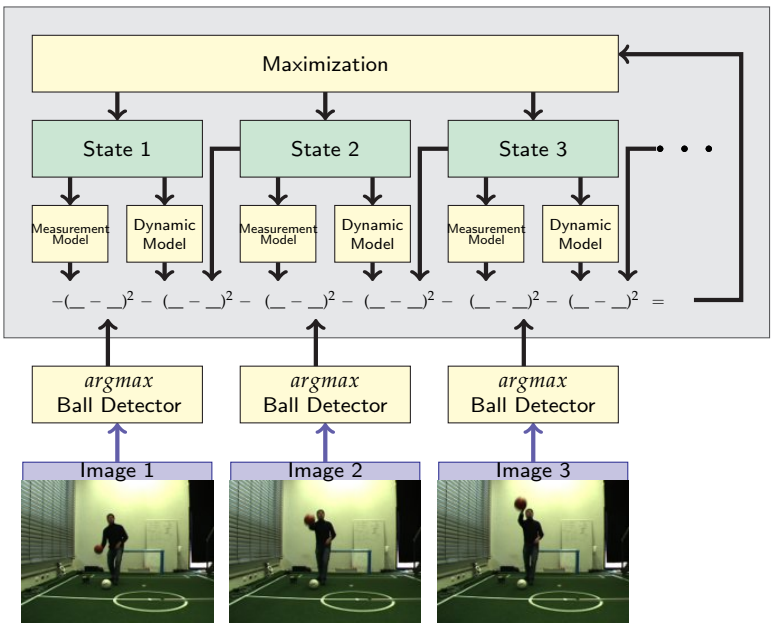


Figure 2.1: Components of a *bottom-up* ball tracker and flow of data between components. Tracking is performed by maximizing the likelihood, which the inverse of the sum of squared errors between detector peaks and the image position obtained from the state propagated through the measurement model and the sum of squared errors due to the dynamic model. As detection is not included in this optimization process (depicted as a gray box), detection results remain fixed.

Other work in tracking balls includes Ribnick et al. [RIBNICK et al., 2007], who were able to detect regions of rapid motion in image space and associate these measurements to a parabolic trajectory using an expectation-maximization algorithm. A system for tracking a soccer ball from broadcast image data was introduced by Ren et al. [REN et al., 2004] with the ability to classify the ball's state over time (e. g. flying, rolling). Using high speed cameras, Shum and Komura [SHUM and KOMURA, 2005] presented experimental results of estimating position and rotation of a pitched baseball. Furthermore, although not relevant for this work, a comprehensive theoretical analysis including determination of conditions for a unique solution of estimating 3D position and velocity of balls from a single view was given by Ribnick et al. [RIBNICK et al., 2009].

To summarize this related work, current state of the art single-target tracking methods for flying balls employ a *bottom-up* approach such as illustrated in Fig. 2.1. Detection of the ball is handled at a stage preceding the actual tracking algorithm. Tracking itself concentrates on adjusting the sequence of states such that they match the detections and the dynamic model of a ball flight. Therefore, the performance of trajectory estimation depends on the quality of the results from the detector stage, which makes these *bottom-up* approaches likely to not make the most out of the information available from the images.

2.3 Single Target Model

The basis for all implemented multiple target tracker algorithms is a single target model capturing the ball's flight properties and its appearance in sensor readings. Additionally, as the camera is not stationary but moving while the robot reaches for the ball, inertial pose estimation has to be performed for determining the changing extrinsic camera parameters.

2.3.1 Dynamic and Measurement Model

Let the state of a ball be its position x and velocity v. The motion of a flying ball can then be described using Newton's laws of motion including gravity. Unfortunately, depending on the ball or flight properties, additional forces might influence the trajectory considerably. De Mestre [DE MESTRE, 1990] and Armenti [ARMENTI, 1992] compiled such effects affecting ball flight in the context of sport science and showed how to formally treat them.

Since the ball has a relatively large cross-section, the only major non-gravitational force to be considered is the drag force. This is important as air drag has a major impact on the predicted catching position. The motion is therefore described by the following two first order differential equations:

$$\dot{x} = v \tag{2.1}$$

$$\dot{v} = g_0 - \alpha \cdot \|v\| \cdot v \tag{2.2}$$

with ball position x, ball velocity v, gravity due to free fall g_0 and the air drag coefficient α, which is a scalar and determined in advance for the specific ball. Furthermore, process noise σ_Q is considered. The corresponding measurement function is denoted as g and is obtained through Euler integration.

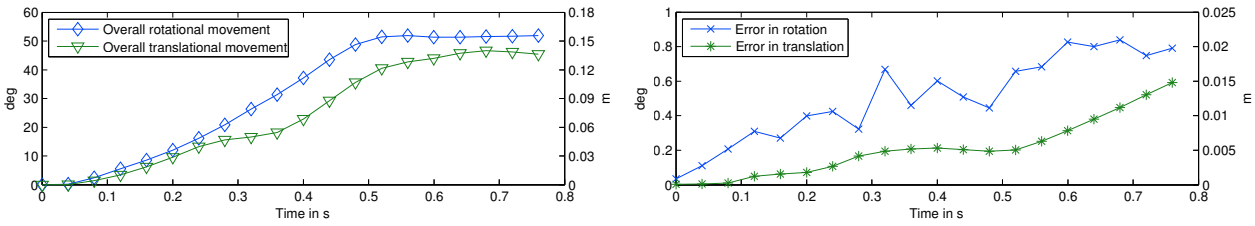
The measurement function h maps a ball state x to a calibrated camera image position and radius. For this, the cross of four points on the ball centered at x with spacing $d/2$ orthogonal to the line of sight is computed, where d is the predetermined ball diameter. Projecting these points into the image plane, the center and radius are determined by computing the mean and standard deviation:

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{4} \sum_{i=1}^{4} \begin{pmatrix} p_{x,i} \\ p_{y,i} \end{pmatrix}, \quad r = \sqrt{\frac{1}{4} \sum_{i=1}^{4} (p_{x,i} - x_c)^2 + (p_{y,i} - y_c)^2} \tag{2.3}$$

The measurement uncertainties $\sigma_{x,y}$ for the circle center and σ_r for the radius of the circle need to be defined. While the uncertainty for the circle center is absolute, σ_r is modeled relative to the circle radius. This helps in capturing the actual uncertainty occurring in detection and avoids linearization problems with distant balls.

2.3.2 Inertial Pose Tracking

As mentioned in the introduction the cameras are not static but move when the robot moves and even vibrate when the arms move due to the reaction forces, see Challenges three to five in Sec. 1.3. Although these effects were reduced by the controller, they were



(a) (b)

Figure 2.2: Overall rotational and translational movement **(a)** and the corresponding error **(b)** of the estimated 6D head pose. The latter was obtained by comparing IMU integration with respect to ground truth during a catch.

neither completely cancelable nor fully observable by the joint sensors and neglecting these effects would drastically impair the tracking precision.

The proposed solution to this problem is to isolate the head-arm system from the rest of the robot and treat it as a self-contained catching device. It is somehow moved by the rest of the robot and the catching device's motion is exclusively obtained from the head-mounted IMU. The estimated displacement of the catching device is used for the tracking and the planning stage.

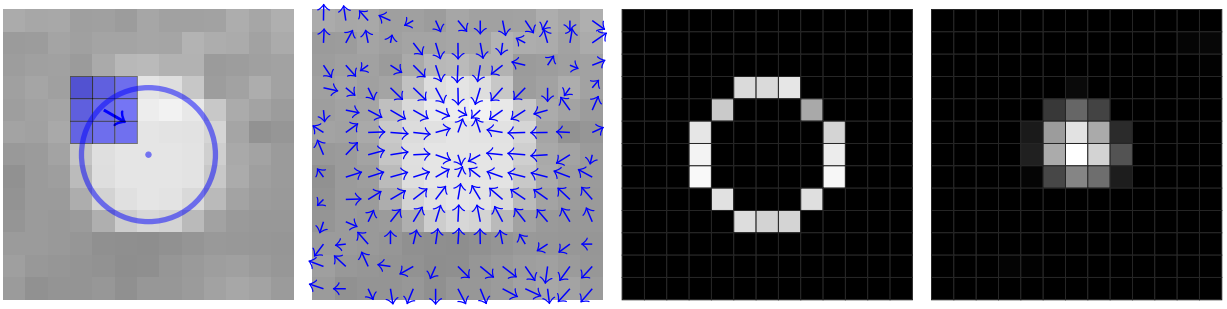
For easy applicability of Eq. 2.2, a fixed world coordinate system for representing the ball state (x, v) is assumed. Based on the IMU pose on start up, the orientation is defined to point opposed to measured gravity, with zero translation. Depending on the catching state, two modes were deployed to account for different catching phases which are available from the robot's operating state:

1. **Orientation estimation:** Orientation estimation is performed when the robot is known to be stationary. For this, the linear velocity is set to zero while gyroscope bias and orientation are estimated. A UKF based orientation estimation scheme inspired from [KRAFT, 2003, MARINS et al., 2001, KIM and GOLNARAGHI, 2004] was employed for this task.

2. **Full pose estimation:** When the robot starts to move, both orientation and translation relative to the world frame are tracked. This is realized by integrating measured angular velocity once and linear acceleration twice over time after subtracting gravity. The integration causes drift which is however reduced by including the estimated bias. Sufficient precision is achieved for typical ball flights, please see Fig. 2.2 for a plot of the typical movement of the head and the corresponding error in estimation over time.

2.4 Circle Detection

The circle detection scheme will be described concisely for completeness, although it is not the work of the author. In each of the cameras' gray scale images balls are detected by their appearance as circles when projected into the image plane. An enhanced Sobel gradient filter C, which performs local image variance normalization, allows evaluation



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

Figure 2.3: Different stages of the circle detector's method of operation: **(a)** example of evaluating gradient information at a circle pixel to measure radial contrast, **(b)** vector image after applying filter to image, **(c)** radial response (Eq. 2.4) at pixels along a fixed size circle, **(d)** circle response (Eq. 2.5) for a fixed radius ($r = 3$) but different circle center x_c, y_c.

of how perturbed a linear gradient is, see Fig. 2.3(a–b), instead of indicating gradient intensity as achieved by the classical Sobel filter.

How much the linear gradient of a local image contributes to a circle can then be defined as a function of a point (x, y) and radial direction α along the circle, see Fig. 2.3(c).

$$R(x, y, \alpha) = \left(\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \cdot C(x, y) \right)^2 \tag{2.4}$$

By integration along the circle, the overall response CR for a circle at x_c, y_c, r is determined as illustrated in Fig. 2.3(d) for a fixed size circle:

$$CR(x_c, y_c, r) = \frac{1}{2\pi} \int_{\alpha=0}^{2\pi} R(x_c + r\cos\alpha, y_c + r\sin\alpha, \alpha) d\alpha \tag{2.5}$$

Exhaustive evaluation of complete images at the circle radii of interest is computationally expensive. To allow real-time operation, two enhancements were made. First, detection starts not directly at the original scale but at lower scales with hierarchical refinement up to the original scale. This allows detection of large circles in repeatedly downsampled images while smaller circles are usually detected at half the original resolution. Second, all equations above were implemented using SIMD (single instruction, multiple data, i. e. SSE) instructions and parallelized for multiple cores for efficient computation. Please see [5] and [8] for detailed information on the circle detection scheme.

2.5 Multiple Hypothesis Tracking

At each time step, the circle detector returns a set of the most circular features of the camera images as measurements. Unfortunately, this set obviously contains clutter and does not necessarily include a measurement which originates from the ball. The task is therefore to associate measurements from ball over time such that a UKF can estimate the trajectory's state using the introduced single-target model. The first method applied to this task is multiple hypothesis tracking (MHT) introduced by Reid [Reid, 1979]. This

algorithm systematically generates as set of hypotheses to account for the different assignments of measurements to targets. These assignments are constrained such that a measurement can only originate from one target and vice versa. The core of the algorithm is the recursive computation of a probability for each hypothesis, which is the product of single probabilities *explaining* the current state of a hypothesis. One way to compute the probability of each hypothesis is the approach by Cox and Hingorani [COX and HINGORANI, 1996]:

$$\begin{aligned} P(\omega_m^k|Z^k) =& \frac{1}{c} P(\omega_{l(m)}^{k-1}|Z^{k-1}) \lambda_N^v \lambda_F^\phi \prod_{i=1}^{m_k} \mathcal{N}_{t_i}(z_i(k))^{\tau_i} \\ & \prod_t (P_D^t)^{\delta_t} (1-P_D^t)^{1-\delta_t} (P_\chi^t)^{\chi_t} (1-P_\chi^t)^{1-\chi_t} \end{aligned} \tag{2.6}$$

Normalized by a factor c, the computed probability is an extension of the parent hypothesis $P(\omega_{l(m)}^{k-1}|Z^{k-1})$ computed in the previous time step. $\mathcal{N}_{t_i}(z_i(k))^{\tau_i}$ represents how well the measurement z_i at time k matches a given target t_i which is available from an integrated Kalman filter. The rest of the parameters constitute the model encoding the various events involving multiple targets that are expected to happen. The density of the appearance of new targets λ_N and the density of false alarm measurements λ_F are exponentiated by the number of new targets v and the number of false alarms ϕ, respectively. P_D^t is the probability of detecting a measurement for track t, while P_χ^t is the probability that track t ends. Finally, the three indicator variables τ_t, δ_t, χ_t are 1 if $z_i(k)$ is assigned to an existing track; if track t, known at time $k-1$, is also detected at time k; and if track t known at time $k-1$ is terminated at time k. In all other cases, these variables are zero, switching off the corresponding subterms.

Unfortunately, an optimal MHT is not feasible due to the exponential complexity of the growing hypotheses tree. The obvious approach is to approximate the entire space by considering only a subset of hypotheses. This is usually achieved in three ways: (1) Ratio pruning removes any hypothesis whose ratio to the best hypothesis falls below a threshold; (2) Generate only the k-best hypotheses right from start which can be realized efficiently by Murty's algorithm as proposed in [COX and HINGORANI, 1996]; (3) N-scan-back pruning removes older hypotheses based on the idea that any ambiguities are resolved after N time steps.

2.5.1 Adaptation to Ball Tracking

For tracking balls the available MHT implementation by Cox and Hingorani [COX and HINGORANI, 1996] was used. A standard UKF employing the single-target model as presented in Sec. 2.3 was implemented for the required underlying single-target propagation.

Measurements obtained from each camera were integrated directly in a sequential manner, i. e. measurements of the left camera are integrated first before measurements of the right camera are considered. This means that MHT is executed twice per time step, but ensures that correspondences from stereo and over time are faithfully considered from the measurements. Alternatively, by matching circle detections from image pairs at each time step, one could simply fed the resulting 3D positions into MHT. While this filter-

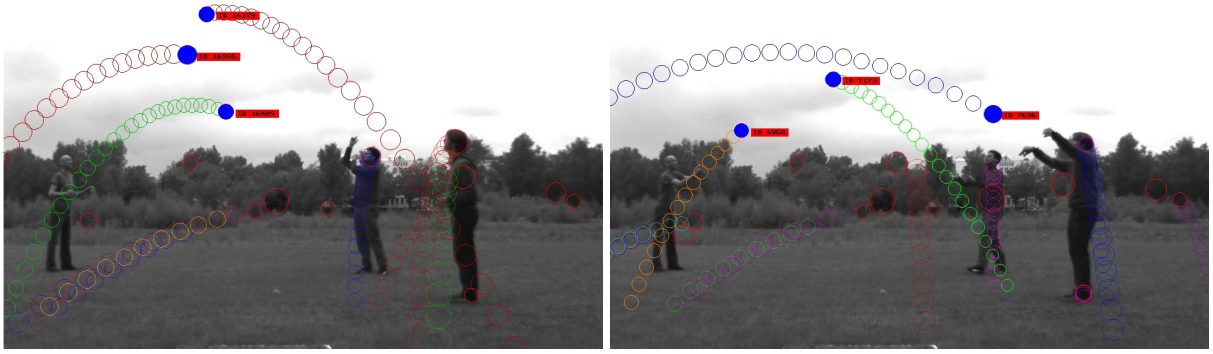


Figure 2.4: Two camera images and corresponding tracking results from an outdoor ball tracking scene where four persons (one outside the image) threw balls towards each other. Detected flying balls are visible by their predicted trajectories, while their recent circle detection is highlighted and annotated by their track number.

ing at circle level would reduce the actual number of integrated 3D features radically, an increased number of features might be missed due to detector inadequacy, which might impair tracking performance.

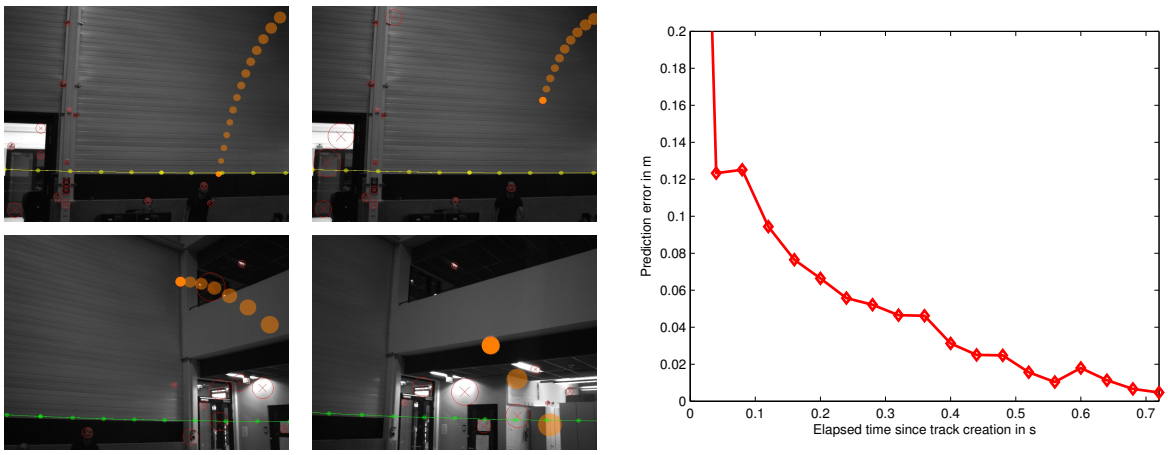
New tracks are initialized at every frame from every available measurement for which an appropriate method needed to be devised. Using the inverse of the measurement function, a position can be recovered from a single measurement through triangulation along the ball diameter. This is not the case for a ball's initial velocity as at least two measurements are required to properly define velocity. Therefore, while initializing, the ball's velocity is set to zero along with a large prior covariance to account for typical velocities of thrown balls. When new measurements are associated with the track in the next iteration, a reasonable velocity is then implicitly computed through the UKF update.

Such a ball tracking system was initially presented in [5], although for a stereo camera system (1024×768 px @ 30 Hz) without using an IMU and therefore intended for stationary use only. For proper tracking, the gravity vector relative to the cameras was integrated as a parameter in the system's state and estimating by throwing a couple of balls during setup.

The system was successfully employed in an outdoor scenario, where four people threw up to three balls simultaneously towards each other. Please see Fig. 2.4 for snapshots of this scene including tracking results. Real-time performance was achieved with an average computation time of 22.5 ms per frame where MHT contributed 10 ms per frame on a Intel Dual Xeon Quad-Core @ 2.5 GHz.

2.5.2 Adaptation to Robotic Ball Catching

For robotic ball catching the approach was refined in two ways. First, two probabilities from the multiple target model are adjusted depending on the state of the target: The probability of detection is set to zero, $P_D = 0$, if the projection of the target's state is outside the image. Additionally, if the target state indicates that the ball has hit the ground, the track is terminated by setting P_χ to 1, while assumed to continue if above ground level ($P_\chi = 0$).



(a) (b)

Figure 2.5: **(a)** Image sequence recorded during a ball catch. Circle detections are depicted as red circles while the detected track is highlighted by its predicted trajectory. The IMU-based head pose estimation is indicated as a colored artificial horizon, where yellow corresponds to orientation estimation and green depicts combined orientation and translation estimation. **(b)** Prediction error with respect to ground truth over time since track creation. Ground truth has been obtained from an external 3D tracking system.

Second, prior information is leveraged at the track initiation phase to rule out detections that do not match the start of a typical thrown ball. For this, a 6D Gaussian is fitted to initial states from trajectories of simulated throws that hit the robot's work space. This encodes information from where (position) and where to (velocity) the ball is typically thrown towards the robot. This is integrated into the track initiation phase where a track is started from each measurement. An initial state and a likelihood of how well this measurement fits the prior Gaussian are computed based on a Kalman filter update. This helps to discard false alarm measurements efficiently and rarely produces unwanted trajectories. Please see Sec. 2.6.2 for elaborate treatment as it was introduced in detail with the GM-PHD filter.

The complete ball tracking approach was first presented in [8] and later included in [2] as part of the complete ball catching system. As a performance indicator, the catch rate was about 80%. Catching failure had been attributed to tracking either due to detection problems or when the ball's predicted trajectory is tangential to the robot's workspace. In the latter, situations arise where the initial imprecise trajectory prediction indicated a catching position within the work space, while the ball actually flew past. This gave the impression that the robot missed the catch.

Figure 2.5 presents visual and numerical results on the tracking quality. This single sample illustrates how the accuracy of predicting the trajectory of the catch point improves as more measurements are integrated. Although the final accuracy is 0.5 cm, measurements obtained $0.16 - 0.2$ s before the time of the actual catch have to be considered due to delays. Therefore, the last available prediction for the planning stage has an accuracy of about 1.5 cm at the catching position, which is enough for catching a ball.

As mentioned in Sec. 1.4, the vision system runs on the robot on an embedded Intel Core 2 Quad Q9000 @2.00GHz, where processing of stereo images takes about 25 ms per

frame while MHT runs between 5 and 10 ms depending on whether the system is idling or tracking.

2.6 Probability Hypothesis Density Filtering

Besides MHT, a conceptually different approach was implemented, namely probability hypothesis density (PHD) filtering which was initially proposed by Mahler [MAHLER, 2003, MAHLER, 2007b]. Instead of explicitly constructing associations between measurements and objects through hypotheses as in MHT, this algorithm unifies all available measurements with all considered targets in a composite-hypothesis manner. This is favorable for real-time applications, as it allows a lower-level control over computational cost than in the hypothesis driven MHT approach.

Being an approximation to the multiple target Bayes filter, the filter recursively propagates the first-order moment statistic of the multiple target posterior in state space. Instead of representing a probability distribution, the posterior PHD denotes an intensity with the important property that integration of it over any region in state space indicates the *expected* number of targets in the considered region.

The PHD recursion makes use of the well known predicting/updating approach for propagating the intensity. Based on the representation of the intensity, two types of filters have emerged. First, the use of particles gave rise to the SMC-PHD filter [MAHLER, 2007b], and second, a mixture of Gaussians resulted in the GM-PHD filter [VO and MA, 2005, VO and MA, 2006]. For the ball catching application the latter was chosen where the density is constructed from a mixture of weighted Gaussians.

2.6.1 Gaussian Mixture Probability Hypothesis Filter

Similar to MHT, the GM-PHD filter makes use of an underlying Kalman Filter with the single-target model. In fact, the same unscented Kalman filter that was already used in MHT including the model from Sec. 2.3 was reused here. Furthermore, both MHT and GM-PHD share the same multiple-target model parametrization having only differences in naming. To illustrate the working principle of this approach, its recursion will be introduced concisely.

Prediction. Based on a prior GM-PHD composed of $n_{k|k}$ weighted Gaussian components

$$D_{k|k}(x|Z^k) = \sum_{i=1}^{n_{k|k}} w^i_{k|k} \cdot \mathcal{N}(x; x^i_{k|k}, P^i_{k|k}) \tag{2.7}$$

the predicted GM-PHD is again a mixture distribution where each Gaussian is predicted according to the single-target motion model and its corresponding weight is scaled by p_S,

the probability of target survival, which is $1 - P_\chi$. Additionally, components are added which correspond to the intensity of target creation. The complete intensity is

$$D_{k+1|k}(x) = \sum_{i=1}^{a_k} \beta_k^i \cdot \mathcal{N}(x; b_{k+1|k}^i, B_{k+1|k}^i) + \sum_{i=1}^{n_{k|k}} p_S \cdot w_{k|k}^i \cdot \mathcal{N}(x; x_{k+1|k}^i, P_{k+1|k}^i) \tag{2.8}$$

where β_k^i, $b_{k+i|k}^i$, and $B_{k+1|k}^i$ define a Gaussian mixture birth intensity with a_k components. This mixture accounts for possible initial target locations in state space. The prediction of existing targets, $\mathcal{N}(x; x_{k+1|k}^i, P_{k+1|k}^i)$, is computed using the unscented Kalman filter's prediction step.

Update. When writing the predicted intensity as a single sum of Gaussians

$$D_{k+1|k}(x) = \sum_{i=1}^{n_{k+1|k}} w_{k+1|k}^i \cdot \mathcal{N}(x; x_{k+1|k}^i, P_{k+1|k}^i). \tag{2.9}$$

the update intensity is

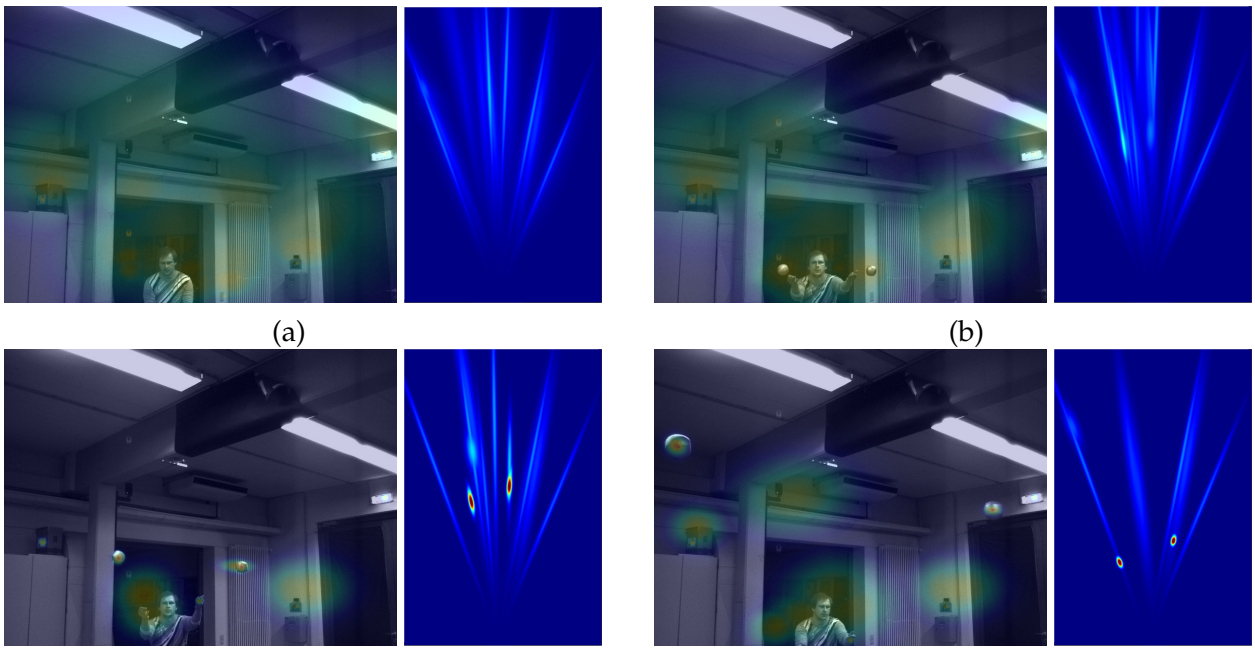
$$D_{k+1|k+1}(x) = \sum_{i=1}^{n_{k+1|k}} (1 - P_D) w_{k+1|k}^i \mathcal{N}(x; x_{k+1|k}^i, P_{k+1|k}^i) + \sum_{j=1}^{m_{k+1}} \sum_{i=1}^{n_{k+1|k}} \frac{s^{i,j}}{s^{*,j} + \sum_{k=i}^{n_{k+1|k}} s^{k,j}} \mathcal{N}(x; x_{k+1|k+1}^{i,j}, P_{k+1|k+1}^{i,j}). \tag{2.10}$$

The sum is separated into two parts. The first one contains only components not updated at all and simply scales their weight by $(1 - P_D)$. The second one is a double sum and is the result of fusing the $n_{k+1|k}$ Gaussians from the predicted intensity with m_{k+1} detections. For this, a regular Kalman filter update is performed resulting in a Gaussian and a factor $q^{i,j}$ which reflects how well the detection matched the prediction based on the Mahalanobis distance. The first defines the updated component $\mathcal{N}(x; x_{k+1|k+1}^{i,j}, P_{k+1|k+1}^{i,j})$. The latter is used to compute the support $s^{i,j}$ of a detection with respect to a component. This is then normalized by dividing by the overall support of this measurement towards other Gaussian mixture components plus clutter. The two types of support are

$$s^{i,j} = w_{k+1|k} \, P_D(x_{k+1|k}^i) \, q^{i,j}, \quad s^{*,j} = \lambda c(z^j), \tag{2.11}$$

where λ is the false alarm density (known as λ_F in the MHT model), eventually spatially distributed according to $c(z)$.

In practice, two distinctive cases can be distinguished. If a detection matches a mixture component, the support of the component in the nominator dominates normalization such that the weight is approximately 1. On the other side, if a component matches no detection significantly, the normalization is dominated by the support s_z^* for clutter in the denominator reducing the component's importance in the intensity considerably.



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

Figure 2.6: PHD as a mixture of Gaussians projected (only position) into image space (left) and its projection onto the floor (right) for a ball throwing sequence. The projected space on the floor is 4 m in width and 6 m depth direction. Four stages can be distinguished during such a sequence. **(a)** When no ball is pitched, only low weighted components recently initialized from false-alarm measurements emerge. **(b)** Detections from actual balls generate components which become peaked as these are supported by measurements in the following images **(c)**. After the integration of several detections, two strong weighted and highly peaked Gaussians denoting the two thrown balls dominate the mixture **(d)**.

Similar to the hypotheses pruning in MHT, a procedure for managing the growing number of fused mixture components must be established. This was done either by gating where the fused Gaussian i, j isn't created due to the Mahalanobis distance exceeding a fixed threshold. Furthermore, Gaussians were merged until the number falls under a certain level [Vo and Ma, 2006]. This was done by merging pairs with similar states first, i. e. that they have a low mutual Mahalanobis distance, in a way which preserves mean and covariance of the mixture.

2.6.2 Prior-Based Track Initialization

Due to the use of a non-linear model, initializing a track through a rough prior such as suggested in Sec. 2.5.1 or by injecting this rough prior in the GM-PHD's prediction step (Eq. 2.8) leads to linearization problems when integrating a measurement. Performing a UKF update linearizes the measurement model at the 1σ-range of the vague Gaussian. As this usually has a high covariance to cover the variety of throws, the mapping of a ball to a circle radius is poorly approximated. This is especially true for the depth mapping and leads to erroneous initial state estimates impairing initial trajectory estimation severely.

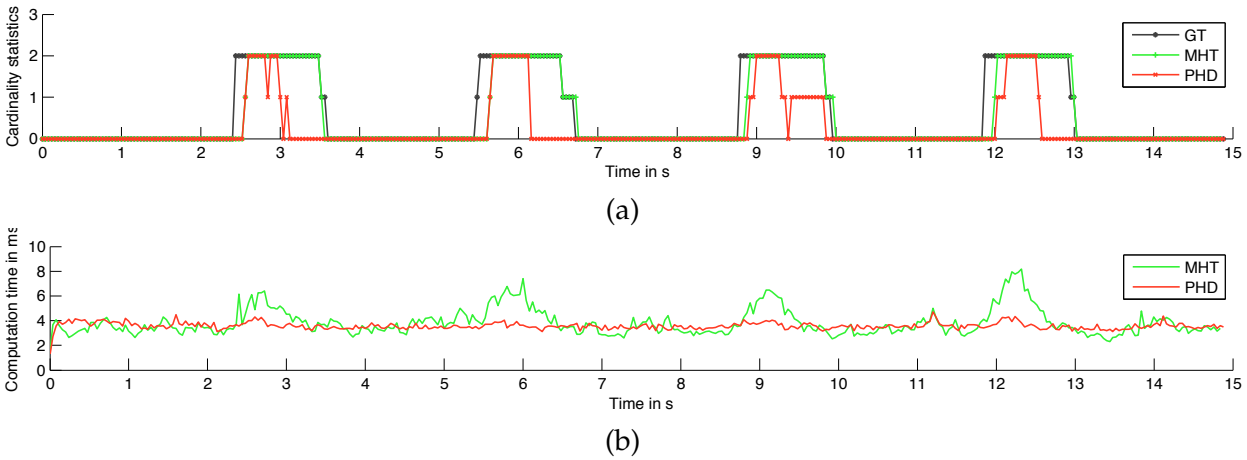


Figure 2.7: Comparison of the GM-PHD filter and MHT for a sequence of thrown balls (four throws, two balls per throw). (a) Cardinality statistics, i. e. estimated number of targets, compared to ground truth (GT) over time. (b) Measured computation time on an Intel Core^{TM}2 Quad Q9000 embedded system on the robot.

To alleviate this issue a special update routine that integrates nicely into the GM-PHD framework was developed. Instead of performing linearization on the vague prior in state space, it is performed on the much more precise 1σ-range of the detection. The procedure consists of two steps: First, the ball's 3D position and its covariance are determined by propagating the noisy detection through the inverse measurement function h^{-1}. In the second step, the resulting 3D Gaussian (position only) is now fused with the initial prior 6D Gaussian (position and velocity) by a linear Kalman filter update. The obtained 6D Gaussian and the factor $q^{i,j}$ are used as the initial state and as a measure of how well the detection matched the prior based on the Mahalanobis distance for proper inclusion in Eq. 2.10. The latter is further used for gating, i. e. rejecting measurements to be fused when they do not match the prior.

Integration into the GM-PHD filter was done by exclusively fusing the birth Gaussian with measurements using the just mentioned method. This functionality was also integrated into MHT to initialize targets from measurements as mentioned in Sec. 2.5.2.

Figure 2.6 depicts the mixture intensity during tracking of a pair of thrown balls. It can be seen how two highly peaked components representing the two detected targets emerge from initially low weighted components created from the special update using the prior.

One drawback of the PHD filter is its unfavorable behavior when confronted with missing detections. When a target has no corresponding measurement, the corresponding Gaussian is only propagated by the first term in Eq. 2.10, effectively scaling its weight by $1 - P_D$. As P_D is usually quite high, this component is not lost but it has a negative impact on the intensity and the cardinality of detected targets. Although the cardinalized PHD filter [MAHLER, 2007a] and its Gaussian mixture instance [VO et al., 2006, VO et al., 2007] were proposed to resolve this problem, these solutions come at the cost of increased complexity as the entire probability distribution of the number of targets is propagated in addition to the intensity. For the application here missing detections do not pose a problem, as a missed target is usually picked up in the next time step and all available

data is retained for proper tracking. Nonetheless, it is clearly an undesirable behavior for a tracking algorithm.

This effect is visible when comparing tracking performance of MHT and GM-PHD in terms of number of detected targets as shown in Fig. 2.7(a). For both approaches, the same UKF and multiple target parametrization has been used. Further, both methods employ the same track initialization scheme using a Gaussian prior as introduced above. While MHT robustly detects any track start after a few measurements and determines the ending of a track accurately, the latter cannot be observed for the GM-PHD filter. This is due to the aforementioned problem of the filter when confronted with missing detections which are common at the image border. This terminates tracks prematurely long before they actually reach the ground, which is no problem in practice as it happens beyond the catch point.

On the other hand, the desired deterministic tracking run time is achieved as can be seen in Fig. 2.7(b). Due to a limitation of the involved components in GM-PHD instead of limiting the number of hypothesis as in MHT, a more fine grained control is achieved. To be precise, GM-PHD only needed 4.3 ms while MHT needed 8.2 ms in the worst case, including Kalman filter evaluation within both trackers. Although no explicit comparison of prediction accuracy between GM-PHD and MHT was performed, comparison of trajectory predictions from GM-PHD to ground truth revealed roughly the same quality in prediction accuracy as with MHT. This is due to the use of the same underlying single-target tracking model. In fact, both filters would estimate the exact same state over time if the number of Gaussians in the GM-PHD filter and the number of hypotheses in MHT were not limited.

2.7 Fully Probabilistic Tracking

Both approaches discussed so far as well as all of the related work in Sec. 2.2 are based on a dedicated detection phase preceding the actual tracking stage. Please refer to Fig. 2.1 for an illustrative explanation. Because of this bottom-up approach, the accuracy of the resulting trajectory depends on the object detector's performance regarding missed, inaccurate and false-alarm detections. In fact, most of recent work focused on these artifacts and elaborate tracking algorithm were developed that try to deal with the errors propagated from the detection phase.

When interested in determining an object's trajectory accurately over time, the aforementioned approaches reveal weaknesses. Figure 2.8 illustrates this for vision-based approaches which focus on appearance cues neglecting motion characteristics on the left, and for classical tracking approaches which estimate the parameters based on a motion model in the middle. Relying solely on the position of detector peaks leads to a trajectory with either poor local accuracy or poor global accuracy, respectively. To alleviate these issues, a tracking algorithm was suggested [1] that integrates single-target tracking using a physical motion model with response-based detection allowing continuous trajectory estimation. This was implemented in [7] and extended by probabilistically modeled trajectory boundaries and trajectory compatibility for handling multiple targets.

Much in the like of the batch Bayesian approaches listed in Sec. 2.2.2, the developed approach searches for the maximum likelihood solution, i. e. $\arg\max_x p(X = x|Z = z)$,

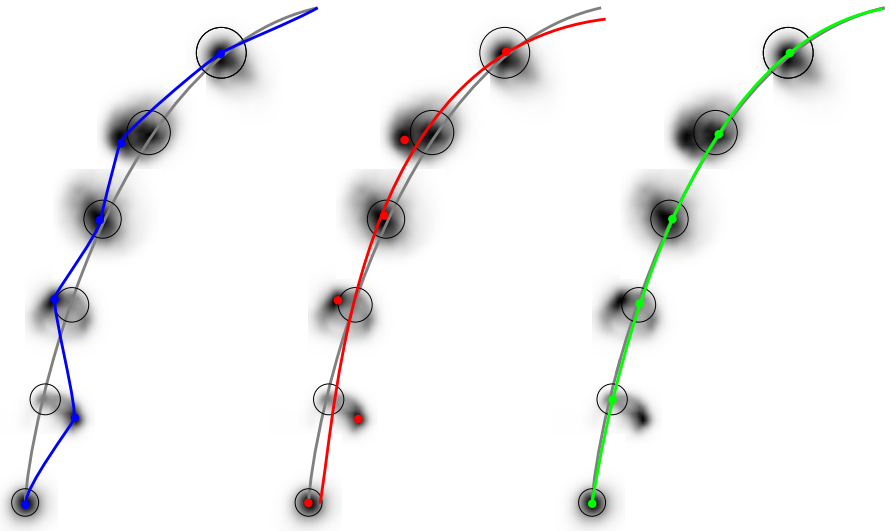


Figure 2.8: Illustration of three different trajectory estimation results when applied on the same raw detector responses over time. Higher detector responses are indicated as darker gray levels and the peak does not necessarily lie on the actual trajectory (gray line) due to detector failure. The three trajectories result from appearance focused tracking **(left)**, classical single target tracking **(middle)** and the developed fully probabilistic tracking approach which integrates the tasks of detection and tracking using a physical motion model **(right)**.

with one major difference: Z is now a sequence of images instead of sets of measurements extracted from images. This allows the task of detection to be integrated directly in the probabilistic optimization process. By keeping images in memory and reevaluating them at the corresponding image portions as suggested by the states along the trajectory defined by a physical motion model, *all* available evidence from the images is used for tracking, not only the detector peaks. See Fig. 2.8 on the right for an illustration of the expected behavior and Fig. 2.9 for an outline of the components and the flow of data.

In detail, given these images, the goal is to recover the unknown number of tracks n_a and each track's states $x^{(a)}_{t^{(a)}_{\text{start}}} \dots x^{(a)}_{t^{(a)}_{\text{term}}}$ between the time of track starting $t^{(a)}_{\text{start}}$ and track ending $t^{(a)}_{\text{term}}$, all this together being called $x = \{(t_{\text{start}}, t_{\text{term}}, x_{t_{\text{start}}} \dots x_{t_{\text{term}}})\}$.

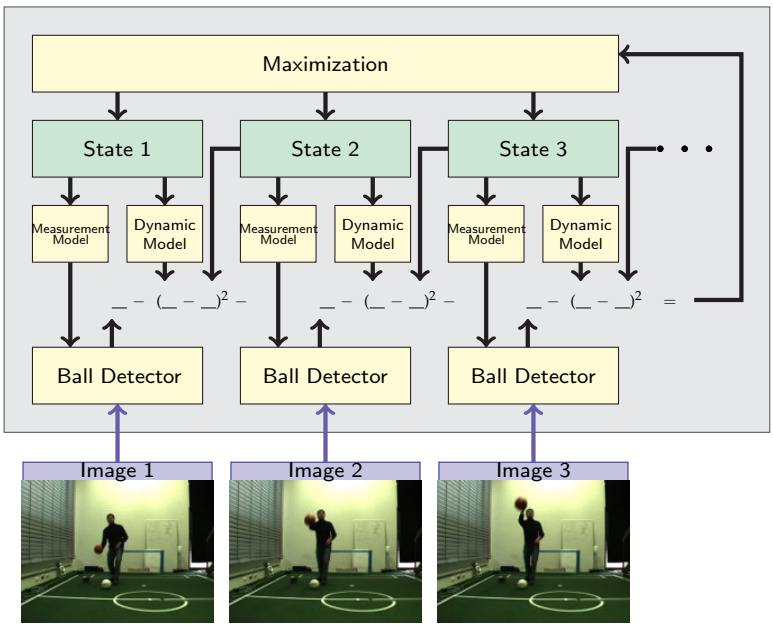


Figure 2.9: Components of the proposed tracker and flow of data between components. The integration of detection into the optimization process (enclosed by a gray box) is central to this approach, as opposed to the bottom-up approach depicted in Fig. 2.1 where detection is performed once and the result is held fixed. Tracking is again peformed by maximization of the likelihood, which is the negative sum of squared errors due to the dynamic model and the detector response at the states along the trajectory.

Here, $p(X = x|Z = z)$ is modeled as the product of likelihoods, i. e. sum of log-likelihoods

$$p(X = x|Z = z) \propto \exp\ L(x), \text{ where}$$

$$\begin{aligned} L(x) = &\sum_{a=1,t=t^{(a)}_{\text{start}}}^{a=n_a,t=t^{(a)}_{\text{term}}} L_{\text{det}}\left(z_t, x^{(a)}_t\right) + \sum_{a=1,t=t^{(a)}_{\text{start}}}^{a=n_a,t=t^{(a)}_{\text{term}}-\delta t} L_{\text{dyn}}\left(x^{(a)}_{t+\delta t}, x^{(a)}_t\right) + \\ &\sum_{a=1}^{a=n_a} L_{\text{s\&t}}\left(t^{(a)}_{\text{start}}, t^{(a)}_{\text{term}}\right) + \sum_{\substack{a,a'=1,a>a' \\ t=\max(t^{(a)}_{\text{start}},t^{(a')}_{\text{start}})}}^{\substack{a,a'=n_a, \\ t=\min(t^{(a)}_{\text{term}},t^{(a')}_{\text{term}})}} L_{\text{exc}}\left(x^{(a)}_t, x^{(a')}_t\right) \end{aligned} \tag{2.12}$$

where

- $L_{\text{det}}\left(z_t, x^{(a)}_t\right)$ indicates support of object $x^{(a)}_t$ in image z_t,
- $L_{\text{dyn}}\left(x^{(a)}_{t+\delta t}, x^{(a)}_t\right)$ indicates the likelihood that an object transitions from $x^{(a)}_t$ to $x^{(a)}_{t+\delta t}$,
- $L_{\text{s\&t}}\left(t^{(a)}_{\text{start}}, t^{(a)}_{\text{term}}\right)$ is a prior indicating how likely an object emerges at $t^{(a)}_{\text{start}}$ and disappears at $t^{(a)}_{\text{term}}$ and
- $L_{\text{exc}}\left(x^{(a)}_t, x^{(a')}_t\right)$ indicates how likely objects are subject to occlusion.

Due to this thorough probabilistic formulation, the approach was given the name fully probabilistic multiple target tracker (FPMTT).

2.7.1 Algorithm

Algorithm 1 was developed to optimize Eq. 2.12 with respect to all the variables and is constructed around three subproblems, namely trajectory estimation, track limit determination, and assurance of mutual exclusion.

Trajectory Estimation: All states along a trajectory from t_{start} to t_{term} are estimated simultaneously using both raw detector responses and a motion model:

$$\underset{x_{t_{\text{start}}} \dots x_{t_{\text{term}}}}{\arg\max} \sum_{t=t^{(a)}_{\text{start}}}^{t=t^{(a)}_{\text{term}}} L_{\text{det}}\big(z_t, x_t\big) + \sum_{t=t^{(a)}_{\text{start}}}^{t=t^{(a)}_{\text{term}}-\delta t} L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) \tag{2.13}$$

The first part is the likelihood L_{det} of observing an object in image z_t at x_t and is provided by the circle detector introduced in Sec. 2.4. Given a state x_t, the corresponding image position is computed using the measurement model and evaluated with Eq. 2.5. Actual modeling of this likelihood is done by considering the ratio between the probability that the obtained response and radius combination is generated by an actual ball P_{ball} and the probability that it is generated by the background P_{bg} inspired by Sidenbladh and Black [SIDENBLADH and BLACK, 2001]:

$$LR(x_c, y_c, r) = \frac{P_{ball}(CR(x_c, y_c, r))}{P_{bg}(CR(x_c, y_c, r), r)} \tag{2.14}$$

L_{dyn} is the quadratic error between propagating the state x_t to time $t + \delta t$ using the dynamic function and state $x_{t+\delta t}$ and considers Gaussian noise as defined in the dynamic model.

$$L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) = -\|x_{t+\delta t} - g(x_t)\|^2_{\sigma_Q} \tag{2.15}$$

For optimization of this combined objective function, the preconditioned nonlinear conjugate gradient (PNCG) method [SHEWCHUK, 1994] is used.

Track Limits: Determining the limits of trajectories is also done individually. Based on the trajectory's sequence of states, the goal is to obtain $\arg\max_{t_{\text{start}}, t_{\text{term}}}$ of

$$\begin{aligned} L\big(\{(t_{\text{start}}, t_{\text{term}}, x_{t_{\text{start}}} \dots x_{t_{\text{term}}})\}\big) = &\sum_{t=t^{(a)}_{\text{start}}}^{t=t^{(a)}_{\text{term}}} L_{\text{det}}\big(z_t, x_t\big) + \\ &\sum_{t=t^{(a)}_{\text{start}}}^{t=t^{(a)}_{\text{term}}-\delta t} L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) + L_{\text{s\&t}}\big(t_{\text{start}}, t_{\text{term}}\big). \end{aligned} \tag{2.16}$$

Algorithm 1 Fully Probabilistic Multiple Target Tracker

Input: Set of prior tracks A
Output: Most likely set of posterior tracks A'
Set of all posterior tracks A

- Insert initial trajectories as new tracks into A, mark them to do only trajectory estimation

for $x^{(a)} \in A$ **do**

- Extend tracks according to dynamic model g
- Determine track boundaries
  Solve $\arg\max_{t_{\text{start}}^{(a)}, t_{\text{term}}^{(a)}}$ of Eq. 2.16
- Estimate trajectory between boundaries
  Solve $\arg\max_{x_{t_{\text{start}}^{(a)}} \ldots x_{t_{\text{term}}^{(a)}}}$ in Eq. 2.13

end

- Ensure mutual exclusivity by stating GIS problem
  Solve $A' \leftarrow \arg\max_{A \subset \{1..n\}}$ in Eq. 2.19
- Prune tracks in A with low likelihood

The likelihood of track appearance and termination is modeled as

$$L_{\text{s\&t}}(t_{\text{start}}, t_{\text{term}}) = \log p_{\text{start}} + \begin{cases} \log p_{\text{term}} & t_{\text{term}} < t_{\text{now}} \\ 0 & t_{\text{term}} = t_{\text{now}} \end{cases} \tag{2.17}$$

where p_{start} and p_{end} denote the prior probability of target appearance and termination, respectively. The problem can be solved using Kadane's algorithm [BENTLEY, 1984] in linear time.

At each time step, already existing tracks are extended at most one step into the past and two steps into the future. The first helps including image information which might be missed by the initialization mechanism. The latter is not only required to keep up with incoming images but also allows revision of tracking decisions in the light of new image evidence.

Mutual Exclusion: Further, an exclusion mechanism has to be established that prevents existence of similar trajectories originating from the same object but allows occasional occlusion of different objects.

For this, a prior probability P_O is employed and is assigned to any two states as a penalty when their projections into the image overlap.

$$L_{\text{exc}}(x_t^{(a)}, x_t^{(a')}) = \begin{cases} \log p_O & \text{if } x_t^{(a)} \text{ and } x_t^{(a')} \text{ overlap} \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

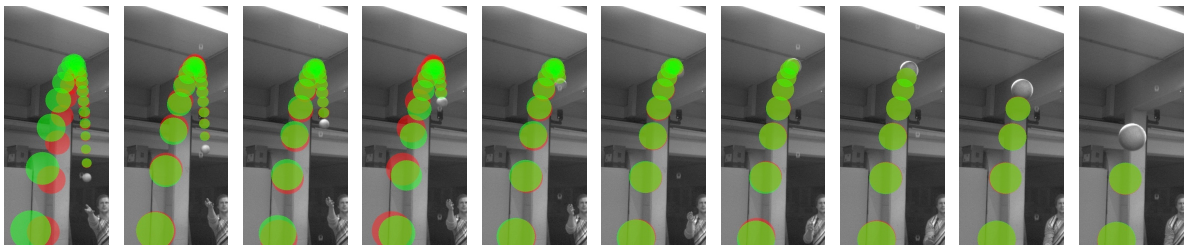


Figure 2.10: Image sequence showing a ball thrown towards the robot augmented by the predicted trajectory over time as computed by MHT/UKF (red) and FPMTT (green). It can be seen that the green trajectory is already quite accurate at the beginning while the red trajectory suffers from early detection inaccuracies and recovers over time to reach the same final accuracy. This impressively illustrates the increased performance at the early tracking stage.

When extended to all states this interaction is the basis for modeling a Generalized Independent Set (GIS) problem [HOCHBAUM, 2000] where each track's likelihood from image evidence as defined in Eq. 2.16 contributes to its existence while an overlap between two tracks penalizes both. The goal of GIS is then to extract the optimal subset of tracks:

$$\begin{aligned}\hat{x} &= \arg\max_{A\subset\{1..n\}} L(\{x^{(a)}|a\in A\}) \\ &= \arg\max_{A\subset\{1..n\}} \Bigg(\sum_{a\in A} L(\{x^{(a)}\}) \\ &+ \sum_{a,a'\in A, a<a'} \sum_{t=\max(t^{(a)}_{\text{start}},t^{(a')}_{\text{start}})}^{t=\min(t^{(a)}_{\text{term}},t^{(a')}_{\text{term}})} L_{\text{exc}}\left(x^{(a)}_t, x^{(a')}_t\right)\Bigg)\end{aligned} \tag{2.19}$$

Unfortunately, GIS is an NP-complete problem. As the number of tracks to consider while tracking is generally low an exhaustive solution is employed. The resulting subset is the most likely hypothesis of tracks given the evidence in the images and can then be passed to the next stage.

Instead of employing an ad-hoc solution for track starting, a MHT/UKF tracker was running in the background and whenever it detected a starting track, a new track was created in the FPMTT. Limiting the number of trajectories to consider is done while solving GIS. A list of the k-best subsets of tracks is maintained and only the tracks from likely subsets are kept up to a fixed threshold. Instead of keeping each response image in memory, only $32 \times 32 \times 32$ px buffer of the circle response was stored in memory and the tricubic approach by Lekien and Marsden [LEKIEN and MARSDEN, 2005] was used for subpixel evaluation through interpolation.

Due to the properties of the tracking problem multiple-target evaluation is non-trivial. As the joint detection and tracking optimization for trajectory estimation is central to the developed approach, single target tracking with respect to tracking accuracy was only evaluated. A proper validation of the multiple-target features is deferred for future investigation.

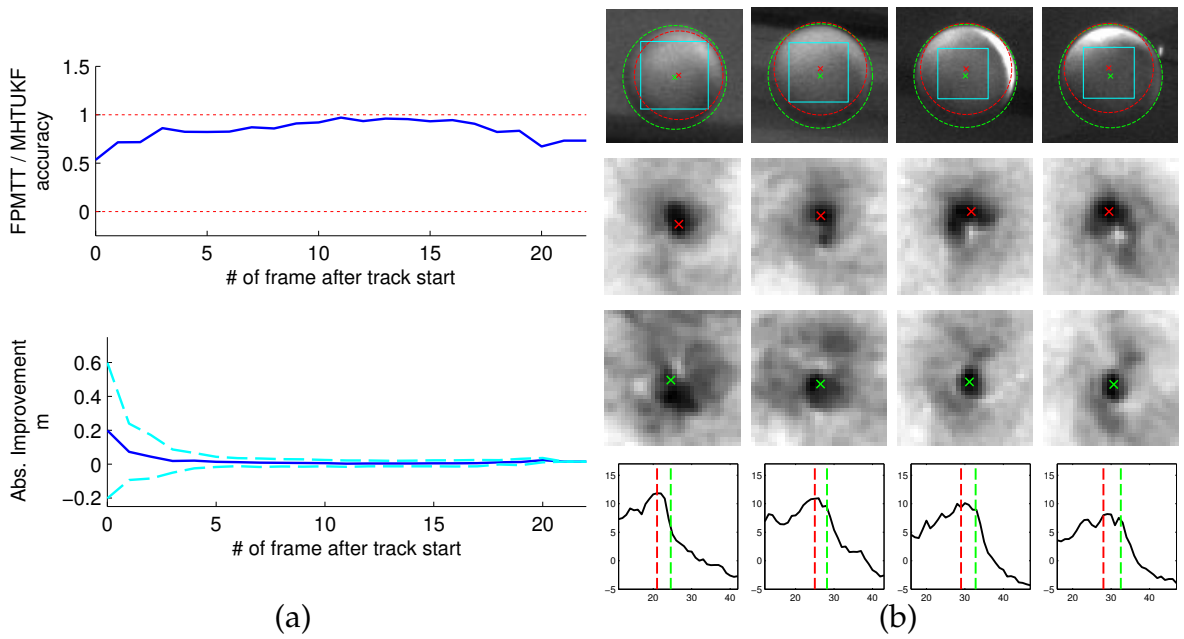


Figure 2.11: **(a)** Geometric mean of the error ratio between FPMTT and MHT/UKF since track start (top). The error is reduced to almost 50 % at the beginning of the trajectory until both methods perform almost the same after seven frames. Average absolute prediction improvement and standard deviation since track start shown as solid and dashed line, respectively (bottom). **(b)** Four examples of erroneous detections and their handling in the developed approach. (1st row) Region of image containing the ball where a red circle denotes the detector maxima and a green one shows the circle determined after FPMTT convergence. (2nd row) Corresponding *LR* volume at detected radius from detector maxima and (3rd row) volume of found circle radius after trajectory estimation. (4th row) *LR* as a function of ball radius over the complete volume for both circles.

Figure 2.10 shows an image sequence comparing the predicted trajectory of FPMTT with MHT/UKF from Sec. 2.5. At the beginning of flight, the trajectory of MHT/UKF is varying due to erroneous circle detections. The global approach of FPMTT allows accounting for that and achieves a more consistent trajectory prediction over the course of the ball's flight. Furthermore, the ability to look back at past frames facilitates refinement as previously unnoticed or discarded detections are considered accordingly.

From a set of 48 recorded trajectories from different throwing sessions Fig. 2.11(a) shows the average error ratio between both approaches since track start on the top. FPMTT halves the important error at track start by roughly 50 % and outperforms MHT/UKF on average by 16.5 %. Absolute error reduction is given in the plot below to give an indication for the performance gain for ball catching. Please see Fig. 2.11(b) for four examples where the detector maxima used as input to MHT are compared to image evidence used by FPMTT. The first have the problem that the found maxima shares only some part of radial contrast of the actual ball resulting in a erroneous circle detection. In contrast, because of the integration of tracking and detection into a joint optimization, detector inaccuracies are resolved using the ball's flight dynamics as context in FPMTT.

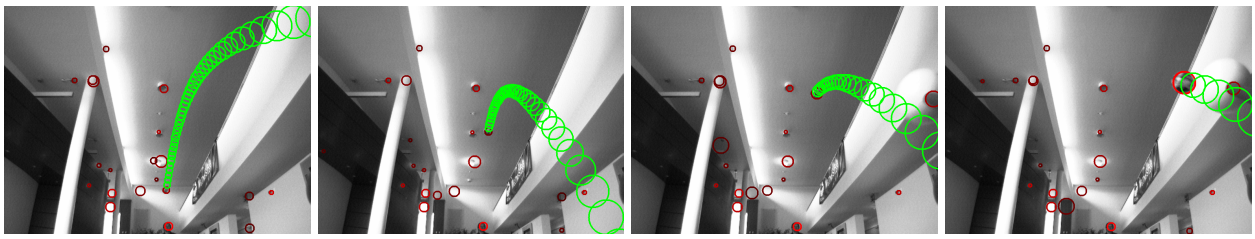


Figure 2.12: Sequence of a ball tracked indoors using the entertainment robot depicted in Fig. 1.4. The prediction (green circles) was computed from the results of MHT/UKF introduced in Sec. 2.5, which was fed by circle detections (red circles).

Real time performance has yet to be achieved as focus was on proper convergence. Using only a single core and exploiting no high level optimizations the average computation time was around 400 ms per frame.

2.8 Summary and Transferability

This chapter presented techniques for tracking multiple balls thrown towards a humanoid robot using head mounted visual cameras for the purpose of catching. Although the difficulty of this tracking problem lies more in accurate tracking and prediction, fully-fledged multiple target solutions were implemented such that every aspect of tracking is considered in a faithful and methodically sound solution.

Based on a UKF single target tracking model, classical MHT was successfully employed for resolving the association between states and measurements for the task of tracking [5] and catching [8, 2] thrown balls. Although the initial accuracy is impaired by the inaccuracy of the circle detection scheme, overall tracking accuracy proved to be sufficient for robotic ball catching including real-time operation through careful tracker parametrization.

Further effort in real-time operability was spent by implementing a GM-PHD filter [6]. As this filter does not enumerate the different combinations of associating measurements to states, more deterministic computational behavior is achieved due to control over the number of tracks (i. e. single target states instead of hypotheses as in MHT) to retain. Although being affected by the missed detection problem, roughly identical tracking accuracy is achieved which is due to the same underlying UKF single target tracking model.

It was further analyzed that the accuracy of these approaches is mainly governed by the accuracy of the used circle detection scheme. To alleviate this issue, the core idea of a fully probabilistic tracking algorithm (FPMTT) was specified in [1] and actually introduced in [7]. It integrates both tracking and response based detection for continuous trajectory estimation. With the use of a physical motion model, this global approach allows reevaluation of the whole past trajectory within the stored images in the light of new information. Results on recorded datasets reveal a roughly 50 % reduced error for the important first estimate when compared to MHT.

The realized ball catching system gained respectable attention resulting in an award nominated video contribution [3], which has more than 350000 views on an internet video website.

Besides porting the tracking methods from the ball catching system to the ball returning entertainment robot (see also Fig. 2.12), the extension of these approaches to further applications is subject to future research. Precise tracking of balls using FPMTT benefits greatly from the rigid motion model paired with the availability of raw detector responses at pixel level. While other objects, e. g. available through CAD models, could be located using response based detection, it has yet to be investigated how much this motion model can be relaxed while the benefits of this approach remain valid. It is especially not well suited for the important class of pedestrian tracking, as pedestrians are not reliably predictable and currently available detectors are concentrating on versatility not accuracy. Nonetheless, one could try to determine the skeletal state of an articulated person from its contour visible in camera images.

Chapter 3

Robot Calibration

When robots are required to perform complex manipulation tasks, possibly in dynamic environments, an accurate interplay of actuation and perception is desired. This is usually accomplished by calibration with the goal of determining geometric and temporal relationships between the perception and actuation components, as well as determining the intrinsic parameters of the components themselves. This chapter focuses on the calibration of a humanoid's head-mounted sensors and their relationship to the robot's complex kinematic chain. Although developed for it, the obtained calibration is not limited to ball catching but is applicable to vision-based manipulation in general. To accomplish the calibration, a textbook style approach is first presented, which combines a series of established practices in sensor calibration. After that, a new approach is introduced that calibrates all relevant parameters in an automated and integrated manner. Finally, additional calibration procedures, which have been developed apart from ball catching, are presented at the end of this chapter.

3.1 Motivation

The goal of this calibration is to resolve the transformation between inertial and visual frames, which allows tracking of balls while moving, and, for actual catching, the determination of the relationship between the frames of the sensing setup and the kinematic chain. Please see Fig. 3.1 for an illustration of the sensor frames involved and parameters to be calibrated.

In robotic systems intended for research both the software and the hardware change over time, requiring frequent (re-)calibration. Furthermore, due to regular maintenance, an invalid state after a collision or worn out components, a previously obtained set of calibration parameters may become invalid and the robotic system has to undergo recalibration.

Such a recalibration is often considered a necessary evil, as it is viewed as the precondition to the actual task the robot was designed for. Therefore, calibration routines often lack the necessary carefulness. For mastering the task of ball catching, a calibration has to be devised that complies with the challenges two (high precision in space and time) and three (moving camera system) from Sec. 1.3. To further motivate the necessity of a precise calibration, major failure modes caused by inaccuracies in the (individual or combination of) calibration parameters of the robot introduced in Sec. 1.4 have been identified:

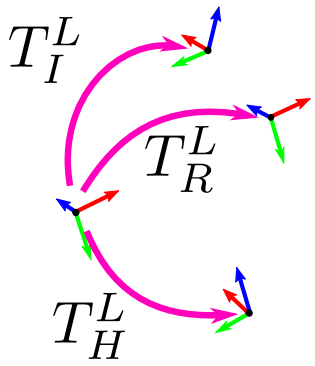


Figure 3.1: View of DLR's Rollin' Justin from behind, including frames of the cameras L & R, the IMU I and the last link of the head H as involved in the calibration process. The goal of the calibration is to resolve the cameras' spatial relationship T_R^L and their intrinsic parameters (focal length $f_{L/R}$, principal point $C_{L/R}$, radial distortion $\kappa_{L/R}$, not illustrated) as well as the transformations with respect to the IMU T_I^L and to the last link of the head T_H^L.

- **Stereo** T_R^L, including intrinsic parameters (focal length $f_{L/R}$, principal point $C_{L/R}$ and radial distortion $\kappa_{L/R}$): While a random rotational deviation between both cameras leads obviously to a failure in tracking right from the start, a change in the cameras' baseline or a divergence in the cameras' vertical axes leads to subtle estimation errors regarding the ball's depth and velocity. When passed to the motion planner, the robot is likely to move to the wrong position in space and grasp at the wrong point in time.

- **Camera (Stereo) – IMU** T_I^L: When using a rigidly attached IMU to estimate a camera's motion, inaccuracies in their geometric relationship influence the performance considerably. As the estimated motion does not match the actual motion, measurements appear at implausible image locations, impairing tracking accuracy and decline even more with increased rotational speed (e. g., due to rapid head movement). Additionally, the gravity acting on the ball as defined in the ball's dynamic model is derived from the result of the IMU's orientation estimation. Therefore, any rotational deviation affects trajectory estimation making the predicted trajectory drift away from the actual one.

- **Camera (Stereo) – Kinematic** T_H^L: Being the interface between perception and actuation, any calibration error between the cameras and the kinematic chain of the robot directly maps to an error in the catching position. This is especially true for catching positions which are not close to the robot (e. g. stretched out arm), as the distance from the robot acts as a lever worsening the error of rotational inaccuracies between the cameras and the kinematic chain.

In robotics, calibration is usually performed by comparing actual sensor measurements with predicted sensor measurements generated from a model given the to be identified

parameters and reference data (e. g. a checkerboard for camera calibration). Thus, the quality of the calibration result is governed by two things. First, the overall structure of the calibration procedure (e. g., static or dynamic) and, second, the ability to generate measurements defining a well-conditioned estimation problem within the procedure.

3.2 Related Work

Camera calibration, introduced to computer vision in Tsai's seminal article [TSAI, 1987] and robustly solved by Zhang [ZHANG, 2000], has become a well understood problem. See [HARTLEY and ZISSERMAN, 2004, SZELISKI, 2010] for a summary of the state of the art techniques. Although it is possible to retrieve the changing pose from a single moving camera (up to a scale factor) and achieve impressive results, adding an IMU is beneficial for two reasons. Either, estimation of the pose using the camera is not the focus and it will be estimated by the IMU solely, or both camera and IMU measurements are used to complement each other for pose estimation. In both cases, geometric calibration between the two sensors is mandatory.

Early work on determining the transformation between a camera and an IMU frame includes the decoupled approach of Lobo and Dias [LOBO and DIAS, 2005, LOBO and DIAS, 2007]. By observing a vertically aligned checkerboard (camera) and gravity (IMU's accelerometer), Horn's method [HORN, 1987] was used to determine the rotational difference between these vertical features and thus the rotational difference between both types of sensors. Translation was then obtained by rotating the IMU's center on a turntable, recording static images of a checkerboard pattern and applying hand-eye calibration techniques. In a dynamic approach [LANG and PINZ, 2005], the rotation was determined using measured visual and inertial rotational differences (assuming no translation) and nonlinear optimization.

In a more elaborated approach [MIRZAEI and ROUMELIOTIS, 2007, MIRZAEI and ROUMELIOTIS, 2008] the full six degree of freedom (DOF) transformation was calibrated simultaneously using an extended Kalman filter. By including time-varying sensor parameters influencing measurements (e. g. gyro bias) and obtaining sensor readings from a setup observing a checkerboard (arbitrarily aligned) in motion, a calibration with a high degree of quality was achieved. On top of that, an observability analysis of the nonlinear camera-IMU calibration system was carried out, revealing that only two of the rotational degrees of freedom need to be excited for successful calibration. Going one step further, Kelly and Sukhatme [KELLY and SUKHATME, 2011] used an Unscented Kalman filter to estimate camera-IMU transformation, metric scene structure and sensor motion without relying on a special calibration object. Further, recent approaches include employment of a passive complimentary filter for rotation estimation under motion [SCANDAROLI et al., 2011] and the determination of the full six DOF calibration by explicit modeling of the trajectory using B-splines and batch optimization [FLEPS et al., 2011].

While classical robotic calibration has the goal of improving the robot's accuracy by identifying deviations in the robot's mechanical structure and using appropriate models in software (please see [ROTH et al., 1987] for a compilation of early approaches and [KLODMANN et al., 2011] for a recent approach), the integration of sensors into robotic systems requires calibration methods to accurately determine the geometric re-

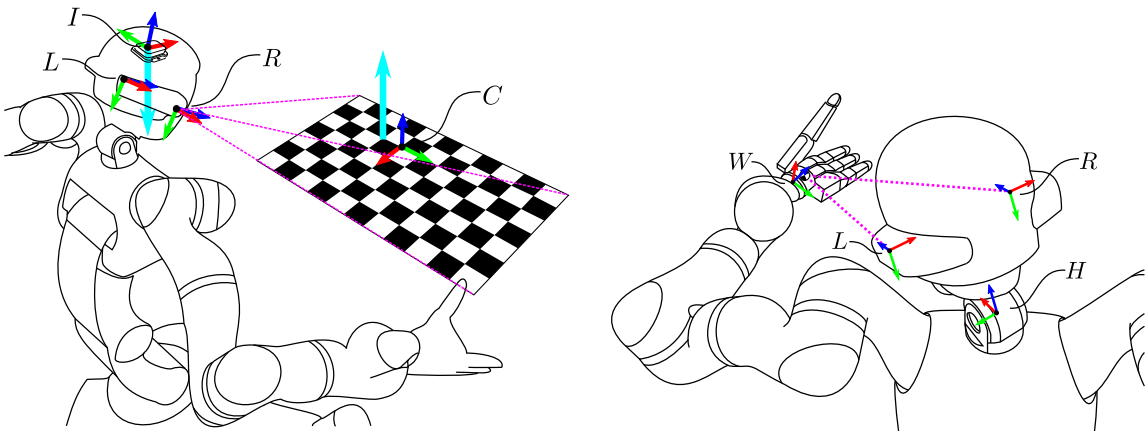
lationship between the actuating and sensing components. In particular, the use of computer vision in robotics raised demand for hand-eye calibration where a visual sensor (usually a camera) is rigidly mounted on a robot's end effector with the goal of calibration being to recover the transformation between these. In general, solutions to this problem are obtained by observing a fixed object from different views in the actuator's workspace. This can be done in many ways, such as sequentially solving the rotational and translational component in a linear least squares manner [SHIU and AHMAD, 1989] or by nonlinear maximum likelihood estimation [STROBL and HIRZINGER, 2006], just to name two.

For the specific case of cameras mounted on moderately actuated platforms (e. g. a pan-tilt unit like it is on the robot's head) the active vision community named this type of calibration head-eye or neck-eye calibration. Obtaining the geometric relationship by observing a reference object was studied by Li and Betsis [LI and BETSIS, 1995, LI, 1998], which is very similar to classic hand-eye calibration. Recovering only the intrinsic parameters was performed through controlled motion [YANG and HU, 1998]. Similarly, the relationship between both geometric and intrinsic camera parameters was recovered in an auto-calibration approach [MA, 1996]. In a more analytical study, Knight and Reid [KNIGHT and REID, 2006] determined the alignment of a camera mounted on a pan-tilt unit by controlled rotation about the actuation axes. In [UDE and OZTOP, 2009] a calibration routine for such an active vision system integrated into the head of the humanoid robot CB-i was provided. The routine consisted of two steps. First, the intrinsic and stereo camera calibration was determined using a checkerboard. Second, the camera-eye relationship was obtained by observing a static checkerboard pattern from different views using the active vision platform.

Besides these head-centric approaches, calibration of more complex robotic systems, such as humanoids, became an active field of research. Garcia [GARCIA, 1999] presented a calibration procedure for the *JANUS* robot prototype to manipulate objects which are perceived by cameras. The calibration routine consisted of three consecutive steps: (1) Calibration of the cameras' intrinsic parameters (but without considering lens distortion) from line and point correspondences and nonlinear optimization; (2) neck-eye calibration by solving the corresponding hand-eye problem (similar as above); and (3), an arm-eye calibration for recovering the relationship between the arms and the camera frame. By placing a visual calibration target at the end effector of each arm the relationship between the arms and the cameras was recovered by applying hand-eye calibration techniques.

For the robot *Robonaut* Nickels [NICKELS, 2003] provided a closed-loop kinematic-visual calibration procedure. Based on predetermined stereo (and intrinsic camera) calibration, a spherical visual feature mounted on a stick was grasped by the robot and observed in a variety of different arm positions. Calibration data was obtained in an automated way and from the resulting joint angle data, visual measurements and the kinematic model the transformation between the chest and the eye coordinate systems was estimated by nonlinear optimization. On top of that, Denavit-Hartenberg (DH) parameters for non-zero components were included to account for kinematic deviations.

Recently, Pradeep et al. [PRADEEP et al., 2010] presented an extensive calibration of Willow Garage's *PR2* robot. Based on *a priori* calibrated cameras, the robot's joint angle offsets and poses of cameras and laser rangefinders were determined. Again, a visual feature, namely a small checkerboard, was gripped by the robot to close the kinematic-visual



(a) (b)

Figure 3.2: The two stages of the static calibration approach. **(a)** In the first stage, the robot observes vertical features (light blue), i. e. accelerometer readings from the IMU I and a leveled checkerboard C, which is also used for intrinsic camera calibration, through the cameras L & R at different body poses (only one shown here). **(b)** In the second stage, the robot observes a point-feature pasted on its wrist W with its two cameras L & R for both arms at different positions enabling hand-eye calibration. Again, only one arm position is shown.

loop. The required data was collected automatically in different arm configurations. Optimization was then done in multiple steps, where in each step the number of parameters was increased to avoid local minima.

3.3 Calibration of a Humanoid Robot's Upper Body

3.3.1 Static Textbook-Style Approach

For the initial operation of the ball catching setup, a simple calibration routine had to be devised. Based on the methods of pair-wise calibration of sensors from literature, an approach consisting of two steps for calibrating relevant parameters of a humanoid's upper body was chosen, as introduced in [8, 12].

The first stage combines classical checkerboard calibration with Lobo and Dias' static approach [LOBO and DIAS, 2005, LOBO and DIAS, 2007] using vertical features, i. e. accelerometer (IMU) readings and a vertical aligned checkerboard (cameras), see Fig. 3.2 (a). This allows calibration of the cameras' intrinsic parameters, stereo offset T_R^L and rotational difference with respect to the IMU R_I^L. The missing translational difference is measured by hand.

The second stage establishes the relationship between the sensor setup and the actuation components. This is done similar to hand-eye calibration approaches. As in [GARCIA, 1999, NICKELS, 2003] a visual feature is placed immediately after the last link along the kinematic chain of each arm, see Fig. 3.2 (b). By locating the feature in the cameras' images, the unknown transformation between the cameras and the last head-link T_H^L is recovered. See Fig. 3.3 (b) for samples of this data (camera images highlighting the visual

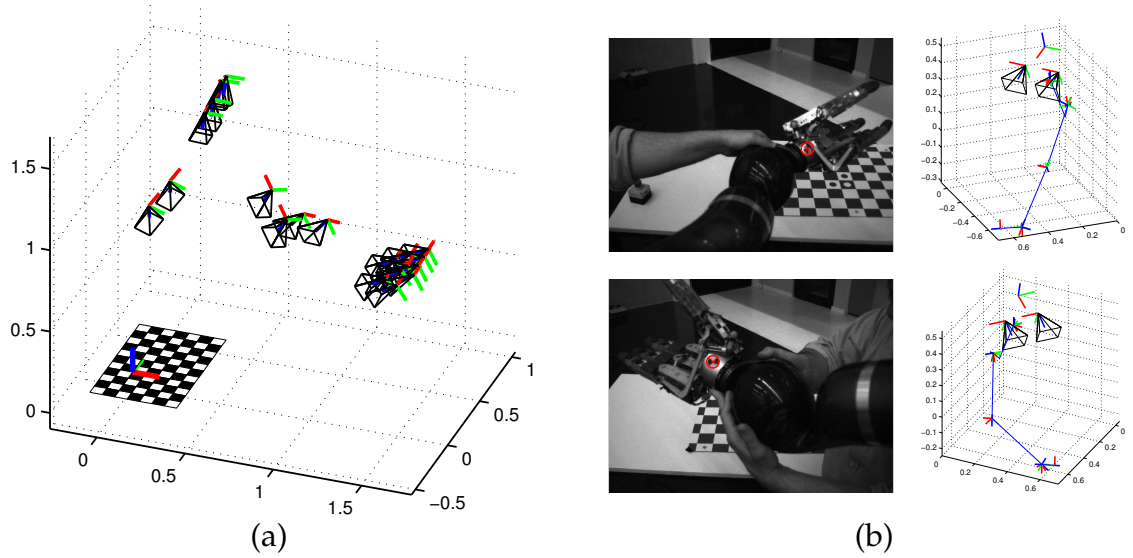


Figure 3.3: **(a)** Illustration of first stage's calibration result showing the checkerboard and the computed extrinsic camera parameters (in m). The cluster on the right corresponds to views of the horizontally aligned checkerboard, while the others belong to freely placed views. **(b)** Example of two arm configurations (as performed in the second stage) observing the visual marker on the robot's hands (left). Corresponding calibration result (in m) including all relevant sensor frames (both cameras, IMU above the cameras) and actuator links as computed by forward kinematics are depicted (right).

feature are depicted on the left). The other of the two unknown transformations (as in traditional hand-eye calibration), is the one from the last arm link to the feature itself. Since a point feature is used, the problem reduces to determine the 3D translation between these.

On top of that, the built-up approach was refined in three ways:

1. Instead of using a vertically aligned checkerboard as originally proposed, a horizontally aligned one is employed. This increases robustness in the estimation of vertical direction: Both DOF available are now employed when observing verticality using a horizontally aligned checkerboard instead of only one DOF through a vertically aligned checkerboard.

2. By using the checkerboard pattern not only for camera-IMU rotation R_I^L estimation but also for stereo calibration T_R^L, both initially separate procedures can now also be combined during estimation. Here, Lobo and Dias' approach [Lobo and Dias, 2007, Lobo and Dias, 2005] together with Zhang's method [Zhang, 2000] is used to obtain the initial guess for an iterative nonlinear optimization.

3. Contrary to previous work [Pradeep et al., 2010, Garcia, 1999, Nickels, 2003], where calibration is performed as a sequence of estimations, the calibration here is performed by joint estimation of all parameters. This avoids propagation of errors from one stage to the other. Furthermore, any parameters that are influenced by two or more classes of measurements (e. g. the camera's intrinsic parameters are determined by the checkerboard and by the hand-eye calibration) make use of *all* data at once for estimation.

Table 3.1: Calibration Results of the Static Textbook-Style Approach

	Intrinsic left			Intrinsic right		
	f_L (px)	C_L (px)	κ_L	f_R (px)	C_R (px)	κ_R
μ	1872.6	857.6,618.5	0.09	1862.6	813.3,615.6	0.10
σ	0.57	0.67, 0.75	$6.7 10^{-4}$	0.56	0.37, 0.73	$7.0 10^{-4}$

	Transformation T_R^L					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	-0.200	0.002	-0.005	0.012	-0.002	-0.009
σ	0.0001	0.0001	0.0004	0.0004	0.0004	<0.0001

	Transformation T_H^L					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	0.062	0.104	0.129	-1.328	1.315	-1.142
σ	0.0005	0.0007	0.0006	0.0007	0.0007	0.0009

	Transformation T_I^L					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	0.0	0.0	0.0	-1.342	1.332	-1.159
σ	n/a	n/a	n/a	0.0037	0.0010	0.0102

Estimation of parameters is usually done using (nonlinear) least squares optimization. When confronted with such a kind of calibration problem, nontrivial difficulties may arise. First, depending on the structure and scale it might become complicated to encode the problem by hand. This is especially true when parameters have impact on different kinds of measurements (as mentioned in the third refinement) or when the sparse structure of the Jacobians, which are required for nonlinear least squares, should be exploited (e. g. for computational efficiency). Second, when non-vectors, such as singularity free parametrization of 3D rotations (e. g. rotation matrices in $SO(3)$), become part of the parameters, special care must be taken while optimizing them.

For this type of calibration a newly developed framework for solving least squares problems, called the Manifold Toolkit for MATLAB (MTKM) [12], is employed. It provides an interface that allows easy setup of problems by defining measurement functions and the parameter variables the functions depend on. After feeding the measurements into the framework, the structure of the overall problem is detected and utilized in the optimization phase to obtain rapid results. Furthermore, besides the ability to use regular (Euclidean) vectors as parameters, it provides types for handling rotations in a singularity free way. By providing a local vector view of a manifold type, classical optimization algorithms assuming regular vectors, such as Levenberg-Marquardt, can be employed.

Despite this advanced framework, actual convergence is dependent on the input data and whether or not they represent a well-conditioned optimization problem. Because all sensors are mounted onto the head, sensor movement and therefore the possibility to generate varying views is limited. It is expected that this limited view of the cameras observing the horizontally aligned checkerboard impairs the estimation of the intrinsic camera parameters. Therefore, a certain number of freely aligned checkerboard images was added. See Fig. 3.3 (a) for a graph showing the camera poses typically used, including the two types of checkerboard views. All data was acquired with human assistance. The freely aligned checkerboard had to be placed manually in front of the robot while the horizontally aligned one had to be manually leveled beforehand. Arm configurations in the second stage were acquired through manual positioning (see Fig. 3.3 (b), left images).

Results of such a calibration (from 11 leveled and 7 freely observed checkerboards, 21 views of both arms) are given in Fig. 3.3 depicting the different camera views and two arm configurations, and in Tab. 3.1 listing actual parameter values. Here, the obtained values and the corresponding 1σ bound are given as reported by the estimator. The required measurement uncertainties were determined in a prior combined estimation routine. Analysis reveals that camera parameters, stereo transformation T_R^L, and their

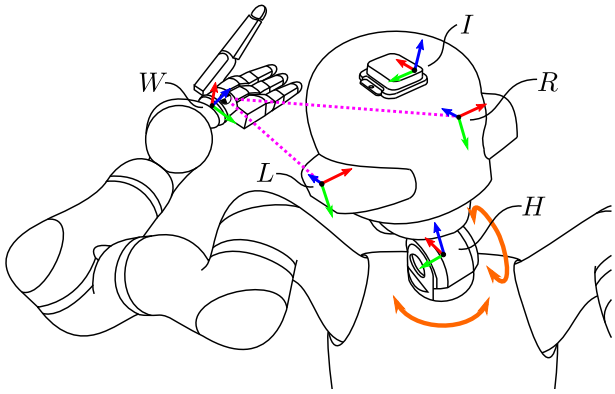


Figure 3.4: Sketch of the automatic self-contained calibration showing DLR's Rolling Justin from the rear including the relevant frames. A visual point-feature pasted on the robot's wrist W is observed (magenta dotted line) by its two cameras (L & R) while moving the head H. Furthermore, measurements from the IMU I and the robot's joint angle and torque sensors are logged. This allows poses of the left camera and the IMU to be calibrated with respect to the head frame, T_I^L and T_H^I respectively. Also, stereo T_R^L, the cameras' intrinsic parameters and joint angle offsets and elasticities of the arms' joints are obtained at the same time.

transformation with respect to the head T_H^L are recovered with a high degree of precision, while the rotation between them and the IMU R_I^L is poorly determined. Combined with manually obtained translation, the IMU's relation to other components poses a weak link in the setup. This precludes rapid head movement (e. g. when the robot follows the ball visually) as it is expected to cause missed catches.

The calibration method given above is part of the examples showing the use of the optimization toolbox MTKM for calibration purposes. Please cf. App. A.3 for details.

3.3.2 Automatic Self-Contained Approach

Based on the experiences from the above mentioned approach and from further analysis of prior literature the calibration of a humanoid's upper body is a rather complex task. Different components contributing to this complexity can be identified:

- **Multiple calibration stages** driven by the use of pair-wise calibration techniques are common. As a consequence, sequential estimation of parameters is used, making the overall estimation results inconsistent.
- **External tools**, such as the checkerboard or the level presented in the method above, are often part of the calibration routine. This leads to less flexibility as these tools have to be carried along when the robot is moved to a different location (e. g. other lab, trade fair). This is especially true for mobile robots.
- **Human assistance** might be required. This is not only the case while obtaining the calibration measurements but also during processing of the data (e. g. extracting features in images manually). Furthermore, negligence of the operator might limit the quality of the calibration result.

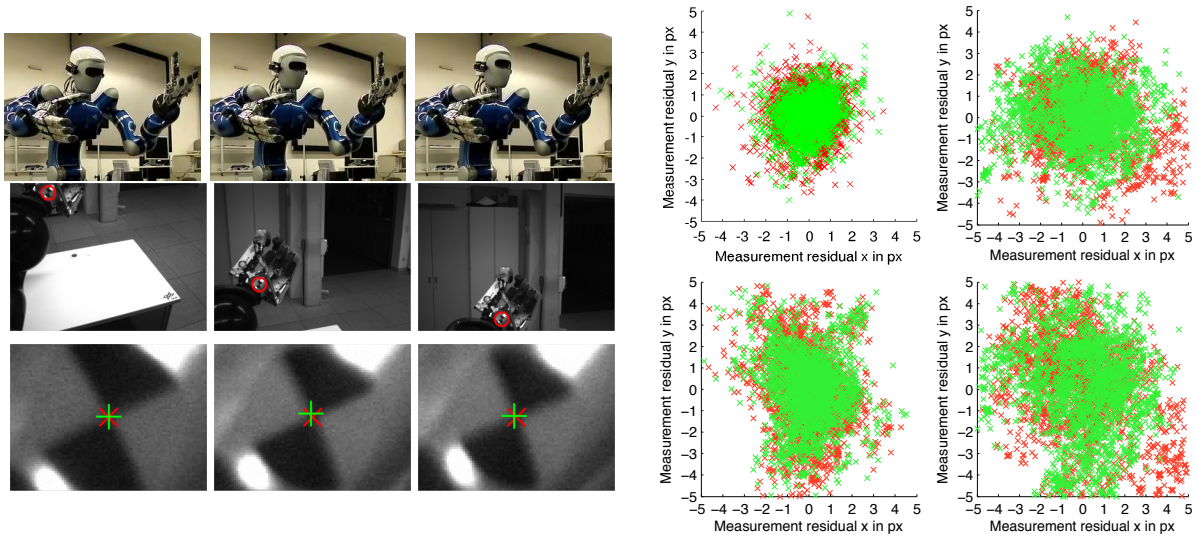
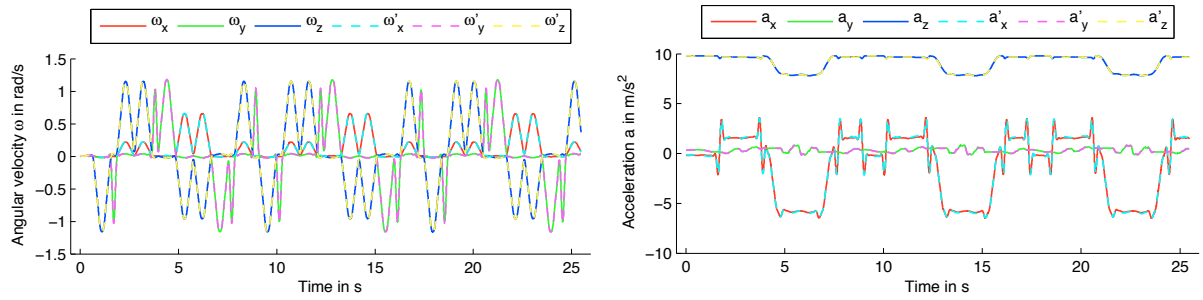


Figure 3.5: **(a)** Snapshots from a calibration process. (Top) External view of calibration procedure observing one arm while moving the head. (Middle) Corresponding view from the left camera. The point feature is highlighted with a red circle. (Bottom) Close-up picture of feature and detected center (red cross $\times$) and predicted center (green cross $+$). **(b)** Vision measurement residuals (left camera depicted by red crosses, right camera by green crosses) from estimations considering all (top left), all but joint angle deviations (top right), all but joint elasticities (bottom left) and all but offsets and elasticities (bottom right) components from the complete set of calibration parameters.

To alleviate these issues, a new calibration approach was developed in [4]. By using the robot's automation abilities all data is recorded by the robot itself in *one* run. Furthermore, only *one* model fitting stage including all models is used where all data are processed at the same time for the corresponding parameters that are part of the calibration process (as already performed in the previous approach). During recording, *no* external tools are employed other than the robot itself or features related to itself. The only human interaction required is *one* button press to initiate the calibration procedure.

In detail, the developed method records two kinds of data automatically (see also Fig. 3.4 for an illustrated explanation).

First, camera images and joint angle positions while the head is observing a visual feature on the robots arm at different arm postures are recorded. Instead of capturing a checkerboard from one view, a single point feature is now registered from many views differing in head rotation, see Fig. 3.5 (a). The scale is not defined through the spacing of the features, but through the robot's forward kinematics. Although not realized here, such an approach would allow using robot properties (e. g. the distance of a stretched out arm) instead of explicit physical units for scale. One drawback of the point feature approach is that the limited workspace does not allow the capture of distant points. Furthermore, as the robot's head movement is only rotational, the distance of the observed feature with respect to the cameras is roughly the same. These two limitations make it hard to distinguish between focal length $f_{L/R}$ and object distance during camera calibration. To compensate for this, several different arm positions are recorded and observed



(a) (b)

Figure 3.6: Angular velocity **(a)** and linear acceleration **(b)** as measured by the IMU (ω, a) and predicted by the model (ω', a') using the measured head joint angles over time.

from most of the image space of the camera during head movements. This also allows determination of joint angle offsets and joint elasticities in the estimation process to account for kinematic deviations. For this, not only the joint angle positions but also the measured torque signal is recorded.

Second, IMU sensor readings and joint angle positions are logged while the head is moving. Here, the corresponding measurement function simply differentiates motion over time to compute angular velocity and linear acceleration that should be measured by the IMU. In this dynamic approach it is now possible to not only recover its orientation but also its position in the robot's head. Please note, that now T_H^I is estimated and the desired parameter T_I^L is now determined indirectly from T_H^L (i. e. $T_I^L = T_H^{I-1} T_H^L$).

The automated recording process actuates around one dozen configurations (taught in advance) for each arm which corresponds to 5 minutes of actual movement of the robot. The maximum velocity of the head movement is limited to $24°/s$ to avoid motion blur in the camera images. While this movement could also be used to perform IMU calibration, a special run of head movement with higher velocities ($60°/s$) is performed. This invokes higher accelerations and angular velocities and allows making use of the complete workspace of the robot's pan tilt unit. To accommodate for structural vibrations, the IMU and joint angle signal are low pass filtered with a cutoff frequency of 5 Hz. Because different sensor measurements are compared during motion, the temporal relationship between these has to be established before the geometric calibration can be performed. This has to be done only once and has been determined by hand in advance, but could also be done automatically by, e. g. correlation, in the future.

Estimation is conducted using MTKM. Once the recorded data is fed into the framework, all parameters are estimated concurrently. Analysis is again performed by reviewing the estimator's output. An analysis of the visual marker residuals is given in Fig. 3.5 (b). It can be seen that the inclusion of both joint angle offsets and joint elasticities as parameters considerably helps the consistency of the residuals. Excluding the joint angle offset leads to inaccurate wrist positions in different arm configurations visible as increased noise, while not considering joint angle elasticities leads to a dominant displacement in the cameras' y-axis caused by the gravity dragging the arm down.

Figure 3.6 compares actual and predicted IMU measurements given the estimated parameters and the measured joint angles showing no substantial deviations. Actual parameter values are listed in Tab. 3.2, where the obtained estimate μ and the corresponding 1σ

Table 3.2: Calibration Results of the Automatic Self-Contained Approach

	Intrinsic left			Intrinsic right		
	f_L (px)	C_L (px)	κ_L	f_R (px)	C_R (px)	κ_R
μ	1869.4	839.7,619.5	0.10	1860.8	817.3,619.3	0.10
σ	0.53	0.54, 0.66	$7.2 10^{-4}$	0.52	0.58, 0.66	$7.6 10^{-4}$

	Transformation T_R^L					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	-0.201	0.001	-0.002	0.012	-0.006	-0.009
σ	0.0001	0.0001	0.0003	0.0005	0.0004	0.0001

	Transformation T_H^L					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	0.066	0.100	0.130	-1.329	1.316	-1.127
σ	0.0002	0.0001	0.0001	0.0001	0.0003	0.0002

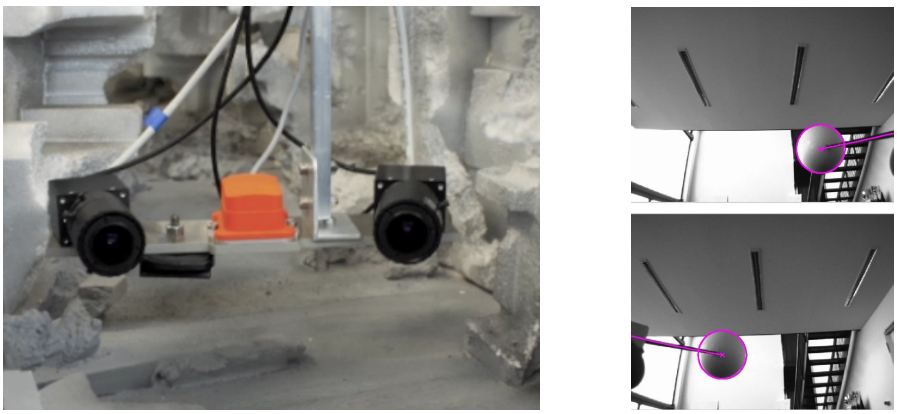
	Transformation T_H^l					
	Translation (m)			Rotation (axis angle)		
	x	y	z	x	y	z
μ	-0.002	0.006	0.236	0.026	-0.014	0.006
σ	0.0007	0.0012	0.0009	0.0006	0.0005	0.0007

bound are given (the required uncertainty of the measurements where determined in a prior estimation). Fortunately, camera intrinsic parameters and stereo accuracy results are comparable to the static checkerboard method (which is considered best practice for this type of problem) from the previous calibration routine. Also, hand-eye calibration is improved benefiting from the ability to generate a large amount of single point data. Most interestingly, the location of the IMU within the robot's head is now determined accurately not only for rotational component but also for the translation.

Since initial publication of the calibration routine in [4], it has been improved in two ways. First, due to the use of a local visual marker detector, only moderate deviations were allowed for the calibration routine such that the wrist-mounted feature is reliably detected in every camera image. A global feature search is now employed based on gradient features. For this the HOG (histogram of oriented gradients) descriptor [DALAL and TRIGGS, 2005], trained from detections of earlier calibration runs, was found to be suitable for detecting the distinctive pattern of the marker. Second, MATLAB's general overhead in evaluating functions combined with MTKM's property of evaluating Jacobians numerically leads to inefficient computation. The estimation was therefore ported to SLoM [HERTZBERG et al., 2013], a framework with the same functionality that shortened the optimization time considerably. In fact, this allows performing a calibration of the robot in less than ten minutes, including recording of all arm configurations as well extracting the visual features from the camera images. Besides the fact that all parameters are now properly determined, this calibration method is a tremendous improvement in time and human assistance when compared to the prior static calibration method.

3.3.3 Designing Optimal Calibration Experiments

As mentioned in the motivation, the quality of the calibration result depends on the obtained measurements. A common approach to determine which measurements to obtain is to design the experiments in a way that the variance as reported by the underlying estimator becomes minimal, a technique known as *optimal design*. For the specific case it is of interest how to select the set of arm configurations such that calibration time (i. e. the number of configurations) and estimator variance (from the information matrix after convergence) becomes minimal. In a recent paper [10] experiments for an optimal calibration design were carried out. The major contribution is the proposal of a *task oriented* selection criterion which encodes the mean squared error of the robot's TCP (tool center point) from the covariance reported by the estimator. This is contrary to other common



(a) (b)

Figure 3.7: **(a)** Stereo-IMU system used for experimental evaluation of the visual SLAM system built from open source components which was calibrated using the method introduced in Sec. 3.3.1. **(b)** Two images from the cameras of the entertainment robot, see Sec. 1.4, during the calibration procedure. The sphere at the end of the bat was used as the calibration feature and the modeled bat is shown as an overlay in the image to verify calibration results.

criteria which combine components of the covariance matrix resulting in a metric which has no physical interpretation. Hence the emphasis is on task, as it directly corresponds to the desired behavior of precise manipulation.

Using this criterion a greedy algorithm was devised which generates an optimized set from a large set of randomly sampled configurations. Although this greedy algorithm is subject to sub-optimal selection of configurations, experimental evaluation shows that this is still considerably better than the taught positions used initially.

3.4 Further Calibration Problems

The calibration methods for the task of robotic ball catching have also been applied to other calibration problems. The stereo-IMU system depicted in Fig. 3.7 (a) used for experimental evaluation of a visual SLAM system built from open source components [11] was calibrated using the static approach from Sec. 3.3.1. Similarly, the calibration between the actuator and the vision system of the entertainment robot introduced in Sec. 1.4 was performed in the same manner as proposed in Sec. 3.3.1, but instead of using a point feature, the sphere at the end of the bat was used as a feature, see also Fig. 3.7 (b). It would also be possible to integrate the detection of the sphere into the calibration process similar to the proposed tracking algorithm introduced in Sec. 2.7. This is considered future work.

Furthermore, two routines for dealing with conceptually different calibration problems have been proposed and will be presented concisely.

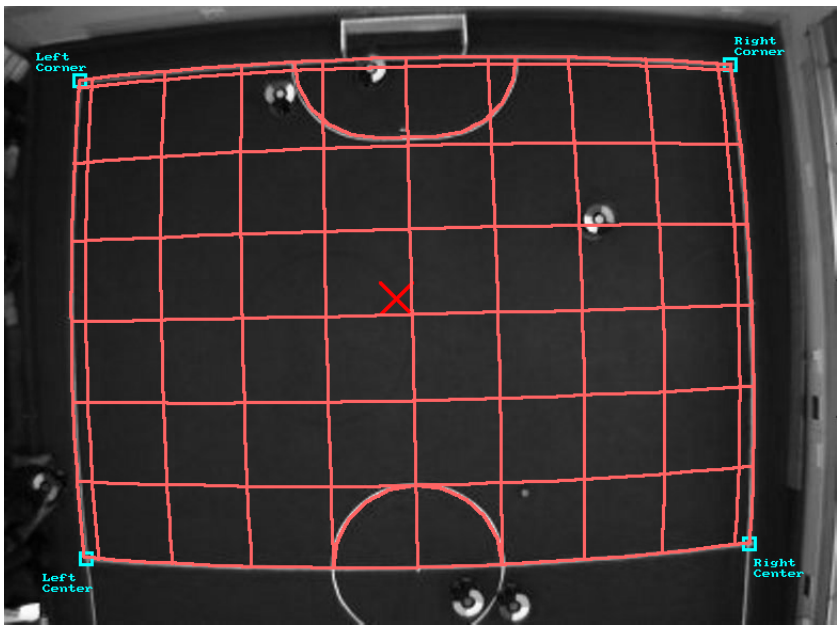


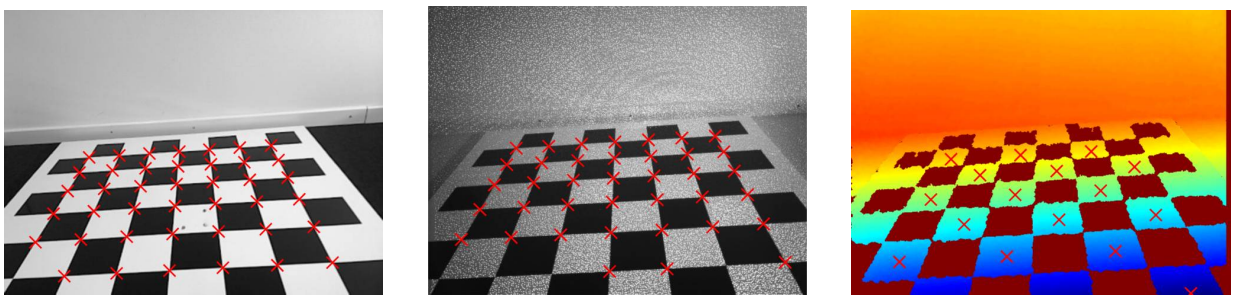
Figure 3.8: Result of camera calibration using SSL-Vision. The field lines, the principal point (as a cross) and a grid are projected to the image plane for visual inspection. The labeled boxes depict the manually selected field corners required for the initialization of the optimization routine.

3.4.1 SSL-Vision Calibration

RoboCup's Small Size Robot League (SSL) distinguishes itself from the other leagues in that it allows the participating teams to use global vision. In practice, every team set up their own hardware and developed their own computer vision software for recognizing robots and the ball on the field. Despite this expanded organizational effort, the software converged mostly to the same set of methods. The organizers therefore proposed a shared vision system, mandatory for all teams since 2010. This software, *SSL-Vision* [13], was introduced and combined best practice methods.

As a part of this software, a new calibration technique which determines the camera's intrinsic and extrinsic parameters with respect to the playing field (see Fig. 3.8) was developed. In the previous approaches of the individual teams, it was common to use special calibration tools, such as a foldable checkerboard pattern spread on the field. This might not only lead to inaccurate and error-prone procedures but is also costly in terms of time and blocks the field for other teams. In the new approach, the known dimensions of the field's line features which are defined in the rules and the easily measurable height of the camera have been used. Usually, as each half of the field is observed by one camera, each camera's image provides sufficient features which is important for a well-defined solution, especially regarding the pose of the cameras.

Initialized by defining the corners of the field through the user and a rough estimate of the camera height, edge detection is used to find field lines in the images. From these and the known field dimensions, a least squares problem is formulated which is solved by using the Levenberg-Marquardt algorithm, enabling fast and accurate calibration. Please see Fig. 3.8 for a visualization of the calibration result.



(a) (b) (c)

Figure 3.9: Kinect sensor output including feature points (highlighted as red crosses) employed for calibration: **(a)** RGB (already converted to grayscale), **(b)** infrared, **(c)** disparity image. For the latter, the data on the black squares was missing and the center of the white ones was used instead.

Besides its designated use in the Small Size Robot League, SSL-Vision is also used in other RoboCup league's as a ground-truth system for evaluation of self-localization algorithms, see Laue [LAUE and RÖFER, 2009, LAUE et al., 2009] and Burchardt et al. [BURCHARDT et al., 2011], and is even used as the basis of a global vision based humanoid league as proposed by Naruse et al. [NARUSE et al., 2011].

This calibration routine has been released as open source as part of the SSL Vision project. Please see App. A.1 for more information.

3.4.2 Kinect Calibration

Recently, combining color (RGB) and depth information gained a lot of attraction. For example, Du et al. [DU et al., 2011] used a Microsoft Kinect/PrimeSense sensor for registration in indoor environments resulting in detailed and accurate 3-D models including texture. Furthermore, Lai et al. [LAI et al., 2011] employed the same type of sensor for object categorization and recognition, concluding that integrating both geometric and visual features from depth and color sensor achieves higher performance than using one of the cues alone. Similarly, Spinello and Arras [SPINELLO and ARRAS, 2011] proposed a detector, named Combo-HOD, that combines depth and color data for the task of tracking people, outperforming traditional vision based detectors.

As with most setups involving multiple (possibly different) sensors, a calibration of these sensors themselves and between them is mandatory. In [12], a calibration for such a setup was presented, namely for a Microsoft Kinect sensor.

For this, the method for jointly calibrating a stereo-IMU setup from Sec. 3.3.1 above was employed (without the need to calibrate these sensor relative to robot). This approach defines the intrinsic parameters of the RGB and infrared (IR) cameras, their geometric displacement (stereo) and the rotation from the cameras to the integrated accelerometer (see Fig. 3.9 (a) and (b) for images of the observed leveled checkerboard pattern). The parameters defining the mapping from depth to disparity are estimated using well perceived points in the disparity image of the checkerboard (Fig. 3.9 (c)) and the depth to the board available from the extrinsic parameter estimate. Again, the Manifold Toolkit for MATLAB (MTKM) was used for modeling the problem and performing calibration.

The developed calibration routine has been released as open source and is included with the release of MTKM, as described in App. A.3.

3.5 Summary and Transferability

This chapter presented two different approaches for calibrating the upper body of a humanoid robot and its mounted sensors. Calibration is considered a precondition for the task the robot was initially designed for and is required due to maintenance, collision or wear over time.

The first approach [8, 12] relied on methods from literature where static measurements acquired from manual positioning were taken. Although a calibration sufficient for robotic catching is generally achieved the approach lacks definition of certain parameters (i. e. translation with respect to IMU) due to its static design. Furthermore, the manual positioning made the calibration costly in terms of time and human resources.

An automatic calibration was therefore proposed in [4]. All necessary data was collected through automated motion, where the moving head observes different configurations of the robot's arm. The ensuing preprocessing and optimization stage are also automated, resulting in a *one button press* calibration. All degrees of freedom of the calibration are defined and an accurate calibration is obtained.

For both of these calibrations MTKM [12] was employed for the task of estimating the calibration parameters from measurement functions defined in the model and the actual measurement data. Due to MTKM's design stating such kind of calibration problems is simplified, as one has only to link measurements and parameters through measurement functions, and let the framework do the rest.

Furthermore, initial experiments for an optimal calibration design [10], the calibration of a stand-alone stereo-IMU system for a visual SLAM system built from open source components [11] and the calibration of the entertainment robot were reported. Finally, two additional calibration problems were introduced in slightly more detail. First, the calibration of a RoboCup soccer field with global vision by using the field lines as features [13], and second, a calibration of the Kinect sensor parameters using the MTKM framework [12].

As for transferability, both presented humanoid calibration routines are not limited to the used robot in the experiments mentioned in this dissertation. In fact, the only notable component that is subject of adaption is, obviously, the forward kinematics of the robot the calibration is intended for. As the interface of forward kinematics is well defined, replacing this should be considered a rather straightforward modification regarding the underlying model for calibration. This is also true for the used visual feature to obtain the hand-eye relationship. By replacing the corresponding measurement function the calibration routine can be matched to use the features proposed in [PRADEEP et al., 2010, GARCIA, 1999, NICKELS, 2003]. Furthermore, because of the flexibility and extendability of the underyling least squares framework (MTKM) the number of sensors can be adjusted with little effort to the additionally installed hardware of the designated robot, such as Kinect sensors or laser range finders.

Chapter 4

Conclusion

This thesis has presented insights into a computer vision system for the dynamic task of ball catching using a humanoid robot. In detail, insight into two components for making a robot successfully locate a ball in its environment was given, namely tracking of a thrown ball and calibration of a humanoid's sensing and actuating components.

Based on sets of discrete circle detections and head position estimation from IMU measurements, it was shown how established techniques (MHT) and recent advances (GM-PHD) from the tracking community combined with a UKF can be employed for this real-time task. Compared to these recursive tracking approaches, a new global tracking algorithm was presented: By leveraging the parabolic motion as context to evaluate the ball's states using the filter response directly from the image, an improved prediction accuracy is achieved. This is especially true at the beginning of the trajectory, which is a critical phase for successful catching.

For the corresponding calibration, an automated and self-contained approach for determining the relation between a variety of sensors on a humanoid's upper body was presented. This procedure consists of one automated data recording run followed by one framework-based model fitting process, employs no external tools such as a checkerboard and is initiated by a single button press. Compared to previous calibration routines, this eliminates the need for human interaction and increases availability of the robot for its originally intended purpose.

The presented work gained respectable attention with two publications becoming award finalists at major robotic conferences and presentation of the developed methods (except the new tracking method) at the Automatica (2010 and 2012) and CeBit 2012 trade fairs, public institute events and numerous lab demonstrations. This is mainly due to the robot performing a benchmark task giving non-specialists the ability to compare the achievements with human performance and to assess the current state of the art in robotics research. Therefore, besides the aforementioned technical contributions the outcome of this work also includes an exhibiting component.

The potential of the proposed tracking algorithm is promising due to the integration of a physics-based motion model in a global fashion. It turns out that this integration of motion as context in detection-based tracking is a powerful tool, and it is assumed that this approach is the way to go when confronted with the task of precise tracking using vision. While this can be generalized to other tracking instances, the efficiency of the algorithm is mainly governed by the stiffness of the underlying motion, as this limits the search space and therefore trajectory convergence. Therefore, it has yet to be determined, how much this motion can be relaxed such that the proposed tracking method still performs

as intended. This is especially true for one important class of motion, namely articulated human motion, where tracking over time considerably helps in resolving kinematic ambiguities and occlusions. Based on a limb/skeletal model, feasible scenarios include tracking short-term regular movements (e. g. walking, waving or mixing dough by hand) and simple cooperative tasks, such as passing a bowl, that require accurate prediction for a responsive behavior of the robot.

While the developed approach is efficient in terms of time and human resources, calibration is still regarded a dedicated task performed only when an inconsistency is recognized. This separation from the main task of the robot is mostly due to reserving the (often limited) computational resources for proper operation. This might be a valid solution for research robots, where calibration is overseen by an expert after the robot undergoes maintenance. Unfortunately, once autonomous service robots have gained maturity and start operating in real world settings, these robots are likely to break, witness an unexpected change in structure or experience wear *during* regular operation. This naturally raises the question: Why not integrate calibration into regular operation? This is not too far-fetched, as a human's cerebral cortex possesses the ability to adapt the motor signal while the body is subject to long-term changes, e. g. when growing during infancy. In fact, this continuously observation of the robot components contributes to the goal of long-term autonomy and, from a safety standpoint, allows the robot to detect any severe defects preventing potentially harming situations, either for itself or for its environment, at an early stage. Furthermore, such an integrated procedure benefits economical operation of autonomous systems.

One can even combine tracking and calibration, using one unified representation so that future robots are not only aware of their surroundings but also of themselves in a consistent manner.

List of Publications by the Author

Reviewed Publications

In Scientific Journals

[1] FRESE, UDO, TIM LAUE, OLIVER BIRBACH and THOMAS RÖFER: *(A) Vision for 2050 – Context-Based Image Understanding for a Human-Robot Soccer Match*. Electronic Communications of the EASST, to appear.

My share is 15%.
This paper proposes and discusses a novel way of image understanding for robotic soccer. I contributed a review of related work and a discussion of work that coincides with my activity in robotic ball catching.

At Peer-Reviewed Conferences

[2] BÄUML, BERTHOLD, OLIVER BIRBACH, THOMAS WIMBÖCK, UDO FRESE, ALEXANDER DIETRICH and GERD HIRZINGER: *Catching Flying Balls with a Mobile Humanoid: System Overview and Design Considerations*. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pages 513–520, Bled, Slovenia, 2011. Supplementary video available at IEEExplore.

My share is 25%.
This paper is an overview of the complete robotic ball catching system. I provided a section that summarizes the computer vision portion of the system including results and insights.

[3] BÄUML, BERTHOLD, FLORIAN SCHMIDT, THOMAS WIMBÖCK, OLIVER BIRBACH, ALEXANDER DIETRICH, MATTHIAS FUCHS, WERNER FRIEDL, UDO FRESE, CHRISTOPH BORST, MARKUS GREBENSTEIN, OLIVER EIBERGER and GERD HIRZINGER: *Catching Flying Balls and Preparing Coffee: Mobile Humanoid Rollin' Justin Perfoms Dynamic and Sensitive Tasks*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3443–3444, Shanghai, China, 2011. Video available at IEEExplore. **Best Video Award – Finalist.**

My share is 20%.
This contribution is accompanied by a video showing the results of work in robotic ball catching and coffee making. I provided the visualization of the computer vision results.

[4] BIRBACH, OLIVER, BERTHOLD BÄUML and UDO FRESE: *Automatic and Self-Contained Calibration of a Multi-Sensorial Humanoid's Upper Body*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3103–3108, Saint Paul, USA,

2012. Supplementary video available at IEEExplore. **Best Vison Paper Award – Finalist.**

My share is 80%.
I implemented the majority of the software, co-conducted the experiments with an actual robot and evaluated the results. The work was presented by me in Saint Paul, USA.

[5] BIRBACH, OLIVER and UDO FRESE: *A Multiple Hypothesis Approach for a Ball Tracking System*. In FRITZ, MARIO, BERNT SCHIELE and JUSTUS H. PIATER (editors): *Computer Vision Systems*, volume 5815 of *LNCS*, pages 435–444, 2009.

My share is 70%.
I implemented the tracking portion with respect to real-time behavior and conducted and evaluated outdoor experiments. Results of this work were presented by me in Liege, Belgium.

[6] BIRBACH, OLIVER and UDO FRESE: *Estimation and Prediction of Multiple Flying Balls Using Probability Hypothesis Density Filtering*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3426–3433, San Francisco, USA, 2011. Supplementary video available at IEEExplore.

My share is 90%.
I implemented the proposed tracking algorithm, compared it to the previously used one and conducted lab experiments. This work was presented by me in San Francisco, USA.

[7] BIRBACH, OLIVER and UDO FRESE: *A Precise Tracking Algorithm Based on Raw Detector Responses and a Physical Motion Model*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013.

My share is 90%.
I contributed parts of the probabilistic model, the complete implementation and evaluated the proposed tracker using experimental data.

[8] BIRBACH, OLIVER, UDO FRESE and BERTHOLD BÄUML: *Realtime Perception for Catching a Flying Ball with a Mobile Humanoid*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5955–5962, Shanghai, China, 2011. Supplementary video available at IEEExplore.

My share is 70%.
I contributed the tracking part and integrated the complete computer vision on the robot's computing hardware, lab experiments were conducted and evaluated using ground truth. Furthermore, I presented this work in Shanghai, China.

[9] BIRBACH, OLIVER, JÖRG KURLBAUM, TIM LAUE and UDO FRESE: *Tracking of Ball Trajectories with a Free Moving Camera-Inertial Sensor*. In IOCCHI, LUCA, HITOSHI MATSUBARA, ALFREDO WEITZENFELD and CHANGJIU ZHOU (editors): *RoboCup 2008: Robot Soccer World Cup XII. RoboCup International Symposium*, volume 5399 of *LNCS*, pages 49–60. Springer, 2009.

My share is 90%.
This work presents the results of my diploma thesis. It is included for the sake of completeness as it provides valuable exploratory work regarding calibration and tracking.

[10] CARRILLO, HENRY, OLIVER BIRBACH, HOLGER TÄUBIG, BERTHOLD BÄUML, UDO FRESE and JOSÉ A. CASTELLANO: *On the Criteria for Configurations Selection in Robot Calibration*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013.

My share is 20%.
This is based on my prior work on humanoid robot calibration. I assisted during design of the proposed method, contributed code and co-conducted the experiments on the robot for evaluation.

[11] HERTZBERG, CHRISTOPH, RENÉ WAGNER, OLIVER BIRBACH, TOBIAS HAMMER and UDO FRESE: *Experiences in Building a Visual SLAM System from Open Source Components*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2644–2651, Shanghai, China, 2011. Supplementary video available at IEEExplore.

My share is 10%.
I co-implemented the technique for calibrating cameras and an IMU in a joint fashion using open source software.

[12] WAGNER, RENÉ, OLIVER BIRBACH and UDO FRESE: *Rapid Development of Manifold-Based Graph Optimization Systems for Multi-Sensor Calibration and SLAM*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3305–3312, San Francisco, USA, 2011.

My share is 25%.
I contributed all calibration cases and prepared them for consideration in the paper. Additionally, I made minor contributions to the proposed software framework.

[13] ZICKLER, STEFAN, TIM LAUE, OLIVER BIRBACH, MAHISORN WONGPHATI and MANUELA VELOSO: *SSL-Vision: The Shared Vision System for the RoboCup Small Size League*. In BALTES, JACKY, MICHAIL G. LAGOUDAKIS, TADASHI NARUSE and SAEED SHIRY (editors): *RoboCup 2009: Robot Soccer World Cup XIII. RoboCup International Symposium*, volume 5949 of *LNCS*, pages 425–436. Springer, 2010.

My share is 25%.
I proposed and co-developed the new calibration method described in this work.

References

[ANDERSON and MOORE, 1979] ANDERSON, BRIAN D. O. and J. B. MOORE (1979). *Optimal Filtering*. Prentice-Hall Information and System Sciences Series. Prentice-Hall.

[ANDERSSON, 1989] ANDERSSON, RUSSELL L. (1989). *Dynamic Sensing in a Ping-Pong Playing Robot*. IEEE Transactions on Robotics and Automation, 5(6):728–739.

[ANDRIYENKO and SCHINDLER, 2011] ANDRIYENKO, ANTON and K. SCHINDLER (2011). *Multi-Target Tracking by Continuous Energy Minimization*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1265–1272, Colorado Springs, USA.

[ANDRIYENKO et al., 2012] ANDRIYENKO, ANTON, K. SCHINDLER and S. ROTH (2012). *Discrete-Continuous Optimization for Multi-Target Tracking*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1926–1933, Providence, USA.

[ARMENTI, 1992] ARMENTI, JR., ANGELO, ed. (1992). *The Physics of Sports*. Springer.

[ARULAMPALAM et al., 2002] ARULAMPALAM, M. SANJEEV, S. MASKELL, N. GORDON and T. CLAPP (2002). *A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing, 50(2):174–188.

[BÄUML and HIRZINGER, 2008] BÄUML, BERTHOLD and G. HIRZINGER (2008). *When Hard Realtime Matters: Software for Complex Mechatronic Systems*. Robotics and Autonomous Systems, 56(1):5–13.

[BÄUML et al., 2010] BÄUML, BERTHOLD, T. WIMBÖCK and G. HIRZINGER (2010). *Kinematically Optimal Catching a Flying Ball with a Hand-Arm-System*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2592–2599, Taipei, Taiwan.

[BEETZ et al., 2006] BEETZ, MICHAEL, J. BANDOUCH, S. GEDIKLI, N. VON HOYNINGEN-HUENE, B. KIRCHLECHNER and A. MALDONADO. (2006). *Camera-based Observation of Football Games for Analyzing Multi-agent Activities*. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 42–49, Hakodate, Japan.

[BEETZ et al., 2007] BEETZ, MICHAEL, S. GEDIKLI, J. BANDOUCH, B. KIRCHLECHNER, N. V. HOYNINGEN-HUENE and A. PERZYLO (2007). *Visually Tracking Football Games Based on TV Broadcasts*. In *Proceedings of the International Joint Conference on Artifical Intelligence*, pp. 2066–2071, San Francisco, USA.

[BENTLEY, 1984] BENTLEY, JON (1984). *Programming Pearls: Algorithm Design Techniques*. Communications of the ACM, 27(9):865–873.

[BLACKMAN, 2004] BLACKMAN, SAMUEL S. (2004). *Multiple Hypothesis Tracking For Multiple Target Tracking*. IEEE Aerospace and Electronic Systems Magazine, 19(1):5–18.

[BORST et al., 2009] BORST, CHRISTOPH, T. WIMBÖCK, F. SCHMIDT, M. FUCHS, B. BRUNNER, F. ZACHARIAS, P. R. GIORDANO, R. KONIETSCHKE, W. SEPP, S. FUCHS, C. RINK, A. ALBU-SCHÄFFER and G. HIRZINGER (2009). *Rollin' Justin - Mobile Platform with Variable Base*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1597–1598, Kobe, Japan.

[BRENDEL et al., 2011] BRENDEL, WILLIAM, M. AMER and S. TODOROVIC (2011). *Multiobject Tracking as Maximum Weight Independent Set*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1273–1280, Colorado Springs, USA.

[BUEHLER et al., 2007] BUEHLER, MARTIN, K. IAGNEMMA and S. SINGH (2007). *The 2005 DARPA Grand Challenge: The Great Robot Race*, vol. 36 of *Springer Tracts in Advanced Robotics*. Springer.

[BUEHLER et al., 2010] BUEHLER, MARTIN, K. IAGNEMMA and S. SINGH (2010). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56 of *Springer Tracts in Advanced Robotics*. Springer.

[BURCHARDT et al., 2011] BURCHARDT, ARMIN, T. LAUE and T. RÖFER (2011). *Optimizing Particle Filter Parameters for Self-Localization*. In SOLAR, JAVIER RUIZ DEL, E. CHOWN and P. G. PLOEGER, eds.: *RoboCup 2010: Robot Soccer World Cup XIV*, vol. 6556 of *Lecture Notes in Artificial Intelligence*, pp. 145–156. Springer.

[BUTTERFASS et al., 2001] BUTTERFASS, JÖRG, M. GREBENSTEIN, H. LIU and G. HIRZINGER (2001). *DLR-Hand II: Next Generation of a Dextrous Robot Hand*. In *Proceedings of the IEEE International Conference Robotics and Automation*, pp. 109–114, Seoul, South Korea.

[BÄTZ et al., 2010] BÄTZ, GEORG, A. YAQUB, H. WU, K. KÜHNLENZ, D. WOLLHERR and M. BUSS (2010). *Dynamic Manipulation: Nonprehensile Ball Catching*. In *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, pp. 365–370, Marrakech, Morocco.

[CAI et al., 2006] CAI, YIZHENG, N. DE FREITAS and J. J. LITTLE (2006). *Robust Visual Tracking for Multiple Targets*. In *Proceedings of European Conference on Computer Vision*, pp. 107–118, Graz, Austria.

[CAPPÉ et al., 2007] CAPPÉ, OLIVIER, S. J. GODSILL and E. MOULINES (2007). *An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo*. Proceedings of the IEEE, 95(5):899–924.

[CAVALLARO, 1997] CAVALLARO, RICK (1997). *The FoxTrax Hockey Puck Tracking System*. IEEE Computer Graphics and Applications, 17(2):6–12.

[COLLINS, 2012] COLLINS, ROBERT T. (2012). *Multitarget Data Association with Higher-Order Motion Models*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1744–1751, Providence, USA.

[COX and HINGORANI, 1996] COX, INGEMAR J. and S. L. HINGORANI (1996). *An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(2):138–150.

[DALAL and TRIGGS, 2005] DALAL, NAVNEET and B. TRIGGS (2005). *Histograms of Oriented Gradients for Human Detection*. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, San Diego, USA.

[DU et al., 2011] DU, HAO, P. HENRY, X. REN, M. CHENG, D. B. GOLDMAN, S. M. SEITZ and D. FOX (2011). *Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera*. In *Proceedgins of the ACM International Conference on Ubiquitous Computing*, pp. 75–84, Beijing, China.

[ERDINC et al., 2009] ERDINC, OZGUR, P. WILLETT and Y. BAR-SHALOM (2009). *The Bin-Occupancy Filter and its Connection to the PHD Filters*. IEEE Transactions on Signal Processing, 57(11):4232–4246.

[ESS et al., 2009] ESS, ANDREAS, B. LEIBE, K. SCHINDLER and L. VAN GOOL (2009). *Robust Multiperson Tracking from a Mobile Platform*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10):1831–1846.

[FLEPS et al., 2011] FLEPS, MICHAEL, E. MAIR, O. RUEPP, M. SUPPA and D. BURSCHKA (2011). *Optimization Based IMU Camera Calibration*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3297–3304, San Francisco, USA.

[FOX et al., 1999] FOX, DIETER, W. BURGARD, F. DELLAERT and S. THRUN (1999). *Monte Carlo Localization: Efficient Position Estimation for Mobile Robots*. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 343–349, Orlando, USA.

[FRESE et al., 2001] FRESE, UDO, B. BÄUML, S. HAIDACHER, G. SCHREIBER, I. SCHAEFER, M. HÄHNLE and G. HIRZINGER (2001). *Off-the-Shelf Vision for a Robotic Ball Catcher*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1623–1629, Maui, USA.

[GARCIA, 1999] GARCIA, CHRISTOPHE (1999). *Fully Vision-based Calibration of a Hand-Eye Robot*. Autonomous Robots, 6(2):223–238.

[GORDON et al., 1993] GORDON, NEIL J., D. J. SALMOND and A. F. SMITH (1993). *Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation*. IEE Proceedings F Radar & Signal Processing, 140(2):107–113.

[GUÉZIEC, 2002] GUÉZIEC, ANDRÉ (2002). *Tracking Pitches for Broadcast Television*. Computer, 35(3):38–43.

[HARTLEY and ZISSERMAN, 2004] HARTLEY, RICHARD and A. ZISSERMAN (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[HERTZBERG et al., 2013] HERTZBERG, CHRISTOPH, R. WAGNER, U. FRESE and L. SCHRÖDER (2013). *Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds*. Information Fusion, 14(1):57–77.

[HIRZINGER et al., 2002] HIRZINGER, GERHARD, N. SPORER, A. ALBU-SCHÄFFER, M. HÄHNLE, R. KRENN, A. PASCUCCI and M. SCHEDL (2002). *DLR's Torque-Controlled Light Weight Robot III - are we Reaching the Technological Limits now?*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1710–1716, Washington, D.C., USA.

[HO and LEE, 1964] HO, YU-CHI and R. C. K. LEE (1964). *A Bayesian Approach to Problems in Stochastic Estimation and Control*. IEEE Transactions on Automatic Control, 9(4):333–339.

[HOCHBAUM, 2000] HOCHBAUM, DORIT (2000). *Instant Recognition of Polynomial Time Solvability, Half Integrality, and 2-Approximations*. In *Approximation Algorithms for Combinatorial Optimization*, vol. 1913 of *LNCS*, pp. 379–405.

[HONG and SLOTINE, 1995] HONG, W. and J. SLOTINE (1995). *Experiments in Hand-Eye Coordination Using Active Vision*. In *Proceedings of the International Symposium on Experimental Robotics*, pp. 130–139, Stanford, USA.

[HORN, 1987] HORN, BERTHOLD K.P. (1987). *Closed-Form Solution of Absolute Orientation Using Unit Quaternions*. Journal of the Optical Society of America, 4(4):629–642.

[HOVE and SLOTINE, 1991] HOVE, BARBARA and J.-J. E. SLOTINE (1991). *Experiments in Robotic Catching*. In *Proceedings of the American Control Conference*, pp. 380–386, Boston, USA.

[VON HOYNINGEN-HUENE and BEETZ, 2009] HOYNINGEN-HUENE, NICOLAI VON and M. BEETZ (2009). *Robust Real-Time Multiple Target Tracking*. In *Asian Conference on Computer Vision*, pp. 247–256, Xi' an, China.

[IKOMA et al., 2004] IKOMA, NORIKAZU, T. UCHIUO and H. MAEDWANG (2004). *Tracking of Feature Points in Image Sequence by SMC Implementation of PHD Filter*. In *Proceedings of the SICE Annual Conference*, pp. 1696–1701, Sapporo, Japan.

[ISARD and BLAKE, 1998] ISARD, MICHAEL and A. BLAKE (1998). *CONDENSATION — Conditional Density Propagation for Visual Tracking*. International Journal of Computer Vision, 29(1):5–28.

[JULIER and UHLMANN, 2004] JULIER, SIMON J. and J. K. UHLMANN (2004). *Unscented Filtering and Nonlinear Estimation*. Proceedings of the IEEE, 92(3):401–422.

[KALMAN, 1960] KALMAN, RUDOLPH EMIL (1960). *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45.

[KELLY and SUKHATME, 2011] KELLY, JONATHAN and G. S. SUKHATME (2011). *Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration*. International Journal of Robotics Research, 30(1):56–79.

[KHAN et al., 2005] KHAN, ZIA, T. BALCH and F. DELLAERT (2005). *MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(11):1805–1918.

[KIM and GOLNARAGHI, 2004] KIM, ANTHONY and M. GOLNARAGHI (2004). *A Quaternion-Based Orientation Estimation Algorithm Using an Inertial Measurement Unit*. In *IEEE Position Location and Navigation Symposium*, pp. 268–272, Monterey, USA.

[KITANO and ASADA, 1998] KITANO, HIROAKI and M. ASADA (1998). *RoboCup Humanoid Challenge: That's One Small Step for a Robot, One Giant Leap for Mankind*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 419–424, Victoria, Canada.

[KLODMANN et al., 2011] KLODMANN, JULIAN, R. KONIETSCHKE, A. ALBU-SCHÄFFER and G. HIRZINGER (2011). *Static Calibration of the DLR Medical Robot MIRO, a Flexible Lightweight Robot with Integrated Torque Sensors*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3708–3715, San Francisco, USA.

[KNIGHT and REID, 2006] KNIGHT, JOSS and I. REID (2006). *Automated Alignment of Robotic Pan-Tilt Camera Units Using Vision*. International Journal of Computer Vision, 68(3):219–237.

[KONNO et al., 1997] KONNO, ATSUSHI, K. NISHIWAKI, R. FURUKAWA, M. TADA, K. NAGASHIMA, M. INABA and H. INOUE (1997). *Dexterous Manipulations of Humanoid Robot Saika*. In *International Symposium on Experimental Robotics*, pp. 79–90, Barcelona, Spain.

[KRAFT, 2003] KRAFT, EDGAR (2003). *A Quaternion-based Unscented Kalman Filter for Orientation Tracking*. In *Proceedings of the International Conference of Information Fusion*, pp. 47–54, Cairns, Australia.

[LAI et al., 2011] LAI, KEVIN, L. BO, X. REN and D. FOX (2011). *Sparse Distance Learning for Object Recognition Combining RGB and Depth Information*. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4007–4013, Shanghai, China.

[LANG and PINZ, 2005] LANG, PETER and A. PINZ (2005). *Calibration of Hybrid Vision / Inertial Tracking Systems*. In *Proceedings of the InerVis: Workshop on Integration of Vision and Inertial Senors*, Barcelona, Spain.

[LAUE et al., 2009] LAUE, TIM, T. J. DE HAAS, A. BURCHARDT, C. GRAF, T. RÖFER, A. HÄRTL and A. RIESKAMP (2009). *Efficient and Reliable Sensor Models for Humanoid Soccer Robot Self-Localization*. In ZHOU, CHANGJIU, E. PAGELLO, E. MENEGATTI, S. BEHNKE and T. RÖFER, eds.: *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots*, pp. 22 – 29, Paris, France.

[LAUE and RÖFER, 2009] LAUE, TIM and T. RÖFER (2009). *Pose Extraction from Sample Sets in Robot Self-Localization - A Comparison and a Novel Approach*. In PETROVIĆ, IVAN and A. J. LILIENTHAL, eds.: *Proceedings of the European Conference on Mobile Robots*, pp. 283–288, Mlini/Dubrovnik, Croatia.

[LEIBE et al., 2007] LEIBE, BASTIAN, K. SCHINDLER and L. V. GOOL (2007). *Coupled Detection and Trajectory Estimation for Multi-Object Tracking*. In *Proceedings of the IEEE International Conference on Computer Vision.*, Rio de Janeiro, Brazil.

[LEKIEN and MARSDEN, 2005] LEKIEN, FRANCOIS and J. E. MARSDEN (2005). *Tricubic Interpolation in three dimensions*. International Journal for Numerical Methods in Engineering, 63:455–471.

[LI, 1998] LI, MENGXIANG (1998). *Kinematic Calibration of an Active Head-Eye System*. IEEE Transactions on Robotics, 14(1):153–158.

[LI and BETSIS, 1995] LI, MENGXIANG and D. BETSIS (1995). *Head-Eye Calibration*. In *Proceedings of the International Conference on Computer Vision*, pp. 40–45, Cambridge, USA.

[LOBO and DIAS, 2005] LOBO, JORGE and J. DIAS (2005). *Relative Pose Calibration Between Visual and Inertial Sensors*. In *Proceedings of the InerVis: Workshop on Integration of Vision and Inertial Senors*, Barcelona, Spain.

[LOBO and DIAS, 2007] LOBO, JORGE and J. DIAS (2007). *Relative Pose Calibration Between Visual and Inertial Sensors*. International Journal of Robotics Research, 26(6):561–575.

[LUBER et al., 2011a] LUBER, MATTHIAS, L. SPINELLO and K. O. ARRAS (2011a). *People Tracking in RGB-D Data With On-line Boosted Target Models*. In *Proceedings of IEEE/RSJ International Conference Intelligent Robots and Systems*, pp. 3844–3849, San Francisco, USA.

[LUBER et al., 2011b] LUBER, MATTHIAS, G. D. TIPALDI and K. O. ARRAS (2011b). *Better Models For People Tracking*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 854–859, Shanghai, China.

[MA, 1996] MA, SANG DE (1996). *A Self-Calibration Technique for Active Vision Systems*. Transactions on Robotics and Automation, 12(1):114–120.

[MAGGIO et al., 2007] MAGGIO, EMILIO, E. PICCARDO, C. REGAZZONI and A. CAVALLARO (2007). *Particle PHD Filtering for Multi-Target Visual Tracking*. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. I–1101–I–1104, Honolulu, USA.

[MAHLER, 2003] MAHLER, RONALD P. S. (2003). *Multitarget Bayes Filtering via First-Order Multitarget Moments*. IEEE Transactions on Aerospace and Electronic Systems, 39(4):1152–1178.

[MAHLER, 2007a] MAHLER, RONALD P. S. (2007a). *PHD Filters of Higher Order in Target Number*. IEEE Transactions on Aerospace and Electronic Systems, 43(4):1523–1543.

[MAHLER, 2007b] MAHLER, RONALD P. S. (2007b). *Statistical Multisource-Multitarget Information Fusion*. Artech House.

[MARINS et al., 2001] MARINS, JOÃO LUÍS, X. YUN, E. R. BACHMANN, R. B. MCGHEE and M. J. ZYDA (2001). *An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors*. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 2003–2011.

[DE MESTRE, 1990] MESTRE, NEVILLE DE (1990). *The Mathematics of Projectiles in Sport*. Australian Mathematical Society Lecture Series (No. 6). Cambridge University Press.

[MIRZAEI and ROUMELIOTIS, 2007] MIRZAEI, FARAZ M. and S. I. ROUMELIOTIS (2007). *A Kalman Filter-based Algorithm for IMU-Camera Calibration*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2427–2434, San Diego, USA.

[MIRZAEI and ROUMELIOTIS, 2008] MIRZAEI, FARAZ M. and S. I. ROUMELIOTIS (2008). *A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation*. IEEE Transactions on Robotics, 24(5):1143–1156.

[NARUSE et al., 2011] NARUSE, TADASHI, Y. MASUTANI, N. MITSUNAGA, Y. NAGASAKA, T. FUJII, M. WATANABE, Y. NAKAGAWA and O. NAITO (2011). *SSL-Humanoid*. In SOLAR, JAVIER RUIZ-DEL, E. CHOWN and P. PLÖGER, eds.: *RoboCup 2010: Robot Soccer World Cup XIV*, vol. 6556 of *Lecture Notes in Computer Science*, pp. 60–71. Springer.

[NICHIWAKI et al., 1997] NICHIWAKI, KOICHI, A. IONNO, K. NAGASHIMA, M. INABA and H. INOUE (1997). *The Humanoid Saika that Catches a Thrown Ball*. In *Proceedings of the IEEE International Workshop on Robot and Human Communication*, pp. 94–99, Sendai, Japan.

[NICKELS, 2003] NICKELS, KEVIN M. (2003). *Hand-Eye Calibration for Robonaut*. Technical Report, NASA Summer Faculty Fellowship Program Final Report, Johnson Space Center.

[OH et al., 2009] OH, SONGHWAI, S. RUSSELL and S. SASTRY (2009). *Markov Chain Monte Carlo Data Association for Multi-Target Tracking*. IEEE Transactions on Automatic Control, 54(3):481–497.

[OWENS et al., 2003] OWENS, NEIL, C. HARRIS and C. STENNETT (2003). *Hawk-Eye Tennis System*. In *International Conference on Visual Information Engineering*, pp. 182–185, Guildford, United Kingdom.

[PIRSIAVASH et al., 2011] PIRSIAVASH, HAMED, D. RAMANAN and C. C. FOWLKES (2011). *Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1201–1208, Colorado Springs, USA.

[POLLARD et al., 2009] POLLARD, EVANGELINE, A. PLYER, B. PANNETIER, F. CHAMPAGNAT and G. L. BESNERAIS (2009). *GM-PHD Filters for Multi-Object Tracking in Uncalibrated Aerial Videos*. In *Proceedings of the International Conference on Information Fusion*, pp. 1171–1178, Seattle, USA.

[PRADEEP et al., 2010] PRADEEP, VIJAY, K. KONOLIGE and E. BERGER (2010). *Calibrating a multi-arm multi-sensor robot: A Bundle Adjustment Approach*. In *Proceedings of the International Symposium on Experimental Robotics*, New Delhi, India.

[REID, 1979] REID, DONALD B. (1979). *An Algorithm for Tracking Multiple Targets*. IEEE Transactions on Automatic Control, AC-24(6):843–854.

[REN et al., 2004] REN, J., J. ORWELL, G. JONES and M. XU (2004). *Real-time 3D Soccer Ball Tracking from Multiple Cameras*. In *British Machine Vision Conference*.

[RIBNICK et al., 2007] RIBNICK, EVAN, S. ATEV, O. MASOUD, N. PAPANIKOLOPOULOS and R. VOYLES (2007). *Detection of Thrown Objects in Indoor and Outdoor Scenes*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 979–984, San Diego, USA.

[RIBNICK et al., 2009] RIBNICK, EVAN, S. ATEV and N. P. PAPANIKOLOPOULOS (2009). *Estimating 3D Positions and Velocities of Projectiles from Monocular Views*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5):938–944.

[RILEY and ATKESON, 2002] RILEY, MARCIA and C. G. ATKESON (2002). *Robot Catching: Towards Engaging Human-Humanoid Interaction*. Autonomous Robots, 12(1):119–128.

[ROTH et al., 1987] ROTH, ZVI S., B. W. MOORING and B. RAVANI (1987). *An Overview of Robot Calibration*. IEEE Journal on Robotics and Automation, 3(5):377–385.

[SCANDAROLI et al., 2011] SCANDAROLI, GLAUCO GARCIA, P. MORIN and G. SILVEIRA (2011). *A Nonlinear Observer Approach for Concurrent Estimation of Pose, IMU Bias and Camera-to-IMU Rotation*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3335–3341, San Francisco, USA.

[SHEWCHUK, 1994] SHEWCHUK, JONATHAN RICHARD (1994). *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Technical Report, School of Computer Science, Carnegie Mellon University.

[SHIU and AHMAD, 1989] SHIU, YIU CHEUNG and S. AHMAD (1989). *Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$*. IEEE Transactions on Robotics and Automation, 5(1):16–29.

[SHUM and KOMURA, 2005] SHUM, HUBERT and T. KOMURA (2005). *Tracking the Translational and Rotational Movement of the Ball Using High-Speed Camera Movies*. In *Proceedings of the IEEE International Conference on Image Processing*, vol. III, pp. 1084 – 1087, Genoa, Italy.

[SIDENBLADH and BLACK, 2001] SIDENBLADH, HEDVIG and M. J. BLACK (2001). *Learning Image Statistics for Bayesian Tracking*. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 709–716, Vancouver, Canada.

[SMITH and CHRISTENSEN, 2007] SMITH, CHRISTIAN and H. I. CHRISTENSEN (2007). *Using COTS to Construct a High Performance Robot Arm*. In *Proceedings of the IEEE International Conference on Robots and Automation*, pp. 4056–4063, Rome, Italy.

[SORENSON, 1970] SORENSON, HAROLD W. (1970). *Least-Squares Estimation: From Gauss to Kalman*. IEEE Spectrum, 7:63–68.

[SPINELLO and ARRAS, 2011] SPINELLO, LUCIANO and K. O. ARRAS (2011). *People Detection in RGB-D Data*. In *Proceedings of IEEE/RSJ International Conference Intelligent Robots and Systems*, pp. 3838–3843, San Francisco, USA.

[STROBL and HIRZINGER, 2006] STROBL, KLAUS H. and G. HIRZINGER (2006). *Optimal Hand-Eye Calibration*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4647–4653, Beijing, China.

[SZELISKI, 2010] SZELISKI, RICHARD (2010). *Computer Vision: Algorithms and Applications*. Springer.

[TSAI, 1987] TSAI, ROGER Y. (1987). *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*. IEEE Journal of Robotics and Automation, 3(4):323–344.

[UDE and OZTOP, 2009] UDE, ALEŠ and E. OZTOP (2009). *Active 3-D Vision on a Humanoid Head*. In *International Conference on Advanced Robotics*, Munich, Germany.

[VO and MA, 2005] VO, BA-NGU and W.-K. MA (2005). *A Closed-Form Solution for the Probability Hypothesis Density Filter*. In *Proceedings of the International Conference of Information Fusion*, pp. 856–863, Philadelphia, USA.

[VO and MA, 2006] VO, BA-NGU and W.-K. MA (2006). *The Gaussian Mixture Probability Hypothesis Density Filter*. IEEE Transactions on Signal Processing, 54(11):4091–4104.

[VO et al., 2006] VO, BA-TUONG, B.-N. VO and A. CANTONI (2006). *The Cardinalized Probability Hypothesis Density Filter for Linear Gaussian Multi-Target Models*. In *IEEE Conference on Information Sciences and Systems*, pp. 681–686, Princeton, USA.

[VO et al., 2007] VO, BA-TUONG, B.-N. VO and A. CANTONI (2007). *Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter*. IEEE Transactions on Signal Processing, 55(7):3553–3567.

[WAN and VAN DER MERWE, 2000] WAN, ERIC A. and R. VAN DER MERWE (2000). *The Unscented Kalman Filter for Nonlinear Estimation*. In *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, Lake Louise, Canada.

[WANG et al., 2007] WANG, YA-DONG, J.-K. WU, W. HUANG and A. A. KASSIM (2007). *Gaussian Mixture Probability Hypothesis Density for Visual People Tracking*. In *Proceedings of the International Conference on Information Fusion*, Quebec, Canada.

[WANG et al., 2006] WANG, YA-DONG, J.-K. WU, A. A. KASSIM and W.-M. HUANG (2006). *Tracking a Variable Number of Human Groups in Video Using Probability Hypothesis Density*. In *Proceedings of the International Conference on Pattern Recognition*, pp. 1127–1130, Hong Kong, China.

[WU et al., 2012] WU, ZHENG, A. THANGALI, S. SCLAROFF and M. BETKE (2012). *Coupling Detection and Data Association for Multiple Object Tracking*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1948–1955, Providence, USA.

[YAN et al., 2006] YAN, FEI, A. KOSTIN, W. CHRISTMAS and J. KITTLER (2006). *A Novel Data Association Algorithm for Object Tracking in Clutter with Application to Tennis Video Analysis*. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 634–641, New York, USA.

[YANG and HU, 1998] YANG, CHANGJIANG and Z. HU (1998). *An Intrinsic Parameters Self-Calibration Technique for Active Vision System*. In *Proceedings of the International Conference on Pattern Recognition*, pp. 67–69, Brisbane, Australia.

[VAN DER ZANT and WISSPEINTNER, 2006] ZANT, TIJN VAN DER and T. WISSPEINTNER (2006). *RoboCup X: A Proposal for a New League Where RoboCup Goes Real World*. In BREDENFELD, ANSGAR, A. JACOFF, I. NODA and Y. TAKAHASHI, eds.: *RoboCup 2005: Robot Soccer World Cup IX*, vol. 4020 of *Lecture Notes in Artificial Intelligence*, pp. 166–172. Springer.

[ZHANG et al., 2008] ZHANG, LI, Y. LI and R. NEVATIA (2008). *Global Data Association for Multi-Object Tracking Using Network Flows*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, USA.

[ZHANG, 2000] ZHANG, ZHENGYOU (2000). *A Flexible New Technique for Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334.

Appendix A

Released Software

During the course of time, parts of the work discussed in this dissertation were not only published as texts but the corresponding software was also made publicly available. This not only extends and supports the text publications but also gives other researchers the possibility of using the software in their work with the possibility of extending it. In all cases, the software was released as open source.

A.1 SSL Vision

In RoboCup's Small Size League (SSL) a shared vision hardware has been introduced using a publicly available vision software, namely SSL Vision. It provides the same set of well established perception features for all teams. The contribution of the author is the specification and development of a new calibration technique, requiring no additional calibration tools (e. g. checkerboard pattern). The software is available at `http://code.google.com/p/ssl-vision/`.

A.2 A Visual SLAM System from Open Source Components

This software release presents the results of building a visual SLAM system from components of open source software, namely OpenCV for feature detection, FLANN for data association, and SLOM for solving the sparse bundle adjustment problem. The software framework and the calibration of the used sensor setup (camera pair and IMU) were provided by the author. The software is available at `http://informatik.uni-bremen.de/agebv/en/pub/hertzbergicra11`.

A.3 MTKM: Manifold Toolkit for MATLAB

The Manifold Toolkit for MATLAB is a framework to define and solve least squares problems in a general and easy way. Furthermore, it provides a way to handle 3D rotations. It is used to solve elaborate problems, such as the calibration of cameras and an IMU relative the kinematic chain of a service robot or operating successfully on SLAM benchmark datasets. The software is available at `http://openslam.org/MTK.html`.